Henley Business School
UNIVERSITY OF READING

Factors Affecting the Quality of Enterprise Architecture Models

HENLEY BUSINESS SCHOOL

THE UNIVERSITY OF READING

Submitted in partial fulfilment for the award of the degree of Doctor of Philosophy

Informatics Research Centre

Cameron Spence

December 2020

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Certificate of readiness to be included in library

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Acknowledgements

I wish to acknowledge, first of all, the tremendous encouragement and assistance from my primary supervisor, Vaughan Michell, who has had the unenviable task of keeping me on track throughout this long, rather drawn-out process. Thanks also to Yin Leng Tan, my second supervisor, for useful hints and questions.

I'm grateful to an ex-colleague, Sam Chong, for nudging me back onto the academic track later in life, and to Prof. Diane Berry for pushing me into this PhD (I think).

A huge thank you to my wife, Cath, for support and encouragement through all the tough times, and for undertaking the onerous task of proof-reading this thesis. All the time I spent on the research was time away from those closest to me, and that was not easy. Thank you also to my dear friends Jane Newton and Russell Ede for simply not letting me give up, even when I desperately wanted to.

Thanks are due to another ex-colleague, Darren Jones, for the deceptively simple question that led to my first research question, and to Prof. Yinshan Tang from Reading University for the discussion that led to my second research question.

I've been fortunate enough to spend the latter years of this PhD in business roles that have actively encouraged my research in this area and so I'm grateful to ex-colleagues at Maersk, and current colleagues in Thames Water for supporting me in my endeavours as well as providing so many interesting ideas, discussions and challenges to my thinking.

I'm grateful to my two examiners, Prof. Erik Proper and Markos Kyritsis, both of whom made the viva process actually enjoyable, and who contributed very helpful suggestions to improve the quality of the final product.

I've also benefited from having access to a leading Enterprise Architecture modelling tool, BiZZdesign Enterprise Studio, which has proved enormously helpful in producing much of the materials used in the survey and elsewhere in the thesis. In particular, I'm grateful to Marc Lankhorst for the provision of the sample BiZZdesign architecture repository that demonstrates in practice the automated creation and visualisation of ArchiMate derived relations.

Dedication

I’d like to dedicate this work to my amazing family; in particular to my mother Phyllis, who always believed in me; and to my three fantastic sons Daniel, Sam and Joshua, of whom I am so proud.

Abstract

We start our research by introducing the subject of Enterprise Architecture (EA), its content and purpose, as well as discussing what we mean by a 'model', and 'quality', building on concepts from semiotics and in particular on conceptual model quality.

We set out to answer three questions. The first deals with how we measure the quality of a set of Enterprise Architecture models, and to answer this we produce a mathematical framework and then test it using a case study. This extends the conceptual model quality work done by Lindland and Krogstie into the realm of Enterprise Architecture, adding new aspects related to completeness of sets of models, modelling maturity as well as conditions for increasing quality. This incorporates mathematical concepts, including set theory and calculus, and proposes three specific metrics for the quality of sets of models (related to truthfulness, syntax and completeness). This uses a simple case study, based upon purely quantitative data, sampling the contents of an existing Enterprise Architecture repository.

The second deals with how we measure the effectiveness of the language used in Enterprise Architecture models. We again use mathematical techniques to construct metrics, this time related to comprehension and utility: the former incorporating a triangulation technique based upon Kvanvig's concept of moderate factivity of objectual understanding, and the latter being a more subjective measure (i.e. self-assessment). From these two metrics we provide a new conceptual visualisation of the effectiveness of language concepts. We then test this framework using a mixed-mode case study, carrying out 68 interviews, based mostly upon quantitative data again but with additional elements of qualitative data. Although the conceptual framework is independent of any particular language, in order to test it we actually need to select an Enterprise Architecture framework, or more specifically, the modelling language within such a framework; the framework we choose for this purpose is ArchiMate. Through the use of alternative modelling notations in the survey process, we gain insights not just into the understanding and utility of various ArchiMate concepts, as perceived by respondents, we also gain insights into the effect of understanding and utility of using the specific notation provided by ArchiMate through the use of differential analysis of the result sets thus obtained.

The final question we address is more practically focused and deals with how we can specify and automate various kinds of changes to Enterprise Architecture models based upon the previous research. We construct a conceptual framework illustrating the kinds of transformations that may be required, given what we have learnt in the previous chapters, demonstrate that these can be deterministic and finally demonstrate, by use of a specific Enterprise Architecture modelling tool (BiZZdesign), that they can be implemented in software, and thus automated.

In the course of our research, we deliver reusable methodologies and frameworks that will assist future researchers into Enterprise Architecture and related frameworks, as well as Enterprise Architecture practitioners.

Table of Contents

Appendices

List of figures

List of tables

Chapter 1. Introduction

1.1 Introduction to Enterprise Architecture and Modelling

The term 'architecture' is perhaps familiar to many in the context of buildings, where an architect may be hired to design and oversee the construction of projects ranging from a small change to a domestic property up to projects taking months or years to complete. In the UK, the Royal Institute of British Architects says this about why one should hire an architect:

Architects are highly skilled and professionally trained to turn your aspirations into reality. They will guide you through the design, planning and construction process whether you are constructing a new building or adapting an existing property.

Architects apply impartial and creative thinking to projects large and small. They add value, whether from maximising light and space, adding functionality, or achieving the best return on your investment. [1]

When working with such an architect, one would expect them to produce a number of diagrams, or models, showing a number of perspectives on what is going to be built, to ensure that you, as the client, are happy that the end results from the project will be what you expect: that the results you are looking for are actually realised.

They might, for example, produce financial models, 'green' models such as the efficiency of heating and/or heat insulation, 3-D models showing the space and lighting, and so on.

They may also oversee the work of various tradespeople, providing them with detailed guidance and specifications (again, many of these will be pictorial) regarding what each of them need to do as part of the overall big picture.

This analogy, and indeed the terminology, carries over into the world of Information Technology.

1.2 Enterprise Architecture in Information Technology

1.2.1 Impact of Enterprise Architecture

Enterprise Architecture (EA) is widely used to model and analyse businesses, to a greater or lesser degree depending on geography [2, 3] and its practitioners wield a significant amount of influence (either "final decision maker" or "great deal of influence") on over $ 10^{12} of Information Technology (IT) related spend, according to Gartner [4].

1.2.2 What is Enterprise Architecture?

The phrase Enterprise Architecture comprises two independent words whose meaning both contribute an essential aspect of the definition of the phrase. The term 'enterprise' itself has a number of definitions, including "*a company organized for commercial purposes; business firm*" [5] and "*an organization, especially a business, or a difficult and important plan, especially one that will earn money*" [6]. One particular publication (an industry standard in the field of Enterprise Architecture) offers a more general definition of 'enterprise' that could in some situations encapsulate only [a large] part of a business, and in others, multiple businesses:

The TOGAF standard considers an "enterprise" to be any collection of organizations that have common goals.

For example, an enterprise could be:

■ *A whole corporation or a division of a corporation*

■ *A government agency or a single government department*

■ *A chain of geographically distant organizations linked together by common ownership*

■ *Groups of countries or governments working together to create common or shareable deliverables or infrastructures*

■ *Partnerships and alliances of businesses working together, such as a consortium or supply Chain* [7]

The use of the term 'Enterprise' normally signals to us, in the context of IT Architecture, that we are considering the whole of an Enterprise, as opposed to just part of it (for more on different types of architecture, see 2.2.1.5 below).

The term 'Architecture' borrows from the original meaning (in the context of buildings and structures) to signify that we are dealing with outcomes, plans and approaches rather than fine detail. In an IT context there are additional aspects to the term 'architecture', in particular the change over time which may be less likely to be relevant to buildings. These additional aspects can be seen in the ISO definition of architecture: [the] *fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*. [8].

When we apply the term 'architecture' to an 'enterprise', the 'system' in the ISO definition is of course the enterprise; thus, Enterprise Architecture deals with the high-level structures of a whole business (enterprise) and its supporting IT. A book published in 2009 on the topic of Enterprise

Architecture suggests that three important perspectives relevant to the role of an enterprise architecture are regulation, design and patterns; and thus offers its own definition of enterprise architecture, incorporating these perspectives:

A coherent set of descriptions, covering a regulations-oriented, design-oriented and patterns-oriented perspective on an enterprise, which provides indicators and controls that enable the informed governance of the enterprise's evolution and success. [9]

It often includes aspects of motivation (desired outcomes) and strategy. It provides a framework in which more detailed design and implementation work can be safely carried out.

1.2.3 What is the purpose of Enterprise Architecture?

The *purpose* of Enterprise Architecture can be seen by analogy with the purpose of Architecture in its classic sense. We hire architects to design houses for many reasons, including: (a) the house must be structurally sound; (b) it must look good (well-proportioned); (c) it must fulfil the needs we have specified (e.g. space of different kinds); (d) it must comply with all relevant regulations; (e) the building must be able to bear the demands placed upon it, such as, for example load-bearing capacity of the upstairs floors.

Many of these also carry through into the world of IT. A business, or its supporting IT, must be structurally sound. This applies in many areas, for example the financial health of an organisation (for example as discussed in [10], hence the introduction of stress testing for banks [11]), the organisational and governance structures; it needs to be able to deliver the services required both by its external and internal customers; it needs to comply with various kinds of legislation (for example employment law and the UK Data Protection Act [12]); it needs to have sufficient capacity in its IT infrastructure to support all the demands upon it (for example, support a sufficient number of electronic transactions, or a sufficient volume of home shopping deliveries, something tested severely at the start of the Coronavirus pandemic in early 2020 [13]).

Thus, we might surmise that the purpose of Enterprise Architecture is to ensure that the enterprise is fit for purpose, designed correctly to achieve the outcomes expected of it. Some useful definitions are offered in the literature that provide quite different perspectives. TOGAF offers "The purpose of Enterprise Architecture is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy" [14]; thus we have, as a goal of Enterprise Architecture, "optimisation" in some fashion. ArchiMate focuses more on stakeholders and concerns in its "why" of Enterprise Architecture, suggesting in its overview section that "An Enterprise

Architecture is typically developed because key people have concerns that need to be addressed by the business and IT systems within an organization" and that "without an Enterprise Architecture, it is unlikely that all concerns and requirements are considered and addressed" [15].

The term has two subtly separate connotations to it. It can be treated as a noun, in which case we use it as a description of the current structure of an Enterprise. It can also be used in a more active sense (similar to a verb), so we have a job role called an 'Enterprise Architect', and we sometimes say 'we need to architect a solution'; in which case we use it to describe the activities required to create a description of the current or future structure of an Enterprise, or to carry out analysis related to this.

1.2.4 Content of Enterprise Architecture

Enterprise Architecture requires the management of complex data sets to satisfy the needs of its various clients, both in the business and technical domains [16]. For example, in the business domain we would include information about services provided and the actors involved; in the technical domain we would normally include applications, platforms and technical infrastructure. This data is assembled and presented in different ways to suit the needs of the stakeholders. For example, some stakeholders will be interested in a financial view; others will be interested in security, or in data replication and duplication. A variety of tools are used to manage the underlying data and build these models, ranging from very rudimentary tools such as basic office productivity software, up to sophisticated software products specifically designed to handle this kind of data (such as those researched by Gartner [17]), for example Sparx Enterprise Architect [18], MooD [19], Troux [20], ARIS [21] and BiZZdesign [22].

Together the set of models built and managed by this kind of tool provide a visualization of the data and relationships comprising our 'body of knowledge' of the architecture of the enterprise in question, along with appropriate analysis of that data.

1.3 All about models

1.3.1 Models in general

The term 'model' is a very widely-used term, and means many things to different people and in different contexts. However, there are three key characteristics to models that we wish to explore, before moving on to the use of models for Enterprise Architecture::Firstly, they represent something else. Models are always <u>of</u> something.

Secondly, they are always imperfect; they are always an approximation to the original. For example, an artist's sketch, or even a photograph, will not capture the original in perfect detail.

Thirdly, they have a purpose. They are there for a reason, perhaps to enable an analysis to be done of the subject matter, or perhaps to convince someone of a particular course of action.

Having laid out three characteristics of models above, it is instructive to consider directly what a “good” model would look like, against each of those characteristics.

- Models represent something else – so a good model would be a good representation, not misleading. This, we will see, is an aspect that will be reflected in the literature on model quality, and in our own research.
- Models are inaccurate, so will not hold every detail, although we would hope that the detail that is included, is correct (see previous point)
- Models are there for a purpose – and so a good model would fulfil the intended purpose. For example, if the purpose of a model was to enable us to reduce the cost of operating our IT, then the model is good if it enables us to do that, or at the very least, gives us the information in a suitable form to make that a possibility!

Because the term model can mean so many things, this can lead to confusion about the intended meaning unless we are very careful to set the correct context. An online thesaurus offers the following synonym for the term ‘models’:

Figure 1 Synonyms for Models [23]

copy	clone	figurine	print	tracing
figure	copycat	illustration	relief	dead ringer
image	ditto	layout	representation	game plan
painting	dummy	look-alike	ringer	knock-off
photograph	duplicate	miniature	setup	spitting image
picture	effigy	mock-up	sketch	visual
portrait	engraving	paste-up	statue	
cartoon	facsimile	pocket	statuette	

All of these give the sense of *representing something else* – our first aspect of models. Cultural contexts may add other of the three aspects mentioned above. For example, in terms of purpose, statues are often erected of historical figures by way of commemoration; their purpose is to remind us of important people in our past. The terms ‘dead ringer’ and ‘spitting image’, which are synonyms mentioned above, as well as being English colloquialisms, have also found their way into UK public life in comedy shows; the former on BBC Radio 4 and the latter on the ITV network; the purpose of both of these was to entertain by means of satire.

1.3.2 Models in the context of IT

The Information Technology used by information-intensive organisations can become very complex over time. The complexity exists not only in the technology itself but also in the way that the technology is used to serve the needs of the business. It is not uncommon, for example, for an organisation to have several hundred IT applications used in various parts of the business, often with overlapping or duplicate functionality, as evidenced by a number of application rationalization studies conducted by the author (for example, one of the UK police forces whose data was analysed in 2011 [24] had 666 separate IT applications) with very significant duplication in application functionality and information being stored. When one considers that organisations may execute many hundreds of business processes, each of which may require a number of applications to support them, it is clear that the interaction between business and its supporting IT is complex and many-faceted.

For this reason, organisations wishing to understand and then change their IT portfolio will seek to build a "picture" or "model" of the business and its supporting ICT infrastructure to see how the two are linked [25], and to then draw conclusions and make recommendations that will bring business benefits (such as reduced risk and cost, and increased speed of change) to those clients. A model is a representation or miniaturisation of something else, that allows one to grasp the essentials of something larger; in this case, we are considering a representation or miniaturisation of the organisation and its supporting ICT. These miniaturisations contain the essential elements of the larger reality, just like a Lego figure or vehicle is recognisable as a representation of a real-life human or vehicle.

The term 'model' can of course mean many things to different people, as we can see from the previous discussion. In an IT context, the following terms can sometimes be used interchangeably:

- Model (in particular if we are talking about a two-dimensional, flat model)
- Diagram
- View
- Picture

These are not necessarily synonymous.

Some models may be two-dimensional representations (for example graphs); others may not be. We might say that diagrams, views and pictures are models; but not all models are diagrams, views or pictures. Indeed, later in this thesis we will be discussing mathematical models of quality and effectiveness.

What about the purpose of models in the context of Information Technology, and Enterprise Architecture? They serve many purposes, including the following:

Table 1 Some Purposes of IT models

	Models enable us to communicate ideas. A single diagram can often communicate concepts faster than it is possible to do so using a string of words, hence the phrase "a picture paints a thousand words".
 [2]	If our models contain some kind of structured information that lends itself to analysis, then we can use models to determine the impact of a change at a given point. We can answer questions like (1) what would be the impact on a business if this particular piece of infrastructure failed, or (2) what would be the impact if we decided to outsource our manufacturing to another supplier?
	Somewhat related to the previous point is the value that models can have in helping us make decisions. If we could see the impact of different choices that we could make, then we could work out the best course of action. One might be analysing the effect on our IT of possible choices of integration strategies (e.g. focusing on microservices [26] vs. Enterprise Service Bus [27] techniques) to see the future impact on our operational costs and speed of development of new solutions.

[1] Image courtesy of PASIEKA/Science Photo Library/Corbis, fair academic use

[2] Image courtesy of prosperityconnection.org, fair academic use

[3] Image courtesy of depositphotos.com, fair academic use

	If our modelling environment allows us to do this, then we can piece together data from a number of sources to assemble, piece by piece, a bigger picture of the data in a particular area. An example would be using (looking ahead to our research) an EA modelling tool to produce pictures showing us the full context in which a particular application is being used, by combining information from business and technical staff into a single picture.

Along with many other disciplines, Enterprise Architecture specifically, and Information Technology more generally, have their own specialised languages that are generally understood by their practitioners and less so by those that rely upon their results. Thus, although many people will be familiar with the term ‘software’ as it has entered our standard vocabulary (for example, according to the Oxford Learner’s Dictionary, this is “the programs used by a computer for doing particular jobs” [28]), there are other terms that have specialised meanings that will not be found in normal dictionaries, but only in specialised publications. Examples of this would include the term “Logical Application Component” which is only found in publications related to a specific Architecture Framework (TOGAF [29-31]). Although most people would have an idea what the individual words in this phrase might mean, the meaning behind this specific combination of words would be unknown to anyone not specifically trained in this Architecture Framework. Thus, it can be seen, like many other disciplines, Enterprise Architecture has its own forms of specialised language. These languages often contain specialised (technical) definitions not just for the nouns in the language, they also specify the allowable relations between the nouns. By way of analogy, the sentence “the cat sat on the mat” contains three parts: a subject (the cat), and object (the mat) and an action (sat on). The action (sat on) here is the relationship between the cat and the mat. If we were to draw this, we would create a picture with two objects on it (a cat and a mat), and then we would do something with the picture to imply that there was a relationship between them (looking forward to Gestalt theory here, see section 2.3.14 below), either by putting them vertically adjacent, or in the manner more normally used in architecture models, by drawing a line between the cat and the mat.

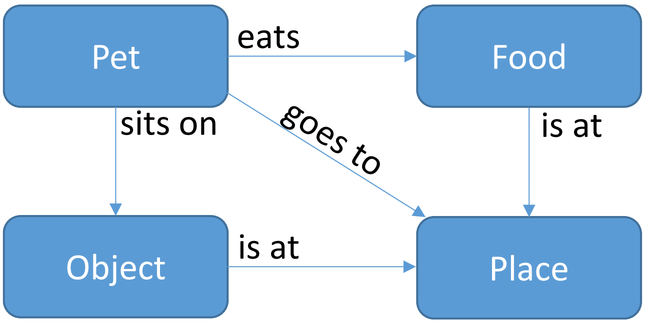
[4] Image courtesy of depositphotos.com, fair academic use

Figure 2 Cats on Mats, Different Notation

Cat and mat related implicitly by adjacency in diagram (model)	Cat and mat related explicitly by line joining them (line represents "sat on")

Thus, we could construct a simple 'modelling language', analogous to an architecture modelling language, that allowed for nouns such as animal (let's say pets, specifically), place, food, and relations such as sit on, eat, put (cat eats food, cat goes to her basket), and so on. We might even construct an "entity metamodel" to describe the allowable terms[32] (entities and relationships), like this:

Figure 3 Entity Metamodel for Pets



We can see that this permits us to "say" things like:

- My cat (pet) sits on his blanket (object)
- My dog (pet) eats my slipper (food?)

As long as the subject, object and action are provided for in our language, then we can say what we need to using the above metamodel (model about models) to guide the construction of our sentence. Broadly speaking, the boxes in the metamodel provide the nouns for our sentences, and the relations (lines between them) provide the actions.

Languages such as ArchiMate do exactly the same, but in the world of Enterprise Architecture, where the nouns are things that makes sense in a business or IT context (e.g. a business process, an application) and the relations show how the different kinds of things are related (e.g. this business process requires that application, or this requirement will be fulfilled by that particular technology).

1.3.3 Models in the context of Enterprise Architecture

These models are often used within the context of a discipline called Enterprise Architecture, which is perhaps an over-specified phrase having multiple different definitions in the literature, however deals ultimately with the structure and evolution over time of a business and its supporting technology [33].

Enterprise architecture is more often concerned about which initiatives should be carried out, for what reason, and so can be categorised as specifying the WHY and WHAT in terms of those initiatives. Solution architecture is concerned with implementing the initiatives, and thus deals with HOW and WITH WHAT (for example the choice of a specific solution component).

1.3.3.1 Communicating using Enterprise Architecture

One of the key purposes of enterprise and solution architecture is that of communication, passing information about existing systems as well as planned changes to those systems. The CHAOS report [34] identified that only 16% of software projects complete on time and budget, and met the original specification (although section 2.2.1.4 below for subsequent doubts raised about that research). One of the key success factors identified was executive management support. Senior managers are key stakeholders in both enterprise and solution architecture and a key purpose of architecture work is communication of intent in a manner that can be clearly understood by the relevant stakeholders in order to ensure project success.

This communication occurs in a number of directions. In some cases a project may wish to demonstrate that its plans adhere to the guidelines and constraints imposed by the enterprise, thus demonstrating compliance or conformance. This can be viewed as communicating ‘up’ to those that have the power to help or hinder a project. In other cases this communication forms part of the hand-off to other teams (perhaps including specifications, patterns or context for further work), such as envisaged by Model Driven Architecture [35], where the communication might be said to be ‘down’ to implementation teams.

In both cases (up and down), it is important that the communication is successful, that is, that the information intended to be conveyed, is in fact conveyed and understood accurately. In other words, the meaning (semantics) that the author of the model intended to convey is accurately received by the reader of the model, as envisaged in the definition of a model being a simplification of reality constructed for a particular person with a purpose in mind [36]. This relies upon the sender and receiver knowing the same language, having the same norms, as discussed for example in [37]. In our use of the term 'model' in the discussion below, we mean it in the more specific sense of a diagram, unless otherwise specified.

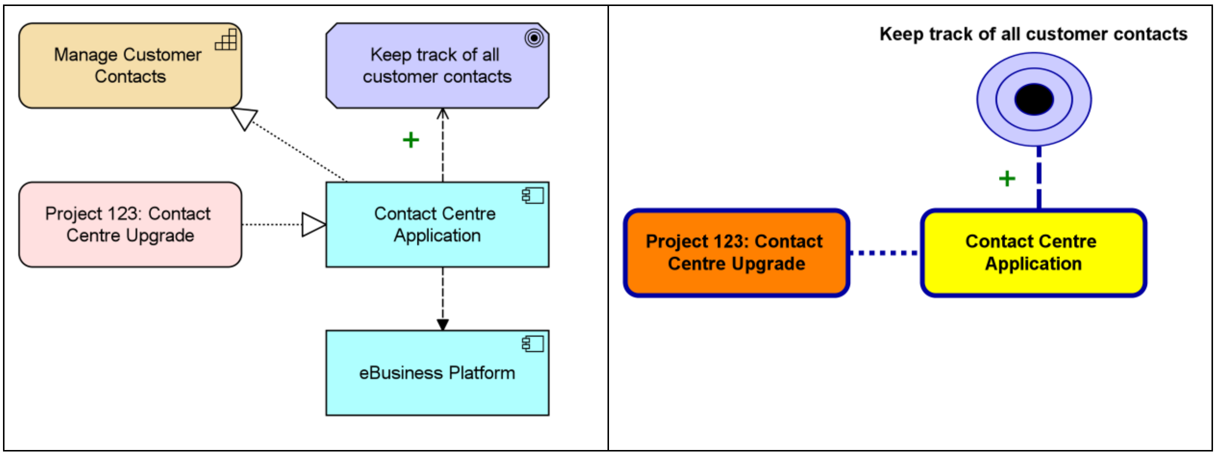
1.3.3.2 Stakeholders and their requirements

The stakeholders who ultimately 'consume' the information may be very different, in terms of both the information that they are looking for and the way that they prefer to see things. When communicating 'up' to perhaps non-technical stakeholders (e.g. CxO executives or programme and project managers), the kind of language that we can use is very different to the language that we can use when communicating to technical staff, or indeed to actual technology, in the cases where our models are actually consumed by software (for example, to auto-generate program code or database schemas).

Consider the example of a system being introduced that is intended to keep track of all communications with customers, by whatever channel (e.g. phone calls, email, instant messaging, and so on). In the role of an IT architect, when communicating with senior management about the purpose of that system, we might emphasise the business benefits (for example, supporting important business capabilities [38], or helping us to achieve a particular goal). One might want to know that this system shares information with our eBusiness platform. This is all in the context of a particular project for which we are seeking investment from the board. Although we might start in a familiar language for Enterprise Architects (ArchiMate), we might tailor the appearance of the model for use with those stakeholders, and thus use slightly less complex representations, for example as shown below:

Figure 4 Business Benefit Model for Contact Centre Upgrade

Version for fellow IT architects	Version for business stakeholders



We might use the one on the left in discussions with fellow IT architects, because they will immediately (if trained in ArchiMate) recognise that we have linked the proposed new application (Contact Centre Application) with a business capability (Manage Customer Contacts), asserted that it will help us meet our goal of "keep track of all customer contacts"; that the application will be realised (put in place) by Project 123, and that information will flow from the new application to our existing eBusiness Platform. To our business stakeholders, however, we would use a picture like the one on the right to help illustrate why they should agree funding for Project 123: that it will provide a new application that they need to fulfil their need to manage and keep track of customer contacts. We can feel free to adjust colours, notation and anything else we need to make the model easier to understand. Either way, the model helps us answer questions like "so what are we doing about tracking customer contact anyway?", and "are there any upcoming projects that might need to make changes to our eBusiness platform?".

One of the differences between the two models is, of course, that the one on the left uses a specific notation (ArchiMate), whereas the other does not. Another example is that some elements that are not meaningful or important to a particular stakeholder in this context have been omitted entirely. For example, the stakeholder may not care about business capabilities, or the underlying eBusiness platform.

Our modelling languages (as provided in architecture frameworks) can play a useful part in the process of communicating, because they provide a standard set of terms (taxonomy) along with a standard (semantic) interpretation to apply to those words, i.e. an ontology. These words can refer to entities (nouns) in our description of the architecture (e.g. an application, a server); they can also refer to relationships between them (e.g. this application is composed of those components). These sets of entities and relations are sometimes referred to as the "entity metamodel" within an architecture framework (the set of allowable nouns and relations between them). This way of communicating works both via the written medium (using words) and verbally. Thus, we can use the word "capability" or "principle" and expect that both the sender and receiver of the communication know what is meant by these words, if they are all using the same framework.

In addition to this, some (not all) frameworks also specify a graphical notation that enable the reader (if they know the notation) to recognise the type of entity or relation being discussed (as demonstrated for example in Figure 4 above, left-hand model). For example, TOGAF [29] does not specify any particular notation to use in diagrams, and so without additional context or labels, it would not be possible to know, merely by inspection, whether a particular symbol represented for example business service or a logical application component. By contrast, ArchiMate [39] and UML [40] both have specified symbolic notation that enables one to identify the type of an element, e.g. recognising that something is a capability just by the shape of the element (the stack of small squares in the top right are the key to recognising this particular symbol):

Figure 5 Example ArchiMate Symbol

Both of these elements (underlying terminology plus graphical notation) are used when communicating with stakeholders. The former is used when giving written reports or giving information verbally; the latter is used typically when presenting diagrams. This thesis will use both.

These models are designed to communicate information to various stakeholders, and so they will visualise, in a number of ways, specific pieces of information in our 'body of knowledge' about the enterprise in question. In the same way that one would describe a situation to a native English or French speaker using different languages or perhaps, to finesse the metaphor, one might adjust one's vocabulary so that one used a simplistic vocabulary with a non-native English speaker, but use a richer, more idiomatic vocabulary with a native English speaker, we may wish to modify our language for various stakeholder groups. These different models are describing the same thing, but just in a different way.

This communication will happen in many different scenarios, for different reasons. At the start of a programme or project we may be interested in whether the initiative is worthwhile attempting at all. Later on in the project, we may be trying to decide between alternative solution possibilities. Thus, the information that we have, the stakeholders that we communicate with, and the purpose of that communication, will change over time.

1.3.4 Model Quality

We have seen in section 1.3.1 that models have three characteristics (stand for something else; serve a purpose; and are only a partial representation); in section 1.3.2 we considered some of the uses of models in Information Technology; we moved on consider models in the context of Enterprise Architecture in section 1.3.3; and now we will consider another aspect of models, and that is their quality.

1.3.4.1 Quality in General

We explore the concept of quality more thoroughly in the literature survey, in particular section 2.2.4 below, however we can summarise quality by saying that it can be seen as many things, depending upon one's perspective, but that it can sometimes be related to the ideas of excellence, value, or conformance to specifications.

The term 'quality' has many definitions. An interesting review of this in 1994 [41] discussed some of the historical roots of our quality definitions, including excellence, value, conformance to specifications, meeting/exceeding customer expectations, and being defined by the customer's viewpoint. It suggests that no one definition is 'best in every situation because each definition has its own strengths and weaknesses.

Taking each of these possible definitions from [41] in turn, it is possible to see how some of them might be applied in the field of Enterprise Architecture models:

1.3.4.1.1 Quality as Value

These focus attention on organisation's efficiency and effectiveness, two characteristics which feature heavily in our research on the effectiveness of modelling languages. It also resonates with the discussion on 'feasible completeness' of sets of models.

1.3.4.1.2 Quality as Conformance to Specifications

This has an obvious, immediate application in model quality, in the conformance to the specified modelling language(s).

1.3.4.1.3 Quality as Meeting/Exceeding Customer Expectations

This one perhaps also encompasses the importance of the customer's viewpoint. It perhaps has a relationship to the completeness of modelling: if our "customer" is expecting that we have modelled a certain domain (subset of our enterprise), then have we actually met that goal?

1.3.4.2 Quality in the Eye of the Beholder?

We cannot of course expect any model to be perfect in its representation of its subject, otherwise it would not be a model according to our definition. That does not mean, however, that it should be wrong either; we would hope that although it might not express the whole truth about a subject, at least, it should not express an *untruth*. There are several reasons why, in practice, a model might not tell the whole truth about a subject. For example, the stakeholder(s) reading the model(s) will have their own particular point of view, their own concerns (these are referred to in architecture frameworks as "viewpoints"), and thus they are likely to want to see only certain aspects of the modelled domain. For example, a business stakeholder with no technical understanding might not want to see technical details of the infrastructure hosting a particular solution. Thus, when producing a model for a business stakeholder, we would only show that part of the picture that was relevant and useful and comprehensible to them.

1.3.4.3 How many models do we need?

Our models are typically produced by practitioners, for example enterprise and solution architects. How do we know when we have enough models? What do we consider a sufficient number? How do we judge the quality of those models as a set?

If we take an analogy of software development, it is customary before writing software, to start from some set of requirements that specify what our software should do so that we can objectively test whether or not our software has met the acceptance criteria.

The question we must consider is, what are the acceptance criteria for our enterprise architecture models? If we do not have any, then we have no objective way of measuring their quality, neither do we have any objective way of measuring whether they are sufficient for our needs. If our practitioners choose to spend one day of modelling per project, or one year of modelling per project, who is to say whether that is too much or too little time?

1.3.4.4 Model Quality and Propositions

Borrowing from the language of propositions, that we explore in our discussion on the topic of "understanding" in section 2.3.16 below, if there exists a set of true propositions p about the model subject S, and a set of false propositions q about the same subject, then we would like the propositions on our model to consist of a subset of p, and not contain any of q. In the language that we use later in this thesis on model quality (see section 4.3 below), we will refer to models containing only propositions from p as being in the domain $\mathcal{D}$, and models containing propositions from q as being outside the domain $\mathcal{D}$.

1.3.4.5 Importance of Model Quality

So, if we have some models describing the current and/or future state of our Enterprise Architecture, does it matter how ‘good’ they are?

It does, for reasons directly related to the purposes of the models. We illustrate, in Table 1 above, some purposes of IT models. Let us consider how poor quality models might affect these various purposes, and the consequences that might result from that.

Table 2 Consequences of poor quality models

	If our models are incorrect, then we fail to communicate what we intend. We might just omit some information, so not giving a complete picture or we might actually give false information. If that communication was designed to inform stakeholders of a proposed course of action, to gain their agreement, then they will have been deceived or misled by the communication, and thus may well feel aggrieved when they learn of the miscommunication, and possibly seek redress and/or seek to alter or stop the course of action.
	If our models give incorrect information, then our analysis of the impact of change or failure will be incorrect. In the case of change analysis, this will result in our estimates of the effort required being incorrect, because we failed to factor in some of the affected components, for example, additional systems that need to be updated. In the case of failure analysis, then we will not understand correctly the impact of failure, which may lead to either wasted resources by overengineering (for example) business resilience solutions, or by unforeseen risk due to lack of provision of suitable failover capabilities.

	If our models mislead us, then we may take courses of action that are less than optimal. We may choose the incorrect solution that will not give the best results, for example wasting money, wasting people's time, wasting computer resources (CPU time, disk storage etc.). We may also set out plans for substantial programmes of change that fail to achieve the stated objectives because we made assumptions that turned out not to be valid.
	If part of the pieces of the 'information jigsaw' that we are assembling are incorrect, then any models, and subsequent inferences, that we take from that set of information are going to be based upon flawed information, and thus lead to any of the above consequences

1.3.4.6 Errors of Commission and Omission

There are different kinds of errors that we could encounter with models. We accept that all models are imperfect in that they are not a complete representation of their subject. They show a subset of its attributes, hopefully relevant to the stakeholder viewing them. It is possible for the models to contain propositions q that are incorrect, that would make the model misleading.

Looking forward here to the discussion on 'understanding' in section 2.3.16 below, and repeating our discussion of propositions p on the subject S, we may consider that there are some propositions that are core to S, and others that are not core to S. It is possible for the models to omit propositions p that are central to the model, and thus would be equally misleading as one would have expected such propositions to be in the models had they been true [5]. It is also possible for the models to contain propositions that are incorrect (commission rather than omission).

By way of example, let us say that we asked someone to produce a model (diagram) that showed all the applications that used a particular type of middleware. If that person knew that the applications

---

[5] In the words of the Book of Common Prayer, "We have left undone those things which we ought to have done; And we have done those things which we ought not to have done" 42. Church of England. *Book of Common Prayer*. 1762; Available from: https://www.churchofengland.org/sites/default/files/2019-10/The%20Book%20of%20Common%20Prayer%201662.pdf.

also supported certain business processes, but did not choose to include those on the diagram, we would not consider this an omission, as they were not needed by the particular stakeholder. However, if that person showed applications that did not meet the given criteria (used a particular kind of middleware), or omitted applications that did meet the criteria, then we would consider this an error in a 'core belief' or 'core proposition' to the question at hand, and would consider the model erroneous in either case, perhaps commission (if a proposition was included that was incorrect) or omission (if a true proposition that should have been included, was not).

1.3.4.7 Other aspects of quality

We discuss the concept of quality more thoroughly in the literature survey, focusing on model quality in section 2.3.1, however apart from avoiding errors of the type "model asserts proposition q which is untrue" or "model should have but did not assert true proposition p as it was core to the model", we can consider other aspects of quality, each of which will have relevance to one of the 'layers' in the semiological ladder discussed later in section 2.3.5 below.

Consider the case of a model (diagram) that has been copied by some means (for example, a facsimile machine, or a low-resolution electronic image) such that we cannot clearly distinguish some of the features of the model. We might say that "we cannot make out" the elements on the diagram. They could be too small, or too blurred, or perhaps the difference in shading or colour between the foreground and background is too small. This would be a failure in the *empirical* layer (unable to make out the individual elements).

Alternatively, the lines in the diagram might be perfectly distinguishable, however they might not form meaningful shapes, or they might connect in ways that are confusing, or just be left unconnected. If we cannot understand what the elements of the model represent, because they do not appear to conform to any particular syntax or notation, this might be a failure in the *syntactic* layer. This particular type of quality issue is one that we include in our own research, in our construction of the syntactical quality metric Q_S.

Errors of commission and omission, as discussed above, really apply at the *semantic* layer, which is also part of our research, and related to the semantic, or truthfulness, metric Q_T.

If the person reading the model infers the wrong thing from a model, then that again is a quality issue, but the fault may not lie in the model itself. It might be that the reader did not understand the notation or did not look closely or carefully enough at the model. It might also be that the reader misunderstood the purpose of the model, or 'read something into it' that was not there and was not intended to be there. Recall our previous discussion about a model being created that did not show processes linked to applications. We were not implying that the applications did not serve any business processes but rather that the stakeholder, for whom the model was produced, was not interested in seeing the linked processes. This links to the *pragmatic* layer in Figure 28 below.

Some of the above issues may be relative to the reader of the model. If we show a model that employs quite technical notation to an IT architect, they may understand it perfectly. If we show the same model to a business stakeholder, they may only partially understand it. The quality of the model has not changed. What has changed, however, is the appropriateness of the model to the reader. Thus, we must consider, when looking at the quality, whether or not the model is appropriate to the intended readers. This is something we will cover in our research into the effectiveness of EA modelling languages.

1.4 Academic work related to model quality

1.4.1 Introduction to academic work on models

The concept of a model is dealt with comprehensively in the work of Stachowiak, relating to General Model Theory, and covered in the literature survey below. Model here is used in the more general sense that we have already discussed, rather than the more limited sense implied by (say) a diagram or a picture. This is discussed more fully in the literature survey, in section 2.2.3 below. The literature here covers the three key aspects of models: (a) they represent, or stand for, something else; (b) they are not a perfect representation of their subject; and (c) they have a purpose.

1.4.2 Introduction to academic work on model quality

There has been some work done in the area of model quality (again in the more general sense, so that it is known as ‘conceptual model quality’), discussed in 2.3.1 below. Moreover, it has been extended into a particular kind of (less abstract) model known as a requirements specification [43].

It has not however been extended specifically into the area of Enterprise Architecture, and would need to be made less abstract, more concrete, in order to make it useable in that discipline, perhaps using a similar approach to its application to software requirements.

There are also additional practical considerations that would need to be worked into such a model of EA quality. This gives rise to the first of the gaps mentioned in section 1.5 below.

1.4.3 Pointer to issue with Enterprise Architecture models

The concept of Enterprise Architecture is not new to the literature, and is discussed in section 2.2.1 below. Enterprise Architecture work is normally based around some kind of standard framework, for example TOGAF [14] and ArchiMate [15]. Such frameworks incorporate a modelling language which specifies the pieces from which we construct our models (the entities and relationships discussed for example in section 1.3.2 above). We are aware of many organisations however where models (in the sense of diagrams) presented to non-technical stakeholders do not use these standard languages (and standard notations, if the language includes such things). Thus, IT architects will often draw the diagrams at least twice: once in a 'proper' EA modelling language, and once for non-technical stakeholders. Some work has been done to understand how to use multiple models to better convey meaning in a particular language, focusing on the notation [44]. However, we have not been able to find in the literature a technique for actually measuring the effectiveness of an arbitrary modelling language. We would like such a thing, because then we could apply it to popular EA modelling languages such as ArchiMate, in an attempt to understand why architects are not using it when communicating with non-technical stakeholders, and maybe to make that continual rework unnecessary.

1.4.4 Why should we care?

There are several reasons why we should care about the quality of the modelling languages that we are using. Note, first of all, that the importance of the quality of the language used in the models is different to that of the quality of the models themselves. The implications of poor quality of the latter was already summarised in Table 2 above. When it comes to the implications of poor quality in the modelling languages, then again we can illustrate these by recalling the purposes of IT models, summarised in Table 1 above, and asking ourselves what would happen if the stakeholders reading the models were unable to understand the models.

Table 3 Consequences of hard-to-understand models

	If our models are hard to understand, then we fail to communicate what we intend. It might be that the readers of the models are given a misleading impression of the subject of the model. Even if the model is technically accurate (and would not be misleading to someone that understood the languages used on the models), then the readers may think it says something else. A perhaps more likely outcome is that the

	stakeholder will tell the person that constructed the model that it is not fit for purpose, because it is unintelligible. In a business context, this may result in an initiative being delayed or cancelled because a key stakeholder has not been convinced of the rightness of the course of action. It may have negative personal consequences for the person that produced the model, if it happens repeatedly, because the (say) business stakeholder decides that there is no point in having further discussions or consultations with someone that cannot 'speak their language'. To avoid this, people waste time and effort drawing everything (at least) twice - once for technical stakeholders and once for non-technical stakeholders.
	If the models are hard to understand, then as well as potentially causing frustration with stakeholders that struggle with them, this will slow down the analysis task and may lead to mistakes when people misinterpret them.
	If our models are hard to understand, then the stakeholders that cannot understand them may not be able to discern the key information clearly enough to take the right course of action. In other words, the case for a particular change may not be compelling, and so fail to achieve consent amongst those whose sponsorship is required.
	Poorly understood models can still be combined into a bigger picture correctly as long as they are technically correct, and so issues of comprehension and utility are less likely to have an impact in this area.

If we come to better understand which parts of our language are difficult to understand – both in terms of the underlying concepts and in terms of the associated notation – then we can avoid using the difficult concepts and notations with certain stakeholders, which will lead to increased quality of communication with key stakeholders, which in turn will lead to increased business benefit as the key benefits of Enterprise Architecture can be realised (tied to organisational motivational context).

1.4.5 Impact of not researching the quality of a set of models

Without finding a way of measuring the quality of a set models, then we have no way of answering the question “how good our are models”. This in turn means that we have no objective way of knowing whether or not the negative consequences outlined in Table 2 above are going to be realised, at a cost to the organisation.

1.4.6 Impact of not researching the effectiveness of modelling languages

Without finding a way of measuring the quality of the modelling languages that we use on the models, we are left to guess what may or not be suitable for inclusion with our stakeholders, and we rule out any possibility of automating the production of models tailored for certain stakeholders. We will not know which elements of well-known frameworks can safely be included in our models and which cannot.

1.5 The literature gap

The gaps in the literature that we intend to address in this work are:

(a) The application of the theory on conceptual model quality specifically to the discipline of Enterprise Architecture modelling, which typically (although not always) is done using two-dimensional diagrams[6], and also contains a measure of ‘expected’ or ‘sufficient’ completeness not currently present in the literature;

(b) The need to measure the effectiveness of Enterprise Architecture modelling languages, used on such models, when communicating with stakeholders who are not trained in such languages; and finally;

(c) The need to understand how the current practice of Enterprise Architecture modelling can and should be modified to take into account the needs of these stakeholders (bearing in mind what we learn from the previous two points).

[6] For an interesting extension of EA modelling into three dimensions, using Virtual Reality, see 45. Oberhauser, R. and C. Pogolski. *VR-EA: Virtual Reality Visualization of Enterprise Architecture Models with ArchiMate and BPMN*. 2019. Cham: Springer International Publishing.

1.6 Research Questions

1.6.1 Introduction to Research Questions

Organisations designing IT solutions often use Enterprise Architecture models to communicate information about the current or proposed architecture with key stakeholders, to show that their needs and concerns have been addressed, and to get their ‘buy-in’ and approval for proposed courses of action. These models can also support analysis and reporting to enable broader inferences to be drawn.

They also serve as a ‘body of knowledge’ capturing and maintaining information about the current and future state of the business and supporting Information Technology for an enterprise.

A practitioner’s paper from 1999 entitled “The Software Architect” [46] describes some of the characteristics of what we might refer to now as solution or enterprise architects. It emphasises the importance of communication to various stakeholders and asserts that the language of choice for representing the architecture is a model. It rightly states that languages (for those familiar with them) provide a rapid transfer of knowledge, amid other benefits.

Our own brief research into the reasons that organisations carry out Enterprise Architecture modelling (specifically within an EA tool), based upon a survey of fellow architects within Capgemini in 2015, suggested the following reasons why people carry out this activity:

Efficiency (faster project execution)

Accuracy (better information on which to base decisions)

Volume (handle information about more assets)

Visualisation (more understandable to stakeholders and consumers of the information)

Less business risk (expose dependencies and ambiguities)

Enabler for large-scale change (e.g. portfolio rationalisation, infrastructure migration) [47]

This information forms a body of knowledge and the models that visualise the data contained therein are used as a basis for decision-making thus there is an imperative to ensure that the models are of good quality. This raises the question of how we measure model quality.

If we make important decisions on the basis of bad models (however we define that), then we risk making poor decisions, which will result in a mis-use of resources (financial, people and otherwise), and potentially fail to achieve the desired outcomes.

What are the characteristics of a model that would lead us to consider it as fit for purpose? We might consider, for example, whether the model is true or accurate in some sense. We might consider whether it is actually intelligible to those stakeholders for whom it is intended. We could obviously ask the above questions of each of the models that we have produced; we might also consider whether we have sufficient models to cover the domain that we are trying to model, and for all the required stakeholders.

If we have a *set* of such models, purporting to describe part, or all of, our enterprise, then how would we measure the quality of the set of models? This leads to the first of our research areas:

1.6.2 Measuring the Quality of Sets of EA Models

Given the uses of models, indeed sets of models to describe part or all of an enterprise, then a natural question to ask is “how good are the models?”, or more specifically, “how would we measure the quality of a set of models?”. What kind of metrics could we use? A search of the literature returned material on general conceptual model quality (for example [48]), however Enterprise Architecture is more specific and there are expectations as to the existence of models that is not covered in the literature. Thus, an extension of the work of Lindland, Krogstie and others will provide a more comprehensive foundation to be able to answer this question.

Thus, the basis for the first research question addressed in this thesis:

RQ1. How can we measure the quality of a set of Enterprise Architecture models?

1.6.3 How Effective are EA Modelling Languages?

Part of the way we measure the quality of a model must bear in mind whether the model is intelligible to the intended consumers.

Successful communication requires that all parties involved have a common language. Two individuals, one of whom spoke only French and one of whom spoke only English, would have difficulty communicating verbally.

It is therefore important to consider, when we look at the kind of models being produced as part of Enterprise Architecture work, not just whether they are technically accurate, but also how *effective* they are in communicating the required messages to the relevant stakeholders, not all of whom will speak the ‘language’ used by enterprise architect practitioners. Enterprise Architecture languages, embodied within Enterprise Architecture frameworks, have evolved continuously; are they getting ‘better’ at communication, or are they proving a hindrance in some situations?

In the author's experience, it is very common for solution and enterprise architects to use technical languages when communicating with each other, for example on graphical models, perhaps drawn in well-defined languages such as UML [49] or ArchiMate [15], and then to use completely different models, often drawn in Microsoft PowerPoint®, to communicate that information to non-technical stakeholders. Why is this necessary? Drawing models twice not only wastes time, the fact that this is normally done manually means that there is a possibility of mistakes being made so that the two models are not consistent. This therefore means that we are giving two different, inconsistent messages to different stakeholder groups, thus (albeit unintentionally) potentially misleading some of them.

The architecture modelling languages (sometimes specified in 'entity metamodels'), as contained within architecture frameworks such as ArchiMate and TOGAF [50], evolve regularly, generally in the direction of increased complexity (see section 2.2.2.5 below), in terms of the number of different types of concepts embodied within the languages.. This complexity, along with perhaps some interesting theoretical underpinnings, might be useful to the small subset of practitioners that understand them, but does it aid the majority of our stakeholders that are not themselves IT architects? When it comes to doing this translation from technical to non-technical models (or representations), how would we know which pieces of the original technical models to carry over to the non-technical models?

These kind of considerations have led to my second research question:

RQ2. How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?

1.6.4 Tailoring EA Models to Improve Communication

The effectiveness of communication of our modelling language with particular groups of stakeholders may depend on a number of factors, which this research will identify. If we knew those factors, is there something that we could do, based upon that knowledge, to tailor our models to better suit particular types of people? In particular, is there something we can do to make the model transformation deterministic, perhaps even automated, so that our architects do not have to waste time hand-drawing models for a second (or third, or fourth…) time?

This practical question led us to our third research question:

RQ3. Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?

These three questions (measuring the quality of sets of EA models, the effectiveness of EA modelling languages, and tailoring EA models deterministically to help them communicate better) form the basis of this research.

The first question is the subject of publications in their own right (see Appendix C); the second was submitted for publication in December 2019 (see Appendix E) but the selected publisher failed to review or process the manuscript in time. The third is demonstrated empirically by the use of the results from the first two research areas in conjunction with ongoing modelling efforts within our current organisation as well as that of a EA modelling tool supplier.

1.7 About Models

1.7.1 Models as Summaries

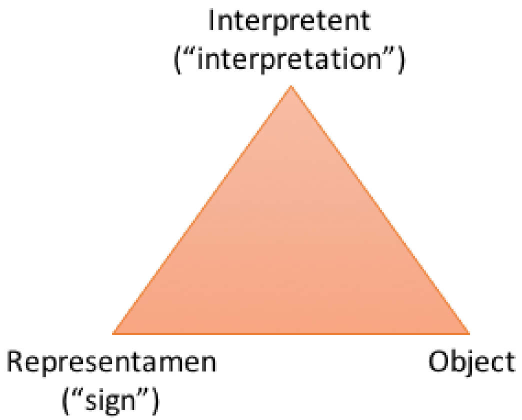
As summarised in a related paper [51], models contain a reduced and hence partial representation of the information available, designed for a particular use at particular times, perhaps to help make particular decisions. Have we used the right summaries for the right stakeholders? Can we identify kinds of information that we should leave out for certain stakeholders, but need to be included for others?

1.7.2 Models as Communication

When we talk to another person, we may speak flawless English. However, that does not guarantee that our communication is effective. Language is of course a set of signs that depend for their effectiveness in communication not just on how clearly we articulate ourselves, but also on the interpretive process executed by the listener.

Models act as a communication medium between the creator of the model and the consumer of the model. We therefore have to take into account the encoding and decoding processes inherent in communicating ideas, and these are covered in the field of semiotics. In their work "The Meaning of Meaning" [52], Ogden and Richards, building upon earlier work on "science of signs" [53] by Pierce, offer the following illustration:

Figure 6 Semiotic Triangle [52]



We can consider a model to contain one or more "signs" that are interpreted by the reader of the model to refer to certain objects. Key to this model is an understanding that the interpretation process is not executed in a vacuum; different individuals with different experiences will "read" different signs as referring to different objects. Two trivial examples are:

The words that we use (both spoken and written) are obviously culturally relative, and the same word may be interpreted differently in different cultures (even those words that are common to more than one culture), for example the word "pants" mean quite different things in American and English.

A another example is the different meaning assigned to colours by various cultures around the world, which has significance for the way that organisations manage their brands [54]. In Western culture, we might easily assume that the meaning that we assign to colours is universal, but we would be wrong.

Exactly the same concept needs to be applied to the way that we communicate concepts related to Enterprise and Solution Architecture. The lead author once worked on a project with another consultant, related to portfolio management, and it became clear very quickly that we were misunderstanding each other, because one of us had a TOGAF [29] background and one an IAF [55] background, and so when one of us discussed the idea of a business service, the other got the wrong idea (had a different object in mind, to use the semiotic terminology).

Therefore, understanding the interpretation process is crucial to the successful communication of ideas. The interpretation that we apply comes often from our surrounding culture, perhaps our family or society (in the case of colours or words), or in the case of Enterprise Architecture terminology, the prevailing nomenclature in our organisation, or the particular framework or notation that we have been taught.

1.8 Measuring the problem

We discuss here how we can measure the extent of the problem.

Although Enterprise and Solution Architecture are different disciplines, there is a lot of overlap between them, and it is frequently the case that models produced by Enterprise Architects are used by Solution Architects, and vice versa. In many cases, the languages provided by EA frameworks span the work of both Enterprise and Solution Architects. For this reason, when we refer to Enterprise Architecture models, that should be taken to include both Enterprise and Solution Architecture models; we will not continue to refer to "EA and SA models" explicitly unless the context requires it.

1.8.1 What is Quality?

We cannot discuss the quality of any aspect of Enterprise Architecture without understanding what we actually mean by quality. This topic is explored more thoroughly in the literature survey (see section 2.2.4 below; however in summary, quality has a number of possible meanings, including excellence, value, conformance to specifications, and also others relating to meeting customer expectations.

1.8.2 EA Model Quality

The motivation behind the first of our areas, the quality of a set of related EA models, came from a question posed of the author by a colleague in a consulting company, as discussed briefly in section 1.6.2 above. We had no immediate way of answering the question "how good are our EA models"; and this to us represented a gap in our knowledge that needed filling. Providing a way of measuring the quality of our set of EA models would enable us to effectively plan how to further develop them as well as giving guidelines for future projects as to how many, of what type, they should produce.

Regarding model quality, there has been some research already done in this area, specifically in the area of conceptual model quality, where the literature (specifically from Lindland [56] and later with Krogstie [48]) uses semiotic concepts are a way of considering different kinds of quality. However, there are some specific gaps we have identified in the literature, that come about when one applies this conceptual model quality theory specifically to EA model quality. Some of these gaps include the concept of where or not a model actually exists (in EA we might expect to see a model but it might not be there); the concept of evolving maturity of modelling; and the notion of applying calculus to model quality to find conditions for increasing quality over time. These gaps remain to be answered, and our intention is to do so in this thesis.

1.8.3 EA Modelling Language Comprehension

Our experience in consulting, using UML, is that people who are non-technical tend to react against technical diagrams such as those contained within the UML standard.

IT staff at a major UK airport have told us that they do their EA modelling in UML, using a particular modelling tool, and that they then redraw them all completely differently in order to communicate with their non-technical stakeholders. This indicates a potential issue of the acceptability of UML to non-technical stakeholders.

We know from practical experience at our current organisation that architects produce architecture diagrams in ArchiMate, using an EA modelling tool, but never present those diagrams to business stakeholders; they always produce diagrams in office tools such as Microsoft PowerPoint®, that look nothing like ArchiMate.

In an initial pilot group interview that we ran with masters-level students that were meant to be somewhat familiar with Enterprise Architecture, when testing the comprehension of some ArchiMate [39] models, there was an interesting reaction against some elements of the language that led us to conclude that the background experience of those viewing the models had a strong correlation with their understanding of specific elements of it. For example, one student that had a project management background said that all of the symbols that denoted some kind of 'service' had no meaning for her. Others had issues with the wide variety of arrows used in this particular language (finding them confusing).

Although not evidenced strongly, a recent paper [57] suggests that there is an issue with models that have not been produced as widely as they might (hence questioning their value) as well as highlighting the fact that although a language may permit many different constructions, not all of them are sensible and of use in communicating with those that need to know. This is an issue both for model quality (previous topic) as well as for the comprehensibility of those models.

1.9 Benefits of Research

1.9.1 Benefits of Measuring EA Model Quality

Understanding all the aspects of EA model quality will enable us to understand objectively how to answer questions about the quality of our models, by constructing metrics that can be measured objectively and tracked over time. This will also allow us to understand better what must be done to improve those models. This in turn will lead to better decision making, where those models are used as inputs to decisions. This is discussed, for example, in Shanks [58], where a particular technique (application of ontological theories) is suggested for information modelling with stakeholders. An example of the practical outworking of this, would be a situation with one of our employers where there was a need to remove a particular kind of middleware [59] because of its high operational cost. One thing that an Enterprise Architecture model would contain (in the more general conceptual sense of the word, not necessarily in a diagram) is a representation of which applications communicate with which other applications, and the mechanism for that communication (for example, via some specified middleware, or some kind of File Transfer Protocol. If the model is inaccurate (for example, does not contain a representation of all the information flows) then the consequences could be serious. Firstly, we would underestimate the effort required to migrate onto a different middleware technology. Secondly, we might turn off the unwanted middleware and then discovered that there were important applications that relied upon it and weren't in our plan because our model of our application infrastructure was incomplete. Therefore, we need to ensure that our models are accurate in order to avoid such issues arising in the first place.

1.9.2 Benefits of Understanding Model Comprehension

Understanding how to tailor our models to better meet the needs of the various stakeholders with whom they are intended to communicate, will enable practitioners to focus their efforts producing the right kind of models, and at the same time avoid the production of unnecessarily complex models. The communication will be more effective (because we do not include aspects in our models that will not be understood by, or even confuse, our readers of those models). This will lead to a better shared understanding between the author and reader of the models.

1.9.3 Benefits of Deterministic Algorithms for Improving Models

If we can produce a set of 'rules' specifying how to adapt models for specific types of stakeholders, then that has two benefits. Firstly, those can then be specified to those people producing models as guidelines or standard procedures, thus leading to more consistency in the models produced. Secondly, there is a possibility that those rules, or algorithms, can actually be implemented in software, thus removing some or all of the manual effort required to tailor models in this way, thus saving time and money and increasing quality (because any manual work holds the potential for inadvertent errors).

1.10 Objectives and Deliverables

In this section we summarise the aims of our research and lay out what we expect to achieve during the research, as well as specific deliverables.

1.10.1 Notation for Research Structure

Recall that the three research questions addressed in this thesis are:

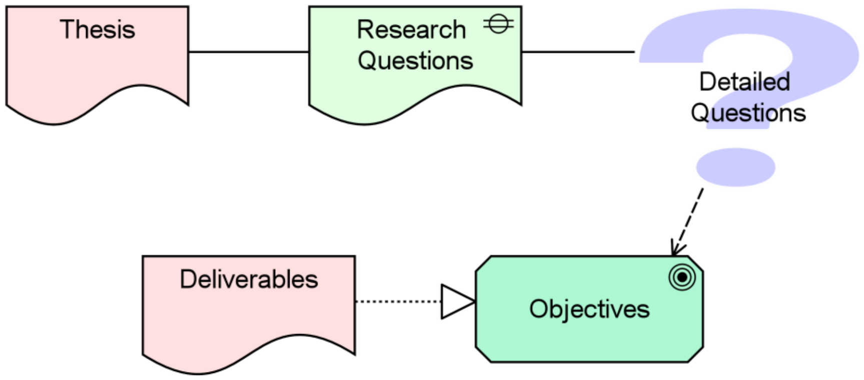
RQ1 *How can we measure the quality of a set of Enterprise Architecture models?*

RQ2 *How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?*

RQ3 *Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?*

Each of these research questions has been broken down into a small number of component (detailed) questions, related to which are a number of objectives and deliverables. We have modelled these in ArchiMate (one of the languages mentioned previously, and which will be subject to a lot more detailed examination in this thesis), along with the relationships between them; and will include the fragments of the overall "thesis model" as appropriate throughout. In other words, we have constructed our own "model" of the thesis structure. The following figure illustrates the 'metamodel' of these various concepts:

Figure 7 Metamodel for Thesis Research Structure

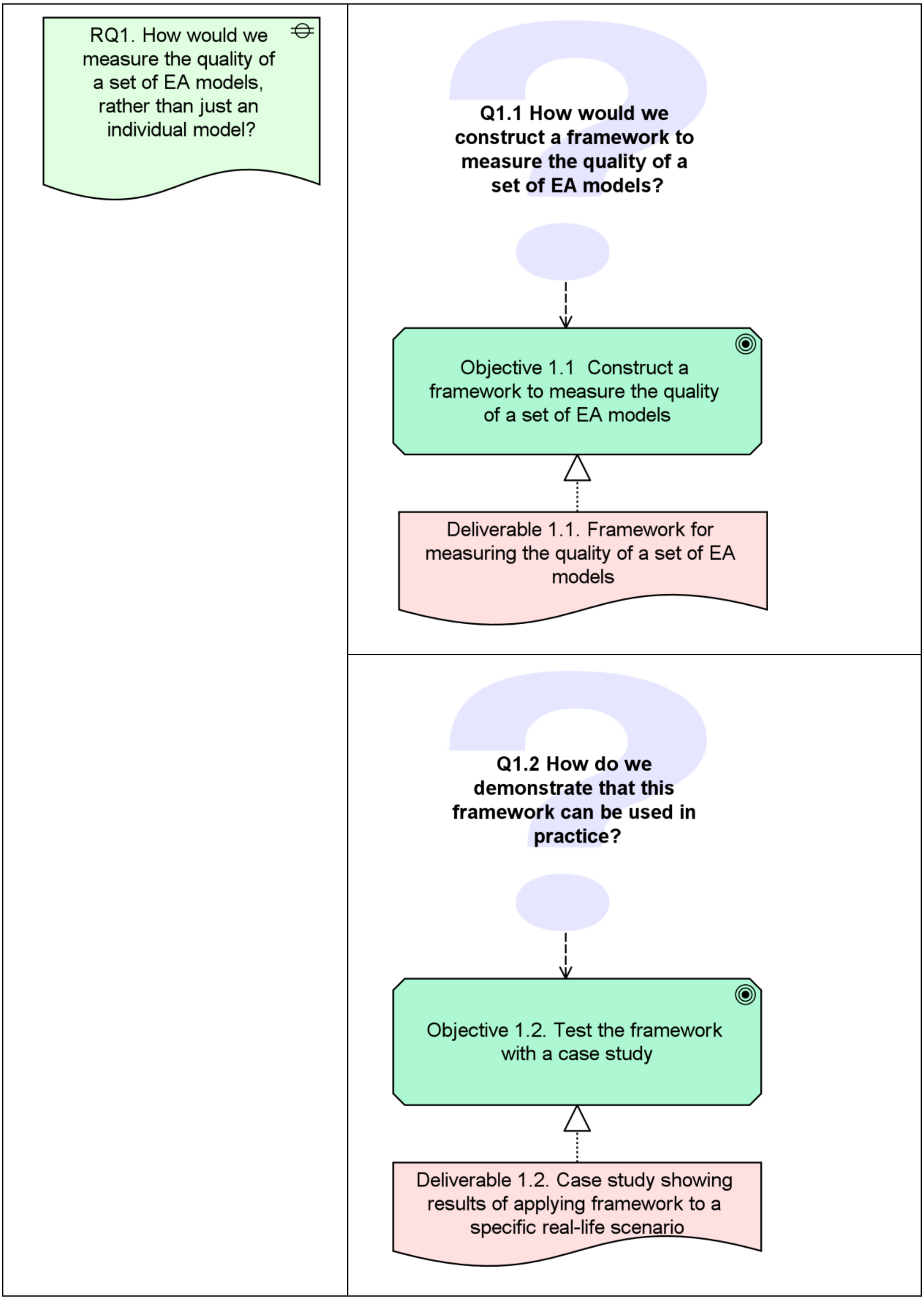


This metamodel tells us that thesis addresses a number of research questions, each of which relates to some detailed questions. Each of these in turn has one or more objectives, which are linked to deliverables that, when produced, will demonstrate that the objectives have been met.

1.10.2 Research related to RQ1

The detailed questions related to RQ1 are shown below, along with their linked objectives and deliverables:

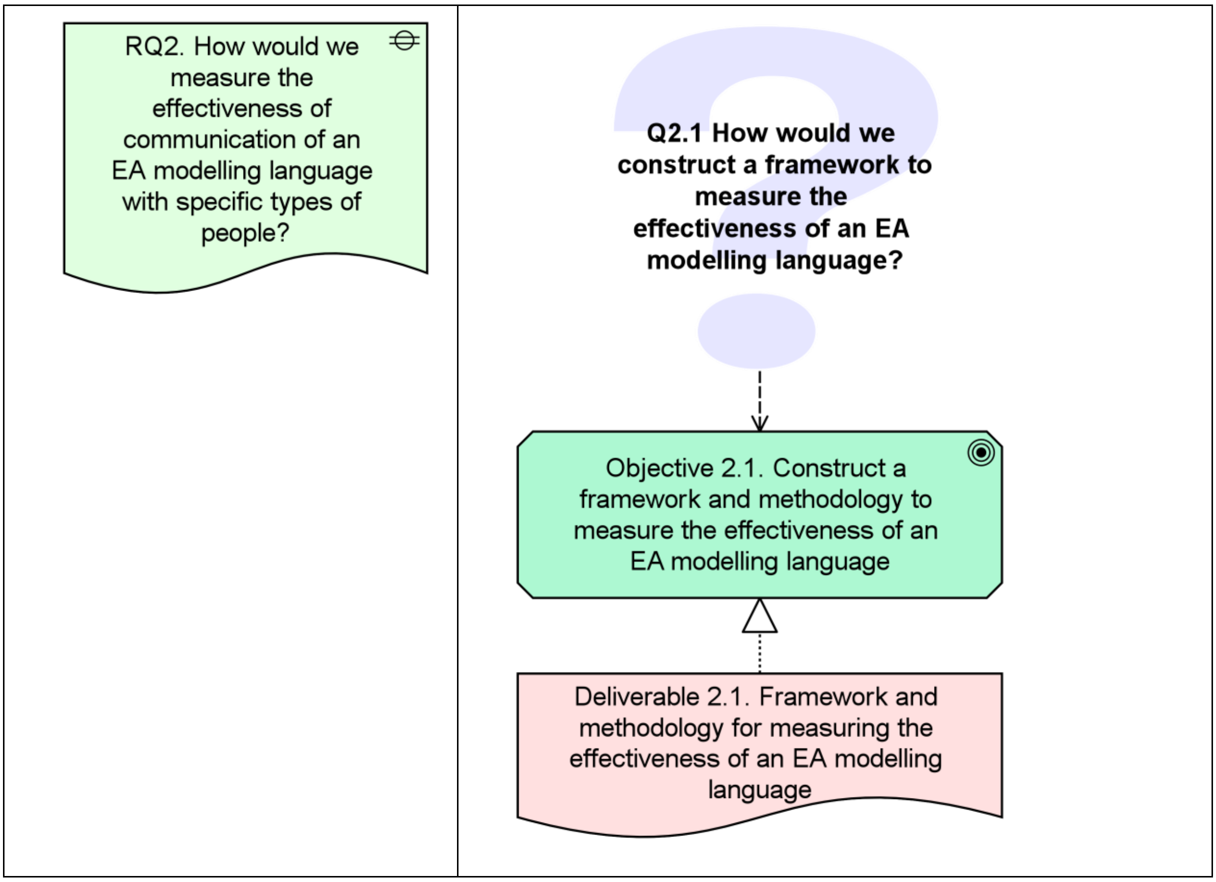
Figure 8 Detailed questions, objectives and deliverables related to RQ1

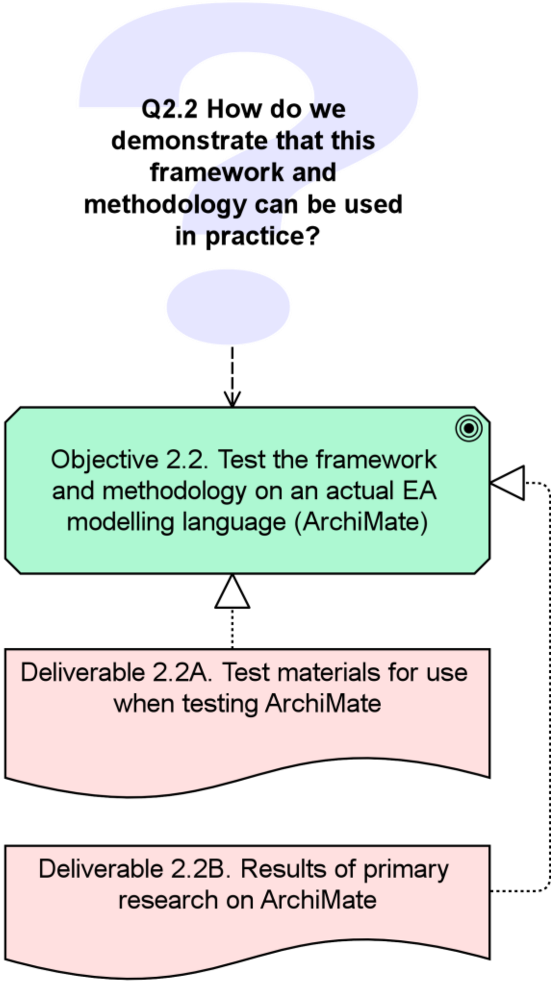


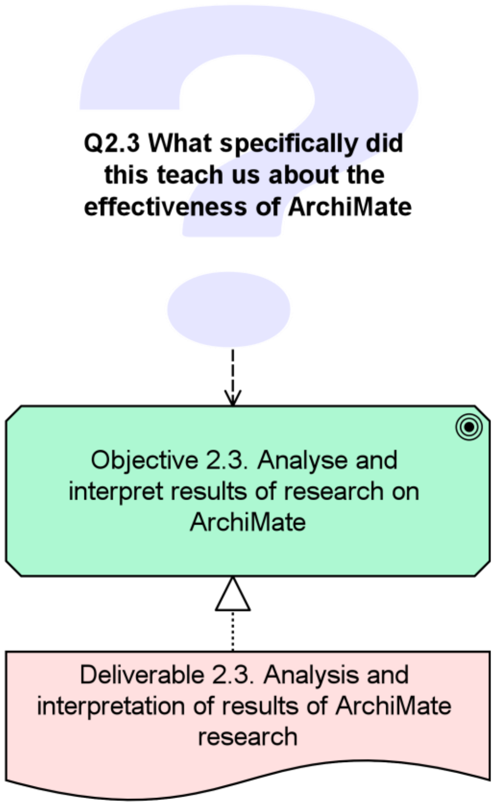
1.10.3 Research related to RQ2

The detailed questions related to RQ2 are shown below, along with their linked objectives and deliverables:

Figure 9 Detailed questions, objectives and deliverables related to RQ2



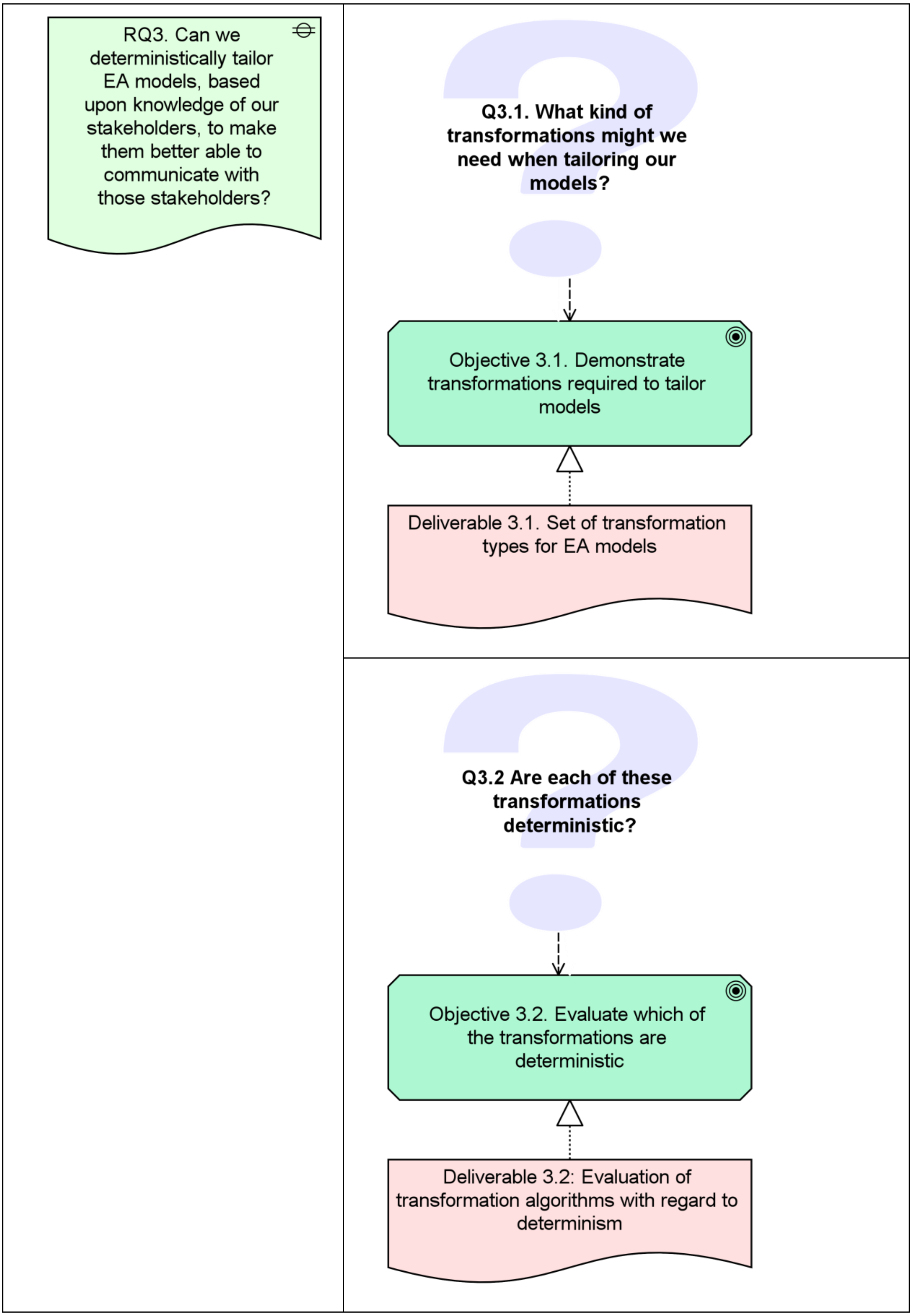
	 Q2.2 How do we demonstrate that this framework and methodology can be used in practice? Objective 2.2. Test the framework and methodology on an actual EA modelling language (ArchiMate) Deliverable 2.2A. Test materials for use when testing ArchiMate Deliverable 2.2B. Results of primary research on ArchiMate

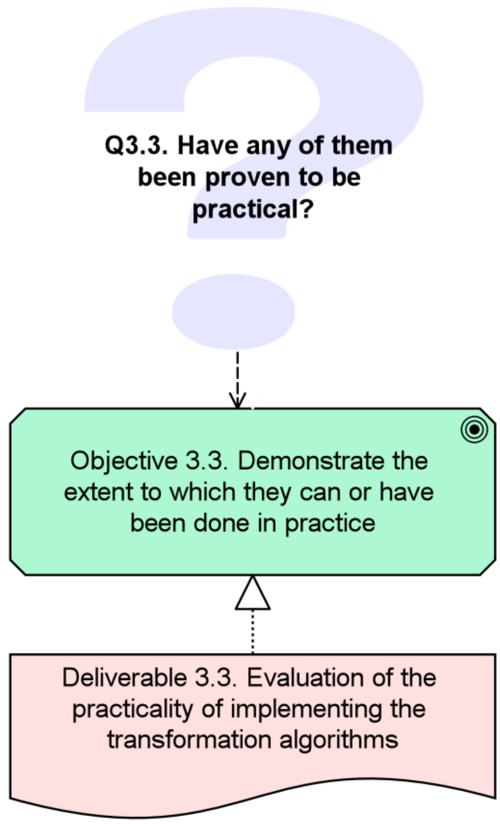


1.10.4 Research related to RQ3

The detailed questions related to RQ3 are shown below, along with their linked objectives and deliverables:

Figure 10 Detailed questions, objectives and deliverables related to RQ3





1.11 Layout of Thesis

1.11.1 Content of Chapters

This chapter (Introduction) gives an introduction to the topic of Enterprise Architecture Models, and to our research. It gives context to, and contains, our three main research questions. It also outlines the structure of the research, giving a list of main and supplementary questions and detailing the various objectives and deliverables that address them.

Chapter 2 (Literature Review) examines the current literature in our research area, highlighting the work that has already been done, establishing the foundation for our own research as well as showing the gap that we intend to fill.

Chapter 3 (Methodology) discusses that way that we approach the research: the underlying frameworks and methodologies that have shaped the way that we have done the research. It also details the theories of knowing (epistemology) that we will be using, our data collection approach, and a discussion of design science and its relevance to our research.

Chapter 4 (Theoretical Approach: Measuring the Quality of a Set of EA Models) contains the first piece of new research within this thesis, addressing the first of the research questions from the introduction. It builds upon existing conceptual model quality theory and provides a new theoretical framework for measuring the quality of a set of EA models.

Chapter 5 (Case Study: Measuring the Quality of a Set of EA Models) tests the framework established in the previous chapter in a case study and demonstrates that at least two of the three quality metrics in the theoretical framework can be used in practice.

Chapter 6 (Theoretical Approach: Measuring the Effectiveness of an EA Modelling Language) addresses the second research question by providing another theoretical approach, this time building upon semiotic theory and the idea of a generic enterprise architecture framework to construct a method for measuring how well understood, and useful, the various concepts in such a framework are to people that might view EA models.

Chapter 7 (Survey: Measuring the Effectiveness of an EA Modelling Language) tests the framework proposed in Chapter 6 in another case study that tests part of the ArchiMate framework.

Chapter 8 (EA Model Evolution) considers some of the conceptual transformations that we would expect to have to make, in the light of the various purposes for modelling established in the introduction, as well as the learnings from the previous research chapters, to our set of EA models and thus addresses our third and final research question. After a brief explanation of EA modelling tools, it then demonstrates that these transformations are actually achievable in practice, by means of examples of the results obtained by such tools.

Chapter 9 (Summary) recapitulates the results of our research, showing how we have filled the current gap in the literature as well as pointing the way toward further research needed.

1.11.2 Content of Appendices

The Appendices contain supplementary material that may be helpful if the reader wishes more detail in some of the areas referenced from the main body of the thesis.

Their contents are as follows:

Appendix A (Architecture Frameworks)

Found here are some technical diagrams from three popular Enterprise Architecture frameworks (TOGAF 9.1 [29], Zachman [60], ArchiMate 3.1 [15]) that illustrate the scope of these frameworks.

Appendix B (Objectives and Deliverables)

This contains a summary of the objectives and deliverables contained with this thesis.

Appendix C (Ethics Approval Documentation)

Contained in this appendix is the documentation that was filled out and agreed before commencing the interviews for the research on modelling language effectiveness, as well as some notes on some slight discrepancies between what was planned for and what actually happened, regarding the interview process.

Appendix D (Publication: EA Model Quality)

A shorter form of the research on the quality of sets of EA models was published in 2016 [61]; this appendix contains a reformatted version of that shorter publication.

Appendix E (Pending Publication: EA Language Quality)

A shorter form of the research on EA modelling languages was submitted to the Journal of Enterprise Architecture in December 2019 for peer review prior to publication.

Appendix F (ArchiMate Viewpoints)

This appendix contains a list of the standard ArchiMate viewpoints (types of view) along with their suggested content. It also shows how the ArchiSurance [62] case study views were taken as source material for inclusion in our own test models.

Appendix G (Questionnaire Used with EA Modelling Language Effectiveness Survey)

Contained here is the blank questionnaire used to capture results from all of the interviews related to EA modelling language effectiveness.

Appendix H (Completed Questionnaires)

This contains all the questionnaires filled out at the time of the interviews.

Appendix I (Constructing the Test Models)

This appendix contains details of how the test materials were constructed, starting with the original ArchiSurance [62] case study material and supplementing with our own original data.

Appendix J (Test Models for Modelling Language Effectiveness)

All the test materials (diagrams, diagram keys, narratives etc.) are contained in this appendix.

Appendix K (Detailed Quantitative Results from Interviews)

In here is contained all of the raw quantitative data collected from the interviews.

Appendix L (Underlying Formulae for Analysis of EA Modelling Language Effectiveness)

The analysis of the data from the EA modelling language effectiveness research relies upon an Excel® spreadsheet containing a large number of formulae. This appendix contains details of all those formulae.

Appendix M (Statistical Treatment of Differential Results)

This contains details of the statistical analysis used to investigate the significance of the differential results obtained from our survey, related to the use of ArchiMate notation on the comprehension and understanding (utility) of ArchiMate concepts.

Appendix N (References)

Our final appendix is a complete list of references used in this thesis.

Chapter 2. Literature Review

2.1 Introduction to the literature review

When considering the literature in relation to this thesis, we need to explore a number of concepts related to the phrases "Enterprise Architecture", "Model" and "Tool". The definitions of these, and research associated with them, will serve as a foundation for our own research. We used a multi-faceted approach to survey the literature which is reflected in the structure of the chapter.

Section 2.2 locates specific terms required in our subject definition (Enterprise Architecture) within the literature, for example "Enterprise", "Enterprise Architecture", "Enterprise Architecture Frameworks", "Models", "Quality", "Tools". It also includes a systematic review of high-impact Information Systems (IS) journals, searching for specifically for research related to model quality, or to understanding of models. This was not done by a search engine; instead, we downloaded certain historic publications going back ten years and check their tables of contents and if necessary abstracts. This review of the high-impact IS journals was based upon the "Senior Scholar's' Basket of Journals [63], and returned fewer matching papers than we had expected.

Section 2.3 discusses specific theories that form the foundation for our own research. These theories were taken from a broader search (using Google Scholar [64]) and not limiting our search to the fixed set of IS journals used previously) against terms in which we were specifically interested; this returned many more relevant pieces of research that were not found in the 'manual' search of the high-impact IS journals.

Section 2.3.17 examines some research from publications known to publish in our specific field, and in which we have ourselves published previously.

Section 2.5 contains some additional background information on ArchiMate, the EA modelling language that plays a major part in our research on the effectiveness of modelling languages.

Section 2.6 outlines a specific difficulty that we have encountered when trying to complete the literature search, and this chapter closes with a review, in section 2.7 below, of what was found in the Literature Survey.

2.2 Specific Review of Terms

2.2.1 About Enterprise Architecture

We start our literature review by looking at the role of IT within a business, and the way that the business and IT infrastructure are related.

2.2.1.1 What is an Enterprise?

A traditional definition of Enterprise would typically focus on a single organisation, for example:

an organization, especially a business, or a difficult and important plan, especially one that will earn money [65]

TOGAF's definition is broader than this, defining it as:

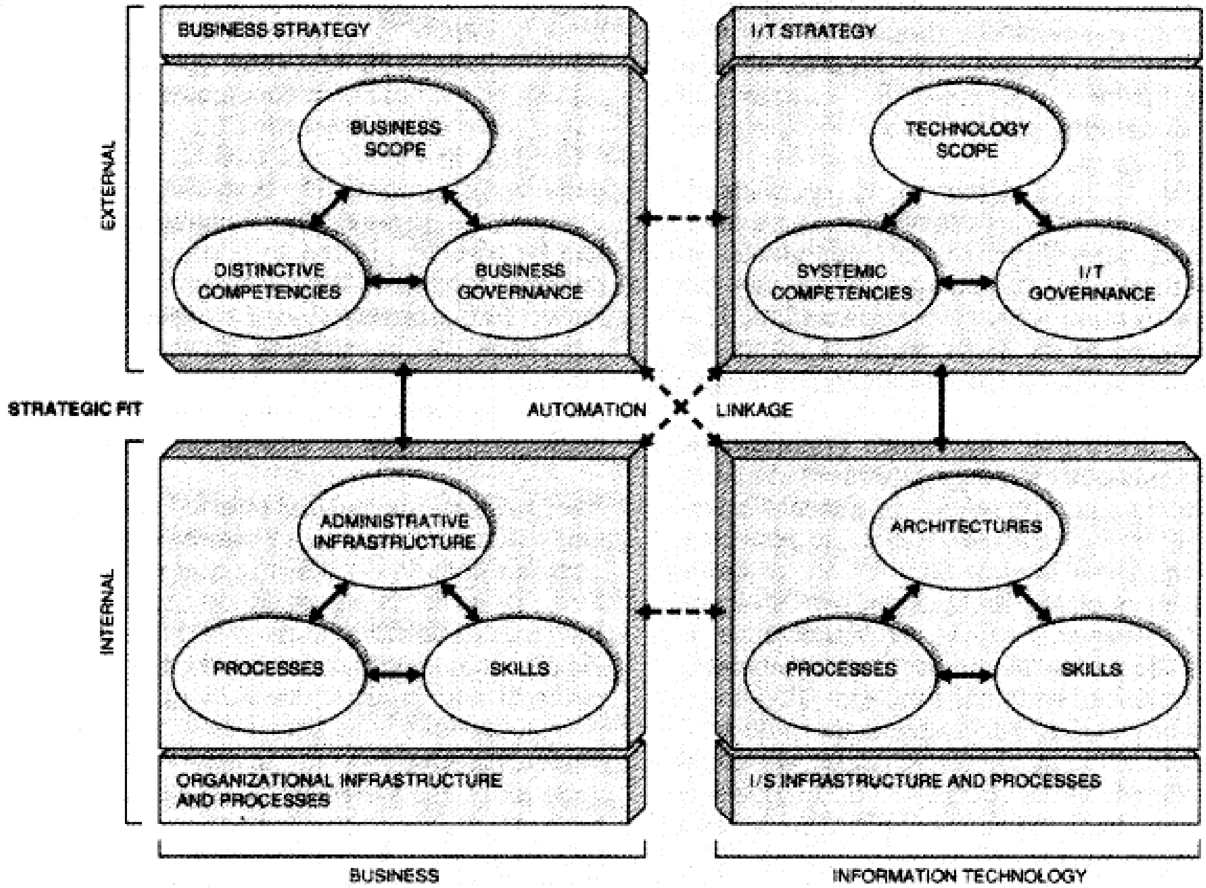
any collection of organisations that has a common set of goals [29]

The reason for this is that organisations with common goals are likely to co-operate in achieving those goals, and thus will be mindful of each other when designing their systems (in the broadest sense of the word). An example of this would be the different branches of the UK armed forces (British Army, Royal Navy, Royal Air Force) who have independent strategies, plans and structures but co-operate in the interests of the United Kingdom and its allies.

2.2.1.2 Strategic Alignment Model

The Strategic Alignment model [66] describes two aspects to each of the business and IT domains: a strategic (planning) aspect and a tactical (operational) aspect, incorporating them into the well-known model:

Figure 11 Strategic Alignment Model [66]



The linkages we are talking about are between the elements of the strategic (upper row) and the tactical (lower row) and between elements of the business (left column) and IT (right column). The paper containing this model describes how to leverage IT for the purpose of transforming organisations, and this gives something of a clue as to the role of Enterprise Architecture, which is a development of this basic model, dealing, as it does, with the components comprising a business and its supporting IT, or as the ISO/IEE/IEEE standard would describe it,

> *[the] fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution [33]*

There is an expectation here, both in the title 'transforming organisations', which is also echoed in the reference to 'design and evolution' in the ISO standard, that we are dealing with a dynamic rather than a static situation. If the business that we are considering is not changing, or changing very little, then there may be limited value in the techniques described herein.

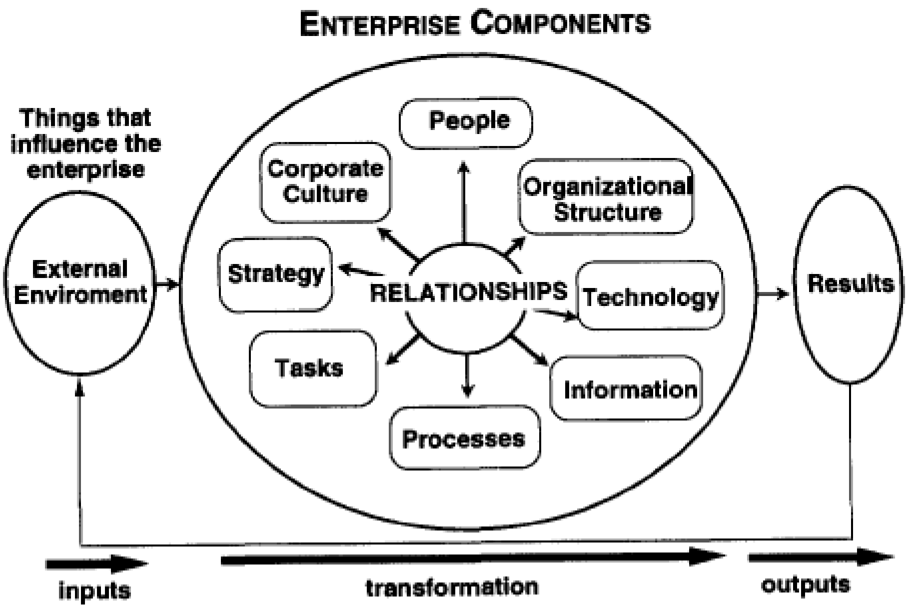
2.2.1.3 Enterprise Architecture and Systems

A system can be considered a set of artefacts or an organisation serving a common purpose [67]. The Architecture of a system describes the constituent parts of a system, the way those parts are used together and the principles governing the ongoing design and evolution of that system [33]. Enterprise Architecture applies this to a system that is a complete Enterprise, in other words something that consists of people, information and technologies, which performs business functions, and has a number of characteristics summarised in an early definition of Enterprise Architecture [68]. Enterprise Architecture therefore deals with the totality of the business and technical infrastructures, as well as the business and technical strategy, and the alignment between them as described in the Strategic Alignment Model [66]. The term Solution Architecture deals with the same layers (i.e. business and technical information) but in the context of a single programme or project rather than the whole enterprise.

2.2.1.4 The Purpose of Enterprise Architecture

An EA Framework (see section 2.2.2 below) comprises many features which serve different needs and purposes. This is a summary of the key needs satisfied by Enterprise Architecture. Rood describes the following model in her 1994 paper:

Figure 12 Rood's Generic Enterprise Architecture [16]



Rood suggests that an EA contains organisational-specific descriptions of each of the items in the central 'bubble', and that an EA:

Is a basis for decision making and planning;

Governs the identification, selection, and development of standards;

Is the mechanism for managing change within the enterprise; and

Enables effective communication about the enterprise [16]

In 1994, the Standish Group produced their famous "Chaos Report" [34] into the causes of project failures, aiming to identify (a) how many software projects fail, (b) the major factors causing failure and (c) what can be done to reduce these failures. Key results from this publication (based upon 365 respondents and 8,380 IT applications), relevant to this thesis, include the following:

Project Resolution Types

16.2% were completed on time, on budget, with the specified functionality;

52.7% were completed late, over budget, and with less than the specified functionality;

31.3% were cancelled at some stage in the development lifecycle

[34]

The three major reasons given by the respondents as to why projects succeed were "user involvement, executive management support and a clear statement of requirements" [34].

Some of the interesting comments made in this paper, reflecting on the reasons for project failure, include:

- The difficulty in getting multiple stakeholders to agree on a set of rules, to get their buy-in for a solution that is best for the organisation, even if it is not best for them;
- Competing priorities causing resources to be redirected away from the project;
- Unclear objectives

It should be noted that subsequent research (e.g. [69, 70] has cast doubt on the results of the original paper, in particular the percentages of projects categorised as failing, partly due to possible selection bias during the survey for the primary research, however it does not remove the significance of the above three factors in correlation with project success or failure.

The 2008 publication "Creating Value by Informed Governance" [9] reflects a number of Rood's themes; in section 2.2.1 of the book we see a discussion about how enterprises find themselves unable to react quickly enough to changes in their environment (in other words, a lack of agility), due to many factors including a lack of understanding of their own portfolio of information and systems (being uninformed), duplication in information and systems, and siloed data that is not shared effectively across systems that need it.

2.2.1.5 Types of Architecture

The term 'Architecture' in the IT domain is often prefixed with terms such as 'Business', 'Information', 'Technical', 'Solution' and 'Enterprise'. We consider here the meaning of the various types of IT-related architecture.

2.2.1.5.1 IT Architecture

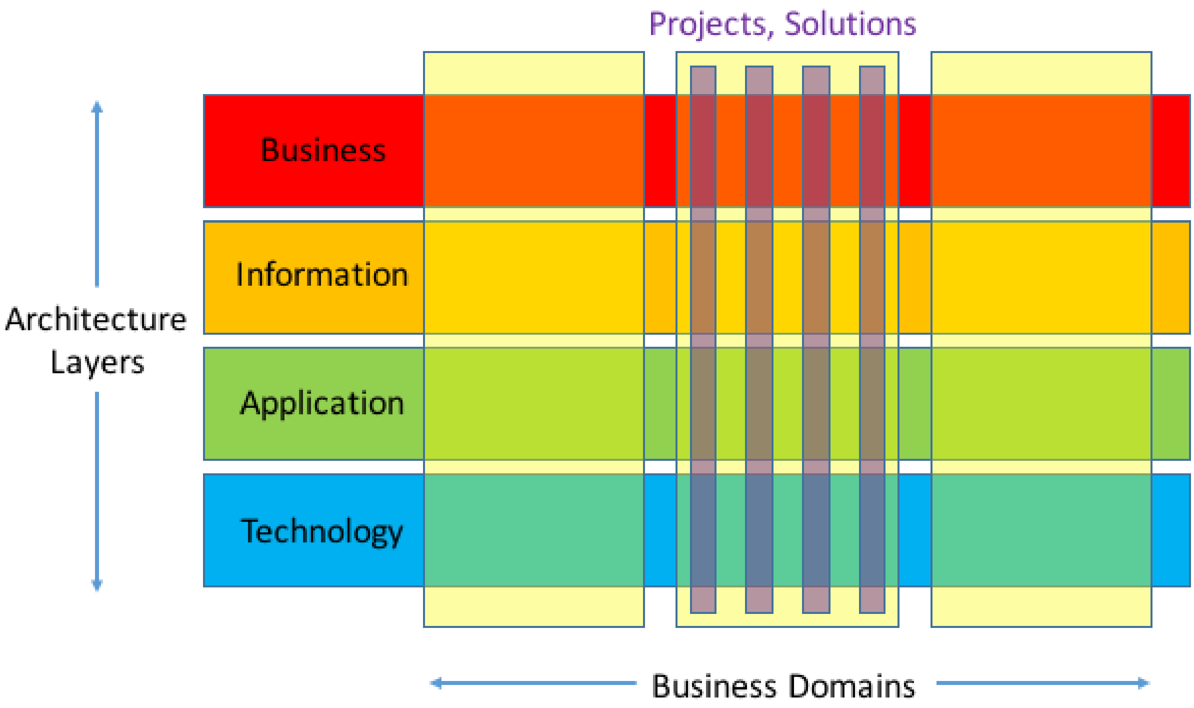
First we note that the term 'Architecture' is actually a protected term, at least in the UK, by the Royal Institute of British Architects [1], referring to of course the physical architecture of buildings and other large-scale constructions, and so to distinguish architecture in the IT sense from the buildings sense, the term "IT Architecture" is used. Thus, we now have the concept of "IT Architect" certifications, which is a little unfortunate as this tends to focus on the IT elements of Enterprise Architecture, which encompass more than just IT (i.e. it also includes the business). We now consider the different types of architecture that fall under the banner of IT Architecture.

In her 1994 paper, Rood [16] suggested that an Enterprise comprises people, information and technology, and that consequently an Enterprise Architecture must reflect this. Subsequent frameworks (including TOGAF [14] and IAF [55]) have tended to standardise on four major divisions or layers: Business, Data, Applications[7] and Technology (worded slightly differently). ArchiMate [71] now has more layers, but still has Business, Applications and Technology in common with TOGAF and IAF. If we consider that an organisation[8] may have major subdivisions as well (e.g. supply chain, manufacturing, retail, marketing etc., specific to a particular industry sector), then we can conceive of the following model to show the various types of IT Architecture:

[7] Referred to as Information Systems both by TOGAF and Rood.

[8] Or possibly multiple co-operating organisations, given the TOGAF definition of Enterprise discussed earlier

Figure 13 Architecture Typology



The four thin vertical strips within the middle business domain represent specific projects or solutions being executed within that business domain, for example, a project to implement an ERP system within the manufacturing domain.

We can use this model to illustrate what is meant by each of the following types of architecture:

2.2.1.5.2 Business Architecture

This deals with architecture work purely in the top layer, so has three possible scopes: (a) the whole enterprise, (b) a particular domain or (c) a specific project.

2.2.1.5.3 Information Architecture

This deals with architecture work purely in the second layer, again with three possible scopes (enterprise, domain or project).

2.2.1.5.4 Application Architecture

This deals with architecture work purely in the third layer, again with three possible scopes.

2.2.1.5.5 Technology Architecture

Unsurprisingly, this is the same but for the fourth, lower layer.

2.2.1.5.6 Solutions Architecture

This takes a vertical ‘slice’ up through the layers in the context of a specific project, and so will encompass elements of all four layers. The Solution Architect will be concerned with the successful delivery of that specific project, along with conformance to the scope and guidelines agreed when it was launched (including enterprise governance work products from Enterprise Architecture)

2.2.1.5.7 Domain Architecture

This considers the whole of a particular business domain and all the projects executed within that domain. Like Solutions Architecture and Enterprise Architecture, it has an interest in all four layers. Not all organisations recognise the need for a Domain Architecture.

2.2.1.5.8 Enterprise Architecture

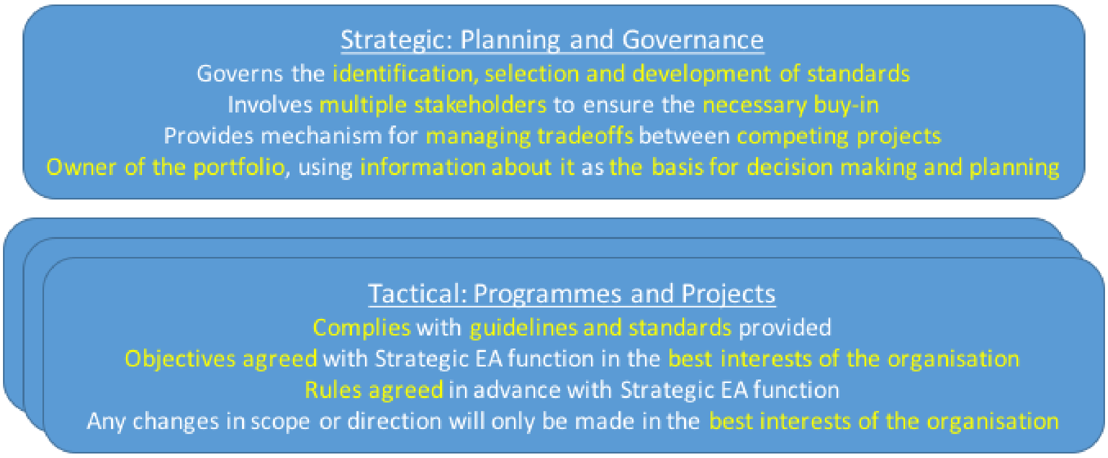
This has the widest scope of all the types of IT Architecture, having an interest in all four layers and across all business domains. However, whereas the emphasis for Solutions Architecture (“SA”) is the execution of a single project, the emphasis for EA is determining the right projects to execute and ensuring that they comply with the enterprise vision and strategy. EA therefore deals at a strategic governance level and SA deals with a tactical execution level. Domain Architecture, where used, represents something of a hybrid of the two.

Drawing upon these, we can formulate our own model of the value of Enterprise Architecture, starting with the following graphic. We are making a distinction between the strategic level, where we determine what projects or programmes an organisation should embark upon, and the tactical level which looks at the execution of an individual project or programme.

Figure 14 Strategic and Tactical Architecture

We can overlay on this model the various statements made in the literature about the value and purpose of EA to see how EA works at both the strategic and tactical level to protect tactical (delivery) projects:

Figure 15 Strategic and Tactical Application of Enterprise Architecture



Enterprise Architecture works at the strategic level (cf. the Strategic Alignment Model [66]) to set the proper scope for individual projects and deal effectively with the "politics" and trade-offs that may be required, in the best interests of the organisation as a whole. At the tactical level, the project can proceed knowing that its scope has been agreed (at a high level) with all necessary stakeholders, and also with confidence that any necessary architectural choices (e.g. the selection of a new identity management component) will be specified in advance so as to keep the overall IT portfolio moving towards what is seen as the ideal "future state".

So to summarise, in a nutshell, the value of Enterprise Architecture is:

(a) To ensure that the business executes the correct programmes and projects to maximise the value to the business (effectiveness, or "doing the right thing"); and

(b) To maximise the chances of those programmes and projects being delivered successfully (efficiency, or "doing the thing right").

2.2.2 About Architecture Frameworks

All Architecture Frameworks do not cover the same ground. There are many different aspects to an Architecture Framework. Some of these are described below. Also shown below is a diagram that summarises the main differences (see Table 4 on page 83).

2.2.2.1 Standard Vocabulary

In practice, to describe the contents of any of the ellipses in the Strategic Alignment Model (e.g. “architectures” or “administrative infrastructure”, we will need a much more extensive vocabulary. For example, when describing “administrative infrastructure”, we are likely to want to describe the organisational structure, and perhaps, also the services provided by the organisation. Part of the purpose of an architecture framework is to provide such a vocabulary. Examples of such frameworks include The Open Group’s Architecture Framework (TOGAF) [29, 31], Capgemini’s proprietary Integrated Architecture Framework (IAF) [72], Zachman’s Enterprise Architecture Framework [73, 74], the US Department of Defense Architecture Framework (DoDAF) [75], the UK’s equivalent, Ministry of Defence Architecture Framework (MoDAF) [76] and the NATO Architecture Framework (NAF) [77]. The TOGAF vocabulary can be seen in Figure 123 below, within Appendix A. These vocabularies, which describe a set of “entities” to be used when describing architectures, also provide suggested relationships between the entities. As an example, we might expect to see a link between an IT application and the service(s) provided by that application. The set of allowable entities plus their relationships constitutes an “entity metamodel”. Different frameworks serve different needs, and so even within the entity metamodels (leaving aside the other content discussed below), we see different emphases. For example, the defence-oriented frameworks (DoDAF, MoDAF, NAF) deal heavily with capabilities – what kind of military missions could be undertaken – and de-emphasise the technical infrastructure, considering only what technical services are required and not concerning themselves with how those services are provided.

2.2.2.2 Reference Material

We can visualise, then, a framework providing a standard vocabulary and our architecture descriptions drawing upon lists or catalogues of entities, for example “the catalogue of all business services” or “the catalogue of all infrastructure components” or whatever vocabulary we choose to adopt.

Some frameworks, for example TOGAF, go further than this and suggest a set of standard entries to go into certain of the catalogues. So, TOGAF has a Technical Reference Model that provides a standard way of describing some technical aspects of an architecture, and as can be seen from Figure 126 below, a suggested set of services (e.g. security, transaction processing) that might appear in an “application platform” catalogue.

2.2.2.3 Development Process

Another strength, in particular, of TOGAF is the consideration that it gives to the process of developing and modifying an architecture. This is encapsulated in its Architecture Development Method, reproduced in Figure 124 below.

2.2.2.4 Reference (Comparison) Frameworks

There are two conceptual frameworks that provide a theoretical basis on which to compare architecture frameworks. The older of these (development of which started in 1995), the Generic Enterprise Reference Architecture and Methodology (GERAM) [78, 79] was itself built upon a number of earlier reference frameworks (CIMOSA [80, 81] for manufacturing industries, GRAI-GIM [82] and PERA [83]. These earlier frameworks were described as "Enterprise Integration Architectures", and GERAM itself also had this integration "flavour". GERAM contains elements of vocabulary, process and reference material spread across its six major components

- Generic Enterprise Reference Architecture
- Generic Enterprise Engineering Methodology (cf. TOGAF ADM)
- Generic Enterprise Modelling Tools and Languages
- Generic Enterprise Models
- Generic Enterprise Modules
- General Enterprise Theories

A number of architecture frameworks have since been compared against GERAM (for example TOGAF [84], and several, including the three used in the development of GERAM and Zachman, mapped together [85]).

More recently, in 2009 another conceptual framework for categorising EA frameworks was published, the EA Framework Framework (EAF^2) , designed " *… to span the space of different EA frameworks in use today*" [86] by looking at the more modern EA frameworks introduced since GERAM. The publication included a table showing how a variety of EA frameworks compared against the domains or areas described within EAF^2:

Table 4 EAF^2 Classification of Frameworks [86]

Metamodel / Framework	TOGAF	DODAF	MODAF	E2AF	FEA	Zachman	Archimate
1. Architecture Governance							
1.1 Architecture Development Process	■	■	■		■		⁄
1.2 Architecture Maintenance Process	■				■		
1.3 Architecture Guidelines/ Principles	■	⁄			■		
1.3.1 Building blocks	⁄				■		
1.3.2 Patterns	⁄						
1.4 Architecture Roles/Skills	■				■		
1.5 Architecture Maturity Model	⁄				⁄		
1.6 Architecture Compliance Guideline & Review Process	■						
2. Modeling Concepts							
2.1 Model Taxonomy	■			■		■	
2.2 Reference Model	■				■		
2.3 Metamodel		■	■				■
2.3.1 Entity Type		■	■				■
2.3.1.1 Attribute Type		■	■				
2.3.2 Relationship Type		■	■				■
2.3.3 Viewpoint	⁄	■	■	⁄		⁄	■

■ The EAF^2 concept considered is presented and detailed.

⁄ The EAF^2 concept considered is mentioned and discussed.

☐ The EAF^2 concept considered is not mentioned.

This is very helpful, because we can now see clearly how the various modern frameworks compare, and how they might be used to complement one another. For example, the Zachman framework contains a metamodel (see Figure 127 on page 348) but no development process and so it would be possible to use TOGAF's ADM as the development process to populate the entities and relationship from Zachman. It should be noted in this context that the EAF^2 paper was published in 2009 when TOGAF was at revision 8.1.1 [30], and that a subsequent revisions to TOGAF, release 9 [31], added a full content metamodel, adapted from Capgemini's Integrated Architecture Framework [55]. Thus, if Table 4 above were to be redrawn subsequent to TOGAF 9, then arguably most of the 'blanks' would have been filled in in the TOGAF column, under Section 2, Modelling Concepts.

Also of note here, is the inclusion of ArchiMate, which focuses mainly on the metamodel, giving a detailed set of entity types, and also differing relations between those types.

2.2.2.5 Increasing Complexity of Enterprise Architecture Frameworks

The EA frameworks tend to increase in complexity (scope + detail) over time. This is shown in the following section, where we look at the evolution of two common EA frameworks, focusing in particular, on the entity metamodels contained within them, as that is the aspect of most interest to our current research.

2.2.2.5.1 Evolution of TOGAF

TOGAF is an Enterprise Architecture Framework owned by the Open Group; it stands for The Open Group Architecture Framework. It has its roots in the Technical Architecture Framework for Information Management (TAFIM) [87] issued by the US Department of Defense in 1993; the first version of TOGAF, 1.0, was issued in 1995.

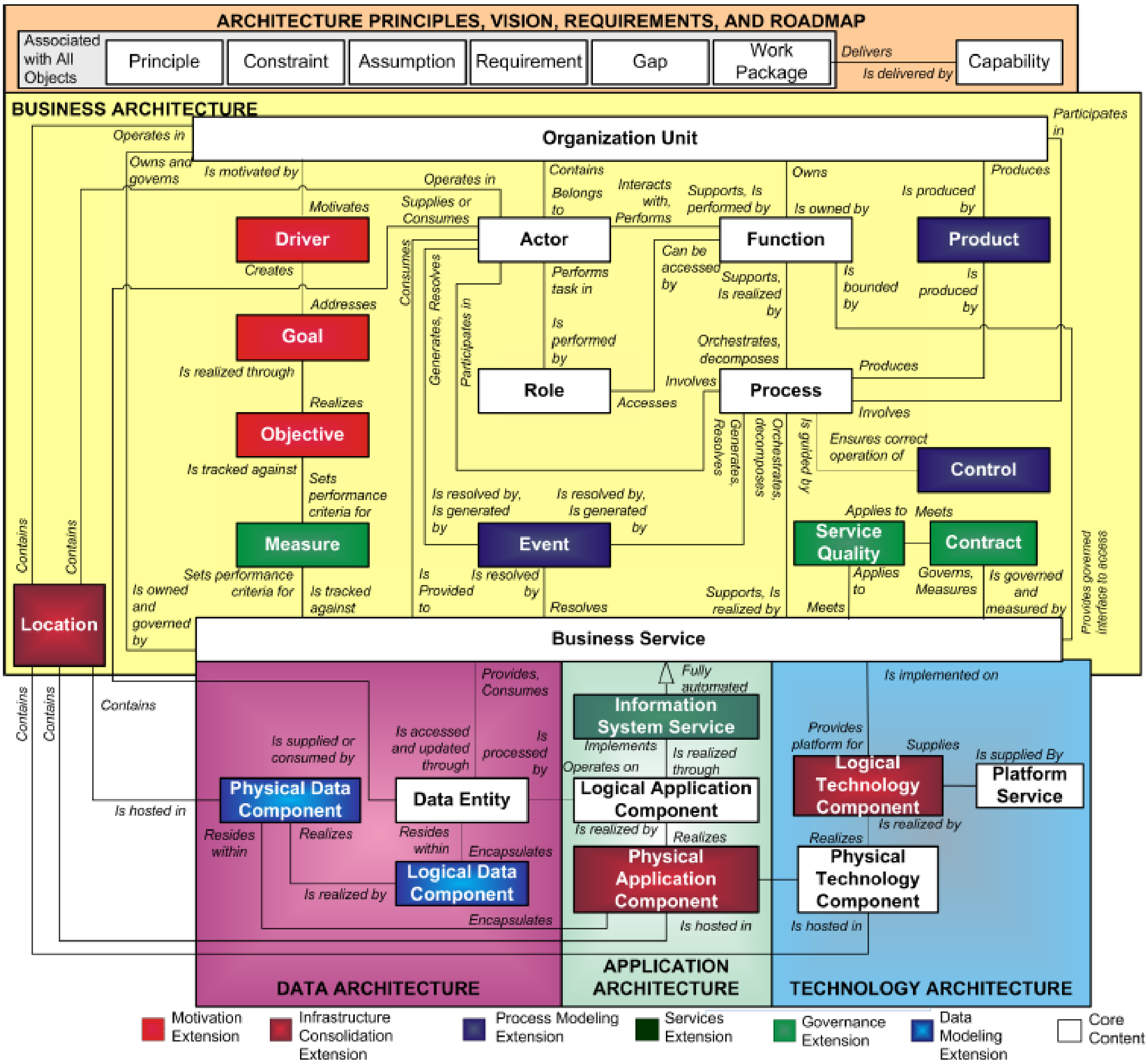
TOGAF 8.1.1

All versions of TOGAF prior to TOGAF 9 did not contain any kind of entity metamodel. The last version of TOGAF prior to TOGAF 9 was TOGAF 8.1.1 [30] released in 2006. We are not interested, for our purposes, in versions of TOGAF prior to 8.1.1 as they all have same metamodel content (i.e. none).

TOGAF 9.0

TOGAF 9 introduced a content metamodel, which has got slightly more complex in subsequent revisions. TOGAF 9.0 was released in 2009 and contained the following content metamodel:

Figure 16 Relationships between Entities in the Full Metamodel (from TOGAF 9.0) [31]

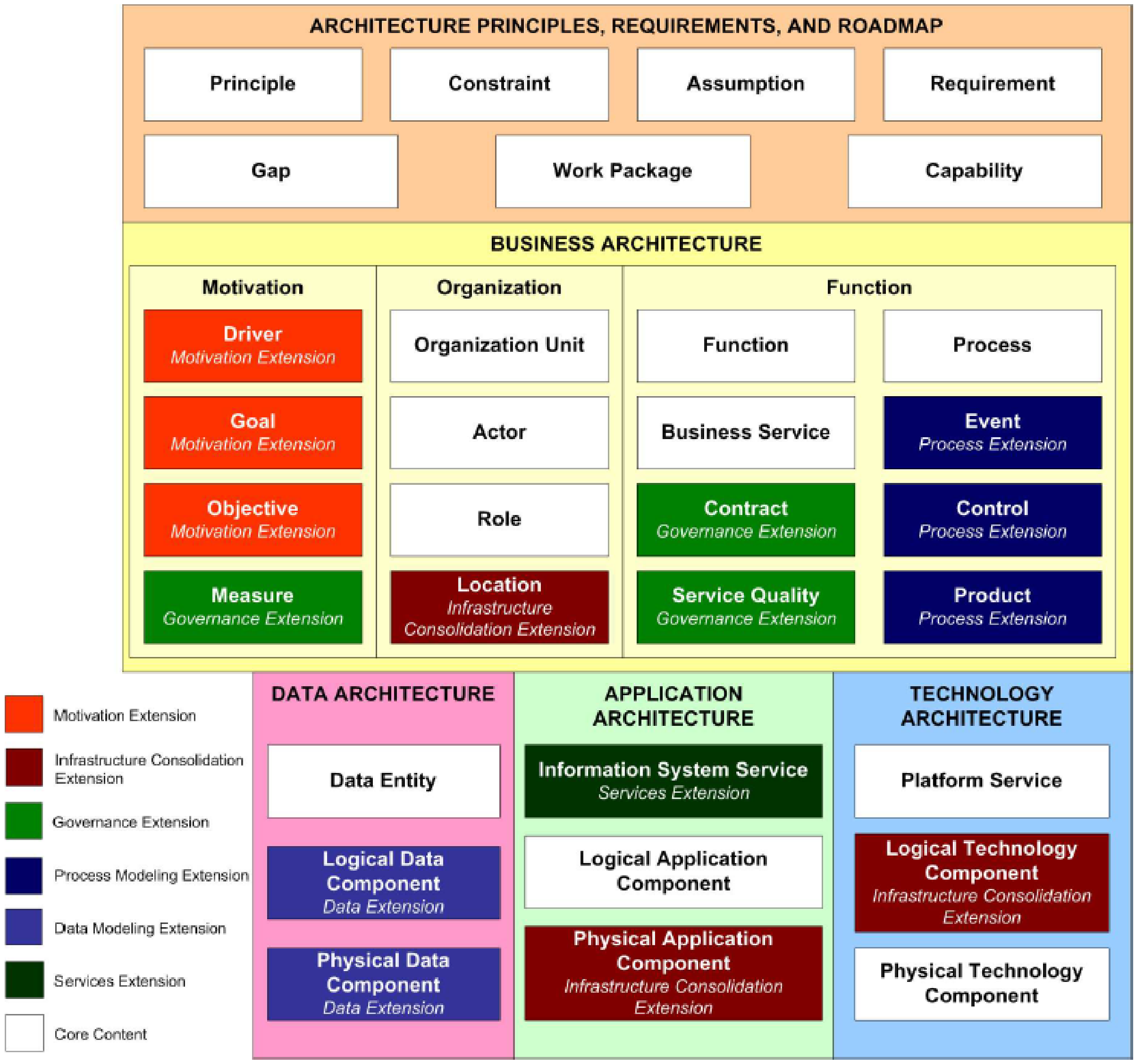


Interestingly, it was noted that this particular information model (i.e. the TOGAF 9.0 content metamodel) was in need of improvement, and so formal ontological techniques were used to propose improvements to TOGAF [88], presumably feeding into the next release.

TOGAF 9.1 why do we care about all these versions? Won't discussion of a table of them do?

TOGAF 9.1 [29], released in 2011, had the following metamodel:

Figure 17 Content Metamodel with Extensions (from TOGAF 9.1) [29]

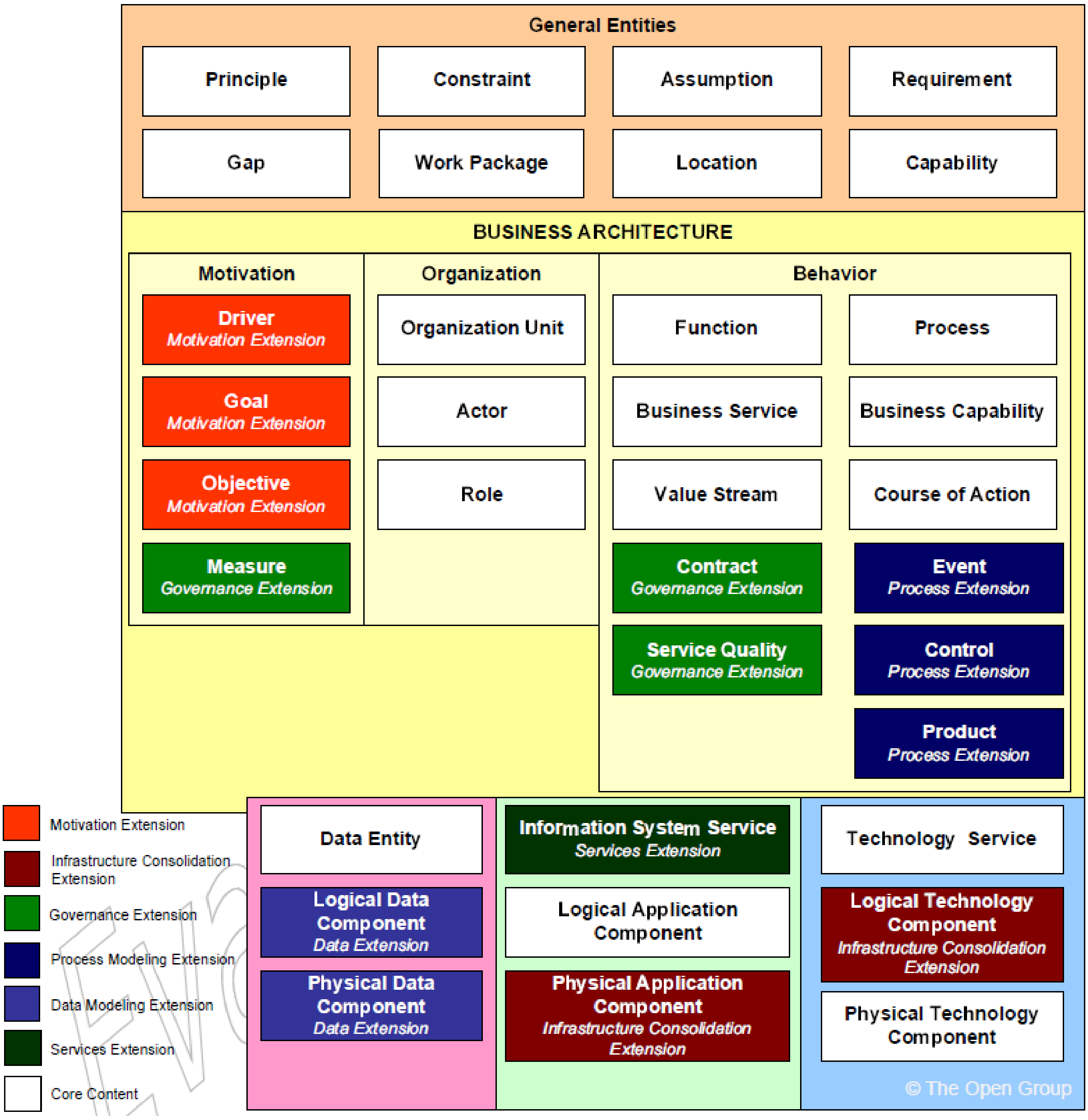


TOGAF 9.1 added the concept of a *Business Service* which was not present in TOGAF 9.0.

TOGAF 9.2

The next release was eight years later in 2019, and contained the following metamodel:

Figure 18 Content Metamodel with Extensions (from TOGAF 9.2) [14]



TOGAF 9.2 added the concepts of *Value Stream*, *Business Capability* and *Course of Action* that were not present in TOGAF 9.1.

Overall Evolution of TOGAF

Thus, we see that the evolution of the content (entity) metamodel in TOGAF is as follows:

Table 5 Evolution of TOGAF Entity Metamodel

Year	Version	Additions to Metamodel	Number of Entity Types
2006	TOGAF 8	No entity metamodel	0
2009	TOGAF 9.0	Initial release	32
2011	TOGAF 9.1	Added *Business Service*	33
2019	TOGAF 9.2	Added *Value Stream*, *Business Capability*, *Course of Action*	37

2.2.2.5.1 Evolution of ArchiMate

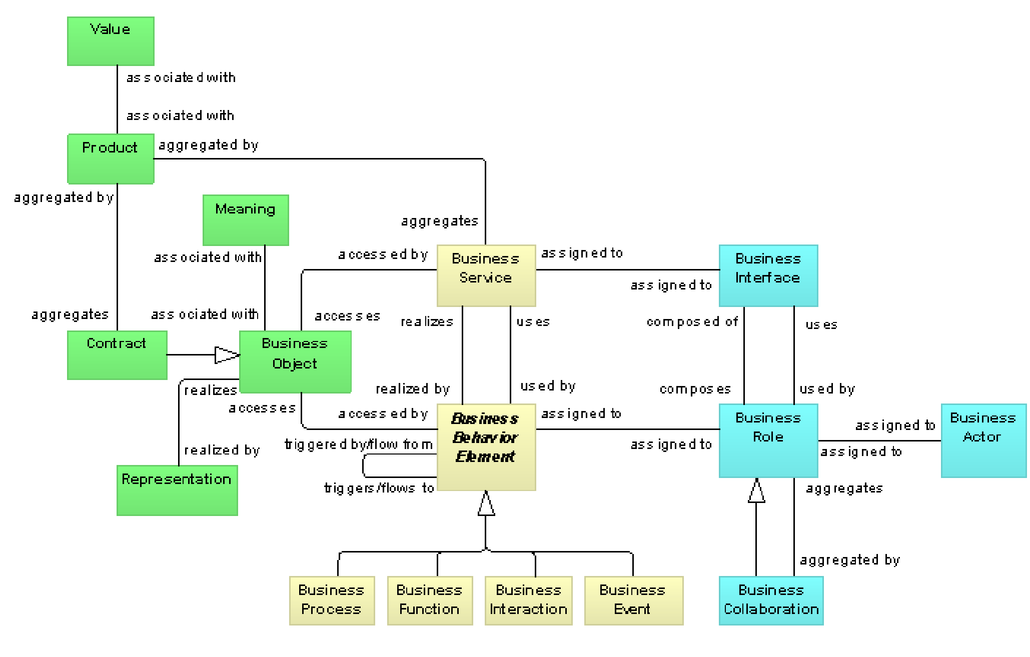
Unlike TOGAF, ArchiMate does not contain much more than a content framework; as illustrated in Table 4 above: it specifies entities, relationships and viewpoints. The entities are grouped into layers and aspects.

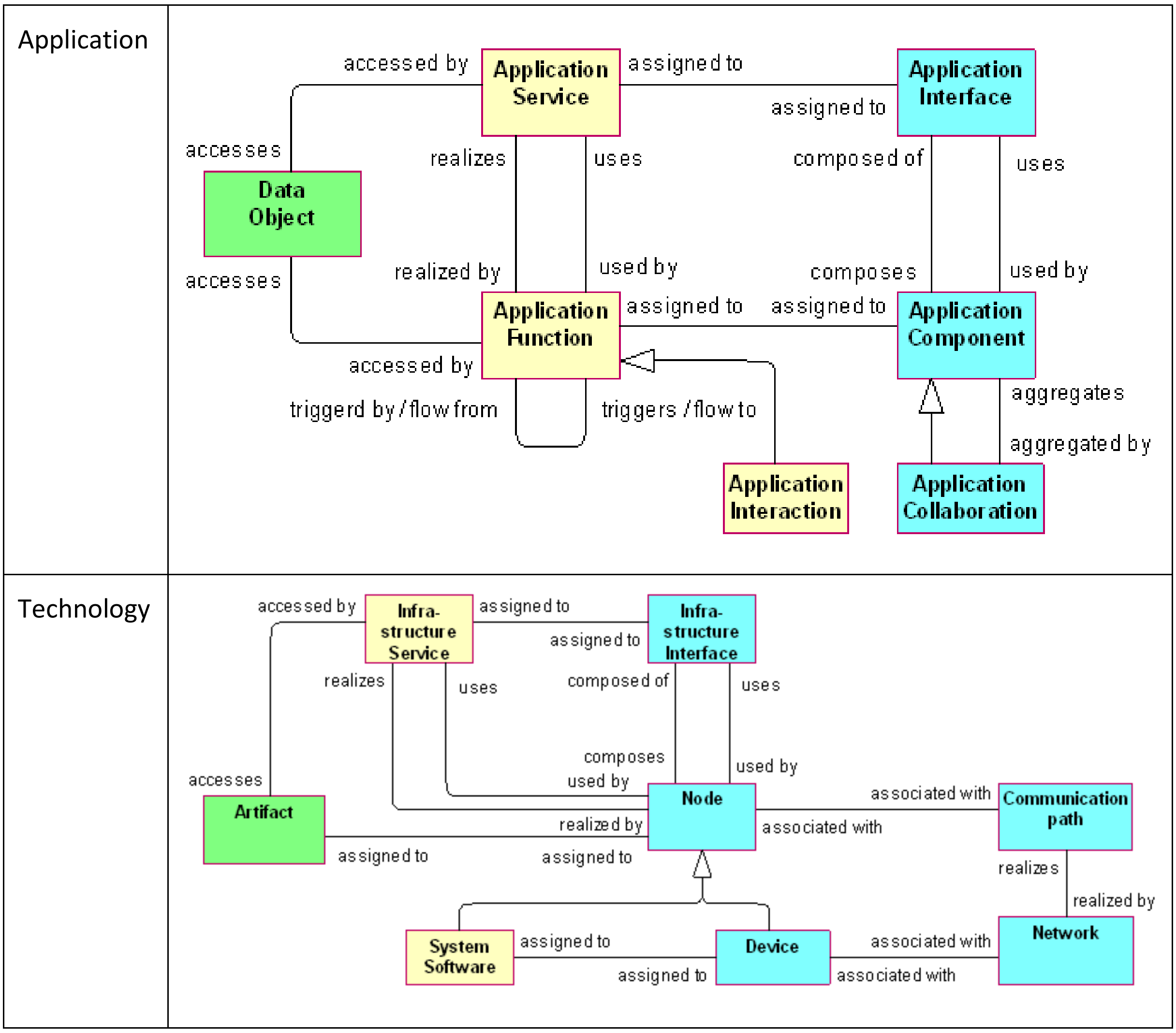
ArchiMate 1.0

ArchiMate 1.0 [89] was released in 2009. It contained three layers, and their metamodels are as shown below:

Figure 19 ArchiMate 1.0 Metamodels [89]

Business





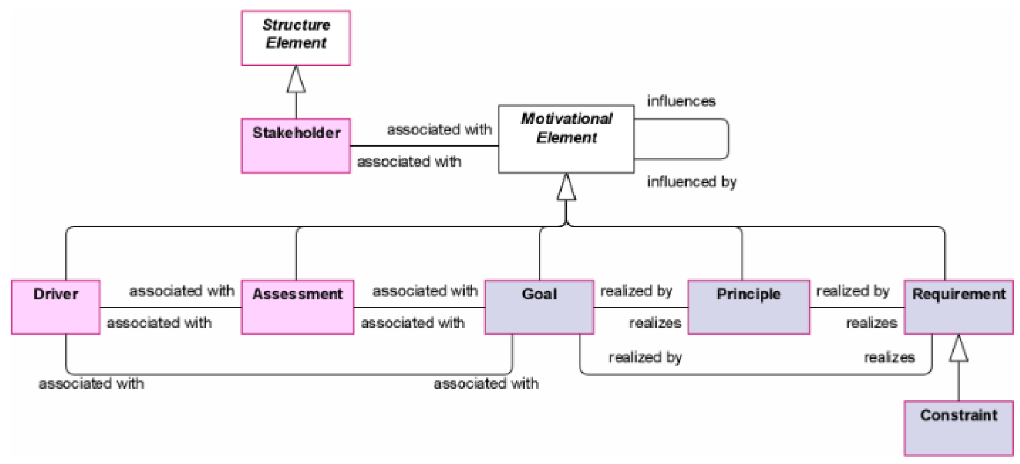
ArchiMate 2.0

ArchiMate 2.0 [90] was released in 2012. It added two new layers to the metamodel: Motivation (similar information to that contained in the Motivation extension in TOGAF 9.0) and Implementation and Migration (designed to work with later phases of TOGAF's architecture process, the Architecture Development Method).

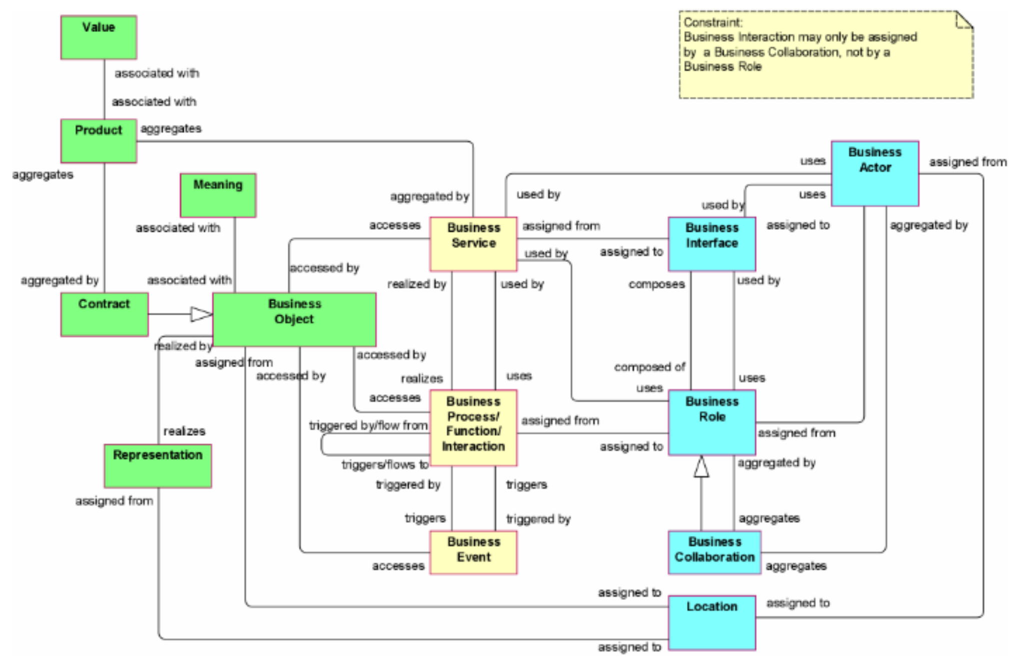
The metamodel for ArchiMate 2.0 is again specified in layers; it is too large to easily represent in a single diagram without requiring excessive space. The layers of the metamodel are as follows:

Figure 20 ArchiMate 2.0 Metamodels [90]

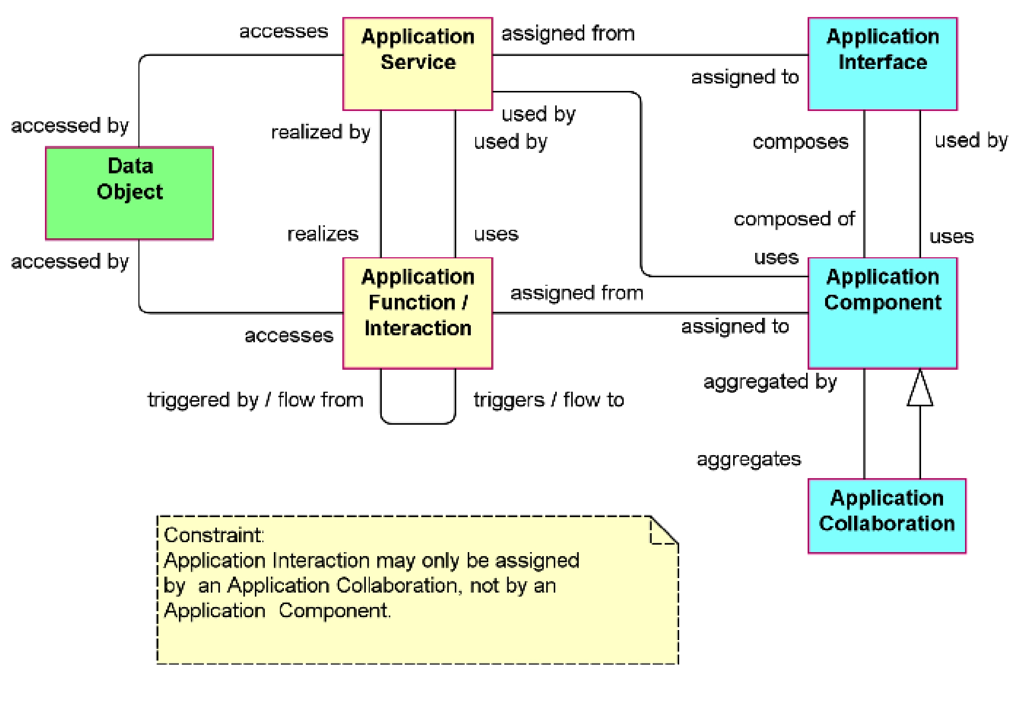
Motivation



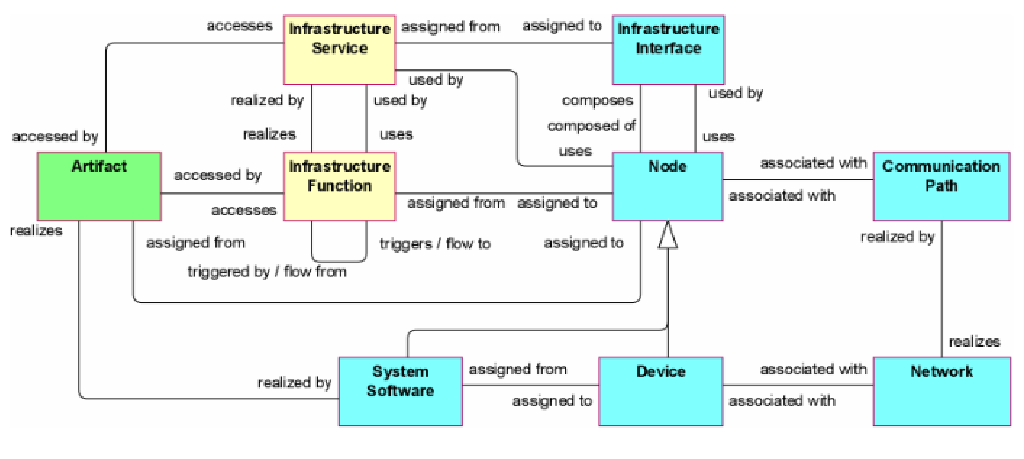
Business



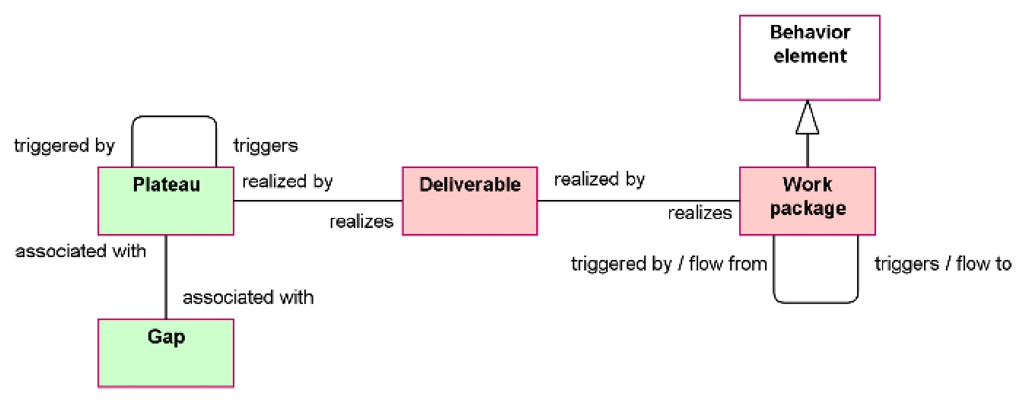
Application



Technology



Implementation and Migration



Thus, ArchiMate 2.0 added the following concepts:

Motivation layer: *Stakeholder, Driver, Assessment, Goal, Principle, Requirement, Constraint*

Business layer: *Location*

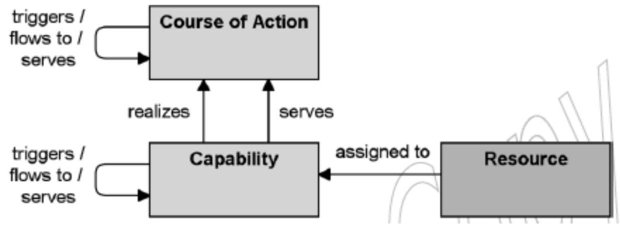
Technology layer: *Infrastructure Function* (later to be renamed to *Technology Function*)

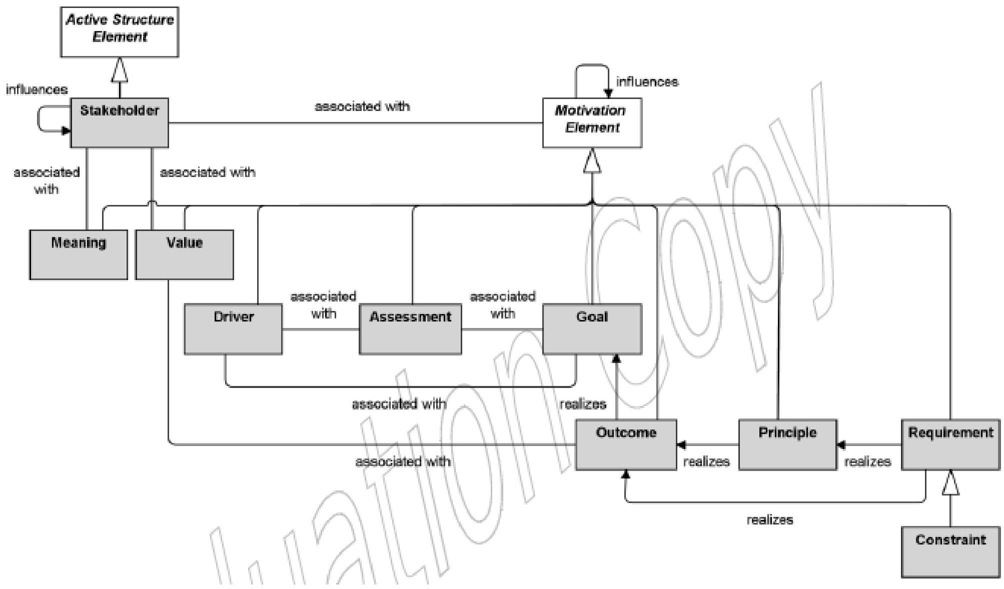
Implementation & Migration layer: *Plateau, Gap, Deliverable, Work Package*

ArchiMate 3.0

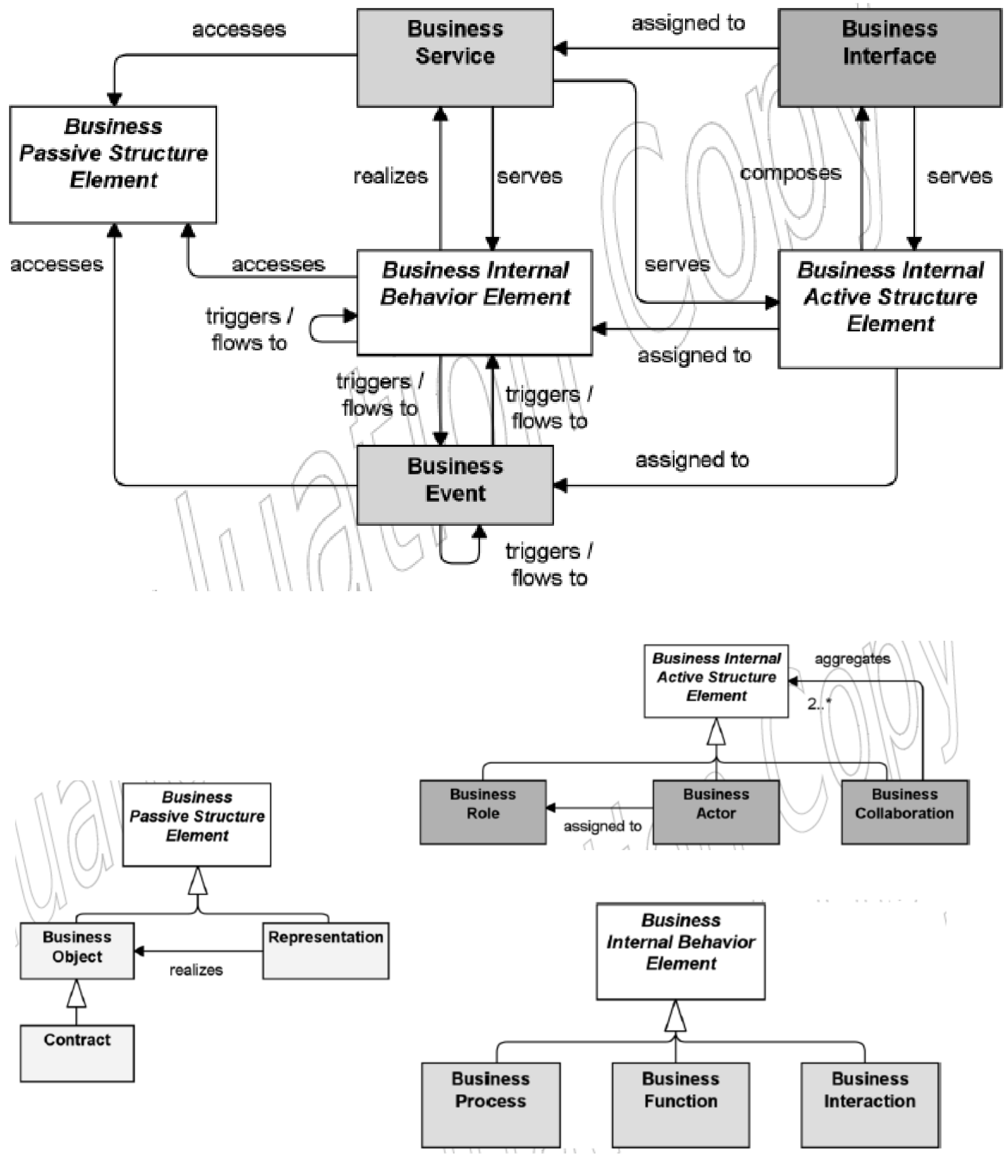
Moving on now to ArchiMate 3.0 [39], released in 2016, more layers have been added: Strategy and Physical. The metamodel for this version is shown below.

Figure 21 ArchiMate 3.0 Metamodels [39]

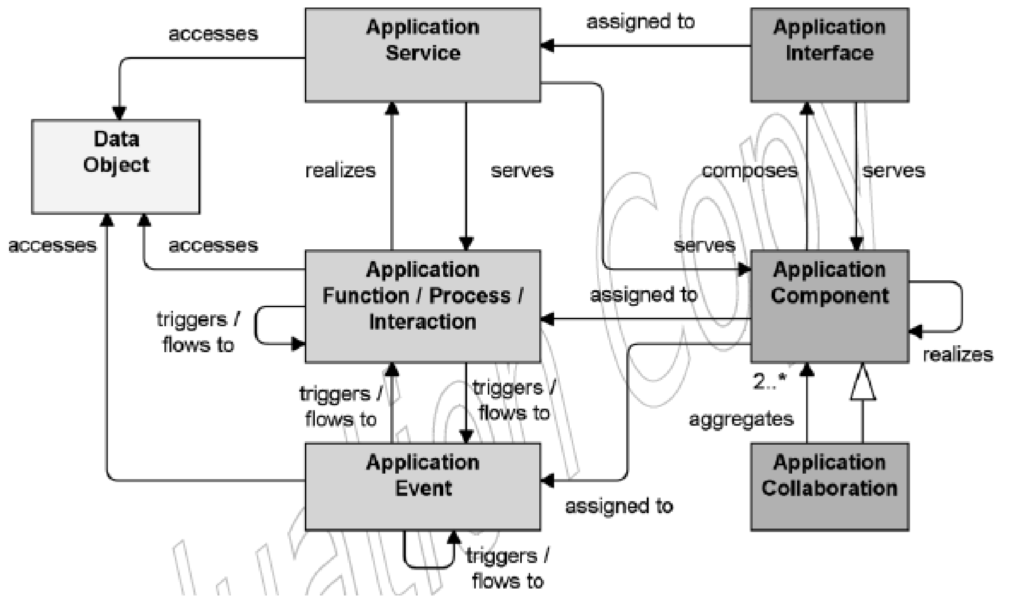
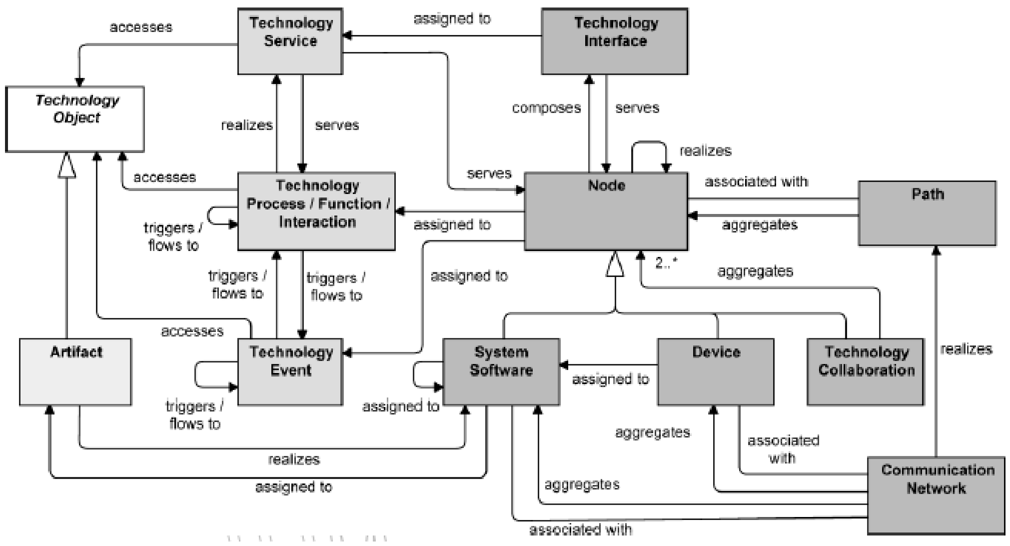
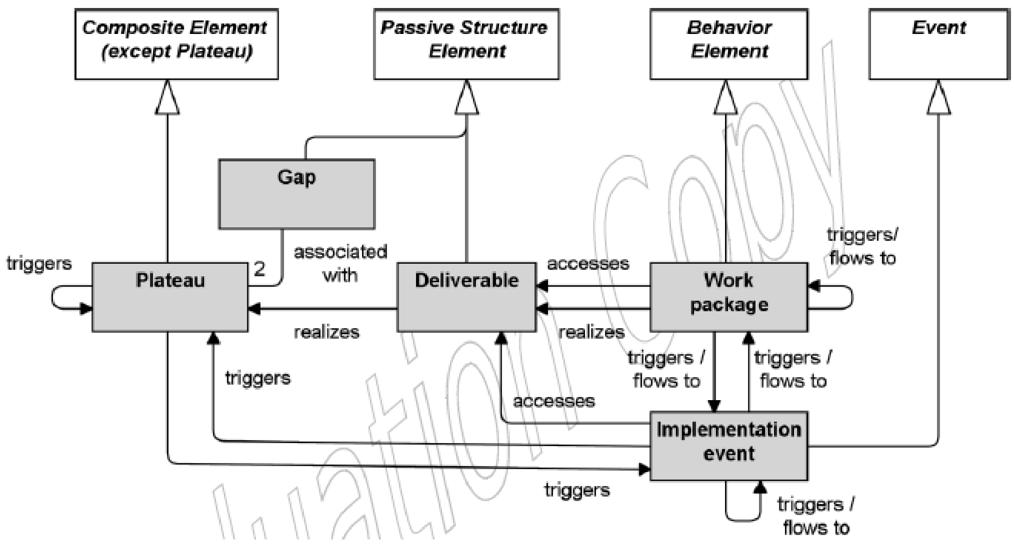
<table>
<tr><td>Strategy</td><td>


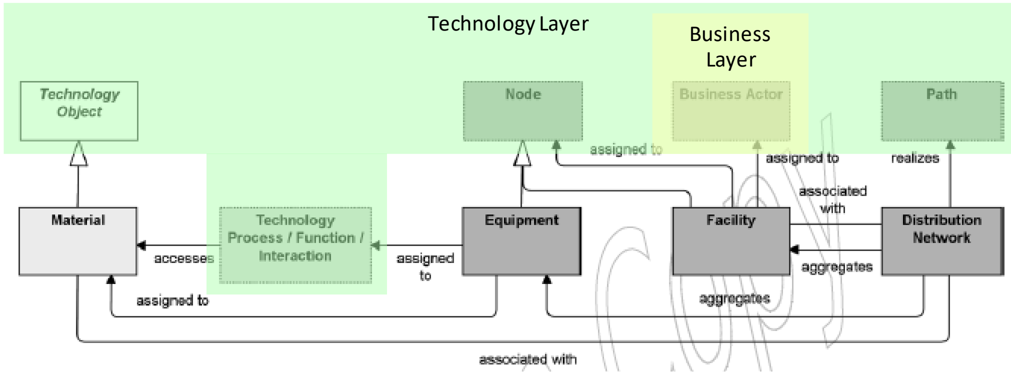
</td></tr>
<tr><td>Motivation</td><td>


</td></tr>
</table>

<table>
<tr>
<td>Business</td>
<td>


</td>
</tr>
</table>

<table>
<tr><td>Application</td><td>
</td></tr>
<tr><td>Technology</td><td>
</td></tr>
<tr><td>Implement-ation and Migration</td><td>
</td></tr>
</table>

Physical	 The original diagram in the ArchiMate 3.0 specification showing the elements in the Physical layer included elements from other layers, and so we have deliberately occluded them somewhat in the above diagram in order to highlight the elements that are actually in the Physical layer.

We can see therefore that ArchiMate 3.0 added (or in some cases renamed) the following elements to those previously available in the previous release:

Strategy layer: Course of Action, Capability, Resource

Motivation layer: Meaning, Value, Outcome

Application layer: Application Process, Application Event

Technology layer: Technology Object, Technology Service (renamed from Infrastructure Service), Technology Process, Technology Function (renamed from Infrastructure Function), Technology Interaction, Technology Event, Technology Interface (renamed from Infrastructure Interface), Technology Collaboration, Path (renamed from Communication Path), Communication Network (renamed from Network)

Implementation & Migration layer: Implementation Event

Physical layer: Material, Equipment, Facility, Distribution Network

ArchiMate 3.1

The release of ArchiMate 3.1 [15] in 2019 made far fewer changes to the metamodel, making only a single addition: that of the concept of *Value Stream* to the Strategy layer.

Overall Evolution of ArchiMate

Thus, we see that the evolution of the content (entity) metamodel in ArchiMate is as follows (showing only the items added; excluding items already present but just renamed):

Table 6 Evolution of ArchiMate Entity Metamodel

Year	Version	Additions to Metamodel	Number of Entity Types
2009	ArchiMate 1.0	Initial release	30
2012	ArchiMate 2.0	Added Stakeholder, Driver, Assessment, Goal, Principle, Requirement, Constraint, Location, Infrastructure Function, Plateau, Gap, Deliverable, Work Package	43
2016	ArchiMate 3.0	Course of Action, Capability, Resource, Meaning, Value, Outcome, Application Process, Application Event, Technology Object, Technology Process, Technology Interaction, Technology Event	55
	ArchiMate 3.1	Value Stream	56

2.2.2.5.2 ArchiMate vs. TOGAF Metamodels

If we compare the size of the metamodels from ArchiMate and TOGAF over time, we see the following trend:

Figure 22 Comparison of Growth of ArchiMate and TOGAF Entity Metamodels

This demonstrates the truth behind our assertion that (at least two) architecture frameworks are getting more complex over time, in terms of the size of their metamodels; ArchiMate more so than TOGAF.

So, what is the impact of this increasing complexity?

On the one hand, it means that those people modelling using (say) ArchiMate have many more terms to use, and those the scope of the language has increased, both in terms of breadth (e.g. ArchiMate 3.0 and later can model strategic elements such as business capabilities) and in terms of depth (more terms added to each layer, so for example events and processes were added to the application and technology layers, also in ArchiMate 3.0).

Architects using the language thus must learn new terminology, although hopefully this is not too onerous a task with proper training. However, if the architects use the full vocabulary available to them, then that also requires the consumers of the models to be able to understand that vocabulary, without having had the training, or perhaps even the inclination. There is thus the potential for the models constructed using this language to become less comprehensible to those not trained in its use; maybe this is one reason why architects avoid using ArchiMate when communicating with non-technical stakeholders. This is one of the focus areas for our research; we are seeking to understand which terms are understood by non-architects, which ones are not and why.

2.2.3 About Models

2.2.3.1 Definition of a Model

Stachowiak [36], translated from the original German by modelpractice [91], distinguishes between three "features" of a model:

> ***Mapping**: Models are always models <u>of something</u>, i.e. mappings from, representations of natural or artificial originals, that can be models themselves.*
>
> ***Reduction**: Models in general capture <u>not all</u> attributes of the original represented by them, but rather only those seeming relevant to their model creators and/ or model users.*
>
> ***Pragmatism**: Models are not uniquely assigned to their originals per se. They fulfil their replacement function a) for <u>particular</u> – cognitive and/ or acting, model using <u>subjects</u>, b) within <u>particular time intervals</u> and c) restricted to <u>particular mental or actual operations</u>. [36]*

We can see here their relevance to Enterprise Architecture: EA models are models of an existing or potential architecture, they capture and present subsets of the data as relevant to the various stakeholder views and they are there to fulfil a purpose, for example to communicate with stakeholders or to make design decisions.

Models are generally defined as explicit representations of some portions of reality as perceived by some actor [92].

The literature does deal with the concept of an *interactive model*, being one where changes in the model results directly in changes to a connected system, however, this is not normally applicable to Enterprise Architecture models, because a change to an EA model will not normally result directly in a change to an organisation.

We should stress that not all models are necessarily graphical. For example, our research contains a number of mathematical models (according to the above definition) that have the purposes of enabling us to measure some kind of quality. However, it is frequently the case that models *are* graphical.

2.2.3.2 Graphical Representations

A paper published in 1996 entitled "External cognition: how do graphical representations work" [93] suggested that at that time, the literature did not contain a well understood and coherent account of how graphical representations worked. The authors went on to propose a new agenda for research into graphical representation, linked to a new theoretical approach within cognitive science linking external (real-world) and internal (mental) representations. This has parallels with our discussion later in the thesis on the semiotic processes, converting external "signs" to internal "referents".

2.2.4 About Quality

As discussed in the introduction, in section 1.3.4.1 above, there are many aspects to quality, and one particular paper [41] discussed some of the historical roots of our quality definitions that are relevant to our research.

2.2.4.1 ISO Definition of Quality

ISO 9000 [94] is a standard for quality management systems. It offers the following as a definition for the term quality: "[the] *degree to which a set of inherent characteristics of an object fulfils requirements*" [94]. This definition depends in turn upon some other definitions, also provided in the same standard:

Characteristics	distinguishing feature
Object:	anything perceivable or conceivable
Requirements	need or expectation that is stated, generally implied or obligatory

We could therefore combine the short version of the ISO definition with the above supplementary terms to provide a fuller definition of quality, still using ISO language:

The degree to which a set of inherent distinguishing features of anything perceivable or conceivable fulfils needs or expectations that are stated, generally implied or obligatory.

There are several aspects of this definition that are relevant to our work.

Firstly, the distinction between perceivable or conceivable. We discuss, in the research related to the quality of sets of models, the fact that we could conceive of models that could (in some cases, should) exist but that do not. Such theoretical but non-existing models could be of good or bad quality. Thus, we agree that quality can not only apply to things that do exist (and are therefore perceivable), but also to things that could exist.

Secondly, the 'needs or expectations' will, in our case, relate to the stakeholders trying to understand the models and we can see that there these can be both implied and obligatory.

By implied, we mean that they are expressed implicitly rather than explicitly and by obligatory we mean that the needs or expectations must be met; they are not optional. These are not opposites (it is not clear if the original paper was suggesting that they are, in context); nevertheless, it is true that there are some needs or expectations that must be met to satisfy stakeholders. These would include the notion of completeness – have we covered all the topics or aspects that the stakeholders are interested in – and of accuracy – even if the models omit certain aspects, are the aspects that they do contain complete, such that the model is correct and complete as regards its core propositions? This is related to our definition of 'understanding', discussed in 2.3.16 below.

These are some of the obligatory aspects; by implied, we might deal with other aspects that are not formally defined or stated but that nevertheless would be expected by certain stakeholders. A number of examples may illustrate this. Firstly, there may be unstated expectations that diagrams will 'look and feel' a certain way, both in element types (concepts) that they leave out (which is why we are interested in which concepts are not understood by non-technical stakeholders), in terms of the visual notation (for example, using company branded colours, avoiding any ArchiMate notation, and a recent request we received, when producing architecture diagrams, to deliberately remove the unique IDs that we had assigned to uniquely identify particular architectural elements such as applications, because certain stakeholders weren't interested in them.

The ISO definition is widely accepted, and at the same time non-specific, precisely because quality means different things to different people.

2.2.4.2 Satisfaction-Based Definition of Quality

Writing in 2009, Wicks and Roethlein [95] survey existing literature, note that there is still no universal consensus on a definition of quality, also noting in passing that the most widely accepted is the ISO definition discussed above. They suggest that one of the reasons for the failure of so many quality initiatives may be the lack of a universally accepted satisfaction-based quality definition. They look at commonalities in definitions of quality in order to determine whether quality or satisfaction is the most important concept for increasing customer retention. In our own case, our customers are of course our stakeholders, and we certainly are interested in retaining our relationship with them. Wicks and Roethlein produce a quality definition related to evaluations by each customer of each object in scope (which in our case will be output of our EA activity, including EA models). The evaluations relate to both internal and external customers. The analogy for the activity of Enterprise Architecture would be to consider other architects or IT professionals as being 'internal customers' and our non-technical stakeholders as 'external customers'. The internal customers will be concerned about some aspects of quality, for example that of value (was it efficient to produce the models?) while the external customers will be concerned to ensure that they can understand what the models are saying, and thus the appearance (look and feel) of the models will be very important.

2.2.5 About Tools

Given that we touch upon the idea of EA modelling tools in one of our chapters, it is worth considering what is meant conceptually by a tool.

The German philosopher Martin Heidegger carried out some important work considering a theoretical framework in which we can examine the various concepts related to tools. In [96], he renders these on p. 64 initially as "useful things", relating them to the Greek word πράγματα (from which of course we derive the word *pragmatic*). He stresses that these are always to a purpose and are understood in relation to other things, indeed their value is in relation to other useful things (i.e. a useful thing, or tool, is useful because it creates X, which in turn is useful because it can create Y, and so on). *A tool is always there for a purpose*. He discussed two aspects of these useful things (which we might translate as tool, or equipment). There is a theoretical aspect, having an understanding perhaps of what something might be used for (German: *Vorhandenheit*, sometimes translated as "presence-at-hand") and then there is what he calls *Zuhandenheit* (translated variously as "handiness" and "readiness-to-hand"), which can only be discovered by actually using it in the intended fashion. Tools are not <u>either</u> *Vorhandenheit* <u>*or*</u> *Zuhandenheit,* rather they oscillate between the two modes: the former being our conceptual model of the tool and the latter being a more primitive knowledge of it as we actually use it.

This matches our own experience of Enterprise Architecture modelling tools; we understand what they are capable of and can explain that to others but, when we actually use them, we have a sense of working in a particular manner, with particular expectations of how they should actually work and how to use them on a day-to-day basis.

This theme is expanded on in [97], discussing Heidegger's work, drawing out the point that equipment (or tool) in Heidegger's analysis, is always in a context that provides it with meaning. Equipment, or tools, derive meaning from the other tools and objects that surround them.

This is a helpful reminder that EA modelling tools, and the activity of using them, derive their meaning from the surrounding context. Modelling in and of itself may be interesting, but the value and meaning of it depends crucially on its context.

2.2.6 Systematic Review of IS Journals

Our systematic search of influential, 'high-impact' journals will be carried out using those included within the "Senior Scholar's' Basket of Journals [63]. In our search, we are specifically looking for papers relevant to our research and thus related either to EA model quality, or to the effectiveness of EA modelling languages. We searched for papers published between 2005 and the current year. The journals we searched were:

- European Journal of Information Systems
- Information Systems Journal
- Information Systems Research
- Journal of the Association for Information Systems
- Journal of Information Technology
- Journal of Management Information Systems
- Journal of Strategic Information Systems
- Management Information Systems Quarterly

From this systematic survey of these journals, we found the following relevant to our research:

Table 7 Results from Systematic Literature Search

Key	Title / Analysis
[98]	An empirical analysis of the factors and measures of Enterprise Architecture Management success
Analysis	This paper suggests that Enterprise Architecture Management (EAM) comprises products (documentation of the EA and related decision making), infrastructure (formal foundation including governance and frameworks) and services (offered to the organisation by the EAM capability). Within the products, are included as-is and to-be architectures, roadmaps and EA principles. A key dimension added here is that of 'organisational anchoring' which deals with socio-organisational, individual and cultural aspects. The paper notes a significant direct correlation between EAM product quality (which will include EA models) and EAM success measures.
Relevance	This underlines for us the importance of producing high-quality models; it supports the assertions made in our introductory section

[99]	Enterprise architecture and enterprise architecture artifacts: Questioning the old concept in light of new findings
Analysis	This paper deals with the Enterprise Architecture artefacts that have been proven useful in practice. By artefact here we mean an EA work product that will contain elements (entities) and relations from some framework, for example value chains, data models).
Relevance	Not directly relevant, as it does not deal with the individual elements of such frameworks. However, as a pointer for further research, it would be interesting to look for correlations between the comprehension and utility ratings of entities (architectural concepts) when compared to the artifacts (as listed in this referenced paper) that would typically use them.

2.3 Theories and concepts Related to our Research Questions

We consider here some theories and concepts that we will be building upon in our own research.

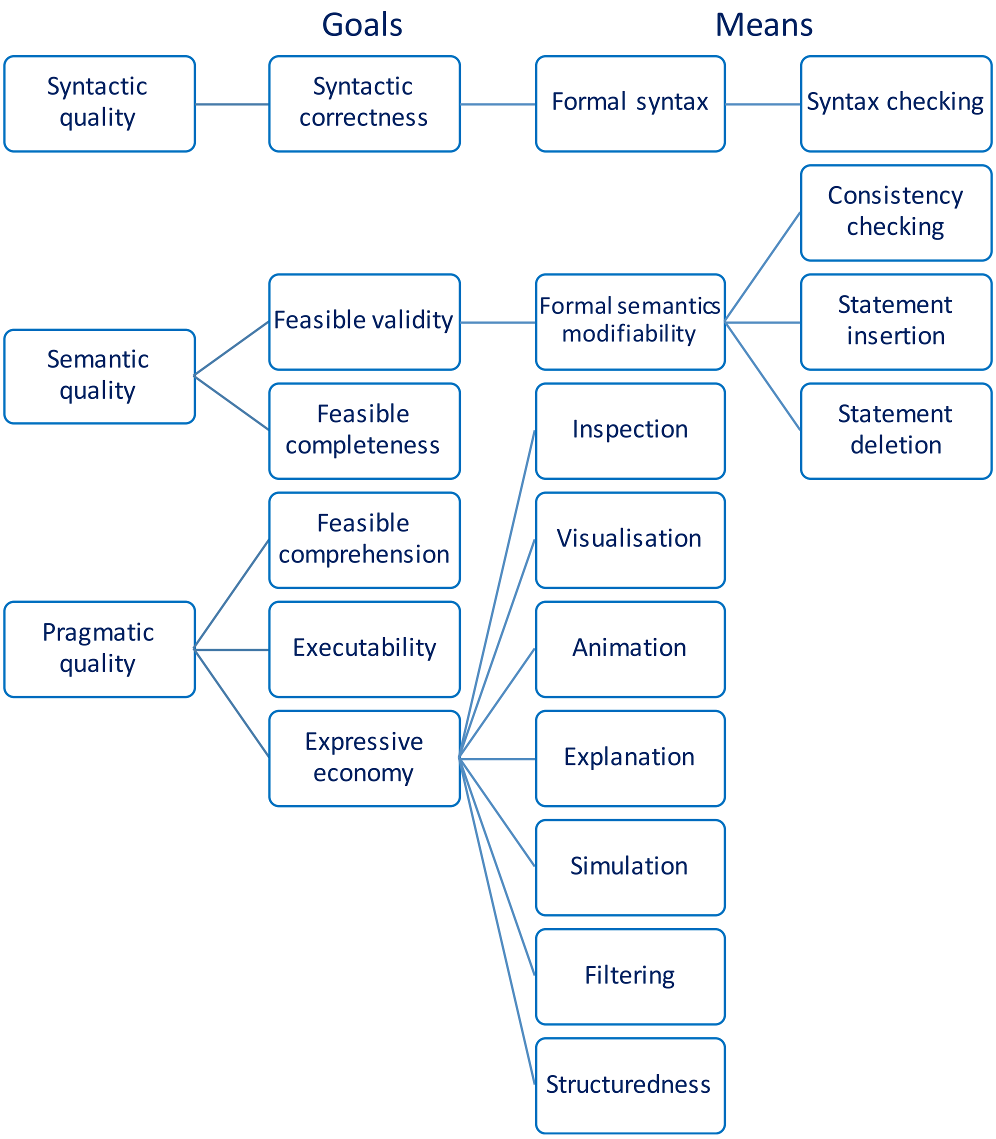
2.3.1 Model Quality

2.3.1.1 Conceptual Model Quality

A general semiotic framework for assessing models has evolved to take in more layers of the semiotic framework: first, three layers, proposed by Lindland [56], then extended by Krogstie [48], given the name SEQUAL, and then again to incorporate more layers of Stamper's semiotic ladder [100] by Krogstie [92]. Recall that in our introduction, in section 1.3, we distinguished between models in the more general sense, from the specific sense of (say) a diagram (or view or picture). Conceptually, models can come in many forms, and as previously stated, have three characteristics (represent something else; have a purpose; are incomplete). Thus, we could argue that a database of customer data within a CRM system (as discussed in [101]) is precisely a model in this more general sense: each record represents a customer; it has a purpose (to track and manage communications with the customer), and it is only a summary of all possible information about the customers). Thus, remembering that models are not necessarily diagrams, we can now examine the literature concerning the quality of conceptual models.

A framework to measure general conceptual model quality was developed by Lindland [56]. His paper lists many ways in which people view quality, including the model being appropriate, complete, conceptually clean, consistent, correct, expressive economy (relates to our *second* research question regarding EA language effectiveness), minimal, testable, traceable and so on. His framework is built upon three of the levels in the semiological ladder: syntactics, semantics (from which he takes his language $\mathcal{L}$) and pragmatics (how the audience will interpret the models). His proposed framework is shown below:

Figure 23 Proposed framework for distinguishing quality-improvement goals and the means to achieve them (adapted from [56])



It contains key concepts of immediate relevance to Enterprise Architecture models in particular, as it deals with compliance with the specified language (syntactic correctness). Lindland's concept of domain is stated thus:

The domain, $\mathcal{D}$, consists of all possible statements that would be correct and relevant for solving the problem. [56]

2.3.1.2 Application of Model Quality to Enterprise Architecture

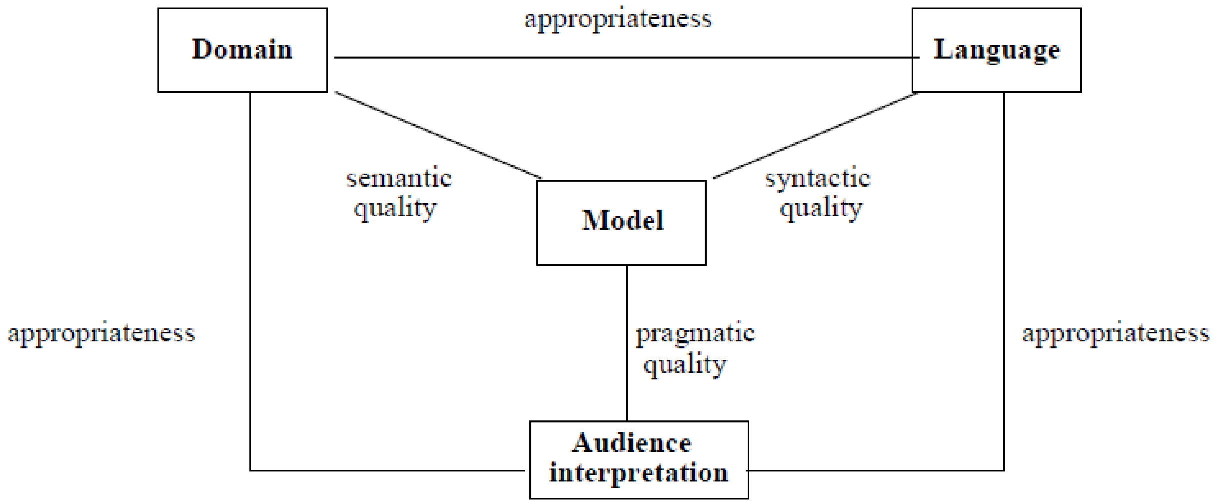
For our purposes, we would wish to broaden this, as it can and should be made more specific in some situations. Looking at Enterprise Architecture models (still conceptual for now, not necessarily dealing with diagrams), the descriptions of our architecture could relate to the past, present or future. The *future* view deals with what reality could look like in the future. Thus, our model might contain 'statements' that might possibly be correct and relevant in the future (as per Lindland's concept) for addressing the issues facing the Enterprise at the current time. However, the *current* view is just as important as the future view, because if we do not understand the current business and IT infrastructure, we will not know the steps required to transform them to a future desired state. Therefore, one extension we will wish to make to Lindland's definition is to widen the definition of domain so that it deals with not just possible future states, but also current states. Thus, in our research, the concept of domain we will use in our quality metrics will be correspondence to reality, either now or in the future.

This framework was later extended by Krogstie [48], to include some missing dimensions from the semiological ladder. It is also interesting to note that the latter paper has a subtly different definition of $\mathcal{D}$, which is closer to what we are seeking:

The domain [is] the set of all statements which would be correct and relevant about the problem at hand [48]

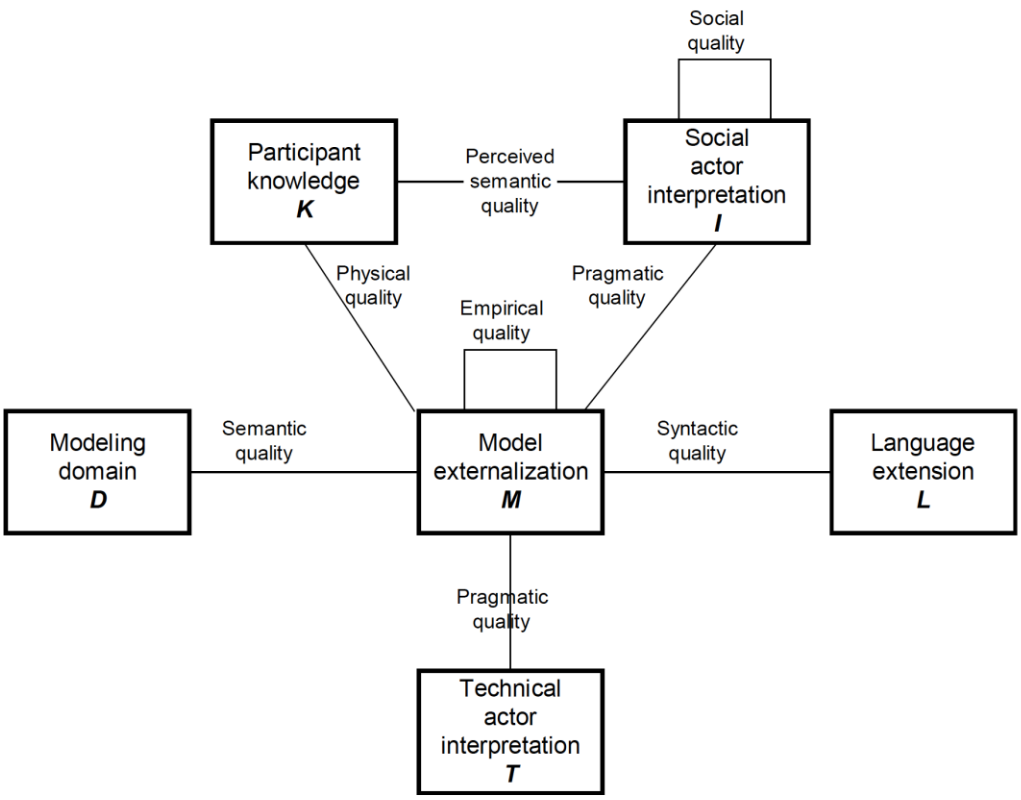
Now Krogstie provides the following diagram illustrating Lindland's use of the three semiological levels:

Figure 24 The framework by Lindland [56]



Krogstie then uses this as a starting point to establish his extended version of Lindland's framework:

Figure 25 Framework for discussing the quality of models [48]



[56]

We can see here the addition of the physical and empirical layers of the semiological ladder. Krogstie adds a refinement on Lindland's "audience interpretation", separating the audience into social and technical actors. We use this distinction in our own research into language effectiveness when we categorise our interviewees into category A (social), B (practitioners/modellers) and C (technical) readers.

We also see in the later paper the formulation (using set theory notation) of more precise definitions of various quality metrics, using the notation that we will continue to use in our own research. Also of interest to us is the appearance of the social and technical actors seen in the above diagram. In the original paper, a technical actor is taken to be software that interprets the model and as previously discussed, a language would have to be rigorous in order for that to be feasible (after all, a programming language is just a specific type of modelling language that describes algorithms). For this reason, no "Participant Knowledge" is required of the technical actor; it is assumed that there "is only one" interpretation of the model, from a technical perspective, assuming that the model has valid syntax (complies with language $\mathcal{L}$).

Key to our discussion is the concept of pragmatic quality, the goal of which is comprehension. More specifically, we are interested in this work in the comprehension of the human (social) actors rather than technical (automated) actors, at least at an enterprise level.

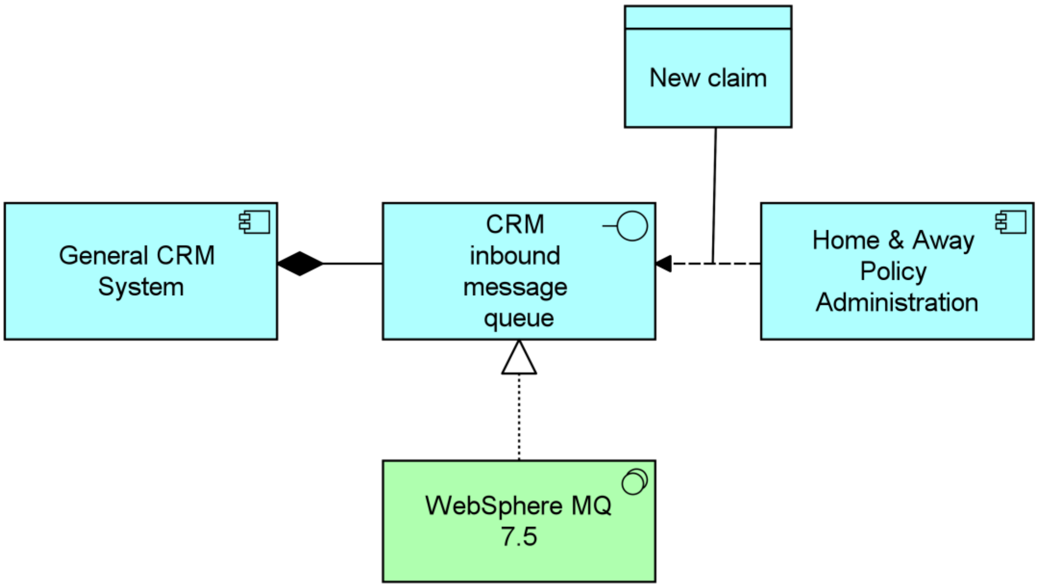
The Krogstie paper suggests that "formal syntax and formal semantics are means for achieving pragmatic quality" [48]. Part of the goal of this work is to question that assumption. Are formal syntax and semantics really the best way of ensuring that all of the stakeholders, for whom our EA models are produced, are able to understand our models? There is perhaps, in the words of the Krogstie paper, a need to balance "expressive economy" (not using too many symbols or words in our modelling language) with a need to maintain sufficient structure so as not to lose the validity of the message.

2.3.1.3 Missing Quality Elements for Enterprise Architecture Models

So, it is worth asking the question at this stage, what is missing from our definition of model quality: why cannot we just apply it directly to Enterprise Architecture models? We have already answered this partially in our discussion of the domain $\mathcal{D}$. There are other aspects can could be elaborated further when it comes to Enterprise Architecture; take for example the idea of Completeness, which relates to Krogstie's semantic quality. When considering the specific type of model known as an 'Enterprise Architecture diagram': if we have a set of such diagrams, when would we consider them complete? It is not clear how to apply the idea of completeness (or feasible completeness) in this situation. Another aspect not addressed is the fact that an organisation's use of formal modelling languages may evolve over time and, it is possible that, at a point in time, there are statements that we wish to make about our domain that cannot be expressed in the current languages available to us. All of these gaps are addressed in our first research question on the quality of a set of EA models.

At this stage, an example of some of these ideas about quality may be helpful. Let us consider the example of two applications, one of which provides information to the other (about a new insurance claim). Let us suppose that there is a message queue used to facilitate this flow of information and that the (valid) ArchiMate model that describes the current state of affairs is as follows:

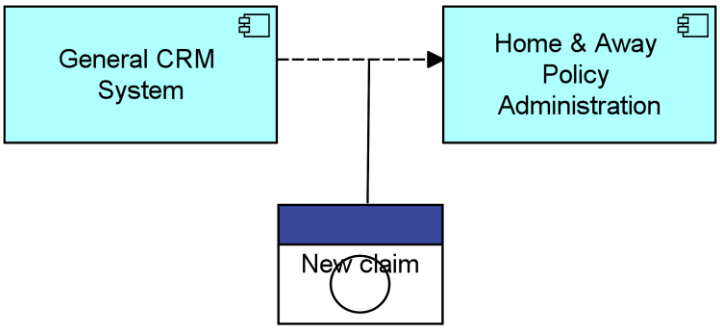
Figure 26 Flow between applications using a message queueing product



We will now give examples of the different kinds of errors we could encounter. Let us start with the pragmatic errors related to user interpretation. This is really related to the effectiveness of the modelling language and whether the concepts that we have used are appropriate for our particular stakeholders. If we are dealing with non-technical stakeholders, that do not understand the concept of an application interface, or system software, then this diagram is likely to be misunderstood.

Let us suppose that we therefore simplify the model to leave out the extraneous details. We provide a simplified version as follows:

Figure 27 Simplified diagram show message flow



This diagram contains two separate kinds of errors. Firstly, *it fails to comply with the specified language* (ArchiMate). The data object (representing the data that is flowing) is using a notation from Entity Relationship Diagrams rather than ArchiMate. Secondly, it fails to comply with the specified domain (in our case, the as-is situation where information flows from the administration system to the

CRM system). In other words, the diagram is “lying” – it says that information flows in the other direction. So we have here in this diagram an example of each of two types of error that will appear later in our mathematical framework for the quality of a set of EA models: a failure to comply with language $\mathcal{L}$ and a failure to comply with domain $\mathcal{D}$. The third category of quality error that could have applied is the completeness one: does the required model, in the form (and with the limited set of concepts for our particular stakeholders) actually exist? If we have not yet constructed it, then it would fail the third test (completeness, so in our notation, outside the set of existing models $\mathcal{M}$).

2.3.2 Modelling Requirements

Krogstie’s paper from 2002 [43] applies the semiotic techniques to a particular kind of model, that of a Requirement Specification. Some of the concepts from the general modelling quality theory (e.g. syntactic correctness; feasible completeness) are used in this quality framework, and in a way similar to the framework that we will develop regarding the quality of sets of EA models.

It uses the same quality framework as the earlier work (as reproduced in Figure 5 above). One of the areas covered by this is the quality of the language itself, encapsulated in the term ‘domain appropriateness’: the goal of this aspect is that “*there are no statements in the domain that cannot be expressed in the language*” [43]. We recognise that as a goal which is not always going to be achieved, and thus this aspect will need to be considered in our construction of a model quality metric (and in fact relates directly to our ‘extension for evolving maturity’ discussed in section 4.3.7 below).

This paper relates two lists of ideas. On the one hand it has a set of quality properties for requirements, taken from a 1993 paper [102]); these properties are:

Table 8 Quality Properties for a Requirements Specification, adapted from [102]

Unambiguous	Complete	Correct	Understandable	Verifiable
Internally Consistent	Externally Consistent	Achievable	Concise	Design-Independent
Traceable	Modifiable	Electronically Stored	Executable / Interpretable / Prototypable	Annotated by Relative Importance

Annotated by Relative Stability	Annotated by version	Not Redundant	At Right Level of Detail	Precise
Traced	Organised			

On the other hand it has the semiotic 'layers' with related quality metrics, shown in Figure 25 above. It then takes the quality metrics from semiotics and relates them to each of the quality properties listed in Table 8. One useful addition here is the recognition that the specification needs to be *complete*. In the context of a requirements specification, this has a very specific and limited meaning, but included in there are statements such as "no sections are marked 'to be determined'". In other words, everything that should have been 'modelled', has been.

It would appear that an approach similar to this would be useful for sets of EA models, because it is the set of all such models that comprise our 'body of knowledge' about our enterprise, its business and its supporting information technology.

This is what we will seek to do in our research into EA model quality.

2.3.3 Types of Architecture Models

We have used the term Enterprise Architecture throughout this work, although the theoretical framework and practical outworking is very similar whether we are looking at Enterprise, Solution or most other kinds of IT-related Architecture (see the typology of IT architectures in Figure 13 above). In particular, the models (diagrams) that we produce will use similar techniques and the same languages. Indeed, in one sense, we might say that our description of the Enterprise Architecture is formed by joining together the various descriptions from other segments of the enterprise, for example various Solution Architecture descriptions (perhaps produced as part of a project lifecycle), or forward-looking domain roadmaps.

Thus, we see little need at this stage to distinguish between, say, Enterprise and Solution Architecture models. The same concepts of quality can be applied equally to any of them.

2.3.4 Impact of Semiotics on Enterprise Architecture Models

As discussed in section 2.2.3 above [51], models contain a reduced and hence partial representation of the information available, designed for a particular use at particular times, for specific reasons. These models act as a communication medium between the creator of the model and the consumers of the model with encoding and decoding processes inherent in communicating ideas. Semiotics, the "science of signs" [53], has as a fundamental concept, the idea that a sign is distinct and different from whatever it refers to. This is often illustrated using the 'semiotic triangle' referred to in our introduction (see Figure 6 on page 59).

Our Enterprise Architecture models contain signs that we hope convey a correct interpretation that points to the correct real-world object. Different individuals with different experiences may "read" different signs as referring to different objects. Two trivial examples are:

- The words that we use (both spoken and written) are obviously culturally relative, and the same word may be interpreted differently in different cultures (even those words that are common to more than one culture), for example the word "pants" mean quite different things in American and English [103]. A another example is the different meaning assigned to colours by various cultures around the world, which has significance for the way that organisations manage their brands [54].
- Exactly the same principle concerns how we communicate concepts related to Enterprise and Solution architecture. We once worked on an application portfolio management project with another colleague and there were misunderstandings between us because one of us had a TOGAF [29] background and one an IAF [55] background, and so when one of us discussed the idea (sign) of a business service, the other got the wrong idea (had a different object in mind).

Understanding the interpretation process is crucial to the successful communication of ideas. The interpretation that we apply comes often from our surrounding culture, perhaps our family or society (in the case of colours or words), or in the case of Enterprise Architecture terminology, the prevailing nomenclature in our organisation, or the particular framework or notation that we have been taught.

The conceptual model quality framework discussed in 2.3.1.2 above, in particular Figure 25, raises some interesting concepts, including the distinction between 'social' and 'technical' actors that will be useful in this chapter.

In the context of Enterprise Architecture models, we might treat 'social actors' as those non-technical stakeholders that will be reviewing our architecture models to ensure that their concerns have been properly addressed. 'Technical actors' might include technical staff who use the architecture models as input to their own technical work (for example, technical architects, developers, testers and so on); as well of course as any systems that might consume the output of the models directly (for example, some models might be used to directly generate application code or database tables).

Architecture modelling languages work with the components comprising a system as well as relationships between them and so both need to be represented in the set of symbols used in our language.

For example, the ArchiMate 3.0 language specifies a large number of symbols (icons, shapes and colours) to denote the various language concepts. The symbols describe both a set of entities or concepts (e.g. business services, application components) and also a set of allowable relationships between entities (e.g. composition, aggregation).

As well as the explicit symbols, we also need to bear in mind the implicit messages (e.g. arising from spatial relationships) suggested from the general layout of the symbols, for example arising from the Gestalt "Gestalt Laws of Perceptual Organisation" [70]. These laws show how we experience larger scale constructs by combining simpler constructs according to certain laws or factors. For example, showing several elements in a similar visual style (e.g. colour) or location suggests a grouping of those elements, directly analogous to the 'group' concept in ArchiMate.

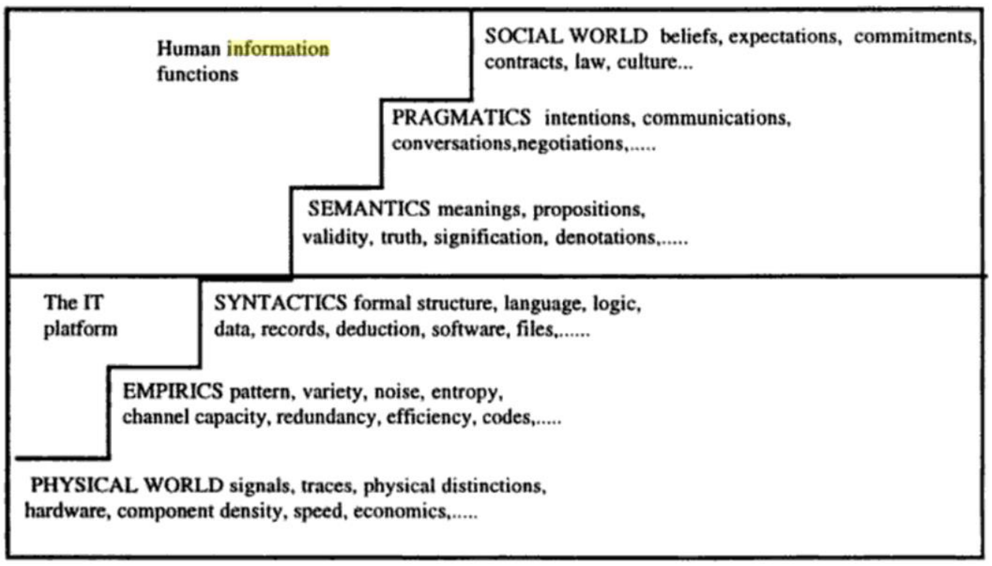
Thus, the elements in our model are both explicit and implicit.

Alongside the explicit and implicit symbols, we also consider models to have a *purpose* (what are they trying to convey) [45], which in an architecture context will depend on the role and specific concerns (interests) of the reader. The graphical symbols may well include several visual elements including shape, colour, icons, line types and endings), all of which have a particular significance in ArchiMate [14] as an example.

2.3.5 Semiological Ladder

The so-called "semiological ladder" from [100] gives us a fuller picture of the different levels of communication:

Figure 28 Semiological ladder between the Physical and the Social World



The definition of our formal modelling languages is really encapsulated in the SYNTACTICS layer, which (obviously) defines the syntax of our modelling languages. The SEMANTICS layer is where we have used our modelling language to represent something that we wish to communicate. However, the understanding of that communication – of that language – can be considered as part of the PRAGMATICS layer. It is here that we would wish to test whether or not the ideas that we wished to communicate to our stakeholders, and thus expressed in a particular 'sentence' in our language, have actually been properly received and understood correctly.

2.3.6 Model Quality for Enterprise Architecture

Quality attributes of Enterprise Architecture are examined in a paper from 2013 [104] that seeks to define the attributes of high-quality Enterprise Architecture products and services. The paper defines the quality of EA products and services as the extent to which the products and services meet the needs of the EA stakeholders. See also Bernus [105], which discusses quality again in terms of outcomes, for example *efficiency* being defined in terms of conveying the intended meaning and *completeness* being defined in terms of how the model can be used to create the intended interpretation. These outcomes could perhaps be employed to help shape the choice of language used for the modelling products and the set of viewpoints that sufficiently represent the interests of our particular set of stakeholders, but will not help us in defining the quality of a set of models, *given* a particular choice of modelling language and viewpoints.

These are obviously of relevance for one of our research questions (RQ1), as we are interested in creating a measure that enables us to calculate, in an objective manner, how close a particular set of EA models is to an ideal state (as we said in the introduction, is not misleading, is accurate as far as it goes, and is fit for the intended purpose).

Shanks [58] also suggest criteria for validating conceptual models: semantic accuracy, semantic completeness relative to the focal domain, no semantic conflict in model parts, no redundant semantics. In this paper, we are assuming that a set of criteria such as these has already been agreed upon as part of the definition of our 'modelling language', and asking how we can measure the quality of a set of conceptual (EA) models as a whole.

Semantic conflict is a major issue for EA analysis as stakeholders and participants frequently use terms in conflicting ways, for example using different terms to describe what is essentially the same concept (e.g. service and function) as far as those using the terms are concerned.

A framework to measure general conceptual model completeness was developed by Lindland [56] and subsequently extended by Krogstie [106]. The original paper considered model quality from three dimensions, and the subsequent model extended this to six dimensions taken from the field of semiotics. Although Lindland distinguishes between explicit and implicit statements ($\mathcal{M}_{\mathcal{I}}$ and $\mathcal{M}_{I}$), we will not be making that distinction in this work; we will be dealing purely with explicit statements (models) whose conformance to the language $\mathcal{L}$ can be explicitly tested.

Lindland [56], and later with Krogstie [48], suggest two key measures for measuring the quality of a particular model.

They consider syntactical quality, which relates to how well the model corresponds to the specified syntax. Using set theory notation (see section 3.3.2 below), they suggest that if a model contains a set of statements, then we could look for all statements within the model $\mathcal{M}$ that are not part of the agreed syntax (or language) $\mathcal{L}$; these would be the syntax errors, and the syntactical quality is obviously greater, the fewer syntax errors we have.

It appeared to us that, when judging the quality of an individual model[9], that this certainly provided an objective measure that could be applied and that furthermore it could be applied to a set of models as well as a single model. However, rather than looking for perhaps a ratio of correct vs. incorrect (from a syntactical point of view) statements within a single model, we could simplify this to just judging a model as a whole to be either correct or incorrect and thus our ratio would be the number of (completely) syntactically correct models, compared to the total number of models. This formed the basis of one of our metrics related to model quality.

This therefore leads directly to the question addressed in 4.2.1 below.

The same literature also deals with semantical quality; the degree of correspondence with the domain, again using set notation, for example saying that the existence of statements outside the domain ($\mathcal{M} \setminus \mathcal{D} \neq \emptyset$) means that the model and the domain (what we are trying to model) are not aligned.

As before, we saw that this could be applied to sets of models, not just a single model. We also saw that this semantical quality had two aspects to it. Firstly, are there propositions (statements) within

[9] The term 'model' in this literature is not restricted to a diagram; it is more abstract than that. However, in our Enterprise Architecture context, we am using it in a more specific sense. We would like to restrict it to a set of (graphical) diagrams (containing 'statements' comprised mainly of nodes and edge). This is not always easy, in particular when the particular set of models being 'tested' is a set of Wiki pages (our case study), rather than a set of architectural diagrams; however the principle still holds.

the model(s) that do not correspond to reality (in some sense, are 'not true')? This consideration led directly to the question addressed in 4.2.2 below.

Finally, are there propositions (that reflect the domain) that should have been present in the model(s), but were not (as per the discussion in 2.2.4.1 above)?

Applied at the level of entire models rather than individual propositions, are there models that we would have expected to be present, but are missing? This leads us to the question discussed in 4.2.3 below.

2.3.7 Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework

Although this paper [107] is focused primarily on the task of programming, using visual programming languages (the examples used are Prograph [108] and LabVIEW [109]), some of the cognitive dimension are also worth considering when building models that will need to be correctly interpreted by human 'technical actors' or, possibly, automated tools downstream of the architecture process. The interesting cognitive dimensions to consider from the paper are:

Consistency

When some of the language has been learnt, how much of the rest can be inferred?

Secondary notation

Can programmers use layout, colour, other cues to convey extra meaning, above and beyond the 'official' semantics of the language?

We consider these in our research on the comprehension of EA modelling languages (consistency in 7.3.1 below ; secondary notation in section 6.3.4.1 below).

2.3.8 Theory of Symbols

Goodman's theory of symbols suggests that, for notational systems, there must be a "one-to-one correspondence [with] their referent concepts" [110]. This clearly borrows from the terminology of semiotics. We make no further use of this, other than to note that we are testing, in our research into modelling language comprehension, to see where, for certain stakeholders, a particular notation is missing a reference concept (because they are not familiar with that concept).

2.3.9 The Physics of Notations

Moody sought to define explicit principles for evaluating, comparing and constructing visual notations [111]. His paper quoted Goodman's theory of symbols and described the following possible anomalies when this one-to-one correspondence did not in fact exist:

> *Symbol redundancy occurs when multiple graphical symbols can be used to represent the same semantic construct.*
>
> *Symbol overload occurs when two different constructs can be represented by the same graphical symbol.*
>
> *Symbol excess occurs when graphical symbols do not correspond to any semantic construct.*
>
> *Symbol deficit occurs when there are semantic constructs that are not represented by any graphical symbol. [111]*

It is one of the goals of this research to establish whether one specific language (ArchiMate) suffers from symbol excess as defined above, at least according to those interviewed.

2.3.10 Importance of Visual Notation in ArchiMate

Interestingly, a key figure in the development of ArchiMate confirmed that the emphasis, when designing the language, was on the semantics rather than the visual notation, which was perhaps something of an afterthought. Quoting from Mark Lankhorst,

> *We do not put the notation of the ArchiMate language central but rather focus on the meaning of the language concepts and their relations. Of course, any modelling language needs a notation and we do supply a standard way of depicting the ArchiMate concepts but this is subordinate to the architectural semantics of the language. [112]*

2.3.11 Improving the Effectiveness of Visual Representations

In a series of related papers, Moody used his 'physics of notations' theory to evaluate the cognitive effectiveness of a number of other languages including UML [113] and i* [114]. We have not been able to find any reference to any similar evaluation being done for ArchiMate, and thus consider this an open question to be addressed in our own research.

2.3.12 Designing Modelling Notation that Users can Understand

Popescu's PhD thesis [44] focuses on how to improve the design of modelling notations, rather than evaluating existing ones. The author examines the effectiveness of two IT-related (but not EA) modelling languages in detail (SEAM and i*), and recommends ways in which the effectiveness of these languages in communicating with non-IT stakeholders can be improved. He highlights confusion amongst his subjects (readers of models) of some of the concepts implicit in the two languages being studied, for example the ideas of system, service and process (section 4.4.1). The purpose of our research is to determine if such confusion also occurs with EA modelling languages such as ArchiMate.

Many of his recommendations deal with <u>sets</u> of models (for example, by using four story phases); his first key recommendation is "the relation with reality, by focusing on the readers' conceptualizations", by, for example, aligning the models to the readers' own views of reality, on what is visible to them, in terms that are natural to them.

There is thus a need in modelling for accommodations, defined as "conflicting interests that appear in situations that do not fully satisfy every stakeholder but are sufficient to enable actions" [115]. In this case the accommodation that needs to be reached is between the modellers, who may have their ideas about what makes a 'pure' model, and the readers (in particular the social actors) whose interests may be better served by it being comprehensible than being completely accurate and unambiguous.

This answers a related question: "how can we improve modelling languages" but does not help us understand "how good are our existing languages at communicating". An explicit aim of this thesis is to measure empirically and mathematically the effectiveness of communication of modelling languages, and hence suggestions for tailoring those languages with particular classes of readers.

This will be useful in the third part of our research where we consider the various techniques we can use to improve the comprehension of our models, especially with social (non-technical) actors.

2.3.13 Why they just don't get it

Two practitioners suggested in 2016 [116] the following six ways in which the creation of 'architecture visuals' (i.e. architecture models) can be improved:

(a) Focus the audience's attention by making the key object(s) stand out;

(b) Use Gestalt theory (see 2.3.14 below), for example grouping related items together;

(c) Use colours to highlight meaning (bearing in mind however that colours are perceived differently in different cultures, as discussed in our introduction);

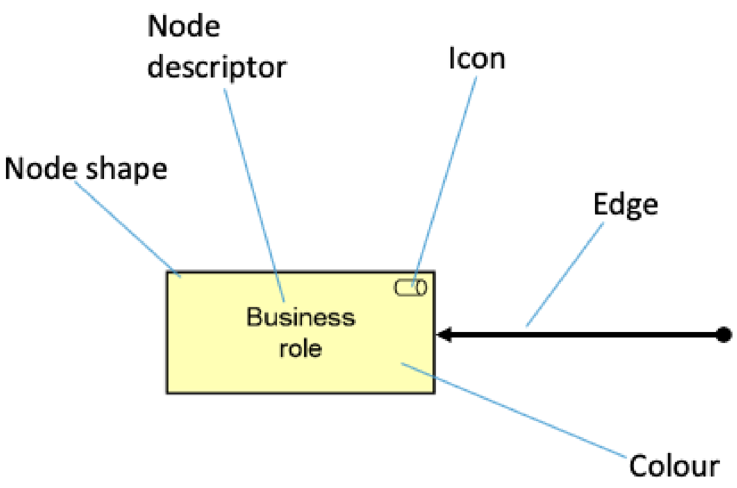
(d) Use icons, logos and images (their paper is critical of ArchiMate for its use of “more than 60 abstract symbols”), ideally from the organisation’s vocabulary, which will be easier to understand;

(e) Applying design thinking and graphic design; and

(f) Using hand-drawn sketches instead of always using words or software to create the pictures, as this can facilitate the building of a common understanding between participants

Just as colours do not always convey messages across cultures, as previously noted, abstract symbols do not either. A particular example of this is the ArchiMate symbol for a business role:

Figure 29 ArchiMate symbol for Business Role



In the above illustration, we see that the notation comprises a number of independent aspects, including:

(a) A shape (rectangle)
(b) A colour filling the shape (pale yellow)
(c) An icon (top right corner of the shape)
(d) A descriptor, a name inside the shape

The particular icon used by ArchiMate to denote a Business Role appears to rely upon an English homonym, whereby a cylindrical shape that could suggest the word “roll” is meant to suggest the completely different concept of a “role” as bearing responsibility for performing specific behaviour. This particular symbol is therefore specific to the English language (although it also works in Dutch), and is therefore is arguably a poor choice of symbol for what is meant to be a global standard. The inappropriateness of this symbol is highlighted as an example in the results section below, where it is mistakenly taken to refer to information (possibly because a confusion with the vertical cylinder sometimes used to denote a database in information systems).

Thus, use of ‘appropriate’ icons, logos and images that are relevant to the culture and experience of those reading the models is likely to increase comprehension of the models and/or recognition of some of the symbols.

We should also mention at this stage that it is not just the different entity types in ArchiMate that have different notations. As can be seen in Figure 128 on page 349, there are also different types of relations (grouped according to whether they are Structural, Dependency, Dynamic or Other), and each of these relations has a different appearance, again which convey a specific meaning to those that understand ArchiMate, but can be confusing to those that do not know how to ‘decode’ them. The focus of our work on language comprehension is on the elements, not the relations however, we will encounter them again in our practical work on model transformation where the different relation types start to feature in the notion of ‘derived relations’ (see for example section 8.3.1 below).

2.3.14 Gestalt Theory

The meaning that we derive from a particular model cannot necessarily be explained simply by the sum of its parts, and this is the idea behind what has been known as ‘structuralism’, as used by the so-called “Gestalt psychologists”. One of the products of their research is the “Gestalt Laws of Perceptual Organisation” [117]. These laws show how we experience larger scale constructs by combining simpler constructs according to certain laws or factors, including:

- Proximity: things that are close are grouped;
- Similarity: things that look similar are grouped;
- Common Fate: things that are moving in the same direction are grouped;
- Simplicity, or Good Figure (original German: “Pragnanz”, meaning regions of figurative stability): we see the simplest possible structure;
- Objective Set (original German: “Einstellung”): original grouping tends to persist;
- Direction (referred to as Continuation when discussed in [118]): we tend to see lines continuing smoothly rather than changing direction;
- Closure: perceiving items as closed shapes in preference to smaller pieces)
- Familiarity: elements that we normally associate together will continue to appear to be associated even in contexts where they are not

In our research on the comprehension and utility of EA languages, we will be following ArchImate in its use of the second of these factors – similarity – inasmuch as there is a grouping of type for elements that look similar.

2.3.15 Information Space

Boisot [119] created the concept of a three-dimensional “Information Space”, or “I-Space”, based upon three different perspectives on data: codification (giving form to phenomena or experience – we might say that this is the same as expressing an idea in a sign, the semiotic process as shown for example in Figure 58 below on page 213); abstraction (discerning the structures that underlie the forms – similar to semiosis in the other direction, going from sign to idea); and diffusion (*the deliberate uptake and use by a given population of data-processing agents [119]* – how the information is “taken on board” by the intended consumers).

Although we have not made use of this work in our thesis, it would seem to be a useful future avenue to pursue, in terms of seeing if there might be a correlation between the position of elements of our Enterprise Language in Boisot’s I-space, and the position of the same elements in our conceptual effectiveness (comprehension/utility) 2-dimensional space as shown in section 7.7.4. This I-space concept was also referred to in Bean’s paper [120].

2.3.16 On Understanding

As part of our research deals with how well people comprehend, or understand, certain concepts used in our modelling languages, we now outline what we mean by the term ‘understanding’.

Kvanvig, writing in 2003 [121], describes a number of kinds of ‘understandings (as summarised in [122]):

(a) *Propositional understanding* (e.g. I understand that I need to submit this thesis to earn my PhD)
(b) *Understanding-why* (e.g. I understand why I feel tired today)
(c) *Objectual understanding* (I understand X, where X is a subject matter, e.g. I understand Enterprise Architecture).

The article from the Internet Encyclopaedia on Philosophy goes on to discuss the concept of Factivity (S knows that p only if p is true) and the question of whether or not understanding is factive. It surveys views in the literature on weak or moderate factivity for objectual understanding, which is relevant for us because we are looking to measure, for our second research question, an objectual understanding by the interviewees of various concepts in our modelling language. The question is raised as to whether understanding a subject matter (e.g. a particular modelling concept) needs us to "know all propositions p about the concept where the propositions are true". In other words, are we looking for weak, moderate or strong (in the terms of the literature) factivity for understanding of these concepts?

Moderate factivity requires (in Kvanvig's view) that the subject's understanding about a particular matter is correct in the central beliefs about that subject; although the difficulty here is defining what constitute the central beliefs about (say) a particular element in our modelling language.

This concept of 'factivity' features both in our case study on model quality and our survey on the effectiveness of modelling languages.

2.3.16.1 Relevance to Model Quality

On one particular aspect of quality – accuracy – we can compare the idea of someone knowing the core beliefs (or propositions) about a subject matter, to the idea of quality being that the object, whose quality we are trying to measure, should be correct in its core assertions, so that for example a model of good quality should be correct and not misleading in its core 'messages' (e.g. if this model shows a business processes, along with all the applications used in the execution of the process, then there are no required applications missing from the model). We might call this 'moderate accuracy', borrowing from Kvanvig's terminology.

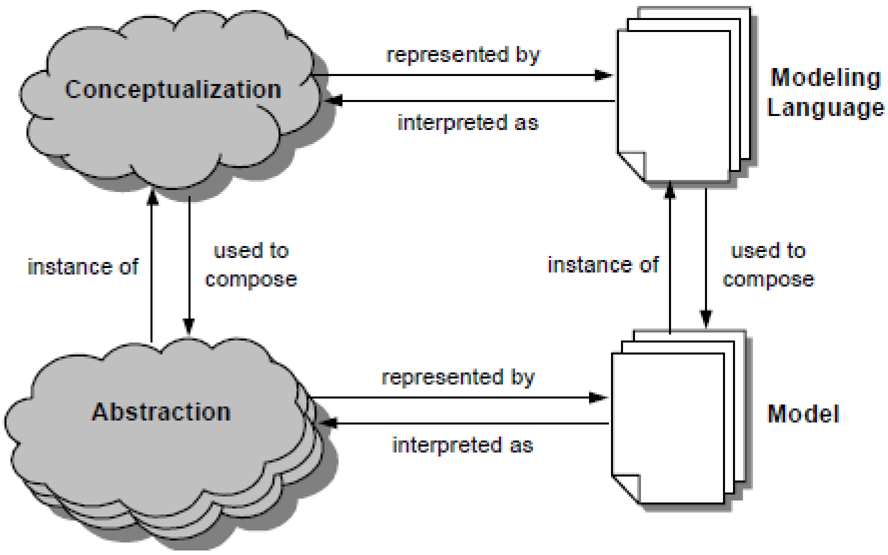
2.3.16.2 Relevance to Modelling Language Effectiveness

This concept of factivity will be used in our survey related to the effectiveness of modelling languages, in particular to help with the process of triangulation (see section 3.1.8.1.9 below)

2.3.17 On Ontology, ontologies, Conceptualizations, Modelling Languages, and (Meta)Models

Guizzardi [123] examines relations between Ontologies in a philosophical sense and ontologies as they are used in the computer science discipline. He examines, in the light of this, criteria that should be met by a modelling language in order to be considered suitable for use in a particular domain; and suggests a framework for evaluating and designing such languages. His paper presents the following model:

Figure 30 Relations between Conceptualization, Abstraction, Modelling Language and Model [123]



In this diagram, we have the following:

The Modelling Language: the set of entities (nouns) and relations (often verbs) comprise the language (an example being ArchiMate) for use on our models (in the diagrammatic sense of the word);

The Conceptualization – the sets of concepts that will be represented in some way within a modelling language;

The Model – the diagrammatic model that contains specific instances of these entities and relations that describes a particular domain;

The Abstraction – the subset of our domain that we want to include in a particular model.

Thus, in terms of our earlier illustration concerning pets, the set of concepts shown in the entity metamodel in Figure 3 above, provide our Conceptualisation; and our Language (which may or not use a specific notation for these various concepts) represents, or is a way of representing, those concepts. An example of an Abstraction, in the above terms, might be the fact that at the moment our cat is eating cat food in the kitchen; and we might choose to 'draw' a model that uses (perhaps) a 'pet notation' and contains specific instances of our concepts to represent a particular pet (our cat), a particular type of food (cat food) in a particular place (our kitchen).

The paper discusses some properties to be reinforced for what it refers to as an "isomorphic mapping between an ontology O representing a domain D and a domain language's metamodel" [123]; these properties include *soundness*, *completeness*, *lucidity* and *laconicity* (each of these having a formal definition). These are similar to some of Moody's principles for evaluating, comparing and constructing visual notations, based upon Goodman's theory of symbols. For example, a deficit in

Guizzardi's language completeness (i.e. not every concept in the domain ontology having a mapping to the language/notation) is broadly the same as Moody's concept of symbol deficit.

Some of Guizzardi's concepts find a direct echo in our own work, for example his concept of language completeness relates to situations where organisations do not yet have a complete set of languages established, as discussed in section 4.3.7 below.

The paper goes on to offer definitions (based upon philosophical constructs such as 'possible worlds'; for a discussion on the history of this, please refer to [124]) of exactly how models, guided by metamodels, 'stand for' or represent parts of various 'possible worlds'.

2.3.18 Opportunities and constraints: the current struggle with BPMN

Recker, writing in 2010 [125], offers a personal viewpoint looking at how Business Process Model and Notation [126](BPMN) has been used in practice, drawing upon surveys and interviews. The author set out to answer several questions, some of which have interesting parallels to our own research. These questions cover the capabilities and deficiencies of BPMN for process modelling practice and how it is used in practice.

Their work on capabilities and deficiencies was carried out using a theory of representation (citing work by Wand and Weber [127]) . Some of the issues found with BPMN are specific to a process modelling language (for example, support for a business rule specification), and thus are not be relevant to our own research, however there are some findings that could apply to any modelling language, for example what the research describes in a section entitled "Events, Events, Events, Events" as the "sheer abundance" of different event constructs. A related paper [128]

reports that users associate a significantly decreased ease of use with BPMN when confronted with multiple event types. This finding suggests that in the case of BPMN the ease of use of process modelling is sacrificed for sheer expressive power. The complexity that comes with selecting the 'right' event construct to use in a given process scenario points to a very basic design advice on basis of the user feedback: the simpler the better [125]

One difference to be noted in our 'target population' for research is that our own studies have focused as much on 'readers' of models as 'creators' of models; whereas Recker's work focuses purely on those that create models. However, it does raise a question as to whether having a large number of different concepts available (and used) in our modelling language might have as much impact on ease of use for readers of the models as for those creating them.

Recker's cites other related work [129] that includes an analysis of how much of the BPMN notation is actually being used in practice, based upon a sample of BPMN models. Based upon the frequency of use of the BPMN concepts, they split the concepts into four groups, with the somewhat provocative names "BPMN Common Core"(concepts found in >45% of models), "BPMN Extended Core" (>25%), "BPMN Specialist Set" (>3% by our estimation) and "BPMN Overhead" (<3%).

We have not found such an analysis done for ArchiMate models; this would be a fascinating piece of research, especially if the results could be compared against the comprehension and utility ratings for the various ArchiMate concepts that we determine later in our own research.

2.3.19 Natural Modelling

The authors of this 2011 paper [130] are making a case for a particular kind of modelling that they denote "natural modelling". Their stated goals are (a) to characterize what they mean as "natural modelling" and (b) identify promising research directions for "computer assisted natural modelling".

To achieve their first goal, they resort to the original meaning of the words employed (by for example giving two meanings of the word 'natural') and suggest some principles that would be incorporated in their characterization of natural modelling: (i) that it is collaborative, a social act, (ii) that it is symbolic, in other words, involves an interaction with symbols, and (iii) the language should be flexible and evolve over time.

They emphasise the human aspect of modelling, and include a very broad historical perspective of how modelling emerged, going as far back as the Stone Age. The purpose for this is link this modelling to "proto-languages", also concepts of "concerns" and "stakeholders". They argue that modelling in the modern age is in fact more restricted, giving the example of highly-structured 'models' on punched cards [131], and suggest that a focus on computerization of models is something of a backward step.

They suggest that "computer-assisted natural modelling" has two principal directions for improvements in current modelling technologies:

(a) *In terms of support for intuitive (non-intrusive) interaction, communication and knowledge creation in modelling, specifically within stakeholder-intensive modelling tasks; and*

(b) *in terms of inferring (metamodel), maintaining and manipulating links between models of different levels of formality and completeness.* [130]

They then highlight particular classes of technologies that show some promise in realizing their vision for "natural modelling", under the headings:

- Natural Interaction in Modelling

This deals with ‘natural’ ways of interacting with models, for example being able to draw free-hand, being able to combined mode (say, pointing and speaking), or physical interaction (haptic). One might compare this with the way that current EA modelling tools work – how people interact with them - and conclude that by these definitions, they are not ‘natural’!

- Flexible Metamodelling

  This deals with the topic of how a metamodel is produced, or emerges, and suggests that having a rigid metamodel can act as disincentive to the adoption of modelling tools (citing for example [132]; it also deals with the production over time of an “emergent metamodel” (inferring the language structure from examples), the adaptability of modelling languages for particular situations (or, we might say, enterprises), and the concept of an “emergent megamodel” that spans multiple metamodels (we might alternatively describe this as “spaning multiple modelling languages”).

- Collaborative Modelling

  This deals with the engagement and participation of multiple stakeholders in the production of a set of models. This can be contrasted for example with situations where a single (say, enterprise architect) creates models on their own.

Enterprise Architecture tools (as discussed for example in section 8.5 below) typically require a modeler (someone creating a model) to specify, at the time of creation, the precise type of an element: thus, in such tools, one is required to “create an Application Component called Retail CRM”, rather than having the flexibility in a tool such as Microsoft PowerPoint® or Visio ®, to “just create a box called Retail CRM” and worry later on about whether it should be an Application Component or something else. This can be viewed as an essential discipline; it can also be seen as an ‘inhibiter’ to “natural modelling” because people may know that something exists before they know its precise nature.

Arguably Enterprise Architecture tools could go some way to reducing the impact of this by allowing the type of an element to be specified or changed at a later time; not all tools provide this capability.

2.3.20 Pragmatics of Enterprise Modelling Languages: A Framework for Understanding and Explaining

In her PhD thesis [133], published in 2018, Bjeković’s research focuses on the use of enterprise modelling languages (of which Enterprise *Architecture* modelling languages such as those considered in this work form a subset), motivated in part by a number of perceived issues:

(a) The emergence of local (organisation-specific) 'dialects', or variants of the languages, which tend to resist initiatives to standardise such languages;
(b) Language standardisation (including visual notation), which some had assumed would aid communication between stakeholders, can sometimes result in generic standards than can be difficult and/or expensive to implement (quoting research by Egyedi [134]);
(c) Quoting Recker's work that we also discuss in 2.3.18 above [125], she suggests that "*While BPMN as a standard aims to accommodate both business and technical purposes of process modelling, the analysis reveals a language which contains a great deal of excessive and rarely used constructs geared towards advanced technical purposes of process modelling, at the same time missing adequate constructs for business-level concerns*". There is, in the language of Moody, evidence here of both symbol deficit and symbol overload.

The author makes the point that whilst analytic considerations require what they refer to as "a normative and formal specification of the modelling language" (which reflects our own experience in developing such analytics based upon EA models), this may not be appropriate when supporting the activity of human modelling, where pragmatics (often ambiguous and subjectivity) come to the fore; thus referring to a new field called "*modelling pragmatics*" (see for example [135]). She thus highlights a tension between the need for a formal structure for 'downstream' analysis and the need more an approach to modelling that is more natural and social (quoting for example [136] on the topic). She also cites previous research suggesting that decisions related to language design can be biased by the experience, knowledge and preferences of those creating the language, citing research by Frank [137].

Interestingly, Bjeković's methodological approach, in contrast to our own which is largely positivist but within recognised limits (see section 3.1.1 below), is socio-pragmatic constructivism, as formulated for example in Wyssusek et al. [138], which short paper suggests, when describing our "semiotic triangle", that the notion of an assumed correspondence between symbol (sign) and reference is itself a positivistic concept. Wyssusek is critical of positivism in this context: that there is meaning "already there" before "experience is to be made", or we might paraphrase, "before any signs are created that refer to it". Thus, the "meaning behind" certain signs might emerge during modelling; a theme recurring from the "natural modelling" paper [130] discussed in 2.3.19 above.

Bjeković suggests in her proposals for future research a number of ideas, one of which is linguistic function and language use for novice modellers, to look for example the effect of "… overload of relationships within modelling language such as ArchiMate …". Our work in this thesis on the effectiveness and utility of many of the concepts within the ArchiMate language is along these lines. A second suggestion for future research is the influence of notation; again, our own research into the effect of using the ArchiMate notation (as compared to just using colours) does precisely this.

2.4 Specialist Publications

2.4.1 Journal of the Association of Enterprise Architects

We searched in this journal for papers on either ArchiMate or Enterprise Architecture modelling, going back to 2010, looking for subjects (in the title, or in the abstract) that related to EA modelling specifically, or to the use of the ArchiMate language – and thus would be relevant to our own research – and found the following:

From 2019, a paper [139] proposing an alternative way of classifying artefacts within an Enterprise Architecture framework, dividing them into six groups according to (a) what they describe, and (b) how they describe. This may be of interest when it comes to looking for possible reasons why certain ArchiMate terms may be more understandable or useful than others.

From 2017, a paper suggesting that "*EA frameworks are far from the practical realities and typically adapted or simplified to fit the needs of specific organisations*" [140]. This suggestion accords with the ideas that prompted our research into the comprehensibility and usefulness of the content of such frameworks.

In the same year, a paper was published proposing a reference architecture for Enterprise Architecture repositories, focusing on "*the technology and tools portion of our profession*" [141]. This uses the ArchiMate notation to discuss common features and functions that we would require of Enterprise Architecture modelling tools (which include implementations of architecture repositories).

In 2015, two separate papers [142, 143] appeared in the journal, both proposing frameworks or models for measuring the value of Enterprise Architecture, and proposing specific metrics for this purpose.

Our own research proposes metrics for Enterprise Architecture *models*, which is obviously part of the activity of Enterprise Architecture and will thus affect the value obtained from Enterprise Architecture.

Bean's paper from 2013 entitled "Producing Enterprise Architecture Content that Counts" [120] covers some of the topics also covered in this thesis. It proposes a classification scheme for three different types (by purpose) of model, as well as covering topics such as the challenge of maintaining

EA content to ensure it remains “accessible, current, consistent and relevant”, and how tools and frameworks help in this regard.

The paper aims to highlight considerations for EA teams to consider in order *to (a) clarify EA’s purpose and contribution towards delivering value and (b) produce and distribute content (both structured and unstructured) that is relevant to its consumers and will contribute towards these aims* [120]. The latter of these considerations is relevant to our own research. Bean’s paper, published in the Journal of Enterprise Architecture, seems at time to be aimed at a conceptual level, dealing with the idea of ‘content’, and at times appears to be aimed at a less abstract level, dealing with idea of models. Bean does refer to the TOGAF distinction between diagrams, catalogues and matrices (see section 8.5.1 below where we discuss the same distinction) but does not make it clear that models (in this particular context) and diagrams are the same thing; this is an assumption easy to make, as the terms are often used synonymously.

Bean suggests that to be successful, models (and in context, this means models in the sense of diagrams) - should meet certain criteria. The following list of criteria was adapted by Bean in her peer-reviewed article from a book on Workflow Modelling by Sharp [144] which is a not an academic one, and so we will take care not to rely too heavily upon this list:

- *They highlight particular facets of the subject that are of interest while masking unnecessary detail;*
- *They employ conventions for adding detail and precision progressively;*
- *They have an obvious logical structure (e.g. starting point and direction);*
- *They use the minimum number of graphic elements that is feasible, so that they are easy to understand without continuous reference to a legend*

The two that are of interest to us are (a) highlighting facets while masking unnecessary detail, and (b) using the minimum number of graphic elements feasible[10]. Part of the purpose of our research is to understand which parts of a specific EA modelling language would count as ‘unnecessary detail’ and, using the minimum number of graphic elements feasible, again points us to the need to understand, of the elements that we use in our models, which ones are actually needed, and which ones may be technically accurate but not adding value to those trying to understand the model.

[10] The other two items in her first list are more specifically relevant for workflow and process modelling, which is not the subject of our own research

This theme is reflected in the next part of Bean's paper, which is a list of aspects of individual items to be considered. In the context of the paper, this again would appear to be describing models (diagrams) rather than more general content (such as might be represented in non-graphical form within an EA tool repository).

Table 9 Aspects of Individual Items of EA Content, from [120]

Aspect	Key Questions
Purpose	What is the main purpose of the content? What else might it be used for?
Audience	Who is the main audience for the content? What actions are expected of them? Is feedback welcome and how should it be provided?
Authority	Has the content been produced by people with appropriate knowledge and skills, and is it endorsed by people with the right level of power in the organization?
Quality	How accurate and complete does the content need to be? Is it sketching a broad direction or is it a more precise description of something concrete? How is it validated?
Lifespan/Update Frequency	Is the content expected to have an ongoing existence, or is it something that is more transient (e.g., related to a specific initiative). When, if ever, is it refreshed? Who needs to be notified when it changes? How will its usefulness degrade naturally over time, if it's not regularly refreshed?
Stewardship & Maintainability	Who will look after this content? Is it easy for them to change and do they have the information to do so?
Findability	How will the presence of this content be signalled and made accessible to people who can benefit from it or who should be using it? What other content is relevant to this content?
Consumability	Is the content easy to access and understand? Are special tools needed? If it is highly codified, is a "user-friendly" explanation available?
Applicability	How widely applicable is the content? Are there any exceptions and what are the restrictions on its use?
Consistency	Are the language and any diagramming notations consistent with other content? If not, will any differences cause problems and how will these be overcome? What key objects are related to objects in other content and how are the relationships maintained?
Security	What privacy and access controls are needed?
Analytic Capability	How easy is it to interrogate the content to perform commonly required analyses?

Although peer-reviewed by the Journal of Enterprise Architecture, this is clearly a practitioner's paper and we cannot necessarily see, with the above table, where the items came from; neither are these concepts further elaborated within the paper. We should add that we believe that these are very helpful questions to ask, even without academic rigour to back up the ideas and can foresee that, in addition to the quality aspect (which does appear in this list), a number of other ideas in Table 9 above could perhaps also become the subject of explicit metrics as assigned to Enterprise Architecture. We return to this theme in our reflections on this particular research question, in section 4.4.3 below.

Quality appears in this list, and Bean asks how accurate and complete the content needs to be. She does not however expand on this further, as this is only one of the aspects that she is considering, so it is not clear how this could be measured or improved upon. Also, in this list is the notion of Consistency which, according to Bean, includes the idea of the language and any graphical notations being consistent with other content. In the (we believe intended) context of EA models, it might have been helpful to have talked about having a *de facto* or *de jure* standard language with which the models might be expected to comply, rather than just saying that the notations used are consistent with other content.

Bean's accuracy and completeness concepts are two of the three concepts that we use in our research on EA model quality and her notion of Consistency, when taken in the sense of consistency with the agreed languages, is the third concept in our EA model quality research. She suggests that narrative presentations of EA are important, and this would fit in with Popescu's research [44] with its concept of having an ordered sequence of models to get across a message.

An earlier paper in 2013 [145], based upon a case study, also covered the topic of models (valuation and metrics) for assessing the value of Enterprise Architecture as an activity. In the same issue of the same journal, the same authors also published a paper on what were current trends in Enterprise Architecture Frameworks at that time. This survey examined many questions round the use of EA frameworks, including motivation, which was the most popular framework, the approach to frameworks (e.g. hybrid, popular/original, consulting firm – and over half the responses indicated a hybrid approach), and what was the most popular framework (answer: *the TOGAF Standard elements were mainly used as a process for building the technology layer, Zachman for taxonomy, Gartner for business architecture, FEAF for reference models and segment architecture, and DoDAF for governance* 2013 [145]). Regarding the popularity of frameworks, it should be noted that the publication date of 2013 is five years after the release of TOGAF 9.0 [31] which contained its first entity metamodel, and four years after the first release of ArchiMate, V1.0 [89].

Finally, the earliest paper in our systematic search of this journal is from 2010 [146], and addresses the topic of Reducing Communication Overhead in Enterprise Architecture. It discusses the concept of "friction" thus: "*Friction is a blanket term that represents diminished relevance to real-world problems as ideas are verbally passed among people or translated among cascading sets of EA artifacts in the course of documenting architectural specifications*" [146]. This has echoes in our work in Chapter 6, looking at how comprehensible and useful are the concepts in a particular set of Enterprise Architecture classes of artefacts.

2.4.2 Business Modelling and Software Design

The last ten years of the publications from these conference proceedings were searched, looking for papers on either ArchiMate or Enterprise Architecture modelling, in particular related to quality. We found one relevant paper, from 2017 [147], considering the relationship between the EA models and the modelled reality, which is one of the key metrics proposed in my paper on EA model quality from 2016 [51]. At the time of writing, we are unable to access this (see explanation in section 2.6 below), so are unable to explore further the relevance to our own research.

2.5 The relevance of ArchiMate

The need for coherent architecture descriptions was the subject of a 2003 paper [148]. Coherency here was seen as a way of joining up descriptions from different architectural domains, perhaps with different notations and languages, and it is here that we can see the beginnings of ArchiMate.

Lankhorst [149] provides a very thorough introduction to the topic of Enterprise Architecture, covering many topics including two key to this work: communication and language. Under the topic of communication, he touches on a Knowledge Transformation Process, of which we find echoes later in this work when we examine model evolution. He also covers the foundation and structure of the ArchiMate language. This book was first published in 2005, and so the ArchiMate metamodel contained in that work is somewhat simpler than the more recent revisions of the standard.

There is an interesting discussion in [150] about the implications of having a rich, complex metamodel: that a more powerful metamodel may be harder to learn, however may be more effective for experienced users. There is perhaps something to consider here when we look at the adoption of languages such as ArchiMate. Is it possible that we could get the best of both worlds – to have the power of the full language available for those that can make use of it, yet be able to use a subset of the language for those people that we do not expect to make use of (or even to be able to understand) the full metamodel?

The need to manage change across the layers of an Enterprise Architecture model (in the broader sense of the word), to maintain consistency and coherence over time, was addressed in a 2016 paper [151].

A more recent paper, on the topic of "Enterprise Architecture Mining" [152], mentions the difficulties of trying to analyse and draw conclusions from unstructured models (section 2.1.3, Automated EA modelling), as well as describing a number of possible sources ("mines") for Enterprise Architecture information, including humans (manual entry) as well as extraction from other IS systems.

2.6 Limitations to Literature Search

At the time of writing, most countries are in a state of lockdown due to the current Covid-19 pandemic. This has introduced some restrictions on access to the literature. Many publications that we would have liked to access are only available under conditions which preclude us accessing them at this time. For example, the proceedings of the Business Modelling and Software Design conferences are not available to us electronically, and require us to use an inter-library loan facility and then for us to pick those hard copies up in person from Reading University. This is impossible at the current time, as the University is closed. The articles are apparently available in the Bodleian library in Oxford in electronic format, however only from a computer in that library, and so again this is not currently possible.

Thus, there may be some relevant literature that we have not been able to survey satisfactorily.

2.7 Recap of Literature Review

We started this review by summarising relevant literature on the topic of Enterprise Architecture, explored some different 'flavours' or 'types' of IT Architecture (Enterprise Architecture being but one of those types), and then discussed what was meant by an "Architecture Framework". We saw that not all Architecture Frameworks provided the same kind of content: that different frameworks had different kinds of content (for example, one in particular contained a process for developing an architecture), and that in fact it was possible to systematically compare content from different frameworks using frameworks designed to compare such frameworks (for example, EAF^2 [86]).

We looked in detail at the entity metamodels underlying two leading Enterprise Architecture frameworks (TOGAF [14] and ArchiMate [15]) and saw whilst the TOGAF entity metamodel had not changed significantly in terms of the number of discrete entity types in the 11 years since it was released (increased from 32 to 37), ArchiMate's count of entity types has increased significantly in the same time period from 30 up to 56, which represents a significant increase in complexity. This led us to ask the question whether this increase in complexity was actually beneficial.

We then considered the topic of models, looking at their three core characteristics (mapping, reduction and pragmatism, according to Stachowiak [36]) . We considered the concept of quality, first of all in the abstract sense, noting a number of definitions (ISO definition [94] and satisfaction-based [95]), and then moved onto Heidegger's somewhat philosophical analysis of the concept of tool [96] (relevant given that our research touches, in particular in Chapter 8, on the notion of an Enterprise Architecture modelling tool).

We then move onto a systematic review of the "Senior Scholars' Basket of Journals [63], looking for recent literature relevant to our own research, so related either to EA model quality or to the effectiveness of EA modelling languages. This highlights two pieces of research, one related to factors and measures of EA management success [98] (which serves to demonstrate the importance of producing high-quality models) and one that focuses on EA artefacts [99](as opposed to individual entities) that have proven useful in practice.

We then discuss more general (conceptual) model quality, highlighting the work done by many in this space including Lindland [56], Krogstie [48] and Stamper [100]. These works propose a framework for defining, measuring and achieving quality in a number of ways. We discuss a direct application of some of these concepts to the specific field of Enterprise Architecture models, reusing some of the notation and concepts from the original works, and giving examples of syntactical and semantic (truthfulness) errors. We also note a distinction between 'social actors' and 'technical actors' which is useful to us when categorising our survey results later on and discuss the relevance of the decoding or interpretation of signs (semiotics) when conveying information in "architecture languages".

We discuss aspects of model quality more directly tailored towards Enterprise Architecture (e.g. Bernus [105]), and then move onto some of the more technical aspects of Lindland's [56], Krogstie's [48] frameworks for measuring model quality, and their potential application to sets of Enterprise Architecture models; this gives us an indication of a gap that we can fill, related to models that should have been present but were not.

We then cover a number of theories related to model content, including cognitive aspects (from a paper related to visual programming languages [107]), Goodman's theory of symbols [110] (which touches on the subject of referent concepts for the symbols), Moody's "Physics of Notations" [111] (dealing with symbolic redundancy, overload, excess and deficit – all interesting when dealing with architecture modelling languages), a review then by Moody, using his "Physics of Notations", of various languages (not including ArchiMate), and then a PhD thesis by Popescu dealing with the design of modelling notations [44] that highlighted confusion in understanding of some of the concepts in the languages he was testing (again, not including ArchiMate).

We reviewed a practitioner's paper from 2016 [116] suggesting a way of improving Enterprise Architecture models, which specifically criticised ArchiMate for its large vocabulary of "more than 60 abstract symbols", and noted ourselves that the ArchiMate notation (which according to one of its founders was not the highest priority of the team developing it [112]) relied to an extent upon cultural concepts (e.g. puns in certain spoken languages) that would not translate well into other languages and cultures, as well as (we would discover later in our survey) clashing to an extent with other symbolic notations.

We then touched upon some other concepts relevant to models, such as Gestalt theory [117] and Boisot's three-dimensional "Information Space" [119], before moving on to examine the philosophical concept of “understanding” [121], which forms a key part of our later research into the understanding of Enterprise Architecture models and concepts.

We then move on to review publications over the previous 10 years from specialist journals known to us, starting with the Journal of Enterprise Architecture. This contained several pieces of research tangentially relevant to our own, including features and functions relevant to Enterprise Architecture modelling tools [141], frameworks for measuring the value of Enterprise Architecture papers [142, 143], and a more directly relevant paper by Bean called “Producing Enterprise Architecture Content that Counts” [120] that covers ideas immediately relevant to our thesis (such as masking unnecessary detail and minimising the number of graphical elements) as well as others that we hope to use in future publications to extend our own Enterprise Architecture model quality framework.

We then return to a discussion of ArchiMate, the importance of coherent joined-up architectural descriptions, the implications of having a complex metamodel [150] and the difficulties when trying to “mine” information from unstructured models [152]. This topic of metamodels and modelling languages is then given a more sound philosophical basis in the work of Guizzardi [123] who examines what it means for a modelling language to be suitable for use in a particular area, and defines some properties or characteristics for the mapping between an ontology and a domain language’s metamodel. Next, we look at a study by Recker [125] that was carried out on another modelling language (BPMN) [126], looking both at its theoretical construction (using a theory of representation), and also in practice at how much of the language actually gets used (using a survey). He opines that there are too many different elements in the language (his particular focus for this being the abundance of types of events), and reports an earlier finding that with BPMN, having more types of elements available for use is associated with a significantly decreased ease of use by those creating models. Although his survey was only for those creating models, there are obviously parallels with our own questions about the large number of elements in the ArchiMate language.

Our next piece of literature surveyed is on the topic of “natural modelling” as discussed by Zarwin et al. [130], where they suggest that modern computer-assisted modelling has become more restricted and less ‘natural’ for those carrying it out; they suggest two specific areas for improvement, one related to a better way for stakeholders to collaborate when creating models, and one related to the way that metamodels can be inferred from models, and perhaps not be so constrained in the early stages of models formal metamodels.

Our final piece in this literature review is another PhD thesis, by Bjeković, published in 2018 [133]. This raises similar themes, suggesting a more “socio-pragmatic constructivist” (SCP) [138] to the use of modelling languages, and culminates in an interesting set of proposals for future research, two of which link to the work done in our own thesis.

Chapter 3. Methodology

3.1 Epistemology

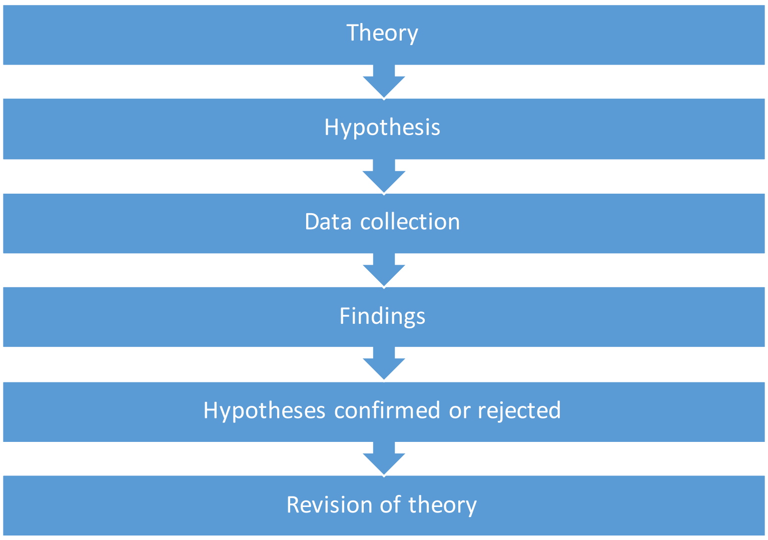
3.1.1 Theories of Research

3.1.1.1 Deductive vs. Inductive Theory

These are two ways of looking at the relationship between theory and research. Deductive approaches use theory to guide the approach whereas inductive approaches tend to form theories as an outcome of their research.

In deductive approaches researchers use, as their starting point, existing theories, or perhaps extensions to those theories based upon certain reasoning, and then construct a *hypothesis* that they wish to test. This deductive approach, as it is normally followed, is outlined below:

Figure 31 The process of deduction (adapted from [153], page 11)



Thus, deductive approaches collect and analyse data in order to test hypotheses formed from existing or new theory. A classic example of this would be the hypothesis issued over 100 years ago by Einstein concerning the effect on the orbit of the planet Mercury due to his newly-published theory of General Relativity [154]. Data collection in this example took the form of astronomical measurements, which indeed confirmed the hypothesis, and thus strengthened the case for his new theory of gravitation.

With deductive approaches we are forming hypotheses, asking ourselves the ‘why’ question, at the start of the process, and then carrying out the measurements or observations to see if our hypotheses were correct.

By contrast, with inductive approaches, we use research to help us find out why things occur and so the theories come at the end of the research process rather than the start. Heit uses the example in his paper of a traumatic event, a burglary, which is a data point from which the victim may form theories about the probability of them being burgled again [155]. This is inductive: the theory concerning the probability of a future event followed the observation (of being burgled the first time).

Closer to our own research, we might be seeking reasons why people find certain concepts difficult to understand or recognise, and then use results obtained to help us form theories about why that might be.

3.1.1.2 Positivism vs. Interpretivism

Various dictionaries and encyclopaedias offer several definitions of the term ‘positivism’, of which the following are typical:

- a theory that theology and metaphysics are earlier imperfect modes of knowledge and that positive knowledge is based on natural phenomena and their properties and relations as verified by the empirical sciences [156]
- A philosophical system recognizing only that which can be scientifically verified or which is capable of logical or mathematical proof, and therefore rejecting metaphysics and theism [157]

This concurs with the definition in Bryman and Bell that positivism is “… *an epistemological position that advocates the application of the methods of the natural sciences to the study of social reality and beyond*”. ([153], page 15). In all of these definitions, we see a form of reasoning employed (very successfully) in the natural (empirical) sciences, being applied in other fields as well (social science, philosophy, theology and so on).

The word “positivism” is taken from a word coined in French, “positivisme”, by the French philosopher Auguste Comte (1798-1857), known also for being the father of social science [158]

Positivism suffers from a number of criticisms, not least that it is self-refuting. For example, is the statement “We should recognize only that which can be scientifically verified or which is capable of logical or mathematical proof, and therefore reject metaphysics and theism” *itself* scientifically verified or capable of logical and mathematical proof? If not, then why should we accept it as being valid? It undercuts itself. It is an *a priori* argument, not itself based upon evidence but taken as an assumed, underlying philosophical approach which although popular early in the 20th century has since been largely discredited [159]. Recognising the limitations of positivism, however, we still feel that it has something to offer in terms of how we carry out measurements of quality.

Interpretivism is seen as an alternative to positivism. At its heart, it recognises that we have to respect the difference between the objects studied in natural science and the people who might do the studying; that the social scientist has to grasp the subjective (as opposed to objective, or observer-independent) meaning of social action [153]. Even natural science, which could be seen by some as the bastion of positivism, has sometimes to take into account the action of the observer; consider the matter of the Copenhagen interpretation of quantum mechanics and the disputed role of the (personal) observer in the collapse of the quantum wavefunction [160].

3.1.1.3 Objectivism vs. Constructionism

These are social ontological theories – theories about what actually exists in terms of social phenomena (e.g. norms, cultures, organisations) – and like the other pairs of terms described above, are opposites.

Objectivism asserts that social phenomena have an existence independent of the social actor – of these people involved in the phenomena. Conversely, Constructionism asserts that the social phenomena is inherently built from and related to the social actors.

These two theories have very important implications. By way of example, take the example of ethics. Objectivists in this area will argue that there are objective values and duties: that certain actions should be deemed to be right and wrong, irrespective of what people think. Constructionists, conversely, would argue that there is no objective right and wrong; that our views on what is good and bad is shaped by our culture and inextricably linked to it. For a defence of an objective view about obligations (duties), please consult [161]. Our own view is that certain things are objectively true, and not socially determined. In some cases this is trivially true: when we fly in an aircraft, we are relying upon ‘hard science’ and we are not concerned (from a safety of flight perspective) of people’s views or opinions about aircraft and the aviation industry in general.

It has also been argued that it is unnecessary to choose between Objectivism and Constructionism on theoretical grounds; that arguments about what is most appropriate have been rendered obsolete by developments in experimental methods as well as computer and mathematical models and that what remains are two types of theory which are both useful in their own way [162].

3.1.1.4 Reflections on Research Theories

We have reviewed a number of theories:

- Deductive vs. Inductive
- Positivist vs. Interpretist
- Objectivist vs. Constructivist

These decisions that we make about theories to use in our research will suggest a suitable research technique.

3.1.1.4.1 Is our Research Deductive or Inductive?

Either could be possible in our case, certainly regarding the second research area regarding the comprehension and utility of EA concepts. If we wanted to ask the question *why* certain concepts are, or are not, understood, then an inductive approach might be useful. However, that is not our question: we are asking *how well* certain concepts are understood, and more than this, *can we objectively measure* how well certain concepts are understood. We have a hypothesis that it is possible to make reasonable measurements about the overall (average) comprehension and utility amongst various groups of people and it is this that we wish to test. Thus, our approach in this area is more deductive than inductive. For the other two questions, people are not involved (apart from ourselves!) and so again those would be deductive.

A reasonable question to ask, however, would be “why make numerical measurements about comprehension and utility in the first place?” The primary reason is the desire for simplicity of analysis and ease of applicability to real-world scenarios. If we know that certain concepts score poorly for all our respondents, then we believe we will be able to generalise and as a consequence, advise practicing IT architects to avoid using these concepts when communicating with non-technical stakeholders. In a way, we don’t actually care why certain stakeholders don’t understand the concepts; it’s enough to know that they don’t.

3.1.1.4.2 Is our Research Posivitist or Interpretist?

Despite the concerns that we have laid out about the over-reach of positivism into the social and other sciences, we are still going to be applying largely a positivist approach in terms of our data collection. For the most complex data set – that related to our second research question – we are trying to measure our interviewees comprehension against a well-defined set of concepts and we are trying to avoid, as much as possible, having the clarity of that measurement altered by social constructs. Whatever people might feel about, for example, what motivates businesses, we have strict definitions (in our modelling language) against which we are testing their understanding.

Thus, overall, our research is positivist.

3.1.1.4.3 Is our Research Objectivist or Constructivist?

This is not a clear-cut decision. One could argue that we are making measurements that are attempting to be clear-cut and attempt to find data points that would be the same whatever people's personal opinion or culture, thus leaning towards objectivist. Given recent literature on this topic, questioning whether this is even an important philosophical question to ask any more [162], perhaps it does not particularly matter.

Thus, overall, we will be mainly deductive, positivist and objectivist, although there may be aspects that cross the boundary into other approaches.

3.1.1.4.4 Three Research Questions, Three Methods

To avoid confusion, we will highlight here the fact that we are anticipating using a different approach for each of our three research questions.

3.1.2 What is Quantitative Research?

Quantitative research fundamentally deals with numbers that can be handled mathematically, for example using statistical techniques. Advertisements that claim "9 out of 10 cats prefer…" are (hopefully) based upon statistical techniques and thus quantitative research.

Quantitative data is simpler in some ways to deal with, as it can be analysed in so many ways. It is not subject to different points of view; one just has data points which can be 'plugged into' mathematical formulae to produce the required analysis.

Quantitative data can be described as 'hard' in its data collection. It is more concerned with measurements – answering questions like "what", "when", "how much" and so forth. Quantitative research methods will typically involve surveys (when dealing with people) and experiments (when dealing with the natural world).

Both quantitative and qualitative (the latter being dealt with in section 3.1.3 below) methods are dealt with in a book on organisational research published in 2005 [163]. The chapter on quantitative research introduces the subject with the text:

Quantitative methods and the scientific method are the foundation of modern science. This approach to research usually starts with a specific theory, either proposed or previously developed, *which leads to specific hypotheses that are then measured quantitatively and rigorously analysed and evaluated according to established research procedures. [164]*

The context for this particular academic book is Human Research Design (HRD), and the authors state that HRD researchers use both quantitative and qualitative methods. Their book suggests that the two methods are often quite powerful when used together, which is precisely what we intend to do with our second research question.

The authors suggest that quantitative research can be viewed as the following five-step process:

1. Determining the basic questions to be answered
2. Determining the participants in the study
3. Selecting the methods needed (variables, measures, overall design)
4. Selecting analysis tools
5. Understanding and interpreting the results

This process is helpful in terms of ordering our own research and we make use of it in later sections of thesis (in particular sections 5.1 and 7.1 below). Our final research question is answered by a combination of theoretical analysis and experiment, and so does not involve quantitative methods.

The authors also mention in their discussion of quantitative methods that they can be exploratory, in other words, used to “discover relationships, interpretations, and characteristics of subjects that suggest new theory and define new problems”, and that when used in this way, research questions are used in place of specific hypotheses.

This is indeed the nature of much of our research in this thesis.

3.1.3 What is Qualitative Research?

Qualitative research deals with concepts that cannot be reduced to ‘mere’ numbers. The data thus gained is more complex to analyse, not being immediately amenable to mathematical analysis, however, is sometimes capable of answering the all-important “why” question.

In a sociology journal published in 2019, Aspers and Corte offer the following definition:

An iterative process in which improved understanding to the scientific community is achieved by making new significant distinctions resulting from getting closer to the phenomenon studied. [165]. This definition is interesting because it includes a number of aspects, perhaps we might say differentiating factors, from quantitative research. It focuses on the benefit (improved understanding), how it is achieved (significant distinctions), and the cause of the improvement (getting closer to what we are studying).

In contrast to the clear process put forward for quantitative research in the previous section, the design of a pure qualitative study is much more iterative. In his chapter on Designing a Qualitative Study, Maxwell suggests that neither a simple progression of stages, or indeed a more complex 'flowchart' of steps, "adequately represents the logic and process of qualitative research" [166] and that qualitative research requires a less restrictive, broader concept of design than we might expect for quantitative research.

If I ask a group of people to put up their hands if they like working in an office, then I am clearly capturing individual data points each of which will be either a yes or a no, and thus quantitative. However, if I interview a group of people and ask them how they feel about working in an office, I am likely to learn the reasons for their preference, not just their preference. This is qualitative data. Analysing such data takes more time. It can sometimes be automated to an extent, for example by searching for key words that may be used (for a description, please refer to chapter 5 in Kuckartz's book [167]).

Maxwell presents in his chapter, what he describes as, a model *of* as well as a model *for* research. Remembering that this is not meant to be executed sequentially, his model for research design consists of five components:

1. Goals – why are you doing the study?
2. Conceptual framework – that will form the backdrop and foundation for your research
3. Research questions – what do you want to learn or answer?
4. Methods – how will you conduct the research?
5. Validity – how do you know that your results will be valid?

3.1.4 Quantitative or Qualitative?

The first choice we are faced with, when it comes to selecting our theory of knowing for our research, is whether to opt for a quantitative or qualitative approach. The authors of one particular paper [168] go so far in the title and content of their paper as to label each of these approaches as dogmas. They suggest that rigid epistemological distinctions between these two methods themselves rest on positivistic dogmas, which tend to suggest that qualitative research is value-laden (as in, biased), and that quantitative research is not.

The differences between quantitative and qualitative research strategies is summarised in the following table:

Table 10 Fundamental differences between quantitative and qualitative research strategies (adapted from [153], page 27)

	Quantitative	Qualitative
Principal orientation to the role of theory in relation to research	Deductive; testing of theory	Inductive; generation of theory
Epistemological orientation	Natural science model, in particular positivism	Interpretivism
Ontological orientation	Objectivism	Constructionism

Our background and training is for certain closer to the natural sciences and we have elected for the majority of this research to focus upon quantitative methods, for example to use mathematical techniques to construct frameworks that can then be tested and evaluated. If we were to elect to base our research on qualitative methods, then we would not then have access to the same mathematical techniques and the concept of measurements.

However, we want to leave the door open, as it were, to any information that might be gained, in the course of our survey (related to the effectiveness of modelling languages), about why people have certain beliefs or understandings. Thus, we are mainly focusing on quantitative data, but will capture some qualitative data within that structure or framework. This puts us, potentially, into the realm of mixed methods, but with a definitive bias towards quantitative methods.

3.1.5 Mixed Methods

One of the most important tasks of Enterprise Architecture is to communicate with people and the effectiveness of that communication will undoubtedly depend, to an extent, on the previous experience of those people. In this question of effective communication, we are primarily seeking to understand *that* a particular person finds a model easy (or hard) to understand. However, in the process of gathering data about this, we would like, if possible, to gain some understanding of *why* this is (but secondary to the main question).

Therefore, our methods will need to include a mixture of both and thus we need to look at some form of mixed methods research. The emphasis is going to be more on quantitative then qualitative, because of the amount of mathematical modelling and analysis that we will employ, and the primacy of the “that” as opposed to “why”, and this means that our epistemological basis is closer to positivism.

We should also note that of our three primary research questions, only one of them (RQ2) will involve gathering partially unstructured data from people (during interviews) and thus be suitable for any kind of qualitative techniques.

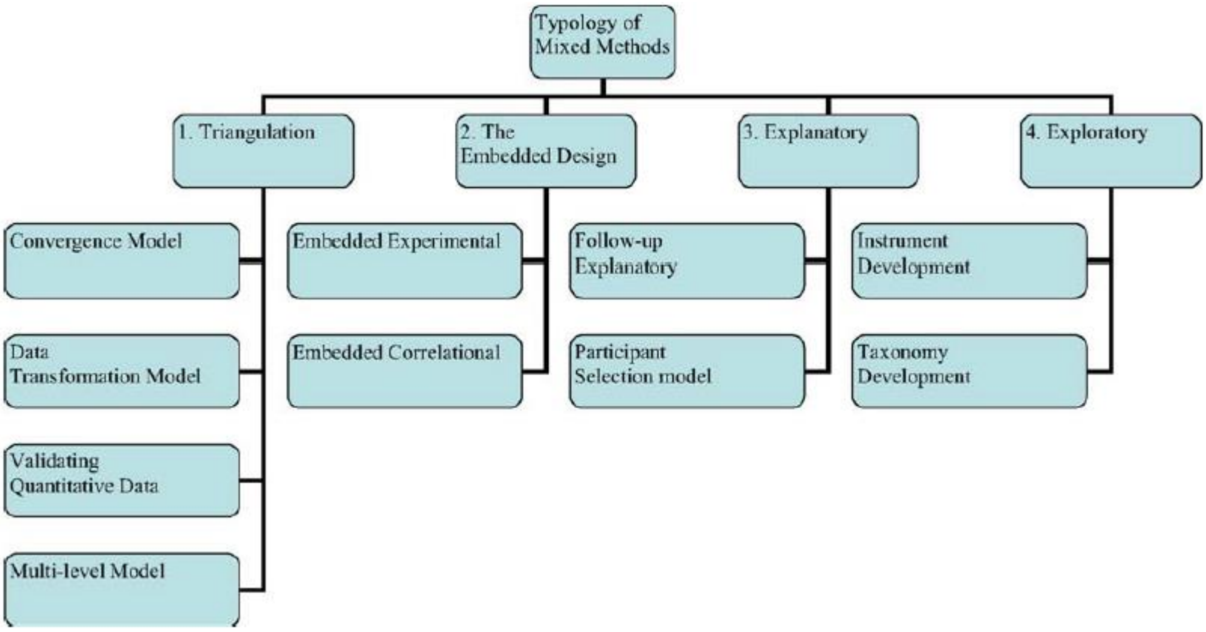
Doyle et. al. [169], writing in 2009, give an overview of mixed methods research and, quoting an earlier paper, repeat the definition as "'research in which the investigator collects and analyses data, integrates the findings and draws inferences using both qualitative and quantitative approaches or methods in a single study" [170]. They support the view of the earlier paper that a blend of the two approaches is to be encouraged for pragmatic reasons. They summarise the main reasons proposed for mixed methods research as being:

- *triangulation* (alignment between quantitative and qualitative data)
- *completeness* (getting a fuller picture)
- *offsetting weaknesses and providing stronger inferences* (suggesting that the combination of approaches can avoid the weaknesses of either)
- *Explanation of findings* (explain results from one mode with results from another)
- *Illustration of data* (qualitative approach can illustrate quantitative findings)
- *Hypothesis development and testing* (quantitative phase can test hypothesis developed during qualitative phase)
- *Instrument development and testing* (similar to previous point; qualitative phase can suggest items then tested in quantitative study)

Given the subject of our second research question, this seems a suitable approach to use, although we expect the qualitative aspects to be subordinate to the quantitative aspects. The interviews involved in this particular piece of research will primarily produce quantitative results, for example the graphs of comprehension vs. utility shown in section 7.7.4 below. However, we are also looking for clues as to *why* people failed to comprehend particular concepts, and this will be difficult to capture using purely quantitative techniques. We were also looking for correlations between the quantitative data and the background and experience of those interviewed; however, this did not prove possible with the volume of data available.

A book cited by Doyle goes into a lot more detail and includes the following illustration, a typology of mixed methods research:

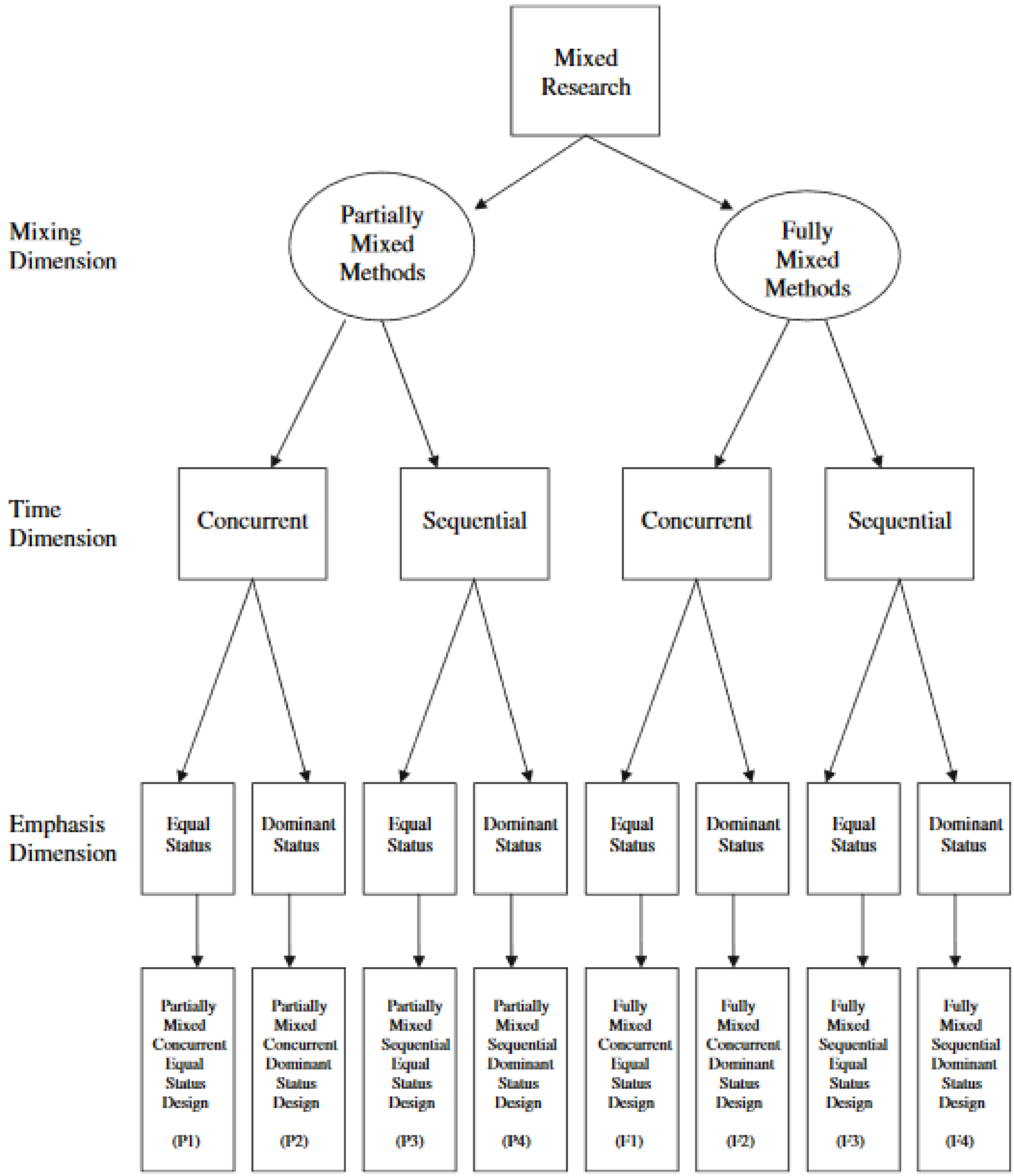
Figure 32 Typology of Mixed Methods Research [171]



Our research in this area falls in the category of embedded design [172], where one method is dominant and the other supports the first; more specifically we will be using the embedded experimental model, where the quantitative method is dominant (capturing comprehension and utility scores) and the qualitative aspects ("what do you think led you to that conclusion") is secondary.

Doyle's paper also cites another one which provides another suggested typology of mixed methods design:

Figure 33 Typology of Mixed Research [173]



This typology has a number of characteristics, the combination of which results in a particular designation; and so we need to consider: (a) are we using partially or fully mixed methods; (b) are we concurrent or sequential; and finally (c) do the quantitative and qualitative aspects of equal or unequal status?

Recall that we have three research questions. The first and third are going to be settled by data capture or observation, and thus are really along the lines of an experiment. The second, which involves capturing data from people, will involve interviewing them and asking them to provide their understanding of a range of concepts. Any ‘reasons why’ will be captured at the same time as their numeric data (e.g. “I fully understand”; or “I haven’t a clue”).

Now according to this typology, a research method is partially mixed if "the quantitative and qualitative phases are not mixed within or across stages" [173]. The analysis will be done separately, and indeed we are placing a lot more emphasis on the quantitative data and not necessarily expecting to capture as large a volume of qualitative data so for this reason, we would characterise this as being partially mixed.

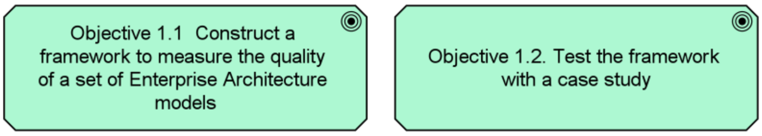
The fact that the quantitative data is of more importance to us signifies that this would be characterised as a dominant design; and finally, the fact that we hope to capture any 'why' information at the same time as the quantitative aspects would lead us to categorise this as 'concurrent'. Thus, according to this schema, our research on the effectiveness of a modelling language would be characterised as Partially Mixed Concurrent Dominant Status Design.

3.1.6 Data Collection Considerations

We have described previously how we are mainly expecting to use quantitative methods but supplemented, in the case of our second research question, with some qualitative data. We will now focus more clearly on what kind of methods are appropriate for collecting data for our research

To answer the question RQ1 (*How can we measure the quality of a set of Enterprise Architecture models?*) we have explained that we have two objectives: the first is to construct a theoretical framework, and the second is to test that theoretical framework:

Figure 34 Objectives related to RQ1



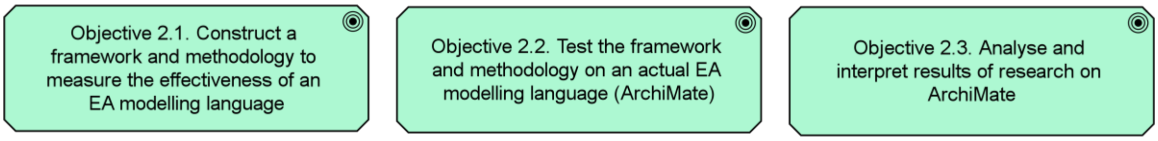
We will shortly be explaining our rationale for using a "case study" to address this objective.

To test that framework requires some in-depth analysis of real-life data. Our aim here is to show that it is possible to use the framework to measure quality. We are not asserting that it can be used in all cases; simply that it is possible to use it under certain circumstances. Therefore, it is sufficient merely to demonstrate that it can be used in a single situation, as long as there are clear indicators that show us that this single situation was not a 'fluke' or some kind of anomaly.

We do not need to use it repeatedly. We expect that using it successfully on one occasion will provide valuable information as to how it can be improved and used successfully on other occasions.

Moving on now to RQ2 (*How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?*), we see that this is substantially different. We are again constructing a theoretical framework, then testing that framework in real life, and finally analysing the results:

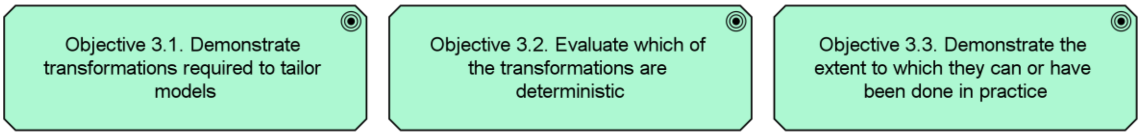
Figure 35 Objectives related to RQ2



As we will see, as we discuss the research in more detail, we are expecting to use mostly quantitative data here, 'decorated' with some qualitative data (hence a kind of 'mixed-methods' approach). What we are measuring here, we want to have a degree of statistical significance. The data points that we are collecting are much simpler (ultimately just two numbers), and we want to collect that data from many people. This will ultimately lead to a different type of data collection from the previous research question.

Finally, our last question is RQ3 (*Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?*); our objectives in this area again are partly theoretical (list types of transformations, determine which of them are deterministic), and then by means of practical demonstrations, prove that at least in some circumstances, that they can be automated:

Figure 36 Objectives related to RQ3



3.1.7 Comparing the Approaches

We can see therefore, differences in our approaches to these three questions. Answering the first one (in particular Objective 1.2), we are just dealing with 'brute facts'; we will not be collecting data from people, but just looking at existing EA artefacts. Thus, we will not gain any understanding *why* the artefacts are in the state that we find them. However, the analysis is easier as a result; it should be straight-forward to demonstrate that we can indeed use the framework in practice. Our research question is, in any case, not trying to answer why the EA models are in a certain state and so not having the 'why' question answered is no hindrance to us answering our question.

For the second one, our primary aim is to measure understanding, across a number of people, of certain concepts used in Enterprise Architecture. We are primarily concerned with numeric measurements of understanding and usefulness, across as wide a range of people as can be managed in the time available; information about why they do or do not understand is secondary. Thus, our data points (“features” in the language of Table 11 below) are very simple, but we will collect this data from many people. Thus, we will benefit again from the ease of manipulation of results collected, and in addition gain a limited understanding of the “why” question.

For our third question, the only data collection required is by way of experimentation and demonstration that some theoretical techniques can in fact be executed (automatically) in reality. This only requires us to demonstrate this once, for each technique in question.

The nature of the differences between the above three requirements for data collection lead us to conclude that the methods for each will need to be different.

3.1.8 Case Studies, Experiments and Surveys

When examining the literature on case studies (given that we expect to use a case study as part of our portfolio of methods), we found that an edited book, published in 2000, summarising the key issues and texts related to case studies, presented a very useful summary of the differences between case studies, surveys and experiments

Table 11 A schematic comparison of case study with experimental and survey approaches [174]

Experiment	Case Study	Survey
Investigation of a relatively small number of cases.	Investigation of a relatively small number of cases (sometimes just one).	Investigation of a relatively large number of cases.
Information gathered and analysed about a small number of features of each case.	Information gathered and analysed about a large number of features of each case	Information gathered and analysed about a small number of features of each case.
Study of cases created in such a way as to control the important variables.	Study of naturally occurring cases; or, in ‘action research’ form, study of cases created by the actions of the researcher but where the primary concern	Study of a sample of naturally occurring cases; selected in such a way as to maximise the sample’s representativeness in relation to some larger population.

Experiment	Case Study	Survey
	is not controlling variables to measure their effects.	
Quantification of data is a priority.	Quantification of data is not a priority. Indeed, qualitative data may be treated as superior.	Quantification of data is a priority.
The aim is either theoretical inference – the development and testing of theory – or the practical evaluation of an intervention.	The main concern may be with understanding the case studied in itself, with no interest in theoretical inference or empirical generalisation. However there may also be attempts at one or other, or both, of these. Alternatively, the wider relevance of the findings may be conceptualised in terms of the provision of vicarious experience, as a basis for 'naturalistic generalisation' or 'transferability'.	The aim is empirical generalisation, from a sample to a finite population, though this is sometimes seen as a platform for theoretical inference.

We realised that this would be helpful in determining, for each of our three research questions, which data collection approach might be most appropriate. In order to help focus on the best option for each question, we can summarise the above table, focusing on the volume of cases and features, in which case we obtain the following:

Table 12 Number of cases, features and approach

Number of cases	Number of features	Approach
Small[11]	Small	Experiment
Small	Large	Case Study
Large	Small	Survey

Now looking at our three research questions, it appears that each question is likely to be amenable to a different approach and so, we will consider them in turn. Looking at the discussion in 3.1.7 above, we can see that for RQ1, we will have a small number of cases (1, in fact), but many features (quality aspects and data points to measure), and so using Table 12, we can see why this suggests that a Case Study is appropriate.

For RQ2, the number of cases is large (we want to interview many people), however the number of features is small (two quantitative data points per concept), and so this suggests a Survey approach.

Finally, for RQ3, we have a small number of cases (we only need to demonstrate each technique once), and we only need to demonstrate a single feature per case. Thus, with low cases and low features, this suggests an Experiment.

Thus, the characteristics of our research questions, and the objectives related to them, guides our selection of the data collection approach in each case.

3.1.8.1 RQ1 – Case Study

3.1.8.1.1 Why a case study?

As we address the question *How can we measure the quality of a set of Enterprise Architecture models?*, we are going to take a single case – the description of (part of) an Enterprise Architecture for a single (public sector) organisation. We do not need to use multiple cases. We are relying upon the following deductive argument:

1. If the method (the application of our framework) works once, it will work more than once

[11] Small can mean just a single case

2. The method works once
3. (from 1 and 2) the method will work more than once

The above is a valid deductive argument, of valid form, and we only need to demonstrate the truth of premises 1 and 2 in order to prove that premise 3 is true. Demonstrating premise 2 will be achieved by the production of deliverable 1.2, and our achievement of that is demonstrated in section 5.4 below.

Demonstrating premise 1 will be achieved by reflecting upon the results of the case study and looking for evidence to show that it could be repeated, from both a theoretical and a practical perspective. This discussion can be found in section 5.5.3 below.

We want to avoid any kind of bias that might imply that we have a reusable quality measurement framework when we do not, and the above deductive argument does this.

3.1.8.1.2 What is a case study?

Case studies, according to Fry, are "… complex examples which give an insight into the context of the example as well as illustrating the main point … used … where an understanding of complex relationships is important" ([175], page 138). They are widely used in business research [176].

From Table 11 we see some typical key features of case studies:

- They involve a relatively small number of cases (sometimes just one);
- For each case, we are interested in a large number of features;
- Qualitative data is seen as more important than quantitative data;
- Theoretical inferences or generalisations might not be present; the aim might just to be understand the cases themselves.

Some of the literature, including Eisenhardt's paper just cited, talks about generating theory from case studies, which is obviously applying qualitative techniques. However, this is not really what we are trying to do with our first question: we are mainly focusing on quantitative data, not subject to ambiguous interpretation, and so we want to test our theory that we can measure the quality of sets of EA models.

3.1.8.1.3 How to select cases

Although not quite as relevant for a single case, a paper from 2008 [177], related to political research, summarises seven different case selection procedures: Typical, Diverse, Extreme, Deviant, Influential, Most similar, Most different. Of these, only Typical, Extreme, Deviant and Influential are applicable to a single case. In our case, before commencing the research, we had no clear way of knowing what we would find. The use of the Typical selection method is "*Confirmatory: to probe causal* *mechanisms that may either confirm or disconfirm given theory*" [177]. This is exactly what we are seeking to do: to confirm our theory that it is possible in practice to measure quality of sets of EA models. In practice, given our role in a consulting business at the time, this particular case was self-selecting: it is the organisation that represented our client.

3.1.8.1.4 Case study methodological issues

Gomm et al ([174], pages 5-6) summarise the following methodological issues from differences in perspective on the nature and purpose of case study: generalizability, causal or narrative analysis, the nature of theory, and authenticity and authority.

The key issue we want to address in our case is that of generalizability: can we form conclusions from a single case study? In the case study we are proposing, we are simply trying to demonstrate that it is possible to measure the quality of a set of EA models. We discuss our own reasoning for why this measurement can be made repeatedly in different scenarios (i.e. is a *generalizable method*) in section 5.5.3 below.

However, we will here give more of a theoretical explanation about the concept of generalizability. The terms 'generalizability' and 'inference' are related. Inference is of course related to the word 'infer' – for example as used in logic, looking at deductive argument forms, where we form arguments such as:

1. **P -> Q**
2. **P**

---

3. **Q**

In other words, if (a) we know that *if* P is true, *then* Q must be true, and (b) we know that P is actually true, then (c) we can reliably *infer* that Q is true.

For an overview of deductive argument forms and the associated nine rules of logical inference, please consult [178]. The above argument is using the first rule, **modus ponens**; and our deductive argument in 3.1.8.1.1 above is an example of the application of this rule.

What we mean by infer is that we can deduce the truth or falsity of some proposition from the truth or falsity of another proposition. The topic of inference and generalizability is the subject of an edited book, in which one of the chapters offers the following definition of generalizability: "*whether results are generalizable to a larger group or to theoretical principles*" [179]. This is a slightly unfortunate definition in that it is circular: the definition of generalizability depends upon the definition of generalizable. However, thankfully the author then goes on to provide a more comprehensive definition: "[generalizability] aims to establish the relevance, significance, and external validity of findings for situations or people beyond the immediate research project" [179].

So, generalization relates to the validity of findings outside the scope of the initial research. What this means will depend on the type of research and it applies as much to surveys and experiments as it does to case studies. It uses an inference of the following type:

Let proposition P represent a conclusion specifically related to the scope of the research, and Q represent the same conclusion related to a wider scope (maybe to a different case, or a different set of people). If the results are generalizable, then we are arguing using **modus ponens** again. The crucial question of course is how we can safely generalize.

For a discussion of generalizability as it applies to surveys, please see section 3.1.8.1.8 below. We now examine the concept of generalizability for our case study answering RQ1.

3.1.8.1.5 Generalization of the Case Study

Although we are describing this first piece of research as a case study, it should be noted that there are no participants involved in this research. Thus, concerns listed in [179] related to inferences across populations are not relevant. The generalizability we are seeking here, to borrow Duff's words, is "validity of findings for situations beyond the immediate research project". In other words, could this method be used in other situations. We are not trying to draw conclusions about people or groups of people; we merely need to establish that the method (for measuring EA model quality) can be used in other situations.

For an analysis of how we have demonstrated the generalization of our method, please refer to section 5.5.3 below.

3.1.8.1.6 Why a survey?

Our next question is *How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?* We are going to interview many people, capturing data points from each of them. The structure of the data will be very simple - the bulk of the data is a number of pairs of ratings - and so a survey appears most suitable for this piece of research.

3.1.8.1.7 What is a survey?

Groves defines these as *… a systematic method for gathering information from (a sample of) entities for the purpose of constructing quantitative descriptors of the attributes of the large population of which the entities are members* [180], quoted in [181].

Given the complex nature of this particular piece of research, it is worth considering some of the aspects of surveys contained in Table 11 above, and the relevance of them to our research.

Study of a sample of naturally occurring cases; selected in such a way as to maximise the sample's representativeness in relation to some larger population. [174]

In our research we are seeking to measure effectiveness against a 'general business population' and by that we mean a typical set of stakeholders that might interact with models produced by Enterprise Architects. Thus, we will select a mixture of people across all of the categories that we are considering, including (a) the mix of technical and non-technical, and (b) a wide range of job roles (not restricted, for example, to just one organisation). The set of cases (interviewees) has been constrained to an extent; we will only interview people that we already know personally, rather than attempting to advertise and interview people not known to us.

The aim is empirical generalisation, from a sample to a finite population, though this is sometimes seen as a platform for theoretical inference. [174]

This is exactly our aim: to attempt to generalise and make statements such as "the concept of an application collaboration is generally poorly understood amongst the general business population – those that might need to understand the work of IT Architects".

As well as relying upon mathematical and statistical techniques, surveys also rely upon other disciplines, including psychology, and a relatively new field called Cognitive Aspects of Survey Methodology (CASM). The latter considers the role of the cognitive processes within the respondent (interviewee) in the survey process (see for example [182]). There is a discussion on mixed-mode methods in relation to surveys in [181], however in this case mixed-mode methods means something completely different to the mixing of quantitative vs. qualitative methods: it deals with mixed means of communicating with the respondents in a survey. The same work gives an overview of design considerations for surveys, dealing with (a) measurement or observation units, (b) analysis units, (c) experimental dimension and (d) temporal dimension.

In our case, as is typical of surveys, our observation units are going be individuals: people that we will ask to participate in the survey in order to gauge their view of certain concepts. Our analysis units are mixed: we will analyse the data by individuals, and also by groups (for example, the understanding by non-technical stakeholders, or the understanding of those that happened to be tested using ArchiMate notation).

3.1.8.1.8 Generalizability of the Survey

For the survey, we need to consider challenges related to generalization that we did not need to face in our case study; this is because our survey will have human participants. Duff's chapter on generalizability, cited previously, says that generalizability, when used with quantitative research, is commonly accepted (when used with appropriate sampling, research design and other measures) as having the potential to yield results that can be generalised. She points out that this is, to an extent, dependent on the surveyed population being representative of the wider population, to whom we are hoping to extrapolate the results. In our context, this means that we would like to be able to say things like "non-technical people find some specific concepts in ArchiMate hard to understand", which is an extrapolation from "the non-technical people that we surveyed found some specific concepts in ArchiMate hard to understand". This obviously depends on how representative our sample population is.

Duff lists some sociocultural variables that might reduce the validity of this kind of conclusion; these factors include institutional context, first language, the relationship between the researcher and respondents, among other factors.

Regarding institutional context, we had originally been intending to select people from a single organisation, and it was in this context that we originally sought ethical approval for this research. After that stage, for a variety of reasons, fortuitously as it turns out (from a generalizability perspective), we changed to interview people across a range of organisations (see discussion in section C.2 below).

Regarding first language, we will conduct the surveys entirely in English. The ArchiMate standard itself is available in a number of languages, however we are only fluent in English. It is certainly possible that different results might be obtained should the research be carried out using test material in another language. We once taught an Enterprise Architecture course in Beijing, with Chinese students (in English), and remember a major discussion before the course commenced on the closest Chinese 'word' to the English word 'principle'. However, any weakening of the generalizability of our results would only be for the specific ratings obtained, and not for the method itself (that produced the ratings).

The relationship between researcher and respondent could also be a factor, and we will need to keep track of this as the research progresses to ensure that this does not introduce any bias (see sections 7.6.1 below).

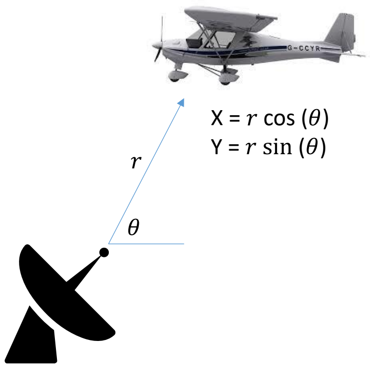
3.1.8.1.9 Triangulation

Triangulation is going to be an important concept in our survey design. Jick [183] quotes Denzin for the definition of triangulation: "*the combination of methodologies in the study of the same phenomenon*" [184]. Another piece of nursing research describes it as "… *a method used to increase the credibility and validity of research findings*" ([185], citing in turn [186]) and then relates this to two other terms: credibility (trustworthiness – how believable some research is) and validity (how well does the research reflect or evaluate the ideas being considered).

Care needs to be taken when researching this concept, as it has a number of completely different meanings in different contexts (e.g. Value Added Tax, Graph Theory and others). In this context, a simple analogy will suffice to illustrate the basic concept. The triangulation metaphor, in the sense that we are using it here, has its roots in the combination of trigonometry and geography: using different data points or perspectives to gain a more accurate figure.

Suppose one wishes to locate the position of some point in the plane from two different perspectives; for example one wished to locate, in the horizontal plane, the location of an aircraft, using some kind of radar system. With one such system, one could take two measurements: one of angle, and one of distance. These two, when put together, will uniquely identify a given (horizontal) location:

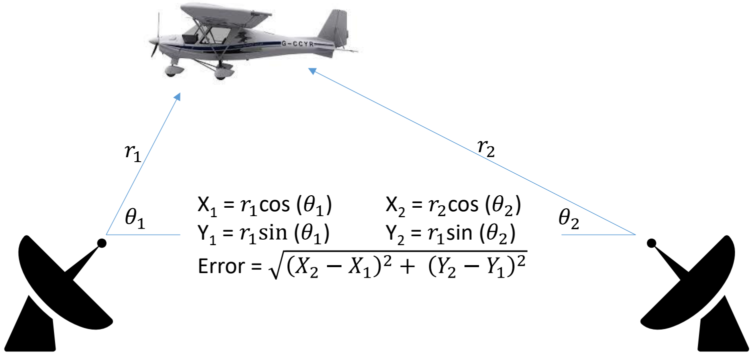
Figure 37 Single radar with aircraft



This is of course one sense of the term triangulation: we have combined two measurements with some formula (trigonometric) to give us a single result. This is the way that we might, for example, measure the height of a hill or mountain and in fact was the technique used to measure the height of Mount Everest [187].

However, there might be errors in the data captured, either in the angle θ (which direction is the aircraft?) or in the distance (how far away is it?). There is no way of knowing how good these figures are. However, if one uses two radar systems, each of which takes independent measurements of the angles (θ_1 and θ_2) and the distances (r_1 and r_2), then we can cross-check the two sets of results to obtain a better idea of how well we understand the position.

Figure 38 Two radars with aircraft



If we get two positions that differ by a significant amount, then we know that at least one of the measurements is incorrect. If the two positions agree substantially, then we have increased confidence in our measurements. This is now a separate use of the term ‘triangulation’. Rather than trying to make a single measurement, we are now using two perspectives in order to assess the confidence in a set of measurements of the same point.

So, we have two meanings of ‘triangulation’: one to do with making a single measurement and one related to quality-checking a measurement (from two perspectives). It is the latter that we are concerned with here.

In our particular case, we wish to employ some kind of triangulation to be more certain about a respondent’s understanding of certain concepts. When asking them if they would prefer if an item was removed from a diagram, then we can take that at face value. If, however, we want to understand whether they understand a certain concept, then we would want to test that understanding, to avoid the potential situation whereby a respondent says “I understand it perfectly”, but in practice does not. Thus, we intend to capture their brief narrative description of a concept’s meaning in order to then ‘score’ it against the official definition.

This will make use of the concept of ‘moderate factivity’, as discussed in section 2.3.16 above, and so we are seeking to test the respondents’ description of the concept in its core beliefs against the official definition of the concepts (in the ArchiMate standard).

For more details, please refer to the description of the data collection process in section 3.2.2.3 below.

3.1.8.2 RQ3 - Experiment

Our final piece of research addresses the question *Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?* To do this, we summarise the key transformations required and then seek to find example of how each of them can be done in practice. The low number of cases combined with the simple result (can it be done, or not, hence a Boolean), makes this an experiment.

3.2 Data Collection

Our approach here is different for the various research questions.

3.2.1 Data collection for RQ1

For the research related to model quality (RQ1), we will collect information from existing computer systems, specifically, looking at the contents of a 'Wiki' [188], which is a collection of linked pages edited collaboratively by a number of people, of which perhaps one of the most famous examples at the current time is Wikipedia [189], so-named because it is a Wiki form of an Encyclopedia.

We will be seeking to demonstrate the truth of the hypothesis: the mathematical framework that we will construct to measure the quality of sets of EA models (in Chapter 4) can be used in practice.

The mathematical framework requires, as its raw data, a set of EA models, and for each of these models we will try and answer three questions about them, each of which has either a 'yes' or 'no' answer. This evaluation is purely deterministic. This is therefore a very simple case study.

3.2.2 Data collection for RQ2

As with the previous research question, this is split into two chapters: the first describes the construction of a theoretical framework, with a mathematical basis, and the second describes the case study that we used to validate the framework as well as producing some interesting information about the ArchiMate language.

This is the only research area that required us to collect any data from people.

3.2.2.1 Why use a Questionnaire?

We had two main choices: ask the respondents to fill in some kind of questionnaire themselves, or interview people. Because of the possibility of misunderstanding the questions being asked, we felt that better results would be produced if we carried out the interviews in person. We did trial the interview process with a group of masters-level students to ensure that they understood what was being asked and that we got a good range of numeric scores (in other words, some concepts were poorly understood, and others well understood). Having established that the approach seemed to work, we then proceeded with the full data capture. We did go through the standard ethical approval process (see Appendix C below) before commencing the interviews.

3.2.2.2 Questionnaire Design

Oppenheim [190] distinguishes between two types of survey: descriptive, enumerative, census-type and analytic, relation type.

The purpose of the descriptive survey, states Oppenheim, is to count, and to infer about the population as a whole, from the subset included in the survey. These surveys can also be used to make predictions. A key aspect to consider in the design of a descriptive survey is having a representative sample of the population as a whole, otherwise the results that emerge will not be representative. By contrast, an analytic-type survey is used to test relationships: where there are two (or more) variables to measure and we are looking for some kind of correlation between them.

In our case, there is one small element of analytic design. Consider that we are looking to see if comprehension of certain concepts (one variable) changes with the use of ArchiMate notation (another variable). Hence our survey falls into the analytic category.

Oppenheim goes on to specify four kinds of variables used in designing an analytic survey:

- Experimental
  - The variables that you will vary deliberately to see their effect
- Dependent
  - The results that you are collecting and analysing and to see how they differ with the experimental variables
- Controlled
  - Variables that you try and keep constant so as not to invalidate the results
- Uncontrolled
  - Variables that cannot be controlled and that can cause ‘correlated biases’, or errors

In our case, the experimental variables – the things that we change – will be number just one: the choice of whether or not we use ArchiMate notation for the test models.

The dependent variables will be the comprehension and utility ratings (described in more detail in Chapter 6 below).

In terms of controlled variables, the main source of concern here would be that our scoring of the respondent's description of the various concepts might become subjective, and our method for controlling that is to refer frequently, during the scoring, to the official definition of the concepts, so that we do not stray into using our own definition.

In terms of uncontrolled variables, we might include in here factors such as the attitudes of the respondents to the survey process, or past experiences or training that they have had that they did not declare.

The data we require here for the case study will be, as described in 3.1.5 above, a mixture of quantitative and qualitative. The quantitative data takes primacy, with a small amount of qualitative data included within that framework. The qualitative elements were really adjuncts or clarifications to the quantitative elements.

Rattray and Jones' 2005 paper [191] on questionnaire design was written in the context of nursing, however it is just as relevant in other fields. They propose a framework that supports the design and development of questionnaires by a structured approach, logical and systematic. In the original context, the authors say that "*when developing a questionnaire, items or questions are generated that require the respondent to respond to a series of questions or statements. Participant responses are then converted into numerical form and statistically analysed*"[191]. This is precisely what we wish to do with our survey; to understand participant's knowledge of certain concepts. They caution that this approach (meaning questionnaires) is not without criticism and rightfully point out that there are assumptions being made about shared language and interpretations, which is in line with the literature on semiotics. They say that "*closed questions may restrict the depth of participant response*", and it is for this reason that we will also invite the participants to share *why* they believe something. In fact, we will be asking three things (for the comprehension part of the survey): (a) *how confident* is their understanding (Likert scale with no neutral point); (b) *what* is their understanding (free text - to allow us to triangulate and perhaps adjust their suggested rating); and finally (c) is there anything they can share that helped or hindered their understanding.

The authors provide a table suggesting key stages in questionnaire development and, with each stage, some related key issues along with examples of measures (quoting other literature, which we will not reproduce here verbatim). These stages and key issues are:

Table 13 Stages in questionnaire development: item generation and scale construction, adapted from [191]

Questionnaire development	Key issues

What will the questionnaire measure?	Knowledge Attitude/beliefs/intention Cognition Emotion Behaviour
What types of scale can be used?	Frequency Thurstone Rasch Guttman Mokken Likert type Multiple choice
How do I generate items for my questionnaire?	Ensure relevance of items? Wording issues Which response format is best? Which types of question are possible? Free text options? Does your measure have subscales? Questionnaire layout

The authors say that Likert-type scales use fixed choice response formats and are designed to measure attitudes or opinions, that they measure levels of agreement or disagreement, that they assume that there is a linearity with the strength or intensity of the experience (or belief, in our case), and that they make the assumption that it is possible to measure attitudes. It is not clear why the changing of words in this paragraph from “attitudes or opinions” to “experience” back to “attitudes”. Nevertheless, this appears suitable for our purposes, as we are asking for the respondents’ strength of agreement with the proposition that “they fully understand what is meant by concept X” (although we will word it slightly differently).

In their section on item generation, the authors say that free text response or open questions may be included in order to allow respondents to expand upon answers and it is our plan to do precisely this. The paper is aiming perhaps at a greater degree of certainty that we have aimed for in our own research. For example, the authors expected that the questionnaire pilot with under 100 respondents, and then when the questionnaire is finalised, then use over 100 respondents.

We are attempting, first of all, to demonstrate that it is possible to measure comprehension of concepts from an EA modelling language and then see if any broad results are immediately forthcoming that might then be the subject of more detailed, focused and rigorous research.

Finally, the authors suggested that the metadata about the respondent might be captured at the end of the process to avoid boredom. We felt that it would be useful to do the opposite, because we are expecting the respondents to have to do some thinking during the processes, and so we felt that giving some easy questions, about themselves, might be a kinder way to start the process.

So, before moving on to the key quantitative data, we intend to capture some brief information about the interviewee (e.g. their name, job title, previous educational attainment and experience). In practice we made virtually no use of this textual data; the data that we used in our analysis was almost completely restricted to the quantitative data and so, although we started this particular piece of research as mixed-methods, in practice it ended up largely as quantitative, apart from a few useful phrases that we were able to pick out afterwards.

We should also specify here that our intention is to use a cross-sectional (measuring unrelated groups of people) design rather than a longitudinal design (measuring the same respondents over a period of time).

We now move on to the core data being collected. The dependent variables we are seeking are numeric: ratings of comprehension (between 0 and 1) and utility (between -1 and +1). There are many ways that we could get to this rating. Factors that we had to consider included:

(a) Time – having to collect comprehension and utility ratings on approximately 10 concepts (per interview lasting maybe one hour) will limit our ability to have an in-depth analysis of the respondent's understanding.

(b) Triangulation – we could have just asked for the respondent's estimate of confidence in their understanding of the concepts – but we are not trying to answer how well the respondents think they understand a concept – we are trying to answer how well they understand it, and so we need to use their description of the concept to enable us to triangulate and thereby potential reset their comprehension rating according to their actual understanding.

For these reasons, the questions for comprehension will be open, along the lines of "what you think things of type B2 represent?". Once they respondent has formulated a response, which we record briefly, we will then rate that response (convert the narrative response into a numeric rating).

The questions for utility will be different, as no triangulation is necessary: whatever the respondents think is the right answer is the answer we want to record, and so the question will be a closed one: "Would the model fulfil its purpose better, worse, or about the same, if everything of that type was removed?". This enables us to record a -1/0/+1 score directly in response to the closed question.

Before commencing the data capture, we will also make clear to the respondents that we are interested why they assign certain ratings and so, when the respondents give a reason, we will also record that briefly.

3.2.2.3 Structured Interviews vs. Self-Completion Questionnaires

It is our intention to use the same questionnaire for all respondents, however we will use a mix of one-on-one interviews in some cases, and group interviews in other cases. The group interviews will actually consist of a group briefing, followed by the respondents filling out the questionnaires themselves, with the interviewer present at all times to answer any questions from the respondents.

In both cases we are intending to give the same information about how to complete the questionnaire – its purpose and method, and so on. However, for one-on-one interviews, the interviewers will write down the words spoken by the respondents, whereas for the group interviews, the respondents will write (type) their responses directly onto a copy of the questionnaire. This latter technique is referred to in the literature as a 'self-completion questionnaire'. This is discussed in Bryman and Bell [192], where the authors discuss some of the differences between structured interviews and self-completed questionnaires.

As well as noting that the latter are quicker to administer, they note that structured interviews have the possibility of introducing bias into the answers given by the respondents and that self-completion questionnaires are less likely to encounter this bias. They also report a finding that self-completion questionnaires, without an interviewer present, are likely to reduce the incidence of respondents biasing their responses related to sensitive data, sensitive in the context of the paper related to sexual behaviour and drug consumption [193]. Asking respondent's opinion about the meaning of IT-related concepts is not as likely to be viewed as a sensitive topic and so we may hope to avoid this kind of bias.

The authors note that a disadvantage of self-completion questionnaires is that no-one is available to help respondents if they wish to ask a question to clarify what is being asked of them. In our data collection we will avoid that by being present with the group filling in these questionnaires.

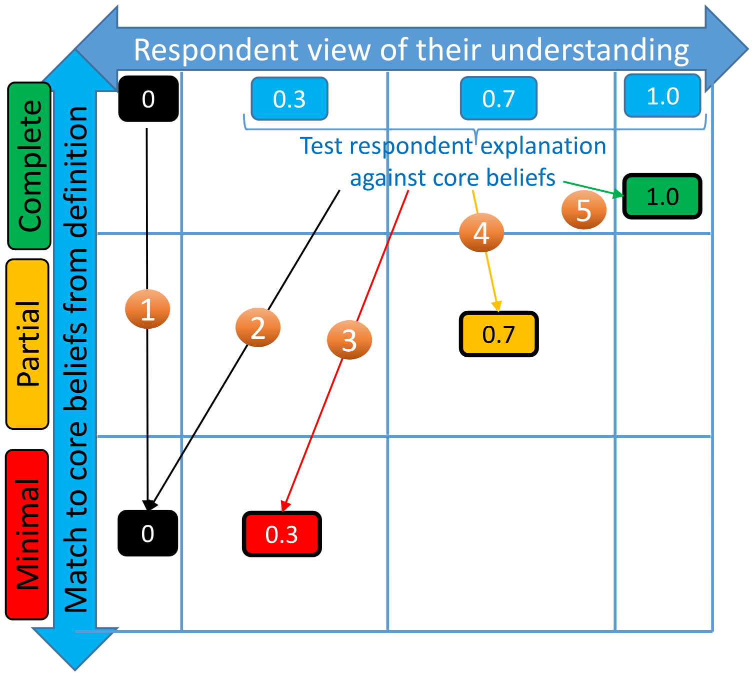
They also refer to issues with the respondents filling out their own questionnaires being able to answer questions in any order; also potential issues with respondents whose English literacy is not necessarily well developed. As we will see in our data collection, this is a factor that we faced with the group interviews.

Response rates, also mentioned in this section, are not a concern for us because we intend to be present in the group interviews, rather than just distribute the questionnaires and wait to see what comes back.

3.2.2.4 Triangulation to Test Understanding

Having established the general concept of triangulation (see 3.1.8.1.9 above) and moderate factivity in relation to objectual understanding (section 2.3.16), we now discuss how these two concepts will be used together to triangulate our measurement of comprehension of a particular concept by a respondent. The process of triangulation is illustrated below:

Figure 39 Triangulation of Comprehension Ratings



This should be read in conjunction with the mathematical description of the process of measuring comprehension, as contained in section 6.4.2 below. As the later section describes, we will first of all ask them if they understand the meaning of a particular type of symbol and for a confidence level of that understanding. If they claim to have no understanding (a rating of 0), then we accept that at face value (path 0 above). If they claim to have some level of understanding (any other score), then we will compare their description of the concept with the official description (in this case, in the ArchiMate reference material), to test that understanding. Depending on whether there is no, a partial, or a full match, in terms of the core elements of the definition (using our definition of moderate factivity here) then we will replace their suggested score with our own rating (1, 2, 3, 4 or 5 above). Thus, the only user-assigned rating that we let pass unchallenged is a rating of 0.

3.2.3 Data collection for RQ3

Answering this question does not involve any systematic data collection at all. Having established in the previous chapters some context for the kind of transformations that we need to make in our models, this single chapter summarises those transformations and then demonstrates that they are not only theoretically possible, but entirely practical as well, by way of demonstration of the art of the possible. Thus, this could perhaps best be described as an experiment, seeking for observations that fit certain desired criteria.

3.3 Mathematical Approach

As much of this research relies upon mathematical analysis of data in order to measure certain concepts, we now proceed to give an overview of some of the mathematical techniques that we will use in our research. This gives some background information that has a bearing on how we will collect the data required (discussed in section 3.2 above).

3.3.1 Metrics

Our first two research questions are fundamentally asking how we might measure something, and thus we need to discuss the idea of metrics. An online dictionary defines a metric as “a standard for measuring or evaluating something; basis for assessment” [194]. We use them all the time; common examples would include miles or metres for distance, kilograms for weight, knots for airspeed, hectopascals for air pressure, and so on). There is literature available on constructing a good metric [195], unfortunately not currently available to us (see section 2.6 above).

It seems self-evident that a good metric uses units that can be measured objectively and easily. For example, trying to measure a musical tone by comparing the frequency of the waves in air with the speed of flapping of a swallow’s wings [196] would fail the ‘easy to measure’ test. By contrast, distance can easily be measured using a tape-measure.

In order to construct these metrics in what is, hopefully, a methodical and rigorous way, we have drawn on some relatively simple mathematics from a number of areas and below we give a brief introduction to the concepts and notation that we use within this work.

3.3.2 Set Theory and Logic

A comprehensive introduction to the topic can be found in [197]. Sets are groups of things that have something in common, for example the set of all planets in our solar system, the set of all chairs in my house, the set of all residents of the United Kingdom, the set of all numbers divisible by 3, and so on. Let us say, by way of a concrete example that we can explore further, that we have two sets of objects (things) which we call A and B and that they are both sets of integers (whole numbers).

Set theory notation gives us the following ways of expressing the relationship between sets A and B:

Table 14 Set Theory Symbols

Symbol	Name	Meaning	Example
{ }	set	a collection of objects	A = { 1, 3, 5, 7 } B = { 5, 6, 7, 8 }
$\varnothing$	empty set	a collection that has nothing in it	C = { }
$A \cap B$	intersection	objects that belong to both A and B	$A \cap B$ = { 5, 7 } (as only these two numbers are in both sets)
$A \cup B$	union	objects that belong to either A or B (or both)	$A \cup B$= { 1, 3, 5, 6, 7, 8 } (numbers that appear in either or both sets)
$x \in S$	belongs to	The single object x is within set S	$3 \in A$ (as 3 is one of the members of set A)
$x \notin S$	does not belong to	The single object x is not within set S	$6 \notin A$ (as 6 is not one of the members of set A)
$A \setminus B$	relative complement	Objects that are in A but not in B	$A \setminus B = \{ 1, 3 \}$ (as the other members of A - 5 and 7 - are also in B)
$\mid A \mid$	cardinality	The number of objects in set A	$\mid A \mid = 4$ (there are four objects in set A)

The following symbols are also useful for working with logical statements:

Table 15 Logical Notation

Symbol	Name	Meaning	Example
$\forall$	Universal quantification	For all	$\forall\ n \in \mathbb{Z}$ for all values of n that are part of $\mathbb{Z}$
$\exists$	Existential qualifier	There exists	$\exists\ x \in \mathbb{Z} : x < 0$ There exists (are) some numbers in $\mathbb{Z}$ that are negative
$\nexists$	Negated existential qualifier	There does not exist	$\nexists\ x \in \mathbb{N}_0 : x < 0$ [12] (there are no non-negative integers less than zero)

3.3.3 Functions (Mapping)

Implicit in the concept of a function, as used here, is the idea that a function takes something (typically a number) and turns it into a different number. This is sometimes referred to as mapping: a function maps one set of numbers to another set of numbers. For example, the function f(x) = 2n, where $n \in \mathbb{Z}$ [13], maps the set of all integers to the set of all even numbers.

The set of allowable *input* values for a function is called its *domain*. For example, assuming that we are staying within the real numbers, avoiding complex for now, then the domain of the square root function is the set of all non-negative real numbers (because we cannot, without resorting to complex numbers, take the square root of a negative number).

The set of possible *output* values for a function is called its *range*. As an example, if we have a function defined as $f(x) = x^2$, then the domain of this function is $\mathbb{R}$, the set of all real numbers (because we can square any real number)[14]. However, the range of f(x) is the set of non-negative real numbers,

---

[12] $\mathbb{N}_0$ is standard notation for the set of all non-negative integers

[13] $\mathbb{Z}$ is standard notation for the set of all integers (of any sign)

[14] We assuming here that we are staying strictly with real numbers, and not straying into the domain of complex numbers $\mathbb{C}$

because no matter what input value we provide to f(x), it will always get mapped to a non-negative value.

When describing a function, the following notation may be used:

f : D → R

This can be interpreted as saying that the function f maps values from D to R.

3.3.4 Calculus

The only concept from calculus used in our research is the concept of a derivative, which fundamentally concerns the rate of change of something with respect to something else. Real life examples of rate of change would include:

The speed of a vehicle is defined as the rate of change of position with respect to time (taken as the limit as the time interval over which we measure it approaches zero). For example, we might say that a vehicle that travels sixty miles for every hour that passes is travelling at sixty mile per hour. Now if position is denoted as x, and time is denoted as t, then we take a small amount of position, dx, and a small amount of time dt, we say that the ratio of the two small amounts represents this derivative, and denote it as $\frac{dx}{dt}$. This is displayed on the dashboards of almost all vehicles (in units of either miles, kilometres or nautical miles per hour).

Another example would be the slope of a road. We might ask how much does a road change in its vertical height (say, above sea level) for every unit length you travel horizontally along it, If the height above sea level was given by y, and the horizontal distance was x, and we then choose to move a small distance along (dx), and measure a small increase in height (dy), then we might describe the slope of the road at the point (x,y) by the ratio of these two small numbers: $\frac{dy}{dx}$.

Another related notation is used when we are dealing with functions of a single variable. If we have a function f that acts upon a variable x, then we use the notation f'(x) to describe the derivative of the value of f(x) with respect to x. Thus, f'(x) is the same as $\frac{df(x)}{dx}$. It is just a more compact notation.

Graphically, the derivative can be viewed as the "slope" of a line on a graph (say, of y = f(x)).

If the slope is positive ($\frac{dy}{dx} > 0$) at a particular point on the line, then as x increases from that point, y also will increase. Conversely, if the slope is negative, then as x increases, y will decrease.

One final piece of mathematics that we will use later, concerns the derivative of a quotient of two functions. If we have two functions f(x) and g(x), and they are both differentiable[15] in some domain (valid range of input values), then if there is a third function h(x) that is obtained by dividing f(x) by g(x), in other words

$$h(x) = \frac{f(x)}{g(x)}$$

then the derivative of the combined function, h'(x), is given by

$$h'(x) = \frac{f'(x)g(x) - f(x)g'(x)}{g^2(x)}$$

This is a standard result obtained by use of the chain and product rules.

We will use this later on when we consider the quality of a set of models as a function of time, and look at the conditions necessary in order for quality to increase with increasing time (in other words, h'(t) > 0).

3.3.5 Statistics

We will use statistics only in one place in our research, and that is to assess the impact of the use of ArchiMate notation on comprehension and utility ratings.

Given any set of N data points $d_1, d_2, \ldots d_N$ we can calculate their average value thus:

$$\text{Average, } A = \frac{\sum_{i=1}^{N} d_i}{N} \qquad (1)$$

In plain English, this means add up all the points (using *i* as an index value), and then divide by N, the number of points.

We may also want to get an idea of how 'spread out' the values are. For example, are all the data points close together, or are they wide apart? To do this, we use a measure called the standard deviation (sometimes denoted with a symbol σ). We take all the differences between each point and the average, squaring each difference in turn (which forces them to be a positive numbers). We then divide the sum of the squared differences by the number of data points and then take the square root of that number. This gives an approximation to the 'average distance' of the data points from the average value. The formula looks like this:

[15] i.e. capable of being differentiated – which means essentially that they are 'smooth' curves

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - A)^2} \qquad (2)$$

Here, d_i represents each of the data values in turn, A is the average, given in formula (1) above, N is the number of data values, and the $\sqrt{}$ sign means take the square root of whatever is inside the sign.

3.4 Types of Science

3.4.1 Natural and Artificial Science

Writing in 1996, Herbert [198] discusses in his first chapter the difference between what he refers to as the "natural" and "artificial" worlds. Natural science is defined as knowledge about things that already exist in the real world, and deals with their attributes, their relationship to each other and their behaviour.

By contrast, in the artificial world, we are dealing with "artifacts" – things that have been specifically created to achieve a particular goal or effect. Herbert is interested in the extent to which approaches that have been used in the natural sciences, for example empirical approaches to modelling nature, can be applied to objects that are artificial and could have been designed differently.

3.4.2 Design and other Sciences

Researchers with an interest in Information Technology have previously defined information as "data that has been processed into a form that is meaningful to the recipient and is of real or perceived value in current or prospective actions or decisions" [199].

This highlights immediately two separate kinds of approach that need to be considered in relation to Information Systems.

Information Systems are created artifacts; they are designed, they are most definitely artificial, could have been constructed differently, and so of themselves are properly the subject of design science.

However, the information they produce is intended to be used (in most cases) by human beings who are not artificial and who themselves are more properly examined using other kinds of science (e.g. natural).

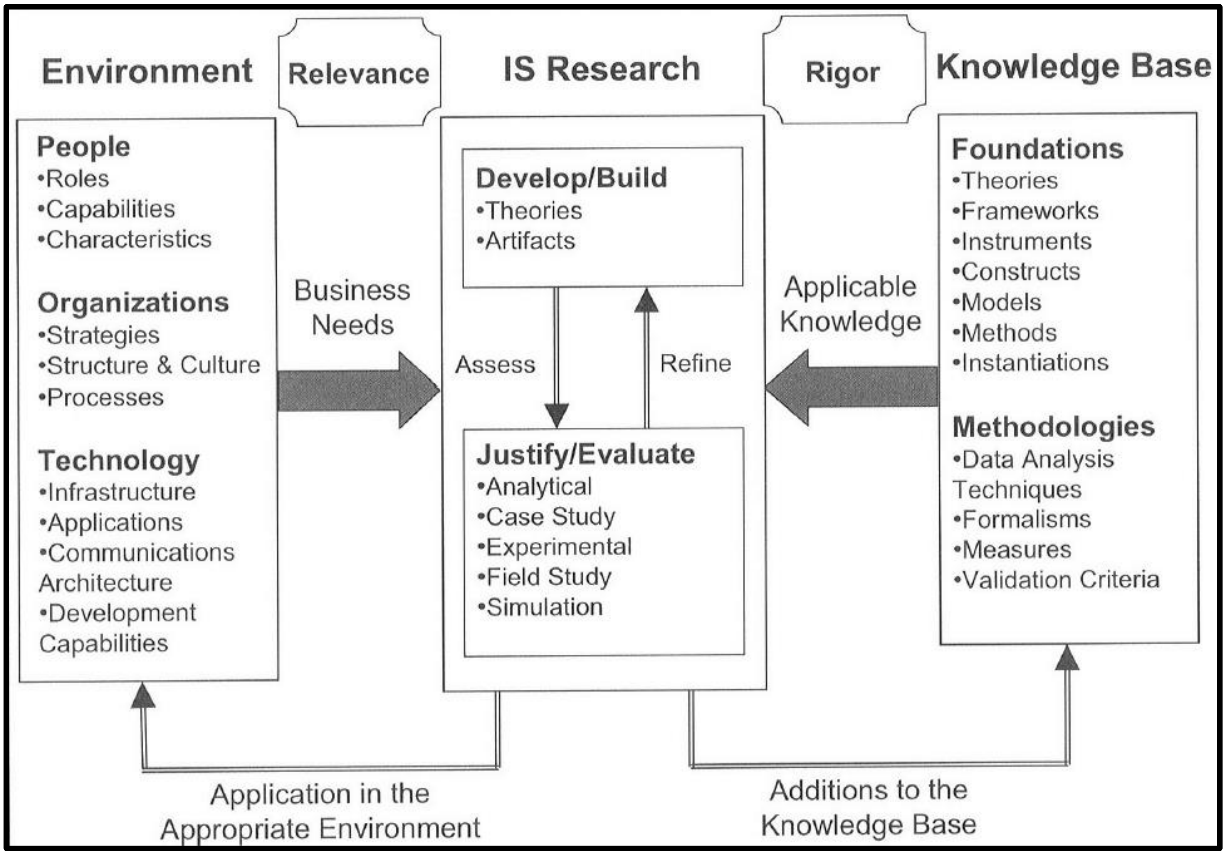
So we can see here a duality, an essential complementarity between these two approaches, and a paper from 1995 actually suggested a two-dimensional framework for IT research that incorporates both design and natural science approaches [200]. However, for our purposes, it will suffice to use design science as a foundation for our methodology.

3.5 Design Science

3.5.1 Design Science Essentials

Hevner's paper from 2004 [201] provides clear guidelines for carrying out design-science research related to Information Systems. The paper presents the following framework for IS research:

Figure 40 Information Systems Research Framework [201]



Hevner's work here has, as its starting point, a perceived need of the environment (people and organisations). Our own research has a subtly different starting point. We are not trying to solve a particular perceived business need: we are trying to create methods for evaluating how well needs are being met (so to speak). Nevertheless, the above framework is a useful starting place for our methodology. We are, in the words of his paper, "building and evaluating artifacts designed to meet the identified business need" – but our artifacts are largely measurement tools, verified by a survey and a case study.

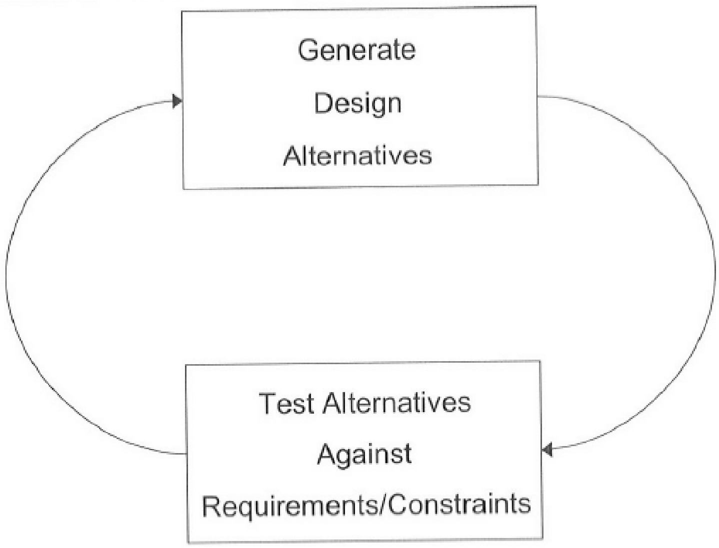
Hevner's research guidelines are summarised below:

Table 16 Design-Science Research Guidelines [201]

Guidelines	Description
1. Design as an Artifact	Design Science research must produce a viable artifact in the form of a construct (vocabulary and symbols), a model abstractions and representations), a method (algorithms and practices), or an instantiation (implemented and prototype systems).
2. Problem Relevance	The objective of Design-Science research is to develop technology-based solutions to important and relevant business problems.
3. Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well executed evaluation methods.
4. Research Contributions	Effective Design-Science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
5. Research Rigor	Design-Science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
6. Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
7. Communication of Research	Design-Science research must be presented effectively both to technology-oriented, as well as management-oriented audiences.

Hevner also includes a diagram illustrating the iterative nature of this approach:

Figure 41 The Generate/Test Cycle (from [201])



3.5.2 Our Application of Design Science

Hevner's seminal paper on Design Science [201] lays out guidelines as to how to carry out research in the area of Information Systems. We examine now the guidelines laid down by Hevner and how we have used them in our own research. In each of the following sections, we start by reproducing the description from Hevner's work and then explain the relevance to our own. We would, first of all, emphasise some words from Hevner on the 'rote' use of his guidelines:

"*… we advise against mandatory or rote use of the guidelines. Researchers, reviewers, and editors must use their creative skills and judgment to determine when, where, and how to apply each of the guidelines in a specific research project. However, we contend that each of these guidelines should be addressed in some manner for design-science research to be complete.*"

Hevner's work discusses "*… how knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artifact*." [201]

3.5.2.1 Design as an Artifact

Design Science research must produce a viable artifact in the form of a construct (vocabulary and symbols), a model abstractions and representations), a method (algorithms and practices), or an instantiation (implemented and prototype systems) [201]

In our own research we are producing all of the above.

We use mathematical models (vocabulary and symbols) for the theoretical frameworks addressing the first two research questions (in Chapter 4 and Chapter 6); the frameworks themselves are arguably models in their own right, in that they represent something else, are a summary (hence incomplete) and have a specified purpose (measuring quality). The practice referred to is demonstrated in the two case studies (Chapter 5 and Chapter 7). Finally, the instantiations are demonstrated in the more practically focused research on model evolution (Chapter 8). However, the software that is in effect showing the state of the art, thus answering the question “are these model transformations actually capable of being automated?”, was not created directly as part of this research. In some cases, they were created during the course of our employment; in some cases, they were created by third parties. Thus, we are not claiming to have used design science principles in the construction of that software. We just refer to the software as evidence for a point of view.

3.5.2.2 Problem Relevance

The objective of Design-Science research is to develop technology-based solutions to important and relevant business problems [201]

The problems we are trying to solve relate ultimately to improving the quality of the models that we produce, in order to lead to better decision-making by those relying upon the models. Technology certainly has a role to play here, in particular producing measurements, of various kinds, of the quality of individual models (EA modelling tools with built-in analytic capability can help enormously here), as well as in the automated manipulation (what we describe later as evolution) of our models to better meet the needs of our stakeholders. Thus we demonstrate that technology-based solutions are possible, although it is outside our scope to propose a specific technology solution.

3.5.2.3 Design Evaluation

The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well executed evaluation methods [201]

Hevner’s paper lays out, in Table 2, some design evaluation methods, grouped under the headings Observational, Analytical, Experimental, Testing and Descriptive. Given that our use of Design Science does not quite fit with the intent of the paradigm, not all of these methods are appropriate for our research.

We have carried out controlled experiments (as described in section 7.5 below) with the method for testing the effectiveness of modelling languages.

3.5.2.4 Research Contributions

Effective Design-Science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies [201]

The design of our mathematical models has clear foundations, in particular in the literature on conceptual model quality and semiotics. We have produced new artifacts, in extensions to the concept of model quality for Enterprise Architecture, as well as a method for evaluating the effectiveness of an EA modelling language.

3.5.2.5 Research Rigour

Design-Science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact [201]

The method we have employed in the construction of the two theoretical frameworks is the mathematics of sets and related metrics which is sufficiently rigorous for the purpose, due to its unambiguity and clear meaning.

3.5.2.6 Design as a Search Process

The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment [201]

This has been less relevant to our own research as there were very few conceptual restrictions to the design and execution of our artifacts.

3.5.2.7 Communication of Research

Design-Science research must be presented effectively both to technology-oriented, as well as management-oriented audiences [201]

The purpose of thesis is to present the research to audiences both with and without a technical background. Any technical content required in order to understand the material, is presented first in the literature survey

Chapter 4. Theoretical Approach: Measuring the Quality of a Set of EA Models

4.1 Research Question

Our first research question (RQ1) is *How can we measure the quality of a set of Enterprise Architecture models*? In this section we propose a theoretical approach that will answer this question.

In our literature survey we covered a number of topics including Enterprise Architecture (section 2.2.1), Enterprise Architecture frameworks (section 2.2.2), models (section 2.2.3), and quality (section 2.2.4). We also examined theory pertaining to conceptual model quality (section 2.3.1), and learnt how that had been applied specifically to software requirement specifications (section 2.3.2). We would like to similarly extend the conceptual model quality framework specifically into the area of enterprise architecture models.

We learnt that EA frameworks provide a common language for describing the enterprise architecture – the high-level structure – of businesses and their supporting information technology and that lack of knowledge in this area can be an inhibiter to change. This gives us one of our reasons for wanting to extend the notion of model quality for Enterprise Architecture to have a measure of completeness, just like it had been for its application to requirements specifications.

We have also covered some of the basic notations and techniques that we will need to do this, for example metrics (section 3.3.1), set theory notation (section 3.3.2) and calculus, specifically the notion of a derivative (section 3.3.4). All of these will now be used to help answer our first research question.

4.2 How to Solve

In our introduction (in section 1.6.1) we considered in outline three questions that we could ask about a set of models:

4.2.1 Are the models syntactically correct?

If we have agreed a language or notation with which are models should comply, then do they actually comply?

4.2.2 Do the models contain propositions that are, in some sense, “not true”?

Are the models accurate, or do they provide a misleading picture?

4.2.3 Are there propositions, that should be contained in our set of models, that are missing?

Are we missing an important piece of the picture?

These form the basis for our mathematical model that follows. Our intention is to produce metrics to allow us to measure these aspects for any given set of models.

4.2.4 Definitions for sets of EA models

In this model, we adapt Lindland's definitions to our new context of a set of models, thus:

$\mathcal{M}$ is the set of all the EA *models* that exist within our architectural description, irrespective of their content (subject) or format, each of the models being a simplified, tailored view of (a possible) reality designed to meet the needs of a particular stakeholder. This is similar to the original definition, where M is a single (conceptual) model consisting of statements, each of which is either syntactically correct, or not; and reflects the domain, or does not. In our adaption of this framework, instead of a single conceptual model, we use the region $\mathcal{M}$ to encompass the models (in the diagrammatic sense) that have been provided; thus equating conceptual model statements in Lindland's and Krogstie's work to diagrammatic models in our own (as summarised in Table 17 below).

$\mathcal{L}$ is the language, i.e., the set of all statements which it is possible to make according to the vocabulary and grammar of the EA language(s) that we have agreed to use in our repository, for example use case diagrams from UML or structural or behavioural diagrams from ArchiMate;

$\mathcal{D}$ is the domain, the boundary of our Enterprise Architecture, both the current state of the business and its supporting ICT, and possible future states (options or alternatives) that we are investigating. More precisely, this is the full set of models that we would expect to see in order to fully and accurately describe the domain of interest. In TOGAF terms, this means the set of models that fully populates the required views and viewpoints for all the stakeholders in scope. An example of a model in $\mathcal{D}$ would be a diagram that showed an existing business service (e.g. Intelligence Management, in the policing sector) relating to a new Intelligence system that is being acquired, because this represents a possible (indeed planned) reality;

Thus, in summary, we have adapted LIndlands' definitions for use in our own research on the topic of Enterprise Architecture by defining the scope of our conceptual model to be the set of all EA models in our control, stored within an architecture repository, so relating the terms from Lindland's original theory to the terms in our paper thus:

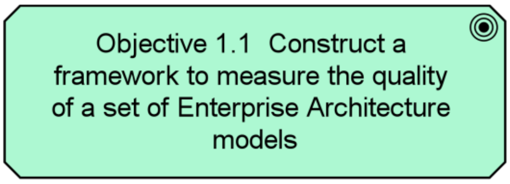
Table 17 Adaptation of Terminology for Model Quality

Original Theory	Equivalent in this Research
Statement within the Model	One of the models within the repository
Scope of Conceptual Model	The set of all models within the repository

We are focused on the syntactic (relating to tokens and language) and semantic (relating to meaning) views. The pragmatic elements are out of scope at this stage.

4.3 Mathematical Model

We are now going to address the objective:



For any given model (ie a diagram) containing information about our Enterprise Architecture, therefore, we can classify its quality in a number of ways, three of which are specified in 4.2 above.

For models that exist, we can say that either it is (entirely) syntactically correct, or that it is not. Thus, the proposition “this model is entirely syntactically correct” has a truth value of either $\mathbf{true}$ or $\mathbf{false}$.

Similarly, the proposition “this model corresponds completely to the domain it is modelling” (or more colloquially, “this modelling is telling the truth”), again has a truth value of either $\mathbf{true}$ or $\mathbf{false}$.

The combination of these two Boolean values could be represented in a number of ways, for example as a truth table (as discussed in for example [202]):

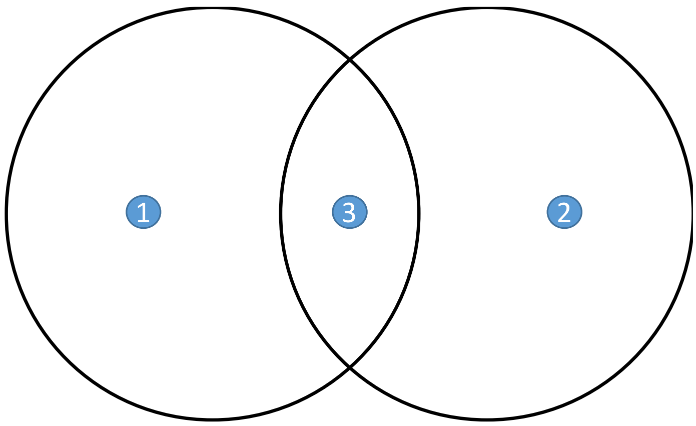
Table 18 Truth Table for Quality of Existing Models

Syntax correct	Telling the truth	Meaning
$\mathbf{false}$	$\mathbf{false}$	Model has incorrect syntax and does not reflect reality

false	**true**	Model has incorrect syntax but is telling the truth
true	**false**	Model has correct syntax but does not reflect reality
true	**true**	Model has correct syntax and is telling the truth

There is an alternative way of visualising the truth of our two 'quality propositions' discussed so far, and that is to use a Venn diagram (for a history of these, please consult [203]). If we create a Venn diagram containing two overlapping circles, each one of which corresponds to one of these truth propositions, (columns in Table 18 above) then we obtain:

Figure 42 Two-part truth table as Venn Diagram



Corresponds to domain ($\mathcal{D}$) Syntax correct ($\mathcal{L}$)

We can visualise each of our models as being a point somewhere on this diagram, in a region determined by our quality propositions as applied to that model.

The specific model (diagram) 1 is in the left-hand circle, and so is deemed to have a truth value of **true** for the proposition "this model corresponds to the domain" (or, " this model is telling the truth"). It is outside the right-hand circle, and so is deemed to have a truth value of **false** for the proposition "this model is in the correct syntax".

The respective truth values are reversed for model 2, which has a truth value of **false** for the proposition "this model corresponds to the domain", and **true** for "this model is in the correct syntax".

Model 3 is in both circles, and so has a truth value of **true** for both the propositions mentioned above.

Put another way, model 1 contains accurate information but is not expressed in the correct syntax (language). Model 2 contains inaccurate information but has the correct syntax. Model 3 is both accurate and in the correct syntax.

It is not clear where models that have a truth value of **false** for <u>both</u> propositions might fit in Figure 42 above. Certainly they would be outside both circles.

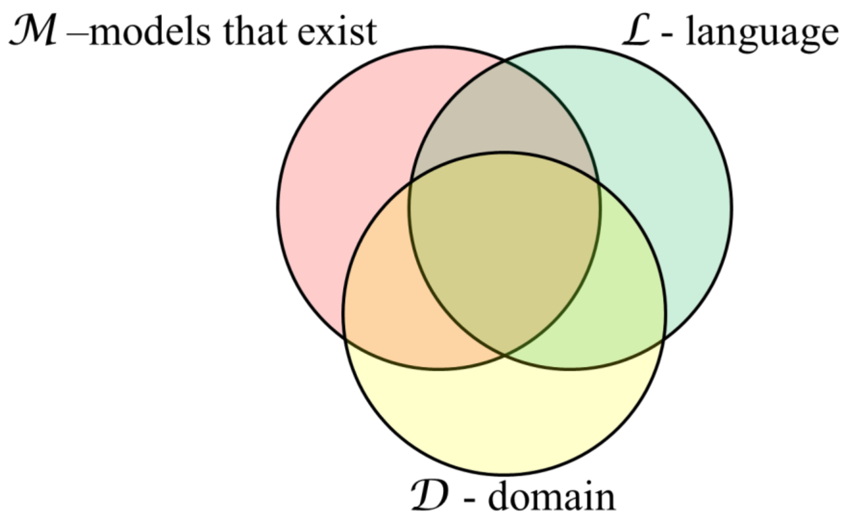
We wanted to find a way of extending the Venn diagram to enable us to account for models that should exist but that did not. This is to enable us to measure the completeness of a set of models. For example, if we would expect to find a particular model that describes the context for each application in our catalogue, but only half of the applications had these models, then we could say that the set of models was only 50% complete.

We therefore added a third circle that corresponds to the proposition “the model actually exists”. This leads us directly to the final form of the Venn diagram and the mathematical model underpinning it, discussed below:

4.3.1 Mathematics of Models, Language and Domain

We use set theory and notation to construct a framework into which we can position individual models (typically diagrams) thus:

Figure 43 Intersection of Existing Models, Domain and Language



Each of the coloured circles represents a particular quality that a model may or may not possess. We can therefore visualise each model as being represented as a point somewhere on the above diagram and the position of the point with respect to the circles tells us whether it not it possesses a particular quality, thus:

(a) If a model (or more precisely, the way the propositions contained within it are expressed) complies with our language that we can visualise it as a point within $\mathcal{L}$; if it does not comply, then it is not within $\mathcal{L}$.

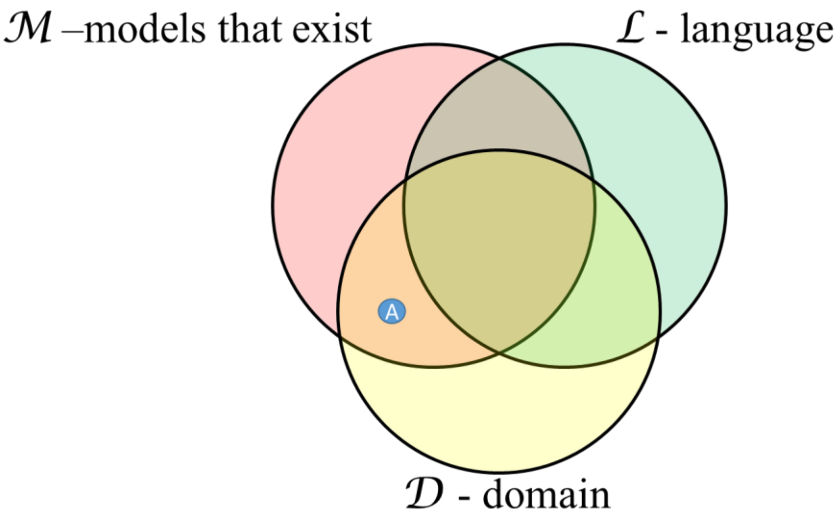
(b) If a model corresponds to the domain (meaning that the propositions contained within the model are 'true' in some sense) then the point is within $\mathcal{D}$; otherwise it is not.

(c) Finally, if there is a reason why a particular model should exist, and it does, then it is within $\mathcal{M}$; otherwise it is outside $\mathcal{M}$.

Using the language of Guizzardi [123], our domain represents a possible world (might reflect reality, or a considered alternative reality – hence 'possible world'); our language is the 'specification language', and models (whether or not they actually exist) are 'logical models' in Guizzardi's terminology.

To illustrate this, consider the following examples. Firstly, if we have a model A that exists, reflects reality (corresponds to our domain), but is not in the correct syntax, then only two of our quality propositions are true, and the model would be positioned as follows:

Figure 44 Exists, corresponds to domain, wrong syntax

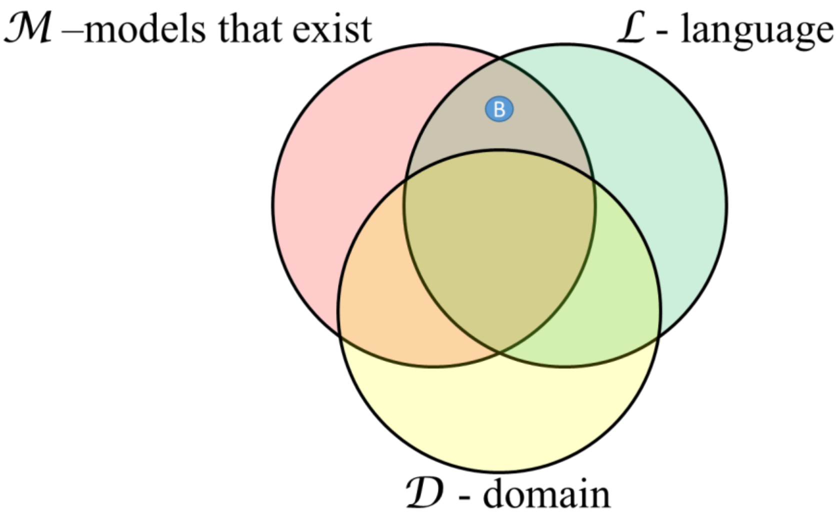


It is in $\mathcal{M}$ and $\mathcal{D}$ but not in $\mathcal{L}$.

This could be for a number of reasons. Perhaps our concept of what constitutes our 'accepted language' is as yet incomplete (see for example discussion in Zarwin et. al. [130] about emergent metamodels and megamodels).

Similarly, if we have a model B that exists, has valid syntax, but does not correspond to our domain, then a different pair of propositions is true:

Figure 45 Exists, doesn't corresponds to domain, right syntax



Looking at these three possibilities, each of which is either **true** or **false**, that of course gives us eight permutations, and it is clear that there is something missing when we compare this with the seven regions in Figure 43 above. The missing permutation relates to models that should exist and do not; that do not correspond to reality (are not true); and that are not in the correct syntax. This is not seen as an important category of models and we shall ignore it henceforth.

What do we mean when we talk about the fact that a model *should* exist? Recall that we can describe the content of an architecture description as comprising catalogues, matrices and diagrams (aka models), as described Figure 129 below. It is entirely possible that we might decide that, say, for our catalogue of current projects, for the purposes of project governance, we would like see a certain set of stakeholder views as standard so, for example a motivation model for the project, possibly a financial (cost) model, a model showing interactions between applications in the project scope, and so on. We might even decide, as part of our project methodology, to allow a project to progress beyond a certain point only if it passes certain criteria, including the production and approval (by the relevant stakeholders) of these standard models (could also be referred to as *diagrams* or *views*, defined perhaps in a set of *viewpoints*). These are possible examples of when we might consider that a model *should* exist. Clearly, in order for any statement involving 'should', there needs to be an expectation –

a norm – as to what constitutes completeness. This should perhaps be considered as part of the task of defining how an organisation does architecture work (maybe to be tackled as part of Phase 0 – Framework and Principles – if using the TOGAF Architecture Development Method to develop our architecture).

Summarising, therefore, if we take a set of models $\mathcal{M}$ and plot each one as a point somewhere on Figure 43 above, then we can use simple ratios to derive our metrics.

4.3.2 Region by Region

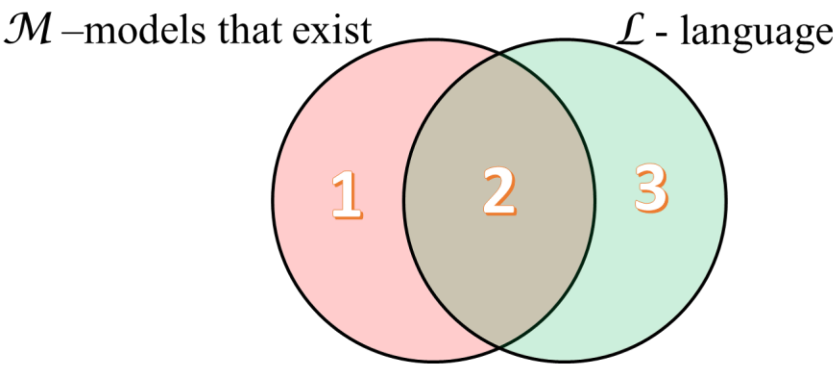
Not every possible region in our Venn diagram (permutation of quality truth values) is useful to us. We consider here each region in turn and discuss its significance. Let us assume for now that anything in the domain $\mathcal{D}$ is, in theory, capable of being modelled in language $\mathcal{L}$. That is, $\mathcal{L}$ is sufficiently comprehensive to be capable in principle of modelling the whole of $\mathcal{D}$.

As previously discussed, each individual model can be considered as a point within one or more of the circles. We start our analysis by considering pairs of circles on our Venn diagram:

4.3.2.1 Regions from Proposition Pairs

4.3.2.1.1 M and L regions

Figure 46 M and L Regions



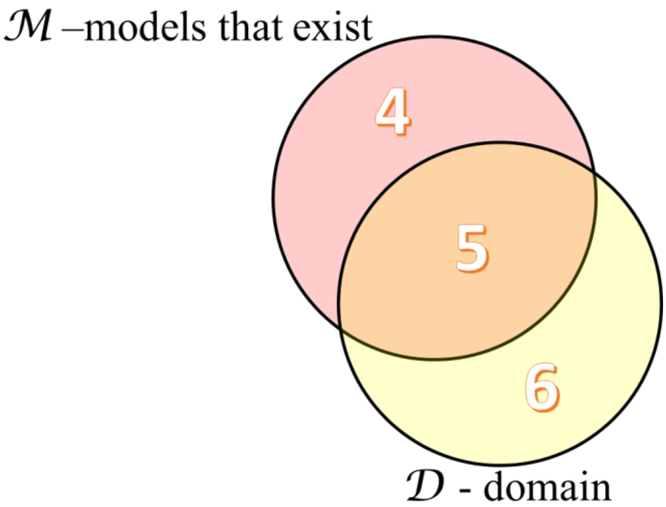
Region 1 represents existing models that are not compliant with the agreed modelling notation (language) $\mathcal{L}$ for our enterprise, or for a particular architecture repository.

Region 2 represents existing models that are compliant and written as (syntactically) correct statements in the modelling language.

Region 3 represents models, or types of models, that would be compliant with our language $\mathcal{L}$, which should have been created (according to expectations and guidelines within the organisation, perhaps as defined within particular standard viewpoints) but which are in fact missing.

4.3.2.1.2 M and D regions

Figure 47 M and D Regions



Region 4 represents existing models that do not reflect the domain, either its current state or a possible future state. This could be a model that was created some time ago, reflecting what was deployed previously, but which is now out of date because the environment has since changed. A very relevant example from our case study would be a logical information model that described the structure of a particular type of record. If the actual implementation (for example the structure of a database table) was modified, but the logical design was not, then the logical design no longer reflects reality, and so has moved (say) from region 5 to region 4.

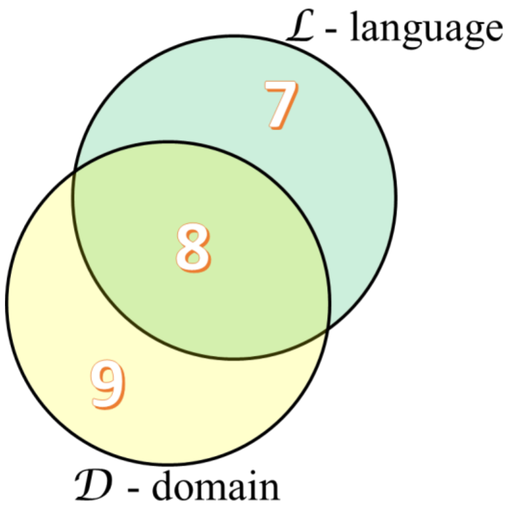
Region 5 represents models that do exist, and that correspond to our domain (in other words, “tell the truth”).

Region 6 represents models that should exist, given the scope (domain) of our architecture modelling, but that do not.

The modelling language or syntax is irrelevant to regions 4, 5 and 6 because the modelling language $\mathcal{L}$ is not included in this diagram.

4.3.2.1.3 L and D regions

Figure 48 L and D Regions



Region 7 represents existing or possible models (note that the set of models that actually exists, $\mathcal{M}$, is excluded from this diagram) that are compliant with our agreed language $\mathcal{L}$, but which are outside the domain – in other words, do not accurately reflect reality, or a possible future reality. They are formatted correctly but do not tell the truth about our domain.

Region 8 represents models (existing or possible) within our domain of interest (actual or possible reality) that are also compliant with $\mathcal{L}$.

Region 9 represents models (existing or possible) within our domain of interest that are in theory capable of being represented in a form compliant with $\mathcal{L}$, but which are not in fact compliant with $\mathcal{L}$. This assumes again that all relevant truth about $\mathcal{D}$ can be modelled in $\mathcal{L}$; we consider how to deal with exceptions to this in a later section.

4.3.2.1.4 Mathematical Definitions for Proposition Pair Regions

Models in these regions are summarized mathematically as:

Table 19 Mathematical Definitions of Two-Part Model Regions

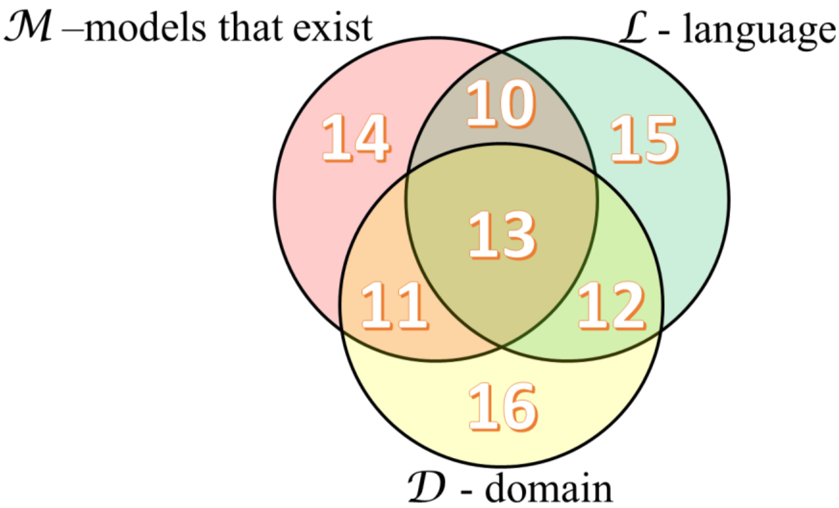
Region	Definition	Region	Definition
1	$m \in M \setminus L$	6	$m \in D \setminus M$

Region	Definition	Region	Definition
2	$m \in M \cap L$	7	$m \in L \setminus D$
3	$m \in L \setminus M$	8	$m \in L \cap D$
4	$m \in M \setminus D$	9	$m \in D \setminus L$
5	$m \in M \cap D$		

4.3.2.2 Full Region Analysis

If we take each smaller region in turn, now using all three regions, then the meaning of each region is shown below.

Figure 49 M, L and D Regions



The smaller regions have the following meaning, and significance:

4.3.2.2.1 Region 10

This represents models that exist (in $\mathcal{M}$), have the correct syntax (in $\mathcal{L}$), but do not correspond to our domain (not in $\mathcal{D}$). Like all the regions in $\mathcal{M}$, these represent existing models and so have significance in that the correspondence to the domain and compliance to the language can be evaluated. An example for this particular region might be a valid ArchiMate model (if that is our chosen language) that showed a flow of information between application components that no longer existed.

4.3.2.2.2 Region 11

This represents models that exist (in $\mathcal{M}$), do not have the correct syntax (not in $\mathcal{L}$), but do correspond to our domain (in $\mathcal{D}$).

An example of this might be a diagram showing the actual interactions between two systems, but failing to use the correct notation.

4.3.2.2.3 Region 12

This region, along with regions 15 and 16, are used to denote models that should exist, according to the standards and guidelines of our particular organisation, but that are missing.

Region 12 is the particular variant that is of most interest, because it highlights “good quality” models (at least, as far as the correspondence and syntactical metrics go) that we wish were present but are not so.

This might be a standard model that we expect for all of the business capabilities in our organisation, showing the business process and organisational units involved in their execution. We cannot of course measure the quality of a model that does not exist and so, for all intents and purposes, we could simply combine regions 12, 15 and 16.

4.3.2.2.4 Region 13

This is for our ideal model – it exists, it corresponds to our domain, and it is in the correct syntax. If all our models were in this particular region, then we would have a perfect score for each of our three quality metrics.

An example might be an ArchiMate model of an application, using the correct elements and relations, that reflected exactly the interfaces and flows connecting it to other applications.

4.3.2.2.5 Region 14

These are for existing models that fail both the other quality tests: they do not correspond to our domain, and they are not in the correct syntax.

An example might be a diagram drawn in basic office software that attempted to describe the contents of a customer record; but that failed to give the correct fields, and was not in an approved syntax (e.g. UML class diagram or Entity Relationship Diagram).

4.3.2.2.6 Region 15

This is not significant in its own right; it is for models that could be constructed (but have not been), that do not correspond to the domain. Models that do not exist cannot be tested, and so this is not a useful region in its own right.

For example, in theory one could create a business service model in the correct syntax that did not reflect the services provided by an organisation. The theoretical possibility of such an activity has no real world significance and so is not considered significant or useful.

4.3.2.2.7 Region 16

Like region 15, this has no real world significance in its own right. One could construct a model that correctly showed the services provided by an organisation, in an invalid syntax. As one cannot test the syntax of a model that does not exist, this is not considered significant or useful.

4.3.2.2.8 Summary of Regional Analysis

All the regions within $\mathcal{M}$ have meaning in their own right. The regions outside of $\mathcal{M}$ (12, 15 and 16), as they are not capable of being tested, can, in effect, be treated as a single region.

4.3.3 Deriving Syntactical Quality Metric

If all of our models were syntactically correct, then we would have a 'perfect score', and in terms of a ratio, we would like this to be 1, representing 100% of our models being in region. We therefore need simply to count the number of models that exist, that are in the correct syntax and compare that with the number of models that exist (irrespective of syntax). The former is given by the population of the intersection of regions $\mathcal{L}$ and $\mathcal{M}$, the latter by the population of region $\mathcal{M}$. This is the origin of the formula:

$$Q_S = \frac{|\,M \cap L\,|}{|\,M\,|} \qquad (1)$$

If all the models that exist (points in region $\mathcal{M}$) are also in the correct syntax (points in region $\mathcal{L}$), then every point in $\mathcal{M}$ is also in $\mathcal{L}$ and so the intersection of the two sets $\mathcal{M}$ and $\mathcal{L}$ is the same as the population of $\mathcal{M}$ (or putting it another way, $\mathcal{M} \setminus \mathcal{L} = \emptyset$, as there is nothing in $\mathcal{M}$ that is not also in $\mathcal{L}$) and the equation (1) will yield a value of 1. Conversely, if every existing model has an invalid syntax, then all members of $\mathcal{M}$ are outside $\mathcal{L}$ (i.e. $\mathcal{M} \setminus \mathcal{L} = \mathcal{M}$), the intersection of the two sets will be empty ($\mathcal{M} \cap \mathcal{L} = \emptyset$), and equation (1) will yield a value of 0. Thus, our syntactical quality metric will always have a value in the range:

$0 \leq Q_S \leq 1$

As this metric is not concerned with correspondence to the domain ("is the model telling us the truth"), the $\mathcal{D}$ proposition does not feature in this metric.

4.3.4 Deriving the Semantic Quality Metric

This is similar to the previous metric, however instead of being interested in syntax (region $\mathcal{L}$), we are interested in correspondence to our domain, so region $\mathcal{D}$. Thus, our metric is of a similar form to the previous one:

$$Q_A = \frac{|\, M \cap D \,|}{|\, M \,|} \qquad (2)$$

If all the models correspond to the domain, then every member of $\mathcal{M}$ is also a member of $\mathcal{D}$, and size (cardinality) of $\mathcal{M} \cap \mathcal{D}$ will be the same as the cardinality of $\mathcal{M}$, thus giving a value of 1 for this metric. Again, if none of the models correspond to the domain, then $\mathcal{M} \cap \mathcal{D} = \varnothing$ and equation (2) will give a value of 0. Thus, we have again a metric that yields a carefully bounded value:

$0 \leq Q_A \leq 1$

4.3.5 Deriving the Completeness Quality Metric

For our final metric, we consider all the models that *should* exist, and compare that with the number of those that *actually* exist (irrespective of their correspondence to truth, or their syntax). Now our use of the set $\mathcal{D}$ here specifically refers to the set of models that we would expect to see, in order to completely (however we wish to define that) model our domain.

This gives us a ratio we can use as our metric. So, for example, if we would expect to find 120 models of various types, but only found 90, then that would give us a ratio of 75% or 0.75. This is the origin of the formula:

$$Q_C = \frac{|\, M \cap D \,|}{|\, D \,|} \qquad (3)$$

As with the other metrics, this is bounded:

$0 \leq Q_C \leq 1$

Thus, we have three metrics that between them, can be used to describe three aspects of the quality of a set of models $\mathcal{M}$.

4.3.6 Change in Quality over Time

The metrics will of course change over time because the number of models that we have and the quality of those models will change over time as new models are created and existing models are updated.

Now, we use the concept of a derivative and ask what conditions need to be satisfied in order for the quality of the set of models to continue to increase over time. Put another way, what needs to be true in order for our models to "get better"?

As we saw in section 3.3.4 above, this requires our derivatives with respect to time to be greater than zero. We use the quotient derivative rule, and apply it to each of the formulae in sections 4.3.3, 4.3.4 and 4.3.5 above.

4.3.6.1 Convergence of Syntactical Quality

For example, dealing with the first metric, $Q_S = \frac{|M \cap L|}{|M|}$, remembering our quotient formula is expressed in terms of two functions f(x) and g(x); if we set f(x) to be $|M \cap L|$, and g(x) to be $|M|$, then applying the formulae we directly obtain the following for the derivative (rate of change) of Qs with respect to time:

$$\frac{d}{dt}Q_S = \frac{|M \cap L|'.|M| - |M \cap L|.|M|'}{|M|^2} \qquad (4)$$

For this derivative (and the other two, as they all have squared terms in their denominator), in order for the overall derivative to be positive (meaning "getting better"), we require the numerator also to be positive, thus:

$$|M \cap L|'.|M| - |M \cap L|.|M|' > 0 \qquad (5)$$

Now if no new models are being created, and all modelling effort is focused on correcting the syntax of the models that already exist, then the total number of models |M| will not change, and so we can see that |M|′ is zero. As a consequence, we just require $|M \cap L|'.|M > 0$. If models are being corrected, so moved into the intersection of M and L, then the cardinality (population) of this region will be increasing, and so overall our metric for syntactical quality will be increasing. If, however, people continue to create new models and those are created with an incorrect syntax, then it will be harder for our overall syntactical quality to increase. We should then require, by rearranging equation (5):

$$\frac{|M \cap L|'}{|M|'} > \frac{|M \cap L|}{|M|} \qquad (6)$$

In other words, the ratio of the rate of change of correctly formatted models to the rate of change of all models needs to be greater than the ratio of the number of correctly formatted models to the number of all models. If there are a low percentage of correctly formatted models, then it will be relatively easy to achieve this inequality. For example, if only 1 in every 10 models is correctly formatted, then we only require:

$$\frac{|\, M \cap L \,|'}{|\, M \,|'} > 0.1$$

In order to increase our syntactical quality metric, the ratio of new correctly formatted models to the number of new incorrectly formatted models just needs to be more than this (relatively low) target. We can see however that as our overall model quality increases, it would be relatively easy for the overall syntactic quality to slip backwards as we require the above ratio to be higher and higher, approaching 1.

4.3.6.2 Convergence of Semantical Quality

By applying the same technique (quotient rule) to differentiate equation (2), we obtain:

$$\frac{d}{dt} Q_A = \frac{|\, M \cap D \,|'.|\, M \,| - |\, M \cap D \,|.|\, M \,|'}{|\, M \,|^2} \qquad (7)$$

Knowing that $|\, M \,|^2 > 0$, we obtain a similar result to equation (5) for increasing semantical quality (correspondence to our domain):

$$|\, M \ \cap \ D \,|'.|\, M \,| - |\, M \ \cap \ D \,|.|\, M \,|' > 0 \qquad (8)$$

Now as before, if no new models are being created, and all effort is directed for a time into modifying the existing models to make them accurately reflect our domain (i.e. describe what actually exists, or might (a solution option)), then as with syntactical quality, $|\, M \,|'$ will be zero, and so from inspection, if any models are being modified to make them accurate, then $|\, M \ \cap \ D \,|'$ will be positive, and thus Q_A will inevitably increase over time. However, if people continue to create new models that do not reflect [a possible] reality, then it will be much harder for the overall semantical quality to increase. We should then require, rearranging (8):

$$\frac{|\, M \cap D \,|'}{|\, M \,|'} > \frac{|\, M \cap D \,|}{|\, M \,|} \qquad (9)$$

Put into words, the ratio of the rate of change of accurate models to the rate of change of all models needs to be greater than the ratio of the number of accurate models to the number of all models.

As discussed in the previous section on syntactical quality, this task may get harder as the overall semantical quality of the model [set] increases.

4.3.6.3 Convergence of Completeness

Applying the quotient rule to equation (3) we obtain:

$$\frac{d}{dt}Q_C = \frac{|M \cap D|'.|M| - |M \cap D|.|M|'}{|D|^2} \qquad (10)$$

Our condition for increasing completeness is simply that the numerator is positive (as the denominator will always be positive, as it is squared), in other words,

$$|M \cap D|'.|M| - |M \cap D|.|M|' > 0 \qquad (11)$$

This is interestingly the same condition that is required for increasing semantical quality.

The only difference in these measures is in magnitude, as the two measures have different (positive) denominators.

The more models that actually exist, the slower it will be, for a given modelling effort, to increase Q_A, whereas the more models that should exist, the slower it will be to change Q_C.

4.3.7 Extension for Evolving Maturity

In organisations where their Enterprise Architecture work is at a low level of maturity (for a discussion on Enterprise Architecture maturity see for example [204]), there may be no clear EA framework or methodology in place, and thus it may not be clear what should be in the EA models that are the subject of our testing. In such circumstances there may be ideas that need to be included in the models, for which we had not yet decided upon the correct syntax.

This might apply, for example, where we want to include an element of business process modelling, but had not yet decided whether to use BPMN [205] or Event-Driven Process Chains [206]. Another example might be the need to model information architecture without having yet determined whether to complement ArchiMate (which does not do detailed information modelling) with UML [40] Class Diagrams or Entity Relationship Diagrams [207]. We would therefore want to exclude our business process and/or information architecture diagrams from the need to comply with a particular language or syntax.

Without being clear about how we want to model these concepts, we cannot simply extend our definition of our language(s) $\mathcal{L}$. The original concept, as discussed in [56] did not make provision for this, however we suggest this is a worthwhile addition to our quality framework, to take into account situations where our choice of modelling language(s) is still evolving. Thus, we want to find a way of excluding from the set of all models ($\mathcal{M}$) those that currently are not capable of being expressed in language $\mathcal{L}$. Thus we introduce a new notation which divides the set of models $\mathcal{M}$ thus:

Models (members of $\mathcal{M}$) falling inside the scope of language $\mathcal{L}$ – that is, models with content that can be expressed completely using language $\mathcal{L}$ – comprise set $\mathcal{M}_S$

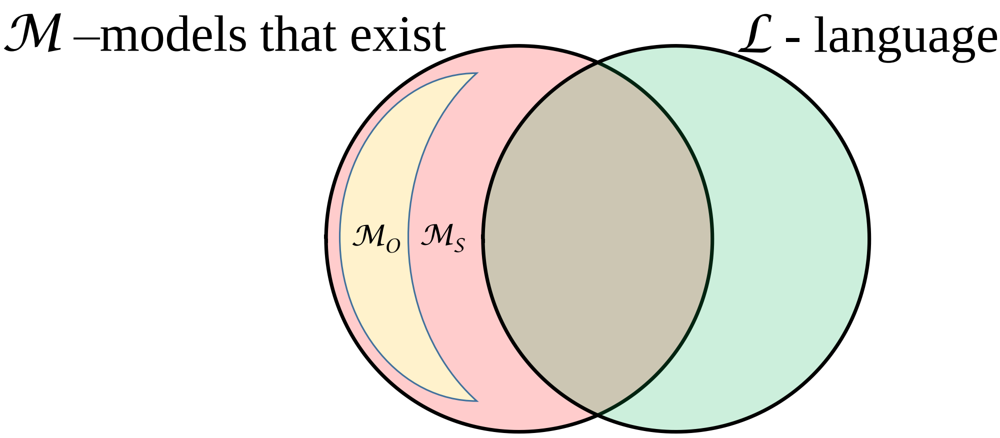
Models (members of $\mathcal{M}$) falling outside the scope of language $\mathcal{L}$ – that is, models with content that cannot be expressed completely using language $\mathcal{L}$ – comprise set $\mathcal{M}_O$

Thus, our set of models $\mathcal{M}$ is divided into these two sets, so that $\mathcal{M} = \mathcal{M}_S \cup \mathcal{M}_O$

Either a model can be fully expressed in our given language(s), or it cannot be; it cannot be both expressible and inexpressible at the same time, so that $\mathcal{M}_S \cap \mathcal{M}_O = \emptyset$

We can therefore view the relationship between $\mathcal{M}_S$, $\mathcal{M}_O$, $\mathcal{M}$ and $\mathcal{L}$ thus:

Figure 50 Relationship between regions of M and L



Any models in $\mathcal{M}_O$ can never, in principle, be expressed in a form compatible with our chosen language (syntactically correct). Therefore, in order not to bias the syntactical quality metric with models that cannot, for good reasons, become compliant with our language, we need to exclude those from our model count. Of course, models in $\mathcal{M}_S$ may or may not be compliant with our language.

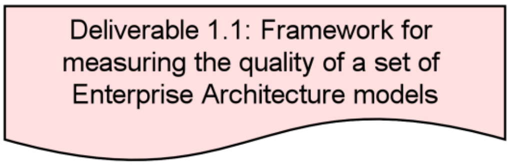
Thus, we replace the overall count of models in $\mathcal{M}$ with the count of models in $\mathcal{M}_S$, to produce modified versions of our equations for our syntactical quality metric:

$$Q_S = \frac{|\, M \cap L \,|}{|\, M_S \,|} \qquad (12)$$

$$\frac{d}{dt} Q_S = \frac{|\, M \cap L \,|' . |\, M_S \,| - |\, M \cap L \,| . |\, M_S \,|'}{|\, M_S \,|^2} \qquad (13)$$

If there is nothing in our (existing) models outside the scope of our agreed languages, then the sets M_S and M are identical.

This completes the framework, and thus the following deliverable:



4.4 Challenges and Limitations

4.4.1 What is truth?

In the words of Pilate[16], what is truth? Some definitions talk about "the property (as of a statement) of being in accord with fact or reality" and "fidelity to an original or to a standard" [208]. If our model is describing something that exists in the real world, then we can compare our model with the reality and see if they match. This is harder if part of what our model is trying to describe is aspect of a possible future reality, for example, going back to our project example, "*If* this project goes ahead, then there *will exist* a [new] interface between these applications".

Another way of interpreting this 'correspondence' property would be to look for correspondence (agreement or consistency) between different models on the same subject.

In our next chapter we describe the case study that we carried out to test the theoretical approach described in this chapter.

4.4.2 Reflections on published paper

In the abbreviated version of this research published in 2016 ([61], attached as Appendix D below) the description of region $\mathcal{3}$ reads:

> *Region $\mathcal{3}$ represents models, or types of models, that would be compliant with our language $\mathcal{L}$, but which have not been created (perhaps because they are not required). An example of this might be UML interaction diagrams, if our agreed notation is UML but we have only hitherto used (say) sequence and use case diagrams. [51]*

This is somewhat misleading. We are concerned, for our completeness metric, about models that should exist but that do not. The description included within the main body of this thesis, in section 4.3.2.1.1 above, has been corrected.

---

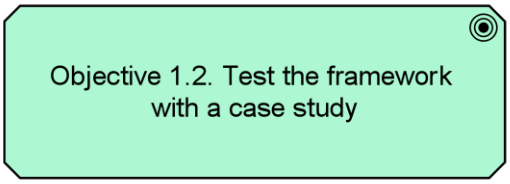
[16] John 18:38

4.4.3 Reflections on other aspects of EA Content

In our literature survey, related to Enterprise Architecture, we reviewed a 2013 paper by Bean [120] that contains, among other things, a table showing aspects to consider for items in our Enterprise Architecture models (see Table 9 on page 129). We have focused our research on three specific quality aspects (completeness, syntax, accuracy). In our view, it would be feasible to construct specific metrics to measure some more of the aspects that she proposes. This would be fruitful area for further research.

Chapter 5. Case Study: Measuring the Quality of a Set of EA Models

In this chapter we select and analyse a case study in the application of the theoretical approach, described in Chapter 4 above, to a real-world situation. This therefore addresses the following objective:

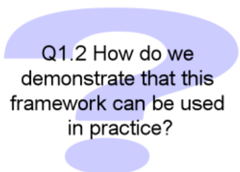


5.1 Application of Quantitative Process

Here we summarise how our research has followed the five-step process outlined in 3.1.2 above.

5.1.1 Determining the basic questions to be answered

This is of course related to our first research question (RQ1), and the specific question that our case study is designed to answer is related to the practicality of the measurement framework:



5.1.2 Determining the participants in the study

In our case, the case study is simply an examination of existing electronic materials, and thus there are no participants.

5.1.3 Selecting the methods needed (variables, measures, overall design)

The methods and design for this research is contained in Chapter 4 above.

5.1.4 Selecting analysis tools

The analysis tools are purely mathematical, using them is simply a matter of collecting the information and using the numbers at the appropriate place in the formulae. The mathematical results are very simple and can be seen in section 5.4 below.

5.1.5 Understanding and interpreting the results

This can be found in section 5.5 below.

5.2 Context for Case Study

An opportunity arose during the course of our employment as an IT consultant, when working with a particular public sector client, to examine an existing set of EA models (specifically related to Information Architecture), and to give an opinion on the quality of those models.

As discussed in section 3.1.8 above (in particular, Table 11), where we look at the characteristics of case studies, experiments and surveys, the characteristics of a case study are:

(a) Investigation of a relatively small number of cases (sometimes just one).
(b) Information gathered and analysed about a large number of features of each case
(c) Study of naturally occurring cases or, in 'action research' form, study of cases created by the actions of the researcher but where the primary concern is not controlling variables to measure their effects.
(d) Quantification of data is not a priority. Indeed, qualitative data may be treated as superior.
(e) The main concern may be with understanding the case studied in itself, with no interest in theoretical inference or empirical generalisation. However, there may also be attempts at one or other, or both, of these. Alternatively, the wider relevance of the findings may be conceptualised in terms of the provision of vicarious experience, as a basis for 'naturalistic generalisation' or 'transferability'. (from [174])

It seemed, considering the criteria listed above, that this particular opportunity was suitable for a limited case study of a particular kind:

(a) We had a single case to examine at that time;
(b) The data appeared to us to be reasonably complex;

(c) We wished to study a real-life 'naturally occurring' case, which was not artificial, and would be reasonably representative of the state of EA initiatives of many organisations (i.e. immature);
(d) Diverging from a typical case study, we were, in this instance, more interested in quantitative data;
(e) We would be interested to know if this could be generalised or transferred to other cases and situations.

In section 3.1.8.1.3 above, we discuss how to select cases for a case study, and see a number of selection procedures, and argue that ours is a Typical selection method, designed to probe causal mechanisms that may either confirm or disconfirm our theory that it is possible to measure EA model quality using our particular novel framework. We argue in section 3.1.8.1.4 above, considering methodological issues, that our key concern is that of generalizability, and we address this by means of a deductive *modus ponens* argument that the practicality of our framework can indeed be generalised to be useful in other scenarios.

This particular public sector client, had hired a team of consultants from our employer that included some software development teams who were documenting their designs in a single Wiki [188] intended to act as a repository for architecture information across the organisation.

The consulting team had been asked to propose a new structure for this architecture information to be used by these various teams and thus, by implication, a new architecture framework and a corresponding representation for the entities and relations in the framework to be held somehow within their Wiki site. The intention was that each entity (instance of a class in the specified metamodel) should be represented as a particular Wiki page, based upon a template that was specific to that class. Furthermore, relations between entities were to be represented as hyperlinks between the entity pages. Different relation types were indicated by hyperlinks being in different (standard) sections of each wiki page.

So why did this suit our choice of a case study? Looking at the above definition of a case study, we can see a match against four of the five criteria:

(a) We are indeed dealing with a single case;
(b) There is potential a lot of data to be captured about the case (given the complex nature of the Wiki that had already been built), although over time we restricted our attention to a particular subset of that Wiki;
(c) The case was 'naturally occurring'; we had not interfered with the data, or the people creating it, in any way;

(d) In this one aspect, this did not match a typical case study, in that we were interested in quantitative data;

(e) We were absolutely interested in conceptualising and generalising the result to understand if this measurement framework could be used in other situations

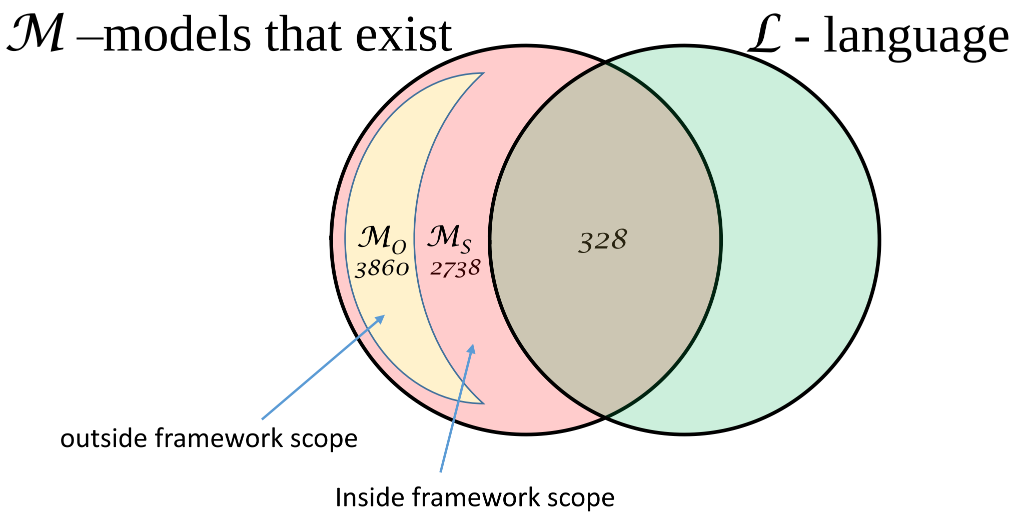
5.3 Data Collection

5.3.1 Collecting Data on Syntactical Quality

The reporting tools built into the Wiki enabled us to do counting of pages and helpfully was able to provide a count of all pages built from a particular template. This meant that we were able to count all pages that were of a particular entity type (class). It also enabled us to report on Wiki pages that had not yet been assigned an appropriate template. This was significant because it was the template that caused the correct structure to be assigned to the page and thus, we were able to conclude that pages that did not have a correct template applied could be deduced to be in the wrong syntax.

Thus, for syntactical quality, our figures were as follows:

Figure 51 Case study figures: syntactical quality



We used the subdivisions of $\mathcal{M}$ thus: knowing that many of the pages in the Wiki held information that was not yet in the scope (language) of the new emerging architecture framework that we were developing, we did not expect pages not in the scope of the framework to comply with the language (syntax). Thus, although the Wiki had a total page count of 6,598, of these only 2,738 were in the scope of the framework. Thus, for our calculations, we used | $\mathcal{M}_S$ | = 2738.

Of the 2738 existing pages in our architectural set of models, only 328 were in the correct syntax, based upon an analysis from the built-in tools. Thus | $M \cap L$ | = 328.

These two figures were all that were required to fulfil the syntactical quality metric.

5.3.2 Collecting Data on Completeness

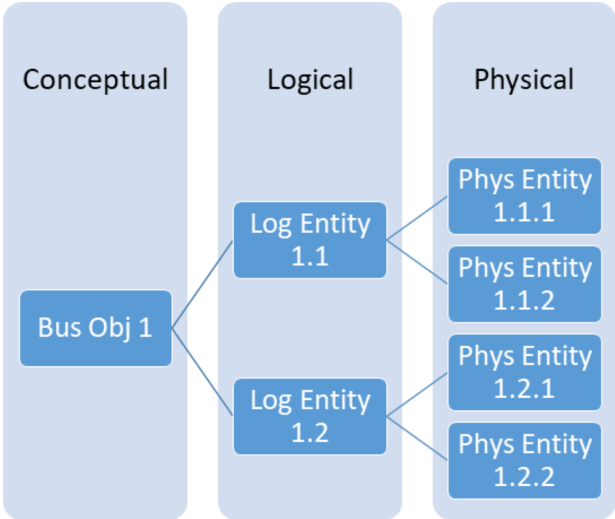
To do this required us to estimate the number of pages required. This was a manually intensive calculation, assisted in part by the built-in tools, using the following generic algorithm:

Figure 52 Algorithm to calculate expected page count

```
  SET expected page count to 0
  FOR each entity in the new metamodel
     IF a complete set of instances exists
  for that entity
     THEN
     Add the number of instances to our expected page count
     OR
     Approximate the number of instances by reference to another
entity
  Add that approximation to our expected page count
     ENDIF
  END LOOP
```

The expectation for page counts by reference to another entity was related to the fact that we were using a custom architecture framework that, in common with some others (in particular Zachman [74]) was split into Contextual, Conceptual, Logical and Physical layers. These layers were applied to a number of types of concept, including, for example Information Architecture, which was one of the major focus areas for this work. The contextual layer was used for overarching governance information and then the detailed entities appeared in the other three layers. Thus, for Information Architecture, we had entities called Business Objects in the conceptual layer, which were decomposed into a number of Logical Entities in the logical layer, and then further into Physical Entities in the physical layer, as illustrated below:

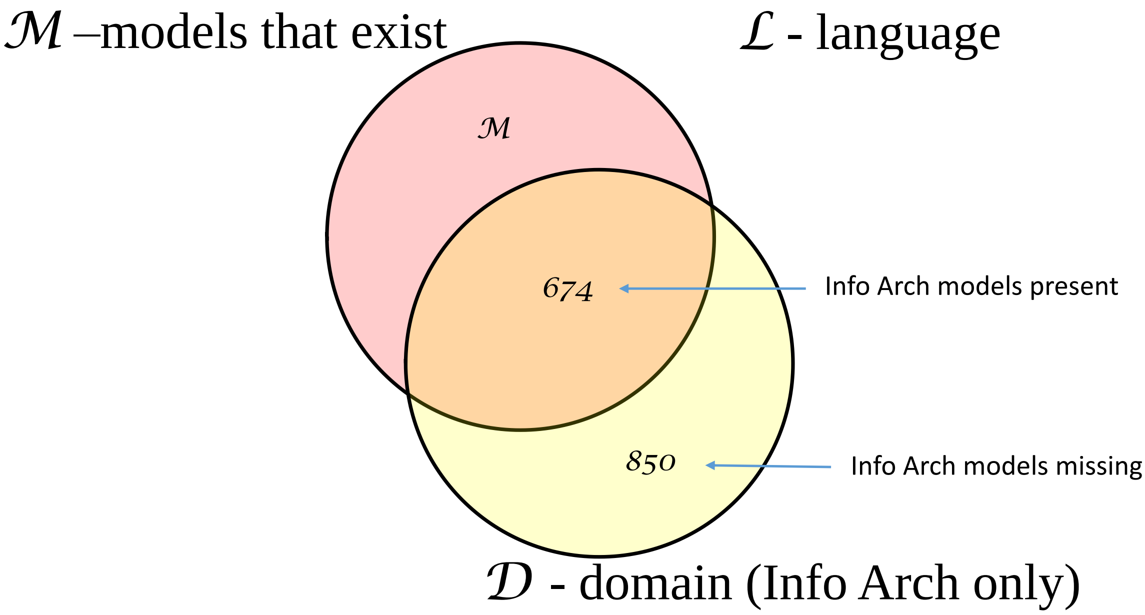
Figure 53 Case Study – Architectural Layers and Entities



We believed that we had a complete set of Business Objects. For some Business Objects, we knew we had a complete set of Logical Entities. For others, we did not. However, we were able to estimate how many there should be by looking at the average number of Logical Entities per Business Object where we did have a complete set. Thus, by constructing approximate empirical ratios to relate the expected numbers of entities in each layer, we were able to make a rough calculation as to how many pages (entity models) might be missing. Obviously, the more complete sets of entities exist, the more accurate this kind of estimation would be.

Across the complete set of entities, for Information Architecture in particular, we found that whereas we would have expected to see 1,524 models (one for each entity instance, if fully populated), in fact we could only see 674:

Figure 54 Case Study figures – completeness



These two figures formed the required data for the completeness calculation, at least as regards the Information Architecture models (thus restricting D to this particular subset of the architecture).

We did not feel that there was sufficient data available to enable us to attempt to measure the correspondence/accuracy metric Q_A.

5.3.3 Data for Accuracy

Insufficient time, plus insufficient technical resources from the project teams, were available to carry out comparisons of the actual database structures compared to the descriptions in the Wiki page (i.e. checking for correspondence with reality), and so we did not attempt to measure this particular metric.

5.4 Mathematical Results

This section we look at the mathematical results obtained by applying the formulae previously derived to the statistics obtained in our pilot study.

From our figures obtained in 5.3.1 above, we were able to obtain a figure for syntactical quality:

$Q_S = \frac{328}{2738} =$ 0.12, or 12%.

For our second metric, completeness, as shown in 5.3.2 above, we had 674 pages (related to Information Architecture) where we would have hoped for 1524, so this leads to the following simple calculation:

$Q_C = \frac{674}{1524} = 0.44$, or 44%.

We were unable to make any calculation for Q_A.

This concludes the case study, and thus our next deliverable:

Deliverable 1.2. Case study showing results of applying framework to a specific real-life scenario

5.5 Research Results

Here we consider what we actually learnt from this research that is applicable elsewhere. The question of whether one example is enough, or whether this could be repeated elsewhere, is specifically covered in section 5.5.3 below.

5.5.1 Theoretical Framework

In this chapter we have successfully constructed a mathematical framework, adopting and extending ideas from the literature, that enables us to, at least in principle, assess objectively the quality of a set of enterprise architecture models in three dimensions:

- Syntactical Quality, Q_S
- Semantical Quality, Q_A
- Completeness, Q_C

5.5.2 Practically Useful

We demonstrated that two of them, Q_S and Q_C, can be used in practice. We have not, as yet, demonstrated that Q_A is practical, although we can see ways in which could feasibly be done.

5.5.3 Repeatability of the Approach

We now want to demonstrate the truth of one of our premises given in the deductive argument given in section 3.1.8.1.1 above. If the framework can be used once in practice (which we have demonstrated), can it be used again, in other situations? Let us consider each of the three metrics in turn, and see whether they could be used in a variety of cases (as opposed to just the single case we tested). We will do this from both a theoretical and practical perspective; the latter will use a similar example from another organisation which is also embarked on an Enterprise Architecture modelling initiative.

5.5.3.1 Repeatability in Theory

- Syntactical Quality, Q_S

Our ability to measure this depends on two things: (a) the existence of an agreed language $\mathcal{L}$, and (b) our ability to measure conformance with that agreed language, across a set of models $\mathcal{M}$ in the particular case. There is nothing special about the specified language in this particular case; it would work with any language, including ArchiMate, or a subset thereof. As long as we are able to examine a set of EA models and check their conformance with an agreed language, this will work.

- Semantical Quality, Q_A

Our ability to measure this depends on two things: (a) the existence of a set of models $\mathcal{M}$, and (b) the definition of a domain $\mathcal{D}$ against which we can check our models for correspondence (i.e. are they telling truth). Now in our test case, we failed to demonstrate the practicality of measuring Q_A, however this was more for lack of time than a failure in the theoretical method. Nevertheless, our deductive argument fails to demonstrate the repeatability of a method to measure Q_A because it fails premise 2 section 3.1.8.1.1 above. However, this does nothing to invalidate the argument regarding the repeatability of the measurements of Q_A and Q_C.

- Completeness, Q_C

Our ability to measure this depends on two things: (a) the existence of a set of models $\mathcal{M}$, and (b) the definition of a domain $\mathcal{D}$ specified in terms of a set of models which *should* exist. If we have a set of models, and we have a definition of what models we think should exist, then that is sufficient to measure Q_C. There is nothing special about this one case study that cannot be repeated elsewhere, as long as we can define what models we should expect.

5.5.3.2 Repeatability in Practice

In our previous employment, we used a tool-based repository to manage our Enterprise Architecture models, and were able to report in detail on the content of the models in that repository, covering a number of aspects including their syntax (for example, using constructions or elements that did not comply with our standards). We actually extended the reporting to cover additional quality aspects (for example, were the models using the right referent when they referred to a particular catalogue item, or was it taken from the wrong place in the repository?). Thus, our notion of Q_S could be extended, and this is one of the aspects that we are anticipating formalising in future research.

In our current employment, we have gone further than merely checking that models that should exist, do exist; and instead, create any reports found to be missing and automatically populate them from data already in the repository (this is described in section 8.6.1.1 below). This gives grounds again to believe that the same techniques illustrated in the initial case study can be used elsewhere.

Thus, by the deductive argument posed previously, we have demonstrated a repeatable method for measuring two of the three quality metrics.

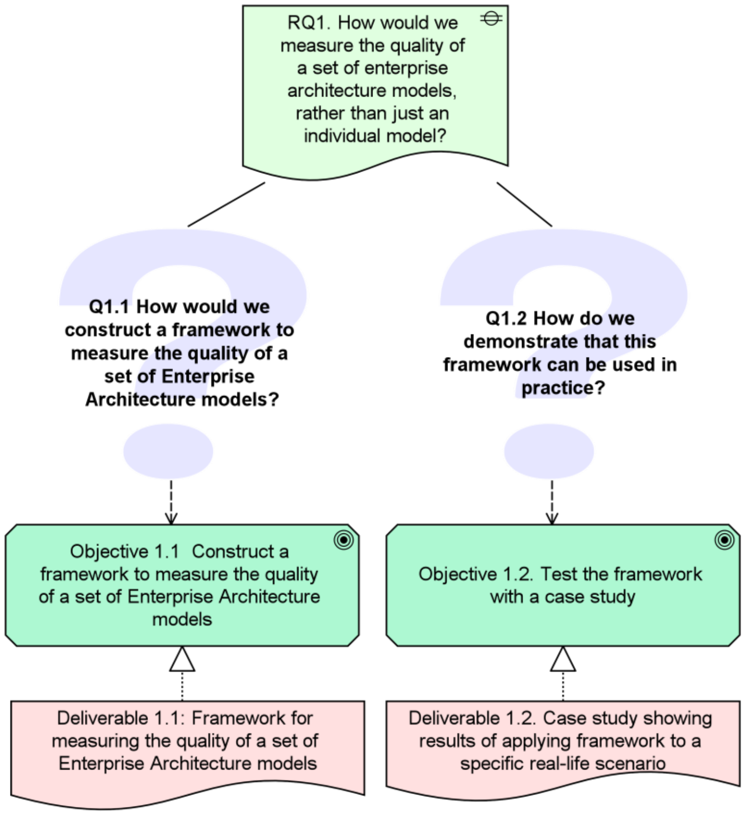
5.5.4 Recapitulation of Research Question

Our research question was RQ1, *How can we measure the quality of a set of Enterprise Architecture models*

We have demonstrated this by constructing a theoretical framework with three quality dimensions (syntactic, semantic and completeness) and then demonstrating that at least two of the quality dimensions work in practice, as well as indications as to how the third dimension could also be used.

We have thus completed the following:

Figure 55 Completion of RQ1



5.6 Reflections

5.6.1 Correspondence/Accuracy Metric

We were unable to perform a measurement of the accuracy metric Q_A. In the case study, there was a clear practical way in which this could have been done, had more time been available: a comparison of the physical database structures against the logical database models forming part of the Information Architecture.

Checking correspondence with the domain – or the subset of reality within the domain – can be harder when what we are checking is not something tangible – something that can be examined directly - but is instead an abstract or conceptual idea, such as a business service. How does one determine objectively what business services are actually being provided by an organisation, when there are many ways that they could be modelled?

Another difficulty with this metric is that sometimes our models are exploring a possible future state, rather than the current state. Does that mean our models are outside our domain $\mathcal{D}$? Are they less 'true' because they describe what may or will become true in the future? This is certainly an area for further research.

5.6.2 Feasible Completeness

The Lindland paper [56] has the concept of 'feasible completeness', recognizing that it may not be worth the effort completing 100% of the model. The incorporation of this concept would affect our target of 1.0 for our completeness metric.

5.6.3 Coherence Metrics

There are other ways that one could measure the accuracy of the content of a set of architecture models, apart from just correspondence with the domain being modelled. For example, one could seek to compare the content of different models and see if they contradict each other. There are also additional techniques, available in particular where a modelling tool is being used to provide a formal architecture repository. For example, do two models that purport to refer to the same entity, actually have the same underlying referent in the modelling tool – see section 8.5.2 below – or are they in fact referring to two independent referents, so they do not in fact refer to the same thing, although they should? This would again be a fruitful area for future research.

5.6.4 Automation of Quality Measurement

We deal, in Chapter 8, with changes to Enterprise Architecture models for various reasons, including the determinism of some of the transformations involved (section 8.4), and the potential to automate those change through the use of Enterprise Architecture Modelling (EAM) tools (section 8.6). Leaving aside the 'truthfulness' metric Q_T, the same kind of automation could be used to automatically calculate the quality of information held in EAM tools. To demonstrate this, we can take two practical examples, one related to the syntactical metric and one to the completeness metric.

For completeness, we might decide that a complete set of views would include a contextual view for every application in our official application catalogue, that showed all the other applications that relate to it in some way. The pseudo-code to implement such a check would look something like this:

Table 20 Pseudo-code to check for completeness of views

```
Set expected view count to 0
Set found view count to 0
For each element in the application catalogue
    Increment expected view count
    For each view based upon that element
        If the view is the contextual one we were seeking
            Increment the found view count
Set metric value to found view count / expected view count
```

Such algorithms are well within the scope of the analytic capabilities of EAM tools such as BiZZdesign and others.

For syntactical quality, we might expect that tools that enforce a particular language might force our models to immediately completely comply with our language. Some more careful thought might reveal the inadequacies of such an argument; here are two examples of why this is too simplistic, both of which represent the actual view of an organisation using ArchiMate for Enterprise Architecture modelling.

Firstly, the agreed language for our organisation might, in some areas, be *reduced* from that of the official standard. One example of this would be the use of ArchiMate "junctions" such as those described in section 5.5.1 of the 3.1 standard:

Figure 56 ArchiMate Junctions

Our previous employer tried strongly to discourage the use of these, on the basis that they made the job of automated analysis of the diagrams much more complex, thus they were deemed to be, in our terms, 'outside the allowable language'.

Secondly, our agreed language might be in some areas be *increased* from that of the official standard. One examples of this would be our current organisation where we felt that we wanted to allow a composition relation from an application component to a technology collaboration, even though this was forbidden in the ArchiMate standard. We therefore customised our EAM tool specifically to allow for this combination.

These two examples illustrate that the agreed language for an organisation may not necessarily be an official standard, even if our EAM tool happens to implement that standard.

Automating the measurement of these quality metrics would not only save time, it would also make it more feasible to demonstrate the progress of an EAM initiative by capturing and graphing this in some kind of dashboard to demonstrate, over a period of time, that the models are getting better. We have not demonstrated this, but it is certainly feasible, given the capabilities of such tools.

Chapter 6. Theoretical Approach: Measuring the Effectiveness of an EA Modelling Language

6.1 Research Question

The research question being addressed here (RQ2) is *How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?*

This is less about quality and more about how well different parts of our modelling language communicate with others and so for this we do not need the model quality theory used to address the previous research question. The key concept used here is the semiotic differentiation between signs and referents. We recognise that the entities in our modelling language refer to concepts, some of which will be familiar to non-technical stakeholders, and some of which will not be. This is one factor likely to have a bearing on how well such stakeholders understand EA models: if the concepts being used are meaningless to the readers of the models – if there is no underlying referent in the minds of the readers – then the readers will not understand the purpose of these entities. Recall that by 'understanding' (see section 2.3.16) we are looking for the core beliefs about a concept being correct – so the reader knowing the core beliefs (within the language definition) about, for example, a TOGAF Logical Application Component or an ArchiMate Application Collaboration. Another factor we also wish to consider is whether the reader finds a particular concept helpful. It is conceivable that a concept could be understandable but not helpful, or vice versa. This is the thinking behind our 'effectiveness grid' (see Figure 57 on page 211).

In this theoretical approach, we will construct a mathematical framework that will yield metrics for both aspects of effectiveness – comprehension and usefulness (or 'utility').

6.2 Approach and Reasoning

In our introduction, we discussed a very common problem whereby Enterprise Architecture models are frequently drawn (at least) twice: one in a form that fellow architects should understand, and once in a form that other stakeholders (for example, non-technical business managers) should understand. To answer the question about modelling the effectiveness of our languages, we might ask supplementary questions such as:

6.2.1 Are the underlying concepts understandable to our audience?

Is it possible that some of the concepts (part of our Enterprise Architecture framework) are not intelligible to some of our stakeholders? Some of the concepts (e.g. a Logical Data Component from TOGAF [209], or an Application Collaboration from ArchiMate [15]), might be too specialist for our audience. It is interesting to note that the latter ArchiMate concept, as discussed in 7.4.1 below, does not feature in the ArchiMate case study material, which may be because it is not as well understood or immediately useful as some of the other concepts.

6.2.2 Are the underlying concepts useful to our audience?

Do our stakeholders find the concepts, as presented on the models shown to them, useful, whether or not they really understand them? We have previously discussed the idea of 'expressive economy'; if we used fewer 'boxes' on our models, would that make things easier to understand? If so, which types of 'boxes' should we remove?

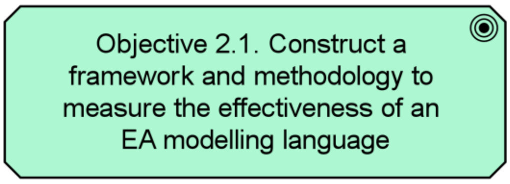
6.2.3 Is our visual notation helping or hindering the understanding of the models?

Some architecture frameworks (notably ArchiMate) include a suggested graphical notation, that indicates (to those that know the language) the type of each element on the model. Do our stakeholders find this notation helpful?

These are the questions that have shaped our research in this area.

6.3 Theoretical Approach

We start by constructing a theoretical and mathematical approach to the problem, to address the objective:



6.3.1 Two Dimensions of Effectiveness

We could perhaps have used the terms "effective" and "efficient" because we are not just concerned with "getting the message through"; we also want to do it in as simple a way as possible, previously described as "expressive economy". Therefore, for the elements of our language, as discussed above, we wish to ask two questions of each element.

Firstly, do the consumers of our models, that include this kind of element, actually understand what this type of element is?

Secondly, do they find it useful, having it included on the model?

These questions bear further explanation:

By *understand* (or comprehend), we are using Kvanvig's moderate version of objectual understanding's factivity (see discussion in 2.3.16 and also 7.6.4 below). In summary, by understanding, we mean, do the respondent's beliefs about the underlying concept match the official (ArchiMate) definition in its core aspects, in the view of the interviewer?

By *useful*, we are asking the respondents to judge whether the model would achieve its stated purpose (part of the test material) better, worse, or the same, were the concepts omitted from the test diagrams. In other words, would the diagram be improved by removing them?

The answers to these questions and should influence our decisions to include, or not include them, in models that we produce for our stakeholders.

Why are these questions important? It is our belief that if a stakeholder does not understand what is meant by a particular set of items on a model (diagram), then there is limited value in including it in the diagram. It might be that they cannot articulate the purpose, but still feel the instances on the diagram have value, in which case maybe they should be left in there (top left quadrant in the above grid); and vice versa also. Certainly, if the reader of the model does not understand the items, and does not think the model would be lessened in value by removing them, then that would give us a clear indication that we should not include them.

Thus, we could bear these concepts of comprehension and usefulness (as pertaining to a particular set of stakeholders) in mind when we come to determine how to construct the models intended for their use, for example using a grid like the following:

Figure 57 Effectiveness Grid

Useful?		
Yes	Maybe	Yes
No	No	Maybe
	No	Yes
	Understandable?	

In other words, if a concept is both understandable and useful to a stakeholder, then use it; if it is neither, then don't use it; and if it is just one, they maybe use it.

6.3.2 Understanding Communication

As previous discussed, modelling languages provide us with a taxonomy of terms (for entities – nouns – and also for relations between entities). For each of these terms, there may be a number of instances of them. For example, if one of our entities in our chosen modelling language is "application", then we may have a number of "applications" on any particular model.

Different modelling languages, as contained within various architecture frameworks, have different sets of entities and relations; examples can be seen in A.1.1 below (TOGAF), A.2 (Zachman) and A.3 (ArchiMate).

What does it mean to be effective in communication? Surely this means that the ideas we are trying to communicate to those receiving our message are received correctly. This has to include the interpretation process, referring back to Figure 6 above. So, we are looking for the ideas that we wish to communicate, being correctly interpreted by the receiver. There are of course a number of interpretive processes involved here, so the picture is more complex than the original single semiotic triangle.

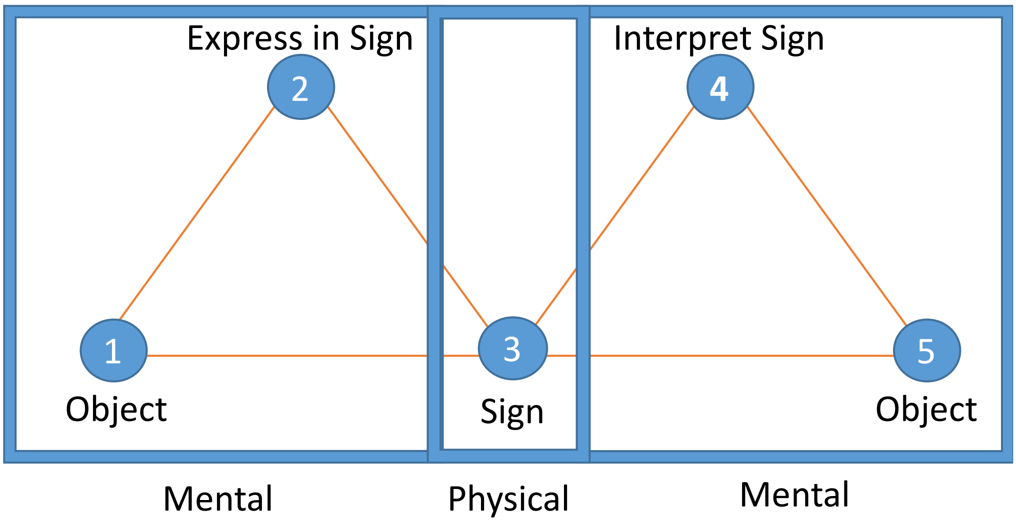
Recall the original idea of a semiotic triangle that we first discussed in section 1.7.2 above, where a sign (representamen) was interpreted as referring to an object. This is of course what happens when we read signs that form part of an EA model (diagram). We need to remember, however, that someone created the model in the first place. Thus, there are two separate processes happening here. In the description below, the numbered circles refer to elements in Figure 58 below.

Firstly, the sender (the person creating the diagram) has ideas in their mind (the objects 1). They search their available vocabulary (again, based upon the training, background and culture 2) for a sign (word, symbol etc.) that they think matches the object (perhaps using signs (notation etc.) taken from an EA framework such as ArchiMate). They then use the sign(s) in a model 3 , in the creation of a physical (real-world) sign that is visible to both the sender and receiver, for example when they use either pen and paper, or a computer, to create the appropriate physical representation of that mental sign.

The receiver, when they see (or hear, or via some other sense) the sign 3 , then will, via some mental process, search their own internal dictionary of signs ("do I recognise this?"), and this interpretation 4 may again be informed by previous experience or training (they might have been trained in ArchiMate!). Hopefully, there will be a match, and they will be able to associate an object, within their thought processes, with that received sign 5.

Thus, we can see that there are two processes: the expression of an object as a sign (perhaps in a model), and then the translation of that back into an object. There are two semiotic triangles here, used in opposite directions:

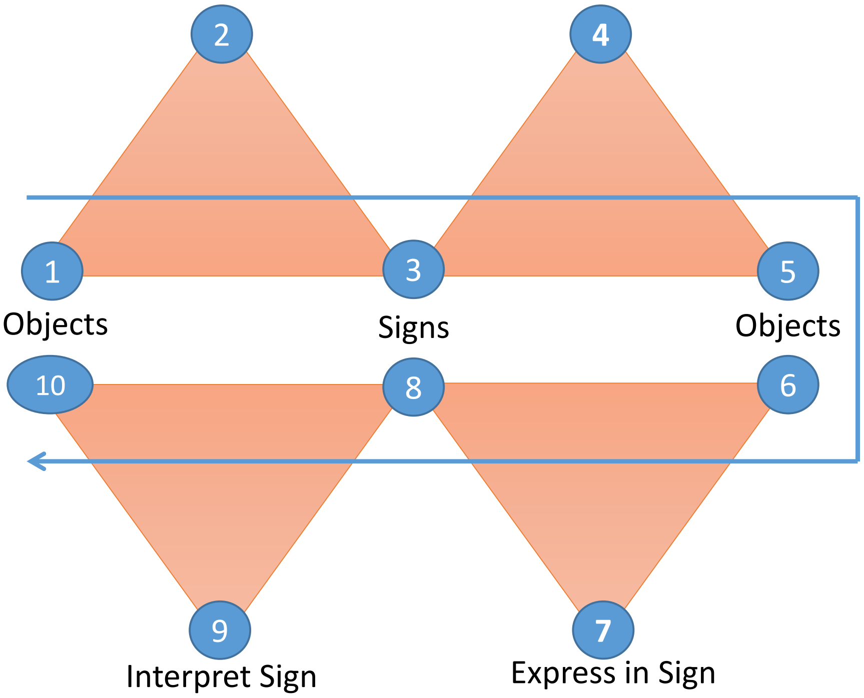
Figure 58 Semiosis for Sender and Receiver



The only part of this process accessible to any third party is the sign 3; the objects that we have in mind (both the sender and receiver), and the process to convert to and from a sign are mental, not visible to the outside world.

How do we then check that it has been correctly understood? We can ask the receiver to tell us what their understanding is of what they were given. Clearly, we need to ensure that they don't simply repeat the same sign that they were given. So, we are looking for some kind of 'round trip' to validate understanding of the information that was sent.

Figure 59 Semiosis for Round Trip Communication



We would then hope that the sign returned from the recipient when they "say it in their own words" matches a concept in initial sender's mind that they can tell corresponds to their original idea. So in fact there are four separate semiotic processes at work here: items 6 to 10 in Figure 59 work the same way (but in the reverse direction) as items 1 to 5 in Figure 58.

6.3.3 Communicating Concepts

In our case, we are not trying to test the understanding of a specific word that has been spoken. We are trying to test the understanding of a concept, without directly referring to it. We are trying to understand whether the concept of, for example, an "application interface" or "business interaction" is effective when communicating with people. Rather than say "what do you understand by an application interface", we are trying to be more subtle and give examples of these, to see if the underlying concept already exists in the mind of the consumer of our message (our models).

An analogy may make this clearer. Imagine that the conversation between an interviewer and interviewee went like this:

Table 21 Imaginary conversation (1)

Interviewer	Interviewee
I am going to **tell** you examples of something and would like you tell me what they are examples of.	
"Lion"	
"Unicorn"	
"Whale"	
	They are all animals

How did this work? The interviewee received a number of signs that they recognised individually, because they already knew the words "lion", "unicorn" and "whale" and knew that they referred to specific objects that they knew about (perhaps from reading a book or watching television). Moreover, from the background knowledge of the interviewee, they would be able to infer that they all had something in common (the fact that they are all animals), and thus, they would be able to express back to the interviewer that fact. Of course, the background and experience of the interviewees may also provide additional meaning to the signs. Someone involved in creating or funding new business ventures will understand something completely different about the term unicorn, being a start-up business with a valuation in excess of $1BN [210].

It should be noted that the signs used here are completely independent of the concepts and referents. The interviewer could just have easily used pictorial signs rather than verbal ones:

Table 22 Imaginary conversation (2)

Interviewer	Interviewee
I am going to **show** you examples of something and would like you tell me what they are examples of.	

	They are all animals

There is of course a semiotic process involved in the transmission of the message "they are all animals" back to the interviewer – steps 6 to 10 in Figure 59 - and this 'quiz' will be successful as long as the interviewer has enough background information to know that this is a true statement about the commonality involved.

We are probing here to see if there is an underlying concept that the recipient (consumer) of the 'message' can deduce from the signs sent. For certain, consumers could be taught the underlying concepts however, we are interested in how effective language is at communicating with (in context) non-technical stakeholders, and so we specifically do not want to teach the consumers about the concepts. We want to see if they already understand the concepts. So, for example, in the concept of

[17] Image courtesy of Kevin Pluck, fair academic use

[18] Image courtesy of entrepreneur.com, fair academic use

[19] Image courtesy of World Wildlife Fund fair academic use

Enterprise Architecture, if we gave examples of, say, a business process, then would the consumer be able to recognise the underlying concept of a business process?

There is of course a complexity here in that we are asking the respondent to express back to the originator (interviewer in our surveys) the underlying concept and the respondent may not know the phrase (from our entity metamodel) that we are looking for. Thus, there will need to be a certain amount of clarification between the interviewer and interviewees to check what they mean by their answer to “what are all these examples of”, in an attempt to gauge how close their idea of the underlying concept is to what we were trying to express.

We should recognise at this stage that the theoretical discussion so far is based upon a mainly positivist stance; and that other approaches to language, and in particular modelling languages, have been proposed, that do not make the assumption that the referent exists objectively and independently of the various signs that may be used to refer to it (see for example the discussion of Bjeković’s work in section 2.3.20 above related to socio-pragmatic constructivism).

6.3.4 Application to a Modelling Language

Building on the previous discussion, we want to understand how comprehensible certain concepts, defined in our chosen architecture modelling language, are to the stakeholders that consume our architectural models. To illustrate how this can be done, we are going to use two modelling languages. We will start with a much simpler one – the fictional one laid out in Figure 3 above, as part of the introduction.

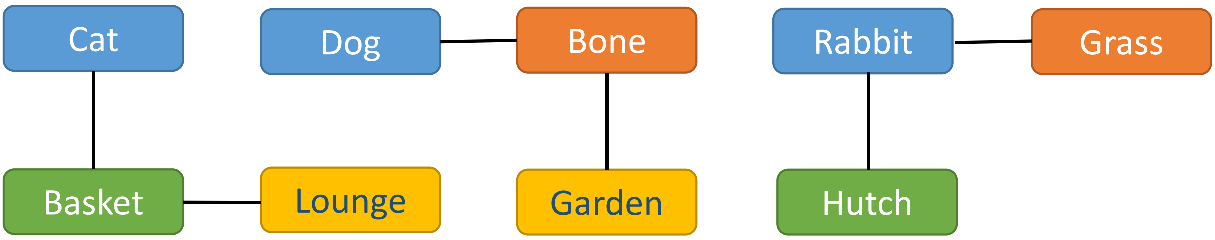
Imagine that we were trying to describe the activities and location of some animals in our household. We could do this in graphical form, limiting ourselves to our fictional metamodel. Recall from section 2.2.2.1 above that our metamodel provides our allowable vocabulary for communication. If we want to ensure that our model complies with the specified language (see for example section 4.3.1 above) then we will ensure that we only use words (concepts) from that metamodel – that is, anything that we try and represent in a model is an example of one of those concepts.

It should be noted that we are limiting the set of signs that we can use when modelling, by adopting a metamodel. For example, if our vocabulary doesn’t provide for the concept of a vehicle (doesn’t appear in Figure 3 above), then we will not be able to express ‘lorry’, ‘van’ or ‘car’ in our modelling language. Thus, our choice of modelling language(s) is very important when it comes to determining what kind of models we want to produce. For example, if we want to produce detailed process models, and our particular entity metamodel does not provide for the relevant concepts, then we may need to extend our set of metamodels to allow this (see discussion related to this in section 4.3.7 above). This is also the reason why architecture frameworks (including entity metamodels) are often tailored by organisations (as specifically provided for in the Preliminary Phase of the TOGAF Architecture Development Method [32]), also in the Language Customisation section of the ArchiMate specification [211]: in order to allow modellers to express concepts that are different to those originally contained within the language(s).

Thus, looking back to our discussion of the semiotic process, and to Figure 58 and Figure 59 above, we will only be able to communicate using signs that are examples of the concepts in our entity metamodel. This might make it easier to 'learn' a language, as the vocabulary is more limited, however it limits the scope of what we are able to express.

We now present a fictional graphical model expressing what our pets are doing; which complies with our fictional 'pet metamodel' given in Figure 3 above:

Figure 60 Modelling Pet Activities and Locations



We could also do this in narrative form, for example "our cat is on the basket in the lounge; the dog is eating the bone that's in the garden; and the rabbit is in the hutch eating grass." Now suppose we wanted to test our reader's understanding of the concepts here. We might ask them "what do you think is represented by the boxes in blue?" Hopefully they will come back and say "pets"; they might also say "animals", or "household animals". This is because they recognise that all the items in blue have something in common: they are animals that belong to our household. We can also see here where the concept of objective factivity (see section 2.3.16 above) comes into play in our theoretical example: do we consider that the core to the concept we are trying to communicate is the idea that our animal is part of a household (i.e. is specifically a pet)? If a respondent says that the examples are all animals, the statement is obviously true, however we need to determine have we got all the 'core beliefs' about a pet in that statement, or not.

Similarly, all the items in orange represent food, the items in yellow are places and the items in green are objects (things). This testing for understanding of the commonality of certain objects, when testing alongside a narrative, is *precisely* what we will be doing with an EA modelling language.

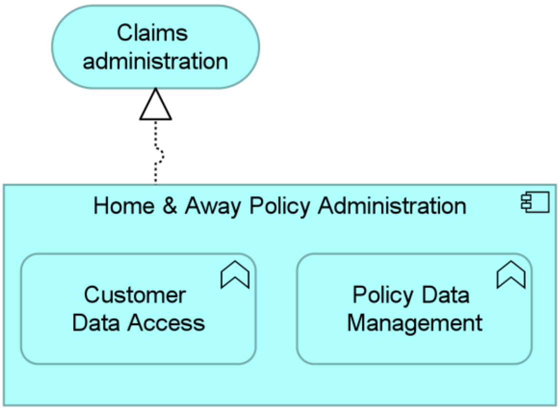
We chose one particular EA modelling language, ArchiMate, for a number of reasons:

- We know it well, having used it intensively for several years as a practicing Enterprise Architect;
- It has been widely adopted; although it is hard to find official figures on this (as it can be adopted with no reference to the Open Group, being an open standard), the LinkedIn group for ArchiMate has, at the time of writing, 11,355 members [212].
- It features heavily in the literature over recent years;
- Unlike most EA frameworks, ArchiMate has its own graphical notation;
- We know anecdotally from conversations with colleagues that it is not always accessible to business users, and as such is of interest to us, to see if we can quantify that difficulty in some way.

Therefore, in the following discussion, we will use concepts that are contained within the ArchiMate entity metamodel.

We will show our stakeholders models that contain examples (instances) of such concepts, in context, to see if the underlying concept is recognised. An actor may of course gain meaning about the underlying concept from a number of sources. Consider the following fragment of an ArchiMate model:

Figure 61 Fragment of application Model (ArchiMate notation)



When we are asking for the meaning underlying the concepts that appear above, for example the ones labelled Customer Data Access and Policy Data Management, there are a number of classes of symbols (or signs) as to their referent:

(a) Their *colour*: anyone who is familiar with this particular modelling language knows that blue colouring is used to refer to concepts in the application layer;

(b) Their *shape* (a rounded rectangle): this is probably not very useful as there are a number of elements in the ArchiMate language that all have the same shape;

(c) Their *icon* (the small drawing at the top right – in this case a vertical chevron); anyone knowing ArchiMate will know that this always refers to a kind of 'function' (for example, an application function);

(d) The *descriptive names* ('node descriptor') assigned to the element (e.g. Customer Data Access)

The comprehension of the concept underlying these may come from their personal background and experience, or alternatively it might come from a visual interpretation linked to previous exposure to the ArchiMate language (or perhaps an alternative language with a different, possibly conflicting graphical notation such as UML [39] or BPMN [205]).

(e) We might illustrate this as:

(f) Experience + Colour + Shape + Icon + Node Descriptor = Meaning

6.3.4.1 Graphical Notation – Help or Hindrance

With any model that we show to our stakeholders, if we want to give them some indication that different elements on the model represent different kinds of concepts (things), then we need to use some technique to highlight these different kinds of concepts (such as described in the discussion on Gestalt theory in section 2.3.14 above). If our chosen modelling language happens to have a specified graphical notation (and ArchiMate does), then this serves as an obvious visual key to enable our stakeholders to distinguish visually the different kinds of concepts on the model(s).

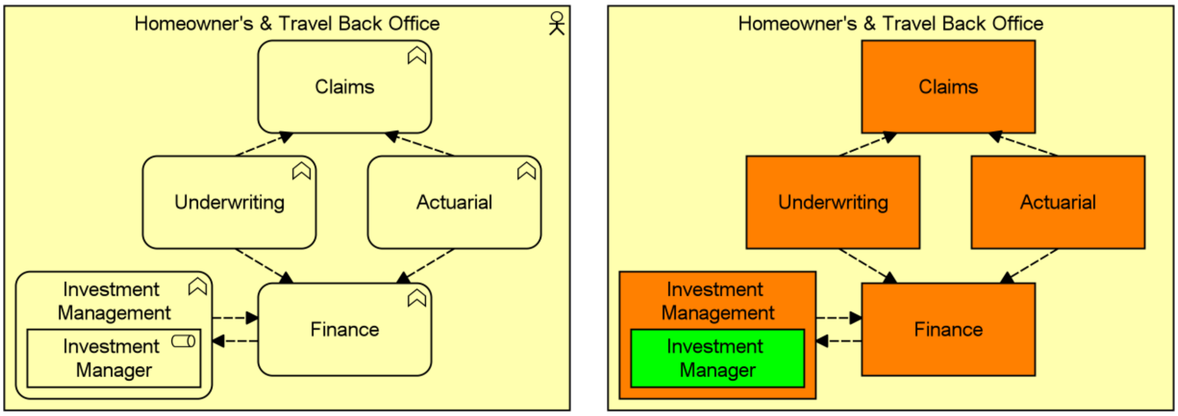
In order to determine whether or not the suggested graphical notation is helpful, then as a control measure, we could produce alternative models with the same essential content but different notations (for example, using a different colour for each entity type, rather than the same colour for all entities in the same 'layer' of the framework, which is what ArchiMate provides). This could be considered 'secondary notation' in the language used in the literature previously discussed on visual programming environments (section 2.3.7 above); although in our particular case, we will experiment with replacing ArchiMate notation with colour in particular, rather than supplementing it.

By comparing the results obtained using both sets of notation we can gain some insights as to whether the use of the default notation is actually helpful, in particular when communicating with non-technical stakeholders.

By way of example, here are two fragments of ArchiMate that have the same essential content but just use a different graphical notation:

Figure 62 Alternative Notations for ArchiMate

Default ArchiMate notation **Bespoke notation using colours**



Here, drastic changes have been made to the four classes of signs:

(a) The *colours* are now different for each underlying class; this is necessary as for these models, it is now the only visually distinguishing feature. The previous ArchiMate standard colour has been removed;

(b) The *shapes* are now identical for each symbol type;

(c) The *icons* have been removed;

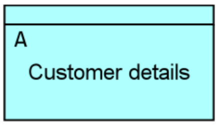
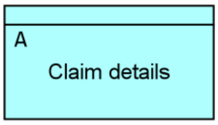
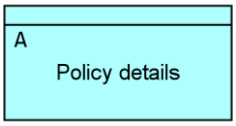
(d) The *descriptive names* have been left unchanged.

If the colour, shapes or icons have had a major impact on the comprehension of the concepts then we would expect to see a change in comprehension scores with the ArchiMate views as compared to the neutral views. If, however there is no marked difference, then this would indicate that these three elements of themselves are not providing extra meaning. Thus, these 'neutral' or 'non-ArchiMate' views provide the equivalent of a control group.

In other words, if by removing Colour, Shape and Icon from our illustrative equation above, we don't alter the meaning conveyed to the interviewees, then we can infer logically that these three classes of symbol were not contributing to the comprehension process (either positively or negatively).

Thus, we extend the imaginary conversations shown above to consider examples like the following. Note that in order to maximise the opportunity for the information to be understood, in the correct context, by the recipient, we are giving the information in both written and pictorial form.

Table 23 Imaginary conversation (3)

Interviewer	Interviewee
I am going to **describe** and **show** you examples of something and would like you tell me what they are examples of.	
The policy administration system has to manage both access to detailed customer data as well as information on their claims and policies.	
	
	
	
	They are all types of information

The blue boxes in Table 23 above are not shown in isolation; they are shown in the context of a larger diagram. The person reading the diagram (model) is asked to distinguish these particular boxes by their shape. This particular shape has a particular meaning in ArchiMate, although this correlation is not known to those interviewed as part of the survey used to test this. Of course, for the interviewers where I used neutral (non-ArchiMate) notation, these three entities looked completely different.

Thus, using Figure 59 again, the steps are:

Table 24 Semiotic Process for Testing Comprehension of ArchiMate Concept

Step	Interviewer	Interviewee
1	I, the researcher, have a concept in mind (which I actually got from the	

Step	Interviewer	Interviewee
	ArchiMate framework, with its definitions); in this example, I am thinking of Data Objects	
2	I am looking for examples of that to use in my test models, and find some in existing ArchiSurance material; some I make up myself.	
3	I create the test models and accompanying narrative and make them available to the interviewee	I read the narrative description and test model(s)
4		I try and make sense of the information I've been given, using my previous experience.
5		What they all have in common is that they are describing some kind of information
6		I think about how to explain this to the interviewer
7		I decide that the phrase 'some kind of information' is the best way of describing what I think they are
8	I hear the words spoken by the interviewee	I speak these words to the interviewer: "They are all some kind of information"
9	I understand what they mean by this	

Step	Interviewer	Interviewee
10	I do a comparison of what I had in mind originally, using the ArchiMate definition, and what they said, and make a judgement of how close these two things are	

6.3.4.2 Pieces of a Model

Architecture modelling languages work with the components comprising a system as well as relationships between them and so both need to be represented in the set of symbols used in our language.

For example, the ArchiMate 3.1 [15] language specifies a large number of symbols (icons, shapes and colours) to denote the various language concepts, as illustrated in A.3 below. The symbols describe both a set of entities or concepts (e.g. business services, application components) and also a set of allowable relationships between entities (e.g. composition, aggregation).

As well as the explicit symbols, we also need to bear in mind the implicit messages (e.g. arising from spatial relationships) suggested from the general layout of the symbols, for example arising from the Gestalt "Gestalt Laws of Perceptual Organisation" [117]. These laws show how we experience larger scale constructs by combining simpler constructs according to certain laws or factors. For example, showing several elements in a similar visual style (e.g. colour) or location suggests a grouping of those elements, directly analogous to the 'group' concept in ArchiMate.

Thus, the elements in our model are both explicit and implicit.

Alongside the explicit and implicit symbols, we also consider models to have a *purpose* (what are they trying to convey) [36], which in an architecture context will depend on the role and specific concerns (interests) of the reader. The graphical symbols may well include several visual elements (including shape, colour, icons, line types and endings), all of which have a particular significance in ArchiMate [39], as an example.

The purpose is useful in terms of giving context to the question: "how useful is a particular concept on the model?". We can phrase it more precisely as "if the concept were omitted from the model, would the model better fulfil the purpose (or the opposite)?"

6.4 Mathematical Model

6.4.1 Definition of Model Content

6.4.1.1 Base Notation

Consider a set of models **M**, let us say D of them, and denote them as $m^1, m^2, m^3, \ldots, m^D$.

Consider also a set of actors (our stakeholders) **A**, let us say there are P of these,

denoted $a^1, a^2, a^3, \ldots a^P$.

Put more concisely,

a set of models **M** with D members m^d $(1 \leq d \leq D)$
a set of actors **A** with P members a^p $(1 \leq p \leq P)$

We are interested here in the explicit symbols on a model (diagram). The explicit symbols on an architecture diagram will typically consist of **nodes** (often, but not always, closed shapes) **n** (denoting elements – the nouns in our language) joined with **edges** (normally lines with possible extra shapes at either end) **e** (denoting relations between the elements), forming a directed graph. There are other implicit symbols (or signs), for example relative position on the page, colour etc., as described in Gestalt theory, however for the purpose of our analysis we are just focusing on nodes, and on edges to give context to the nodes. In an ArchiMate context, for example, the nodes might represent applications and services provided by the applications; and the edges might represent the relations between the two (indicating which applications provide which services).

We refer to these nodes and edges using the following notation:

$n_x^d \in N^d$	describes the set of nodes on model M^d
$e_y^d \in E^d$	describes the set of edges on model M^d

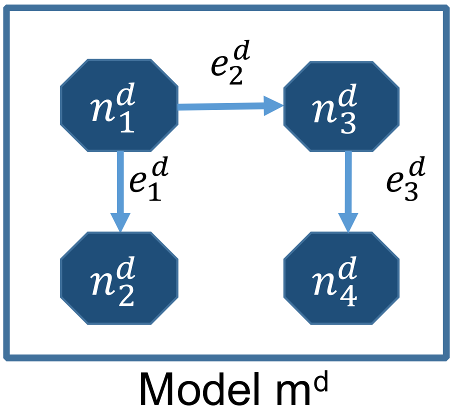
For example, the second node on the third model would be denoted n_2^3.

If we combine both of these we get our overall set of symbols on the models, denoted as S^d:

$s_z^d \in S^d = N^d \cup E^d$	the set of all (explicit) symbols on m^d

Here, x, y and z are just index variables that denote the range of the set of nodes and edges. To illustrate this notation, consider the following diagram (or model) consisting of four nodes connected by three edges:

Figure 63 Models, Nodes and Edges



For an example of what this scheme looks like when applied specifically to ArchiMate models, please see Figure 98 below on page 302.

We refer to the number of symbols on a particular model m^d by L^d where *L* can be taken to be the length of the set of symbols on this particular model; we can break this down into $L^d = L_n^d + L_e^d$, where L_n^d is the number of nodes on model m^d and L_e^d the number of edges. Thus, we have, for any one model m^d:

$L^d = | S^d |$ $\quad L_n^d = | N^d |$ $\quad L_e^d = | E^d |$

So, for example, if our 4th model contained 5 nodes and 6 edges, then we would have:

$L^4 = 11$ $\quad L_n^4 = 5$ $\quad L_e^4 = 6$

6.4.1.2 Mapping Functions

EA modelling languages like ArchiMate have a specific set of entities (e.g. application functions, business processes) and relations (e.g. realises) already defined and so we would expect each of the symbols on our model to map to one of these predefined concepts (that is, if the model actually complies to the language, as discussed in the previous chapter); the nodes should each be mapped to a particular entity type, and the edges to a particular relation type. We thus have mapping functions, which we denote with a lowercase t, with the *domain* being the set of all elements and relations on our models, and the *range* being the set of element and relation types defined by our particular choice of language. Thus:

$\forall\, n_x^d \;\exists\, t_N :\; n_x^d \rightarrow T_{EL}$, where T_{EL} is the set of element types;

$\forall\, e_y^d \;\exists\, t_E :\; e_y^d \rightarrow T_{RE}$, where T_{RE} is the set of relationship types; or combining these;

$\forall\, s_z^d \;\exists\, t_S :\; s_z^d \rightarrow T_{CO}$, where T_{CO} is the set of concept (element + relationship) types.

Obviously, if there is a symbol on a model for which no such mapping exists, then the model is non-compliant with the language (outside region $\mathcal{L}$, regarding our previous definition of model quality).

6.4.1.3 Notation for Concepts

If we define I_c^d to mean the set of all instances of a particular concept (element or relation) **c** on model m^d, then we have:

$$I_c^d = \{ \mathrm{s}_\mathrm{z}^\mathrm{d} \in S^d : t_S(\mathrm{s}_\mathrm{z}^\mathrm{d}) = c \} \qquad (1)$$

We are here using the notation of a function with t_S. This says that we take all explicit symbols $\mathrm{s}_\mathrm{z}^\mathrm{d}$ on model m^d (which is what S^d means) as long as they satisfy the condition that when mapped to an underlying concept **c**, that is the concept we're interested in.

Now it is entirely feasible for any particular concept (for example, an ArchiMate Business Service) to appear on more than one of our models. If we sum this over all the models, then we obtain the set of all instances of a particular concept **c** on our set of D models:

$$I_c = \bigcup_{d=1}^{D} I_c^d = I_c^1 \cup I_c^2 \cup I_c^3 \ldots I_c^D \qquad (2)$$

The length of this set of concept instances across all models is therefore:

$$L_c = \sum_{d=1}^{D} | I_c^d | = | I_c | \qquad (3)$$

This is the total number of occurrences of a particular concept (element or relation) across all models in our set.

6.4.2 Measuring Comprehension

Having established the overall approach above, we are then able to ask a particular actor a^p, for each type of element (concept c) on a model m^d, do they understand what is meant by symbols of the type $\mathrm{s}_\mathrm{z}^\mathrm{d}$? More specifically, do they understand the meaning behind this particular kind of symbol?

This is of course a subjective question and so will require triangulation (see section 3.1.8.1.9) to ensure that we catch 'false positives': people who claim to understand which concept is being represented when they do not in fact understand it. We are less concerned about 'false negatives', as people are less likely to say that they do not understand a concept when they in fact do.

Enterprise Architecture diagrams do not purely contain graphical symbols: the elements (and sometimes relations) are given descriptive names to indicate the specific instance of a concept (for example, the name of a particular business process or capability.

In the survey used to test this, in our questionnaires we focused only on the elements (nodes), and ignored the relationships (edges), because the elements represented the core of the language and (in our view) the relationships have no real meaning outside of the elements that they connect (for example, the term 'booking', as a business process element, has some meaning in its own right, whereas the relational term 'comprises' has no inherent meaning unless attached to something else). Thus, in the subsequent analysis, although we will still refer to concepts, they can be understood specifically to refer to elements, rather than relations.

We do not do this purely by asking the interviewees to look at a diagram; we also give them a printed narrative, that describes the content of the test model(s) that they are given. This narrative refers to each of the elements by something close to their descriptive name (that is also on the diagram), and gives some context to their use, such that anyone already familiar with such constructs (the underlying concepts) would recognise them, as per Table 23 above.

There is a close correlation, hopefully close enough for the respondents to recognize, between the words in the narrative and words on the actual nodes in the diagram, as in the following further example:

Table 25 Correspondence between Narrative and Descriptive Text in Diagram

Narrative Fragment	Node with Text
policy administration system	Home & Away Policy Administration
detailed customer data	Customer Data Access
information on their claims and policies	Policy Data Management

It can be seen that there are nodes on the diagram corresponding to the policy administration system as well as two things that the system does.

So, for model m^d, which contains a number of different element types (denoted t_c^d), we have:

$$\forall\, n_x^d\ \exists\, t_N :\ n_x^d \rightarrow t_c^d \qquad \text{where } t_c^d \in T_{EL} \qquad (4)$$

In other words, for every node $\mathrm{n_x^d}$ on model m^d, we are able to map it to a specific type $\mathrm{t_c^d}$ which is one of the terms in our vocabulary of elements T_{EL}.

Our question therefore relates to each type of element $\mathrm{t_c^d}$ on model m^d, as read by the specific actor a^p. We ask the actor a^p if they understand the purpose or meaning of the kind of elements with a given appearance (to enable them to distinguish items of the same type visually), to help us derive a score to assign to our comprehensibility score denoted by $\mathrm{h_c^{dp}}$. We offer the actor (interviewee) a choice of levels of understanding to choose from:

- No clear understanding (score 0)
- A partial understanding of how the items relate (score 0.3)
- Can articulate the commonality but are not familiar with the concept (score 0.7)
- Can articulate the commonality and are familiar with it (score 1) (*Q07, Q10, Q13*)

In this discussion, the question numbers in red indicate which of the questions in the questionnaire (reproduced in Appendix G below) are used to capture this specific piece of data.

If the actor claims to understand the concept in part or in whole (i.e. selects any option apart from the first one), then a further question is asked to validate this. The actor is asked to try and characterise this concept, to describe the purpose of the elements of this type; what they have in common. This is obviously more complex than a binary answer and we search in their narrative response for synonyms or key phrases that correlate to the correct underlying concept. The actor is not told if they have it 'right' at the time, as this risks frustrating the user or making them feel that they have made a mistake. If their response indicates that they do understand the concept, in part or whole, then we assign an appropriate score to $\mathrm{h_c^{dp}}$: 0.3 if there is a weak understanding, 0.7 for a stronger understanding, and 1.0 when it is clear that the concept has been understood.

We wish to weight the scores by the number of instances of a particular concept on each model, which we denote:

$$\mathrm{w_c^{dp}} = |\ I_c^d\ | \qquad (5)$$

This is so that we may more attention to models that contain many instances of a particular concept than to models that contain few. In practice, for the tests models used in this particular research, each concept only appeared on a single model, and so the weighting factors were not required when calculating the results; however we feel that if further, more comprehensive research was carried out in which a particular concept appeared on more than one model, then this weighting factor will be relevant.

For any models m^d that do not contain concept t_c, we will of course obtain a weight of zero for that concept, thus:

$$w_c^{dp} = 0 \text{ if } I_c^d = \emptyset \qquad (6)$$

This is repeated for each model m^d, and for each actor a^p.

We can see that for any concept t_c that the highest possible comprehension score for actor a^p would be the sum of all the weighting values, which we denote thus:

$$H_{max}^{cp} = \sum_{d=1}^{D} w_c^{dp} \qquad (7)$$

The actual score will be the sum of the responses, each multiplied by the weight for that model, and so is given by:

$$H_{actual}^{cp} = \sum_{d=1}^{D} h_c^{dp} \cdot w_c^{dp} \qquad (8)$$

In order to normalise the 'score' for this concept, for this actor, we then divide the actual score by the maximum possible score, to arrive at a normalised rating for this particular concept by this particular actor, in the range [0, 1]:

$$H_c^p = \frac{H_{actual}^{cp}}{H_{max}^{cp}} \qquad (9)$$

We can sum this over the comprehension scores from all actors a^p, and then divide by the number of actors *P*, in order to get an overall average figure for the comprehension of a particular type of concept t_c; this would be:

$$H_c = \frac{1}{P} \sum_{p=1}^{P} H_c^p \qquad (10)$$

Scores close to 1 would indicate concept types that are understandable and familiar to most users interviewed; scores close to 0 would indicate concept types that are not understandable to most users interviewed. Of particular interest, however, would be correlations between particular types of actors a^p and particular concepts t_c; as we wish to investigate whether particular kinds of actors find certain concepts more comprehensible than other actors. This understanding may come about through two kinds of analysis. Quantitative techniques may give correlation; however, the additional comments captured (as to WHY the concept was understood) will yield more immediately useful knowledge.

6.4.3 Measuring Utility

Let us denote the purpose of model m^d (probably expressed as a single sentence) as r^d. The purpose is included in the test material, along with the narrative; it is typically a single sentence in each case.

To understand how useful a given concept type is for an actor, we start by asking the actor if they believe, for each model m^d, whether that model as a whole fulfils the given purpose r^d. (*Q08, Q11, Q14*). This purpose will be given to the actor as part of the interview process, in writing, along with the narrative mentioned previously. If they do not believe that the model fulfils the purpose as currently shown, then no utility scores are recorded against this model for this actor. This is because the methodology we are adopting for the utility of symbols is to examine whether a model is improved or degraded by the removal of certain symbols and if the model starts out in a state (as far as the actor is concerned) where it substantially fails to achieve the stated purpose, then adding or removing specific symbols is unlikely to improve the situation.

So if the actor affirms that the model m^d does fulfil the given purpose r^d, then for each type of concept on the model (again denoted by t_c^d), we ask them to imagine the model without that type of concept included, and ask the question each time: would the extent to which this model fulfils the stated purpose be reduced, unchanged or increased by the omission of the specific concept (element) type (*Q09, Q12, Q15*)? This will yield one of three answers; we score +1 for reduced (losing the concept type meant the model was not as useful), 0 for unchanged (losing the concept type had no effect) and -1 for increased (losing the concept type actually increased the usefulness of the diagram, perhaps by making it easier to understand).

So, for example, if the interviewee states that removal of all elements of a particular type would mean that the model did not fulfil the stated purpose as well, then we would record a score of +1 for that concept.

The mathematics flows much as before; we denote here the score for the utility of a particular concept t_c on model m^d for actor a^p as u_c^{dp} (analogous to h_c^{dp} but considering utility rather than comprehension). Given the three possible responses, we can see that $-1 \leq u_c^{dp} \leq 1$. Applying weighting in just the same way as for comprehension, then we see that the highest possible utility scores for actor a^p for a particular concept t^c would be given by the sum of all the weighting factors taken across all models, which we have already calculated in equation (7) above:

$$U_{max}^{cp} = \sum_{d=1}^{D} w_c^{dp} \tag{11}$$

However, the lowest possible utility score is not 0; it is in fact given by:

$$U_{min}^{cp} = -\sum_{d=1}^{D} w_c^{dp} \tag{12}$$

This would be obtained when an actor a^p says that whenever concept t_c is removed on a diagram, it increases the degree to which the diagram achieves what it was intended to (in other words, it just got in the way of the message).

The actual score would be the sum of the trinary values, each multiplied by the weighting factor, thus:

$$U_{actual}^{cp} = \sum_{d=1}^{D} h_c^{dp} \cdot w_c^{dp} \qquad (13)$$

We normalise this to get an answer for the utility of concept t_c for actor a^p, in the range [-1, 1]:

$$U_c^p = \frac{U_{actual}^{cp}}{U_{max}^{cp}} \qquad (14)$$

We can also average this out across all actors to get an overall average figure for the utility of a particular concept t_c, thus:

$$U_c = \frac{1}{P} \sum_{p=1}^{P} U_c^p \qquad (15)$$

Concepts with a utility value of zero or below can be considered not useful when communicating with stakeholders, and for negative scores, could be considered worse than useless, as they actually hinder communication.

As before, in the actual test models used, each concept only appeared on a single model (albeit multiple times) and so the weighting factors were not required for this particular set of test results.

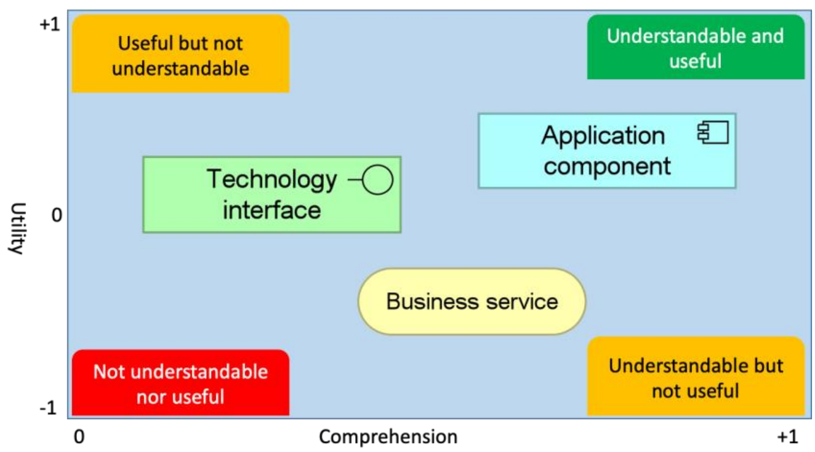
6.4.4 Combining Comprehension and Utility

Each concept in our target language (say, an ArchiMate Driver, part of the Motivation layer) can be considered, when measured in this way, to have an ordered pair of values:

$t_c = (H_c, U_c)$, where $0 \leq H_C \leq 1$ and $-1 \leq U_C \leq 1$.

These are the two scores: H_c is the comprehension score and U_c the utility score. These are averaged over all the respondents. We can visualise these in a two-dimensional bounded space thus:

Figure 64 Conceptual Visualisation Space for Scoring Element Types



We can think of each concept as fitting on such a diagram, with the 'best' concepts sitting in the top-right corner, being both understandable and useful.

Of course, this conceptual view might well be different across our stakeholder groups.

If we could plot each of the concepts on such a grid, then that could guide us as to which items should be included when modelling for particular stakeholders.

6.4.5 Differential Ratings

One simple way of determining the effect of some factor f on our ratings, and indeed on our visualisation spaces for these concepts, is to compare the ratings for the data that included this factor with the ratings for the data that did not. If for a particular concept C we separate out $\mathrm{H_C}$ and $\mathrm{U_C}$ using the notation:

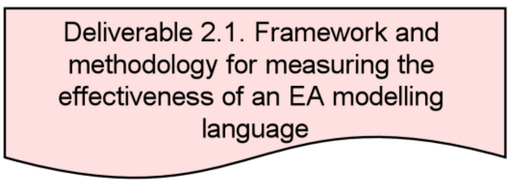
H_C^f	comprehension rating when factor f is present
$H_C^{\neg f}$	comprehension rating when factor f is not present
U_C^f	utility rating when factor f is present
$U_C^{\neg f}$	utility rating when factor f is not present

Then by plotting the points $(H_C^f - H_C^{\neg f}, U_C^f - U_C^{\neg f})$ on our scatter diagrams, we can see visually the effect that this factor is having on our comprehension and utility. Given that it is common practice (as discussed in 1.6.3 above) to redraw diagrams for business stakeholders, in non-ArchiMate notation, an obvious application of this technique is to do a differential analysis to see the impact of using ArchiMate notation, thus plotting on our scatter diagrams the following points, where the A suffix indicates that ArchiMate was used, and the negation of A indicates the opposite:

$$(H_C^A - H_C^{\neg A}, U_C^A - U_C^{\neg A})$$

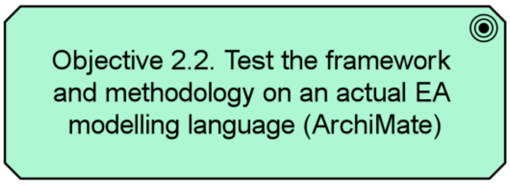
This technique provides a way of answering Bjeković's suggestion for further research into the effect of notation (see discussion in section 2.3.20 above).

This discussion completes the framework and methodology, and thus our deliverable:



Chapter 7. Survey: Measuring the Effectiveness of an EA Modelling Language

Having described the theoretical approach, we now develop a survey that we can use to test this approach. Our intention is to both test that the approach is feasible to implement and also, in the process, see what the results can teach us about one particular modelling language (ArchiMate) thus addressing the objective:



7.1 Application of Quantitative Process

Here we summarise how our research has followed the five-step process outlined in 3.1.2 above.

1. Determining the basic questions to be answered

   The two questions we are trying to answer with this survey are: (a) does the method we have constructed actually work, and (b) what do we learn about one specific language (ArchiMate) by using this method? These are our formal objectives, numbers 2.2 and 2.3 (see 1.10.3 above).
2. Determining the participants in the study

   This was originally going to be drawn from people in one organisation, however later was extended across multiple organisations (see discussion in 7.6.1 below).
3. Selecting the methods needed (variables, measures, overall design)

   The methods have been described in the previous chapter.
4. Selecting analysis tools

   The analysis tools, which consist basically of Excel, using its built-in charting capability, were the obvious choice, given the simple structure of the quantitative data captured. We could have considered other tools had we wanted to automate the processing of some of the qualitative data captured.
5. Understanding and interpreting the results

   This was the final part of our research; this discussion is captured in section 7.7.4 below.

7.2 Context for Survey

Unlike the previous data collection exercise, which was carried out in the course of a consulting engagement, the context for this data collection will be purely academic, for research purposes.

7.3 Constructing the Test Models

7.3.1 Source Material

We are looking to test our theoretical framework on an actual architecture framework (in our case, ArchiMate), and so we need to ensure that the examples that we give, for each concept, are as close as possible to the original intent of the owners of the framework. The way we have done this is to use, wherever possible, examples from official ArchiMate teaching collateral (in particular, the ArchiSurance [62] case study), supplemented with additional examples only where they are entirely absent from the case study.

The scenario examined in the test models is based upon a fictitious insurance company and makes heavy use of the materials provided by the Open Group in teaching ArchiMate. In particular, the models are modified versions and extensions of those provided in the ArchiSurance Case Study [62]. This was done to ensure that we were measuring the concepts as intended by the authors and maintainers of the language.

The following design criteria were used when constructing the set of test models:

(a) The sets of models should cover the complete set of symbols contained within ArchiMate 3.0 (completeness);

(b) The definitions used (which affect how the symbols are used) are intended to adhere as closely as possible to the standard (one reason why the "ArchiSurance" models were chosen as a starting point).

ArchiMate has a wide-ranging scope, broken into a number of layers, for example strategic, business, application and so on. Trying to test all the elements in the language would be too time-consuming, and so a decision was made to focus on three particular layers: Motivation, Business and Application. These were chosen because the first two are more likely to be accessible to people with little IT background, which is part of the motivation for this research. The latter was chosen to provide a more familiar domain to people working directly in IT.

Each of the test models was constructed by forming a composite of all the ArchiMate concepts belonging to a particular layer from the ArchiSurance [62] material. The ArchiSurance material was notably missing some of the concepts from the ArchiMate language and so, where that was the case, examples of these concepts were added to the test models so that each test model contained at least one example of each concept from the layer being studied.

Specifically, the following example concepts were missing from the ArchiSurance sample material and so were added to the composite in order to complete the test models: to the Motivation layer were added examples of Value, Meaning, Assessment, Outcome and Constraint; to the Business layer were added examples of Business Collaboration, Business interface, Business interaction, Business service, Business object, Contract, Representation and Product; and to the Application layer were added examples of Application collaboration, Application Interaction and Application interface. The precise composition and provenance of the test models is discussed in Appendix H below. In summary, we gathered all the fragments of extant ArchiSurance material we could into three views (Motivation, Business and Application), compared the concepts used in these views against the language specification, noted the language concepts that were missing, and then constructed examples of those missing concepts that hopefully made sense in the context of the broader model into which they would be inserted.

Thus, the test models consisted of three views: a complete set of examples of Motivation concepts (elements and relations), a complete set of examples of Business concepts, and a complete set of examples of Application concepts. Each one was accompanied by a single sentence that described the purpose of the model and then a narrative description that tried to say the same thing as the picture, but in words. It was a challenge finding significant practical uses for some of the concepts (which may be why they do not feature in the training material), and so some of the concepts (in particular, the collaborations and interactions that apply in both the Business and Application layers) only had a single example in the test models.

It should be noted that there is a commonality of notation used in ArchiMate, which can help the reader understand new notation, in situations where they have encountered similar notation in a different context. This is the ‘consistency’ sought after in the discussion on visual programming environments, in section 2.3.7 above. For example, leaving aside the very simple example of all application-layer concepts being blue, all technology-layer concepts being green, and so on, there is a commonality and re-use in the ArchiMate icons, as can be seen in the overview of the notation shown in A.3 below (on page 349). For example, the concept of a ‘process’ appears in the business, application and technology layers, and in all cases, the same icon – a block arrow – is used to signify this:

Figure 65 Types of ArchiMate Process

Thus, someone who had already encountered a business process using this notation might well interpret the other symbols as being a type of process. This is useful perhaps for those reading the models with a knowledge of the ArchiMate notation; it has no relevance to those reading models that do not use the notation.

7.3.2 Narrative and Purpose

The narrative description, was not taken from the ArchiSurance material; it was generated by the author, by the simple process of trying to describe, in words, what was contained in the picture, in as natural a way as possible, ensuring that there were specific words or phrases in the narrative that corresponded to each instance of each of the elements appearing on the model. The purpose was also constructed by the author.

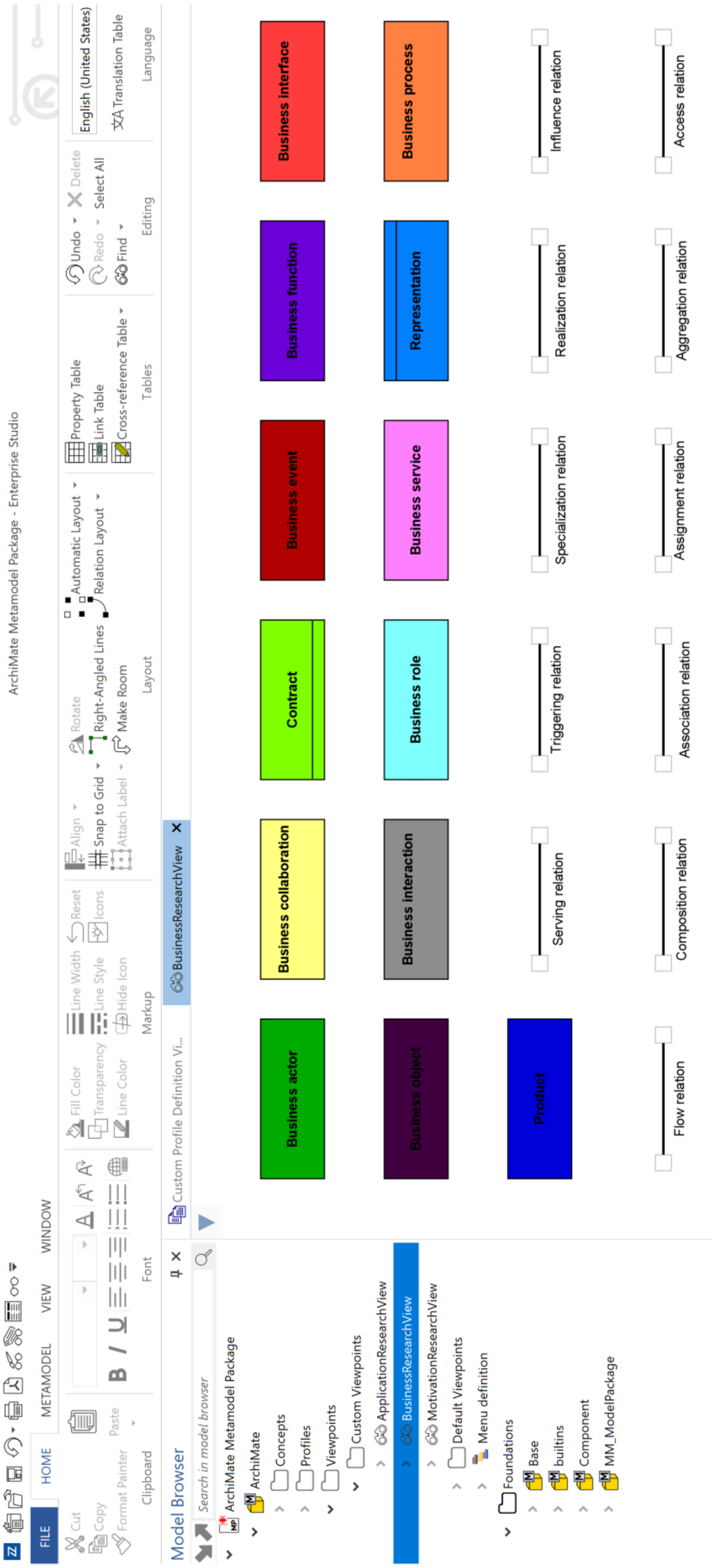
The narrative was necessary in order to provide another set of 'signs' to allow the respondents to have another opportunity to understand what the model was trying to say. In a way this was a form of triangulation: giving the respondents two pointers to the same information. We had to construct it because it was not included in the original ArchiSurance case study material [62].

The purpose closely follows the stated aims of the diagrams from the original case study.

7.3.3 Building neutral (non-ArchiMate) test models

To implement these alternative models, we configured BiZZdesign Enterprise Studio, the EA modelling tool that we used to produce our test models, to produce customer viewpoints that, rather than using the standard ArchiMate notation and colours, used bespoke (i.e. chosen by us) colours to distinguish between entities in the same layer. A screenshot of the modelling tool showing the definition of this custom viewpoint is shown below:

Figure 66 Creating Custom Neutral Viewpoints



This completes the test materials (shown in detail in Appendix J), and thus the deliverable:

Deliverable 2.2A. Test materials for use when testing ArchiMate

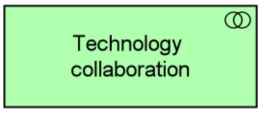
7.4 Challenges and Limitations

7.4.1 Collaborations and Interactions

There was a specific difficulty encountered with these. The paired concept of 'collaboration' and 'interaction' appears in three separate layers in the ArchiMate framework. The definitions below are taken from ArchiMate 3.0 [39], which was the version of the framework that we tested:

Table 26 Collaborations and Interactions in ArchiMate

Type	Appearance	Definition
Business Collaboration		Represents an aggregate of two or more business internal active structure elements that work together to perform collective behavior.
Business Interaction		Represents a unit of collective business behavior performed by (a collaboration of) two or more business actors, business roles, or business collaborations.
Application Collaboration		Represents an aggregate of two or more application internal active structure elements that work together to perform collective application behavior.
Application Interaction		Represents a unit of collective application behavior performed by (a collaboration of) two or more application components.

Technology Collaboration		Represents an aggregate of two or more technology internal active structure elements that work together to perform collective technology behavior.
Technology Interaction		Represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes.

None of the ArchiSurance case study material available at the time contained any examples of these. We found it difficult to think of real-world examples of these that we could introduce into the test material, and it is conceivable that our difficulty in finding examples was perhaps shared by others, which may account for them not being included in the ArchiSurance case study. We did not use the Technology (green) layer in our research and we managed to create one example of each of the others. These items are often encountered in pairs (because an interaction specifies what is done by a collaboration). We manage to invent two suitable pairs: one in the business layer, which we think was a good example and one in the application layer, which was somewhat more contrived. Having more than one example of each might have helped the interviewees more readily grasp the underlying concepts.

7.4.2 Grouping Actors by Job Role

Our original intention was to gain some insight as to what concepts from the ArchiMate language 'worked' (i.e. were both understood and found to be useful) by various classes of actor (categorised by their job role). Having completed the data collection, we realised that we did not have a sufficient volume of data to allow us to produce meaningful results categorised by class of job, neither had we established a satisfactory understanding of the literature to enable us to even carry out this categorising. Thus, in practice, we did not make any use of the data that we captured with regard to job role. This does however explain why the data about job role was captured as part of the questionnaires.

7.5 Controlled Experiment

As suggested in the Design Evaluation part of Design Science, we felt it appropriate to carry out some testing on our approach to measuring the effectiveness of a modelling language. Therefore, having constructed the test model for one particular layer (Motivation), we then tested the method by interviewing (one on one) approximately fifteen masters-level students (studying for IT-related degrees) to ensure that the students were able to understand what was being asked of them, and gave back a range of scores not obviously biased towards a 0 or a 1. This was indeed the case, and thus we felt confident that the methodology was practical enough to allow us to proceed with constructing the full set of test models.

7.6 Data Collection

7.6.1 Selection of interviewees

Our intention was to pick a cross-selection of respondents to interview that fairly represented a 'typical' business. In particular, we wanted to include people that were not IT architects (Enterprise, Solution or any other kind) because understanding how to present information to such non-technical stakeholders is a key motivating factor behind this part of our research. When starting this part of the research, we had intended to interview people purely within one organisation, however as the research progressed it became clear that we would need to interview people outside our current organisation, partly triggered by a change of employment on our part. Partly for this reason, and partly in an attempt to broaden further the cross-section of people that were represented in our sample survey, we chose to interview some people that were already known to us, outside our organisation, in a professional or personal capacity. In addition, we interviewed some current Masters-level students (in groups, in order to speed up the process of data capture), and also some individuals working in another organisation (in various roles related to start-up organisations) were recommended to us to be interviewed as well. Interviewing a combination of people familiar to us, as well as strangers to us, also provided a way of avoiding bias due to our relationship with those being interviewed (a potential concern raised in section 3.1.8.1.8 above).

7.6.2 Contextual Information about Interviews

There were a number of factors that we thought might affect the understanding of concepts from ArchiMate:

(1) if they were IT architects themselves then we thought this might have an impact, as they should be familiar with more of the concepts. Related to this was our particular interest in understanding how well these languages 'work' with non-technical stakeholders.
(2) we thought that formal education and training might affect understanding, particular for some of the more abstract concepts where a background in conceptual thinking may be useful.
(3) we thought their job role might also be relevant.

We will need to keep track (this is not data that the participants need to provide) of whether or not the participants were previously personally known to us; this is in order to minimise any bias, responding to the earlier discussion in section 3.1.8.1.8 above). The actual statistics on this factor that we generated, following the survey, can be found in 7.7.1 below.

In the discussion that follows, we have annotated the discussion with the number of the question on the questionnaire (shown in Appendix G below).

We specifically sought information about the interviewees in these three areas:

(1) Role with respect to EA work (*Q03*)

Bearing in mind our motivation mentioned above, and our desire to understand how well non-technical stakeholders understand certain concepts, we obviously needed to distinguish between technical and non-technical stakeholders, and thus this particular piece of information needed to be captured. This relates to the distinction between social and technical actors in Figure 25 above.

We therefore asked our respondents to self-select where they fitted, in one of three categories: (a) those people who are informed by models and make decisions based upon them (often non-technical stakeholders); (b) those people who produce such models (practitioners) and (c) those people who use these models as a basis for their own work (e.g. designers and developers).

(2) Formal education and training (*Q04, Q05, Q06*)

Based upon early pilot measurements, we were interested to see if people who have had certain kinds of training are more or less likely to understand certain symbols. Given that students specialise significantly in their educational choices, typically, from the age of 16, we will therefore ask for the equivalent of British 'A'-level subjects as well as degree choices and subsequent professional training, to see if there is any correlation here. In practice, for a number of reasons we were not actually able to make significant use of this piece of data, save for observations about the lack of formal architecture training of the category B (IT architect) respondents (see note in section 7.7.1).

(3) Job title (*Q02*)

This is more a reflection of their current role and interest than their previous experience, and like item (2), turned out not to be useful in practice. Nevertheless, we list these last two items as they were actually asked for and recorded in our interviews.

7.6.2.1 . One-on-One Interviews

The majority (approximately $\frac{2}{3}$) of the interviews were carried out one-on-one and the remainder in a group setting. As we can see in our literature search (see discussion in 3.2.2.3 regarding different modes of collection), this has the advantage of allowing clarifying and probing questions to be asked by both parties, and ensures that we actually get data back, at the (low) risk of introducing bias through the presence of the interviewer (the topics are not sensitive).

Each interviewee was given a choice as to which of the three ArchiMate layers they felt they would be able to answer questions about (they were able to select multiple if they wished). There are advantages and disadvantages to letting the interviewees self-select the layers on which they wished to be interviewed.

The advantages would be that they would be most likely to pick layers with which they felt a degree of familiarity and comfort. A non-technical interviewee might have very little idea about any of the concepts used to describe application architecture and might become uncomfortable having to answer, effectively, “I have no idea”, for each concept in turn.

There is also a potential disadvantage. A business stakeholder might prefer to answer questions in their domain of expertise (say, the motivation or business layers). However, we do also want to know how best to communicate technical concepts to business stakeholders and allowing them to elect not to answer questions about the application layer risks denying us the opportunity to understand how much of this layer they might understand.

The interviewees were given one of the test models (a pseudo-random choice as to whether it was ArchiMate or neutral – we tried to alternate to maintain a balance of the two), plus the narrative and purpose, and also a reference sheet giving code numbers for each concept on the diagrams, either ArchiMate or neutral as appropriate for their test models (see J.4 below). In these interviews, we were able to discuss the methodology as much as necessary with the interviewees.

During these one-on-one interviews, the interviewees gave their answers verbally; we were able to clarify where necessary, and then assign comprehension scores ourselves, attempting to compare their description of a particular concept with the official ArchiMate definitions.

The utility scores were also given verbally, although these were easier in that no judgment or analysis was necessary on our part, as their answer was taken as definitive (their opinion as to whether the item was useful).

7.6.2.2 Group Interviews

It had been our original intention to do all one-on-one interviews, however we decided after carrying out the majority of the one-on-one interviews that it would help boost the volume of data if we used a more efficient method of collectio, and the group interviews with their self-completion of questionnaires met that objective. Thus, approximately $\frac{1}{3}$ of the interviewees were dealt with as a group (described in section 3.2.2.3 as 'self-completion questionnaires). They were in fact students from Reading University, sitting in the same room, spaced out in much the same way as they might be for an examination. They were given instructions as a group, allowed to ask questions, and then they proceeded to describe in writing (by contrast to the previous group) their understanding of the underlying concepts. They were asked to assign their own comprehension scores, which we changed if we felt that their level of understanding (based upon their written explanation of the various concepts) was not reflected in their scores. They also assigned utility scores which we did not change, as these did not need any further analysis on our part.

This method had the advantage of capturing data more quickly, however carried the risk of more misunderstandings as there was less scope for intervention and discussion between the interviewer and respondents. This potential for misunderstanding went both ways. The respondents were told that they could ask questions while they were completing the questionnaire, however to the best of our recollection none did. However, we were unable to ask questions to clarify what they had written, as we did not collect, read and 'score' the data until some time after the group had completed the interview and left.

7.6.3 Questionnaire Design

After the initial information about the interviewee themselves, the remainder of the questionnaire simply contained spaces for responses to questions about the various concepts contained on the model. The blank template questionnaires can be seen in Appendix G below. The following tables show the different sections of the questionnaire, along with an explanation of the purpose of each of the pieces of data that is being captured.

Again, Bryman and Bell are helpful when it comes to questionnaire design; in their section already cited on Self-Completion Questionnaires [192], there is a discussion on the presentation and layout of the questionnaire. The ease of use of the questionnaire is undoubtedly more of a concern for those self-completing it than it is for the interviewer doing one-on-one interviews. Nevertheless, the suggestions are helpful. They include:

7.6.3.1 Uncramped presentation

The suggestion here is that the temptation to compress the questionnaire layout to take up as little space as possible should be avoided (citing research by Dillman [213] that includes postal surveys, where the respondents have time and opportunity to reflect upon whether or not they wish to complete the survey – an opportunity that was not afforded to our group of students)

7.6.3.2 Clear presentation

This includes techniques such as using consistent styles for questions and closed answers, including font sizes and effects, with the aim of making the questionnaire 'easy on the eye', not requiring so much concentration from the respondent that they give up because it is too hard to follow it.

7.6.3.3 Likert Scales

The 'tips and skills' in this part of the book also includes suggestions as to how to ask, and to capture the response to, closed questions. As we were originally intending to capture the data on behalf of the respondents, we were less concerned about the appearance of this part, and so we simply provided a box to record the ratings, along with an explanation at the top of each part of the questionnaire as to what 0, 0.3, 0.7 and 1 meant.

Table 27 Purpose of Fields in Questionnaire: Metadata

Interviewer	This was recorded because at one stage we were considering using multiple interviewers, in order to increase the rate at which interviews could be conducted and produce a larger volume of data to analyse. We would have used this to check for bias in results between different interviewers. In practice, just the author carried out the interviews, so this was not necessary.
Interviewee	This was captured to enable us to immediately identify the person concerned. It is not reproduced in the data summaries.
Interviewee Number	We included this as initially we thought it might be useful to help us organise the results, as well as providing a means of obfuscating the personal details. Ultimately it was not used.
Date/Time	These two fields enabled us to refer back to previous interviews, for those occasions where we carried out multiple interviews with the same person.
Location	
Used ArchiMate diagrams?	This enabled us to separate the results for ArchiMate notation from the results not using ArchiMate notation.

Table 28 Purpose of Fields in Questionnaire: Interviewee Background

Q01. Where do you work?	This was background information that we requested, which when put together with the next field, helped provide a fuller picture of the interviewee. In hindsight this particular field was not particular useful.
Q02. What is your job title?	This was captured to enable us to try and form hypothesis about correlation between personal background and training vs. comprehension of various concepts within the ArchiMate language. As it turns out, we did not succeed in this analysis, so along with Q01, this was not useful.
Q03. Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	This categorisation enabled us to separate out the results for IT architects from the other stakeholders.

Q04. What did you study at A-level? Q05. If you went to university, what did you study there? Q06. What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	With all of these questions, we were seeking to build a picture of the interviewee's education and training, knowing that this would undoubtedly affect their interpretation of the architecture models. This again was seeking to understand the correlation between certain types of background/experience and certain concepts. With sufficient data this would have been useful; in our case, we were not able to make significant use of it.

The questionnaires then have a reminder of the scoring to be applied for the comprehension questions:

Table 29 Comprehension scoring matrix

0	Don't understand what these are
0.3	Have some understanding of what these are
0.7	Understand what they are but not familiar with this kind of thing
1	Understand what they are and are familiar with them

We now move onto the actual quantitative data capture to measure the comprehension and utility of concepts in each of the Motivation, Business and Application layers. The questionnaire for each layer starts with the comprehension scores, and contains entries like these (the full questionnaire is reproduced in Appendix G below):

Table 30 Purpose of Fields in Questionnaire: Comprehension Data Capture

Q07. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	This is the rating suggested by the interviewee, and validated by us (or in some cases suggested by us) from Table 29 above	Entered here is the explanation by the interviewee of what they think all things of type "M1" have in common. This is likely to be a summary, agreed with the interviewee, of what they think is their "final answer", as there is likely to be a period of reflection while they consider their answer. For this particular box labelled M1, we will have asked the interviewee what they think of things of type "M1" have in common. Referring to the test models (diagrams) in Appendix J below, they will be studying a view (model) of the motivation layer (either ArchiMate, Figure 165, or 'neutral', Figure 166), and also using a key to show them what things of type "M1" look like(either ArchiMate, Figure 171, or 'neutral', Figure 172).
		Recorded in here may be two things. Firstly, the interviewee will have been asked why they think the answer is as given in the previous box, and what in their experience or training has led them to that conclusion. Secondly, we also on some occasions use it to make notes as to why we assigned a certain score to this item; and whether we ourselves on

		reflection modified the score if it became clearer later in the interview that they understood the concept better than we had at first thought.
M2		We then continue with the same structure for every concept in the particular ArchiMate 'layer' being tested, so the user's description goes here…
		… and their explanation of why, plus any comments from us about the scoring, go in here.

The following check is then made before continuing with the utility part of the interviewee:

Table 31 Purpose of Fields in Questionnaire: Check Prior to Utility Capture

Q08. Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	This refers to the purpose as listed on the narrative descriptions accompanying the models; we just record a Yes or No here.

We now move on to record the utility ratings against the same concepts. Unlike the comprehension ratings, these are specified entirely by the interviewee and not subject to testing by the interviewer. They reflect a purely subjective opinion that needs no justification.

Q09. For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	We record here a -1, 0 or +1 as specified by the interviewee	We record here any reason they may give for the rating, which may or may not be related to the interviewee's background or role.

As with the comprehension ratings, this continues for each concept in the particular ArchiMate layer being tested.

The questionnaire then has sections just like the ones above, to capture data from the other two layers (Business and Application).

7.6.4 Assigning Ratings

For the core quantitative data, being the comprehension and utility ratings, there was certainly a degree of interpretation and judgment required on behalf of the interviewer. Recall our previous discussion (see section 2.3.16 above) on understanding: we are in this research looking to adopt Kvanvig's version of a moderate view of objectual understanding's factivity. Therefore, for the comprehension ratings, we are looking to test whether the interviewee's beliefs about a concept matches the central aspects of the definition proposed by the modelling language in question (in our case study, that means ArchiMate 3.0). Thus, we are not looking for a complete description but rather, that the interviewee 'gets' the core idea(s) embedded in such a concept. This is the basis of how we assigned the ratings. In some cases, the interviewee actually said the name of the concept from the ArchiMate language. Assuming that the name is in general use and is not a technical term likely to be misunderstood, then in those circumstances we assigned a 1. Thus, when testing the concept of a stakeholder, given that this is a well-known term found in any dictionary, an interviewee that used the term scored a 1.

7.7 Mathematical Results

7.7.1 Volume of Data

Each of the interviews covered a single layer (one test model) and took between 30-60 minutes to complete. As each interview was completed, some key fields about the interviewee, plus their ratings, were entered onto an Excel® spreadsheet. In this spreadsheet, each row represented one interview, and each column contained the results for a single concept. We used separate tabs to separate out data from the Motivation, Business and Application layers:

The structure of each of these tabs, when first capturing the data, before starting the analysis, was as shown in the following example:

Table 32 Sample Data for ArchiMate Effectiveness in Business Layer

					Comprehension		...		Utility		...	
Date	Job Title	Group	Category	ArchiMate	B1	B2	...	B13	B1	B2	...	B13
16/02/2019	Delivery Manager	No	A	Yes	0.3	0.3	...	0.3	0	1	...	1
02/03/2019	COO	No	A	No	1	0.3	...	0.7	-1	1	...	-1
20/03/2019	Principal Engineering Manager	No	A	Yes	0.7	0.3	...	0	1	1	...	-1
28/03/2019	Service Architect	No	B	Yes	0.3	0.3	...	0	0	0	...	0
18/06/2019	Freelance IT Consultant (now Pa	No	C	Yes	0.7	0.3	...	0.3	1	1	...	0
16/07/2019	Travel Buyer	No	A	Yes	0.3	0.7	...	0	1	1	...	0
18/06/2019	Enterprise Architect (Customer)	No	B	No	0.7	0.3	...	0.3	1	1	...	0
18/06/2019	Data Governance Architect	No	B	Yes	1	0.7	...	0.3	1	1	...	1
06/05/2019	Programme Lead - Governance	No	A	No	0.3	0.3	...	1	0	1	...	1
09/05/2019	Credit Controller	No		No	0.3	0	...	1	0	1	...	1
20/06/2019	Platform Architect (telephony)	No	B	Yes	1	0.3	...	0	1	1	...	-1
05/10/2019	Welfare Benefits Advisor	No	A	No	0.3	0.3	...	0	1	1	...	1
31/08/2019	Senior Sister (nursing)	No	A	Yes	0.3	0	...	0.3	1	1	...	1
30/10/2019	Managing Director (retired)	No	A	No	0.7	0	...	0.7	1	1	...	1
30/10/2019	Initiative Assistant	No	A	No	0.3	0.7	...	0	1	1	...	-1
30/10/2019	Teacher, Consultant	No	A	No	0	0	...	0.7	0	1	...	1
31/10/2019	Product Manager	Yes	A	Yes	0.3	0.7	...	0	1	1	...	0
31/10/2019	Data Manager	Yes		No	0.7	0.7	...	0	1	1	...	1
31/10/2019	Group	Yes		No	0	0.3	...	0.3	1	0	...	1
31/10/2019	Credit policy analyst	Yes	A	Yes	0.7	0.3	...	0	1	1	...	1
31/10/2019	Junior management consultant	Yes	A	No	1	0.3	...	0	1	1	...	0
31/10/2019	Group	Yes		Yes	1	0	...	0	-1	0	...	-1
31/10/2019	Group / Associate	Yes	C	Yes	0.3	0.3	...	0.7	-1	1	...	1
31/10/2019	Group	Yes	A	No	0	0.3	...	0	0	1	...	0
31/10/2019	Supply chain analyst	Yes	A	No	0.3	0.3	...	0	-1	1	...	1

This data is anonymized; the only personal data visible is the role (job title) of the interviewee. The comprehension and utility ratings from the questionnaire are copied directly into this spreadsheet. The data for each column is derived thus:

Table 33 Source of columnar data in spreadsheet

Date	The date of the interview, taken from the completed questionnaire
Job Title	The job title given by the interviewee
Group	Although not in the questionnaire explicitly, we used a notation against the name of the interviewee to indicate whether they were interviewed in a group or individually.
Category	This came from the questionnaire. In some cases the interviewee did not answer this question, or did not feel that they fitted into any of the categories, and in these cases we left this blank.
ArchiMate	This was again taken from the questionnaire.
Comprehension ratings	There is one column for each concept being tested. The ratings are taken from the questionnaire.

Utility ratings	There is one column for each concept being tested. The ratings are taken from the questionnaire.

Excel was used to sum the number of rows containing data meeting certain criteria. For example, to count the number of rows where the interview used ArchiMate notation, the COUNTIF formula was used: =COUNTIF(F$3:F$26,"=Yes")

Thus, we were able to produce the following statistics about the volume of data collected.

Table 34 Data Volume Collected

Layer	*Total*	**ArchiMate**	**Neutral**	**One-on-One**	**Group**
Motivation	23	12	11	14	9
Business	25	12	13	16	9
Application	20	9	10	11	9
	68	33	34	41	27

Here, an interview is defined as being for a particular layer so, if one interviewee was asked about two layers, even if those two layers were dealt with in the same meeting, that counts for our purposes as two interviews. There is one missing from one of the total ArchiMate Neutral notation columns because we forgot to record, for one interview, which notation we used.

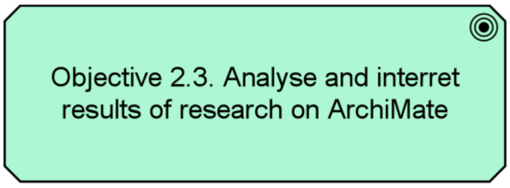
Of the 68 interviews carried out, 33 were with respondents previously known to us, and so there is a pretty even balance which goes some way to address the concern raised about generalizability and the relationship with the interviewer. Also of interest is the fact that of the category B respondents (IT architects), looking at their formal education, only two of them (out of the 8 interviews conducted with category B respondents) have had training in an EA framework (one in TOGAF; one in Zachman). Possibly more exposure to EA frameworks might have produced different results here.

This data (shown in detail in Appendix K) constitutes the results from the surveys, and thus meets deliverable 2.2.b

Deliverable 2.2B. Results of primary research on ArchiMate

7.7.2 Data Analysis

We now move on to the analysis and interpretation of the data gathered:



This section details with how the raw data was analysed to produce the final statistics that we were seeking; and the following one deals with the visualisation and interpretation of those statistics.

The raw data in the spreadsheet (one tab per ArchiMate layer), plus the intermediate results (scores), can be viewed in appendix k

7.7.2.1 Introduction to Data Analysis of Effectiveness Ratings

A number of filters were then applied to this data. In other words, if we wanted to understand how the results for category A users different from the other categories, then we produced subtotals for each category.

Excel was then used to produce graphs of this data, in a form compatible with what we had envisaged in Figure 64 above.

In most case, the results were simply a filtering of the raw data by some category. For the “ArchiMate vs. Neutral” diagrams, we used two sets of totals and subtracted one from the other. Thus, for example, if the average score for comprehension of a particular concept when using ArchiMate was 0.2, and the same score when using neutral (non-ArchiMate) notation was 0.5, then we would score this as 0.2 – 0.5 = - 0.3 for the relative comprehension of ArchiMate: a negative score.

We now consider each section in turn of the analysis to show how precisely how it was done. We will illustrate this with the data from the Application tab (layer); exactly the same techniques were used to analyse the Motivation and Business data, although the cell references will be slightly different as each layer has a different number of concepts being tested (columns) and a different number of interviews (rows).

This will require us to refer to specific “cells” within the spreadsheet holding the data and so we have included the row and column headers from Excel in the raw data shown in Appendix K, in Table 71 (Motivation Layer Detailed Analysis), Table 72 (Business Layer Detailed Analysis) and Table 73 (Application Layer Detailed Analysis) to enable the reader to see exactly how our Excel calculations have been carried out.

7.7.2.2 Summing the Ratings

In order to perform the calculations described in sections 6.4.2 (Measuring Comprehension) and 6.4.3 (Measuring Utility) we need to sum the ratings and then divide by the number of ratings, in order to achieve a normalised score, between 0 and 1 for the comprehension ratings and between -1 and 1 for the utility ratings. We cannot rely upon every column having the same number of rows. This is because occasionally, an interviewee did not give a rating for a particular concept (possibly our fault if we forgot to ask for it). There was also one interview where we forgot to record whether or not the test models used ArchiMate notation and so again we have left the cell blank.

Before each of these calculations we will prefix the discussion with a representation from the spreadsheet of the cells that we are populating; this will enable us to keep track of how the calculations relate to the overall results.

Cells E24-F25

ArchiMate	9
Neutral	10

We start by summing the number of interviews (in this particular layer). The total number of interviews carried out using ArchiMate is given by the number of rows that have “Yes” in the “ArchiMate” column. This is therefore accomplished by use of the formula `=COUNTIF(F$3:F$22,"=Yes")` in cell F24 (in the Application tab; this is F27 in the Motivation tab and F29 in the Business tab; for the sake of brevity and clarity we will only give references in the Application tab from now on).

The number of interviews carried out using non-ArchiMate (neutral) notation is calculated using a similar technique, but here we are looking for a “No” in the “ArchiMate” column, and so the formula used is `=COUNTIF(F$3:F$22,"=No")`, within cell.

We now consider the data (ratings) assigned to each concept, and to illustrate the techniques, we will focus on the first concept (labelled A1, which equates to the ArchiMate concept of “Application Component”). Recall that each concept is represented in a different column within this spreadsheet. Note also that each concept appears twice in the spreadsheet because it has two ratings. The first column it appears in contains the comprehension rating; the second column contains its utility rating. Thus, in the Application layer, we will be focusing on columns G and P, as those are the ones that contain the two ratings for concept A1 (Application Component).

We now want to calculate the overall ratings for the concept, as well as calculating the ratings just for the ArchiMate, and non-ArchiMate, interviews, and so populating the following cells:

	E	F	G	P
26	Count Rows Populated		20	18
27	Total Score		16.9	13
28	Count Rows ArchiMate		9	7
29	Score Rows ArchiMate		6.6	5
30	Count Rows Neutral		10	10
31	Score Rows Neutral		10	7

For the first two rows above, we want the overall averaged ratings, and so we want the total number of ratings (cell G26) divided by the sum of the ratings (cell G27). G26 is obtained by counting the number of numeric entries in the ratings cells, so using the formula: [G26] =COUNT(G3:G22)[20]. G27 is obtained by summing the numeric entries, so using the formula [G27] =SUM(G1:G22). We carry out the division further down the spreadsheet. The formulae in cells P26 and P27 are similar, operating on the contents of column P rather than column G, so having the contents [P26] =COUNT(P3:P22) and [P27] =SUM(P3:P22) respectively.

To do the same operations of counting ratings and rating entries for just the ArchiMate interviews requires us to filter the rows so that we only include those where the ArchiMate column was set to "Yes". Thus, for the comprehension figures, the count of ArchiMate rows calculated in cell G28 uses a COUNTIF function to achieve this filtering, and contains the formula: [G28] =COUNTIF($F$3:$F$22,"Yes"). Similarly, the summed ratings in cell G29 uses a SUMIF function to include only ArchiMate results, and the formula used is [G29] =SUMIF($F$3:$F$22,"Yes",G$3:G$22). The utility figures use a similar technique, but in column P rather than column G. There is a specific adjustment that we should mention at this stage. When we tested the Application layer, as previously mentioned, there were two occasions where we failed to collect utility ratings, and so our spreadsheet is missing two rows (rows 6 and 19 in Table 73 below) from the utility scores, for interviews where the ArchiMate column holds a "Yes". For this reason, there is the -2 added in the first of these formulae: [P28] =COUNTIF($F$3:$F$22,"Yes")-2 and [P29] =SUMIF($F$3:$F$22,"Yes",P$3:P$22), in order to avoid having two extra counts of "Yes" for ArchiMate, which would have incorrectly biased the figures by adjusting down the ratings for the ArchiMate models.

[20] By this notation of [<address>] =<formula> what we specifically mean is, the cell in the spreadsheet referred to by the given address is set to =<formula>.

Finally, moving on to the ratings for the neutral (non-ArchiMate) models, there were no missing rows of data, and so for the comprehension totals, we used: [G30] =COUNTIF($F$3:$F$22,"No") and [G31] =SUMIF($F$3:$F$22,"No",G$3:G$22); likewise for the utility totals, we have [P30] = =COUNTIF($F$3:$F$22,"No")[21] and [P31] =SUMIF($F$3:$F$22,"No",P$3:P$22) .

We now move on to consider the difference between the interviews carried out in groups, and those carried out on an individual basis, and thus to calculate the following values:

	E	F	G	P
32	Count Rows Group		9	8
33	Score Rows Group		7.3	4
34	Count Rows Non-Group		11	10
35	Score Rows Non-Group		9.6	9

The red triangles here were used to record our comments that we were missing some utility ratings for the Application layer, and this time one was ArchiMate, one for neutral notation; and so we subtracted one from the row count for each, as we shall see below.

The count of rows for group interviews is given by the formulae [G32] =COUNTIF($D$3:$D$22,"Yes") and [P32] =COUNTIF($D$3:$D$22,"Yes")-1 (the minus one being there for reasons discussed in the previous sentence). For the total ratings for group interviews we have the formulae [G33] =SUMIF($D$3:$D$22,"Yes",G$3:G$22) and [P33] =SUMIF($D$3:$D$22,"Yes",P$3:P$22).

Similarly, for the non-group (individual) interviews we are looking for "No" values rather than "Yes" in column D, and so the count of rows for these individual interviews are given by: [G34] =COUNTIF($D$3:$D$22,"No") and [P34] =COUNTIF($D$3:$D$22,"No")-1 (minus one again for the missing row of utility ratings). Finally, the summed ratings for non-group interfaces are calculated by [G35] =SUMIF($D$3:$D$22,"No",G$3:G$22) and [P35] =SUMIF($D$3:$D$22,"No",P$3:P$22).

Our final filtered version of data focuses on those interviewees who self-identified as being in either the A or B category. We did not do the filtering for category C interviewees because the volume of data was too low to make this worthwhile. We are thus populating the following cells for the Application layer:

[21] No adjustment down was required as we were not missing any rows of data that had a "No" in the ArchiMate column.

	E	F	G	P
36	Count Rows Type A		10	9
37	Score Rows Type A		8.3	8
38	Count Rows Type B		3	3
39	Score Rows Type B		3	2

This time we selected data based upon column E (category) rather than column F (ArchiMate?). Apart from that, the formulae are almost identical. The row count for category A interviewees is given by [G36] =COUNTIF($E$3:$E$22,"A") and [P36] =COUNTIF($E$3:$E$22,"A")-1 (again a -1 because an interviewee of this category failed to provide utility ratings). The summed ratings for category A interviewees is given by [G37] =SUMIF($E$3:$E$22,"A",G$3:G$22) and [P37] =SUMIF($E$3:$E$22,"A",P$3:P$22).

For category B users we simply replace the “A” with “B” in the above formulae, and so the row counts are given by [G38] =COUNTIF($E$3:$E$22,"B") with [P38] =COUNTIF($E$3:$E$22,"B"); and for the summed ratings we have [G39] =SUMIF($E$3:$E$22,"B",G$3:G$22) with [P39] =SUMIF($E$3:$E$22,"B",P$3:P$22).

The complete set of formulae used in all three layers can be seen in detail in Appendix L below.

7.7.2.3 Normalising the Ratings

The mathematics involved here is simpler than first envisaged in our discussion in sections 6.4.2 6.4.3 above, as each concept type only appears on a single model. Thus, the task of normalising the ratings in the standard ranges (0 to 1 for comprehension; -1 to 1 for utility) is simply a matter of dividing the summed rating values over all actors (interviewees) by the number of ratings (p), to give us the values H_C (comprehension) and U_C (utility). In the next section we will discuss the visualisation; but we mention here that we wish to collect these normalised ratings in rows that have the comprehension and utility ratings adjacent to each other, because we will be using X-Y scatter plots to produce charts in the form we are seeking in our ‘conceptual visualisation space’ (see Figure 64 above).

Thus, having carried out the normalisation to both the comprehension and utility ratings, we then copy the utility ratings to the rows immediately beneath the corresponding comprehension ratings, as illustrated below (the blue arrows show the extent of the copied cells):

Figure 67 Transposition of Utility Ratings

	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
40		Element	A1	A2	A3	A4	A5	A6	A7	A8	A9									
41	Overall	Comprehensior	0.85	0.19	0.48	0.30	0.13	0.61	0.48	0.32	0.52	0.72	-0.03	0.61	0.50	0.25	0.72	0.78	0.67	0.61
42		Utility	0.72	-0.03	0.61	0.50	0.25	0.72	0.78	0.67	0.61									
43	ArchiMate	Comprehensior	0.73	0.21	0.32	0.24	0.13	0.68	0.32	0.44	0.43	0.71	0.36	0.71	0.57	0.36	0.57	0.86	0.71	0.57
44		Utility	0.71	0.36	0.71	0.57	0.36	0.57	0.86	0.71	0.57									
45	Neutral	Comprehensior	1.00	0.19	0.60	0.38	0.06	0.54	0.56	0.20	0.54	0.70	-0.20	0.50	0.60	0.10	0.80	0.70	0.70	0.60
46		Utility	0.70	-0.20	0.50	0.60	0.10	0.80	0.70	0.70	0.60									
47	Group	Comprehensior	0.81	0.13	0.52	0.28	0.10	0.52	0.43	0.22	0.37	0.50	0.00	0.75	0.25	0.38	0.50	0.75	0.50	0.63
48		Utility	0.50	0.00	0.75	0.25	0.38	0.50	0.75	0.50	0.63									
49	Interview	Comprehensior	0.87	0.24	0.45	0.32	0.15	0.68	0.51	0.39	0.64	0.90	-0.05	0.50	0.70	0.15	0.90	0.80	0.80	0.60
50		Utility	0.90	-0.05	0.50	0.70	0.15	0.90	0.80	0.80	0.60									
51	ArchiMate	Comprehensior	-0.27	0.02	-0.28	-0.14	0.07	0.14	-0.24	0.24	-0.11									
52		Utility	0.01	0.56	0.21	-0.03	0.26	-0.23	0.16	0.01	-0.03									
53	Int vs. Grou	Comprehensior	0.06	0.10	-0.08	0.04	0.05	0.16	0.08	0.17	0.27									
54		Utility	0.40	-0.05	-0.25	0.45	-0.23	0.40	0.05	0.30	-0.03									
55	Type A	Comprehensior	0.83	0.16	0.38	0.28	0.09	0.57	0.39	0.33	0.64	0.89	0.11	0.44	0.56	0.33	0.89	0.89	0.78	0.67
56		Utility	0.89	0.11	0.44	0.56	0.33	0.89	0.89	0.78	0.67									
57	Type B	Comprehensior	1	0.2	0.67	0.3	0.2	0.57	0.77	0.57	0.43	0.67	0.67	0.33	1	0.67	1	0.33	1	0.67
58		Utility	0.667	0.67	0.33	1	0.67	1	0.33	1	0.67									
59			Application component	Application collaboration	Application interface	Application function	Application interaction	Application process	Application event	Application service	Data object									

We also add in the final row (59) the full name of the ArchiMate concept that we are testing (recall for example that A1 corresponds to Application Component), so that our scatter charts can have the correct names associated with the data points plotted therein.

No copy operations were required for rows 51 to 54. This is because these two pairs of rows are derived from other rows in the same section. They are used to calculate differential scores between the results depending upon (a) whether or not ArchiMate notation was used (rows 51 and 52), and whether or not the interview was done in a group setting (rows 53 and 54); we will discuss them later in this section.

Returning to the data for concept A1, we want to populate the following data fields:

	E	F	G	P
40		Element	A1	
41	Overall	Comprehensior	0.85	0.72
42		Utility	0.72	
43	ArchiMate	Comprehensior	0.73	0.71
44		Utility	0.71	
45	Neutral	Comprehensior	1.00	0.70
46		Utility	0.70	
47	Group	Comprehensior	0.81	0.50
48		Utility	0.50	
49	Interview	Comprehensior	0.87	0.90
50		Utility	0.90	

Column P is also for A1 (utility scores), although the column label showing this is not shown in the above fragment of the spreadsheet.

The normalised ratings for comprehension and utility are just the summed ratings divided by the number of ratings: [G41] =G27/G26 and [P41] =P27/P26. The utility rating is then copied diagonally down so that it is underneath the comprehension rating: [G42] =P41.

For the ArchiMate and Neutral normalised ratings, we use the same technique but take data from different rows that have already had the appropriate data filtered out, so we have for ArchiMate:

Comprehension is [G43] =G29/G28 and Utility is [P43] =P29/P28;

For non-ArchiMate (neutral) models we have:

Comprehension is [G45] =G31/G30 and Utility is [P45] =P31/P30 (also copied diagonally to [G44]).

For the filtered views that show ratings for group and then individual interviews, we use similar techniques.

For the group interviews we have:

Comprehension is [G47] =G33/G32 and Utility is [P47] =P33/P32 (also copied to [G48]).

For the one-on-one interviews we have:

Comprehension is [G49] =G35/G34 and Utility is [P49] =P35/P34 (also copied to [G50]).

Other filtered views, related to categories of users, are as follows:

	E	F	G	P
55	Type A	Comprehensio	0.83	0.89
56		Utility	0.89	
57	Type B	Comprehensio	1	0.67
58		Utility	0.667	

For type (category) A interviewees we have:

Comprehension is [G55] =G37/G36 and Utility is [P55] =P37/P36 (also copied to [G56])

For type B interviewees we have:

Comprehension is [G57] =G39/G38 and Utility is [P57] =P39/P38 (also copied to [G58]

7.7.2.4 Differential Ratings

The final part of our calculations relate to differential ratings – seeing the *difference* between two groups of ratings – as discussed in 6.4.5 above. In order to understand the impact of choosing to use

ArchiMate notation, we are going to subtract the ratings for the non-ArchiMate interviews from the ratings for the ArchiMate interviews. We used the same technique in terms of individual vs. group interviews in the spreadsheet. However, we have not chosen to further visualise and interpret this differential data at this stage, and so those formulae are not shown here.

The formula used for ArchiMate vs. non-ArchiMate notation simply subtract the latter from the former, for both comprehension and utility. We are populating these cells (recall we are only showing the data for one concept, being the Application Component, referred to during the interviews as concept A1).

	E	F	G
51	ArchiMate vs. Neutral	Comprehension	-0.27
52		Utility	0.71

The formulae are [G51] =G43-G45 and [G52] =G44-G46.

All other columns follow the same pattern.

7.7.3 Data Visualisation and Analysis

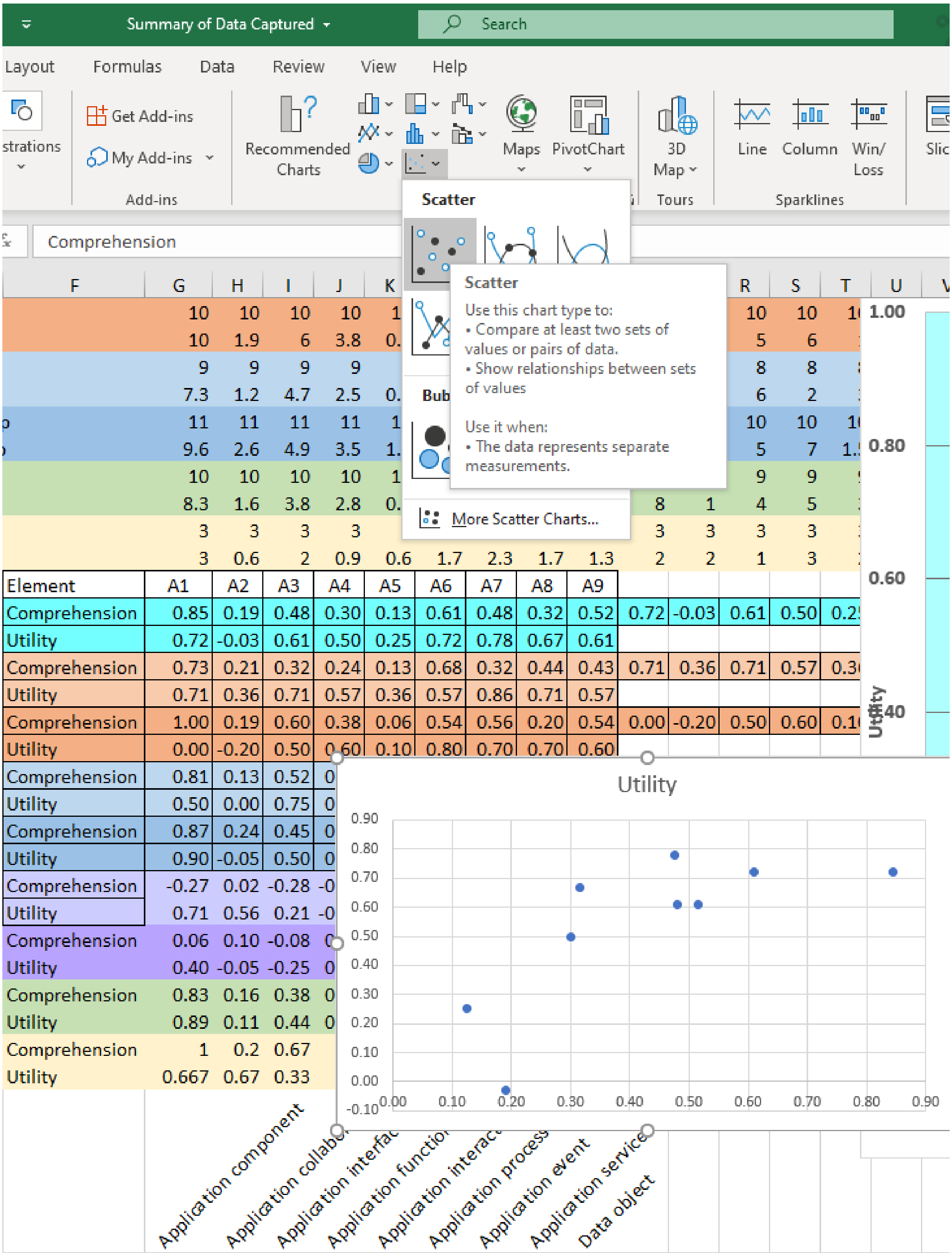
The visualisation technique used is a scatter graph, where the X co-ordinate is based upon the comprehension rating and the Y co-ordinate is based upon the utility rating. This matches the orientation of our conceptual visualisation space. To illustrate how this is done, we will step through the creation of the overall comprehension vs. utility graph for the Application layer, included below as Figure 81.

Recall that the utility scores in each case are immediately below the corresponding comprehension scores. The cells required, therefore, are drawn from three rows. To create the graph, select the two rows of data. In the example below, I am using the data for the Application layer:

Comprehension	0.85	0.19	0.48	0.30	0.13	0.61	0.48	0.32	0.52
Utility	0.72	-0.03	0.61	0.50	0.25	0.72	0.78	0.67	0.61

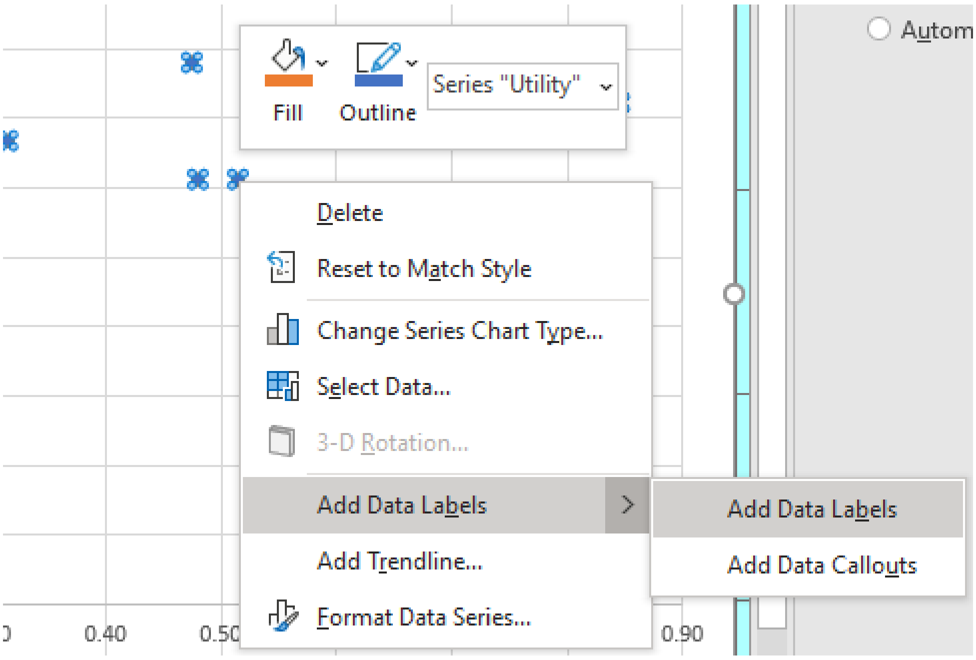
Having selected these two rows of data, I then selected an X-Y chart from the menus in Excel and selected the Scatter diagram, as can be seen below:

Figure 68 Creating Initial Scatter Diagram from Concept Ratings



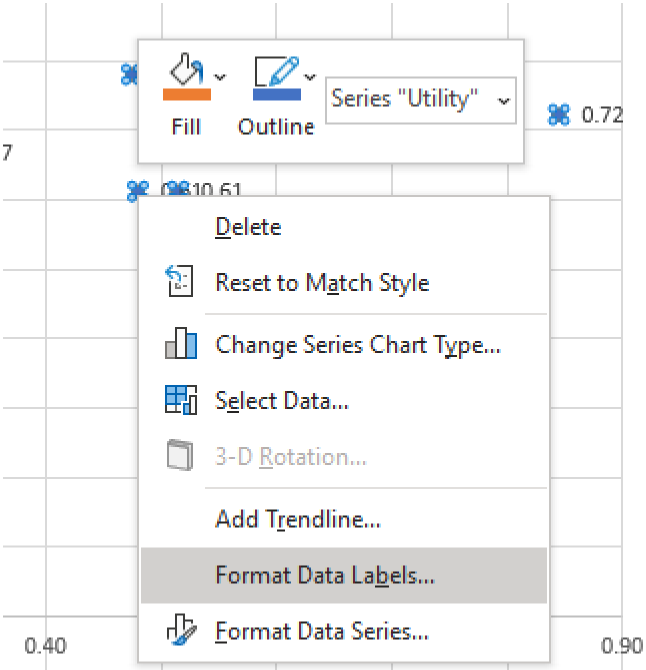
This creates the initial graph with points in the appropriate locations. The next step is to ensure that the correct concept names (e.g. Application Component) are used as labels for the data points. We have placed the concept names in the same columns, visible at the bottom of Figure 68 above. To associate these with the data points, we select one of the data points, use a right click and select Add Data Labels:

Figure 69 Selecting Data Labels



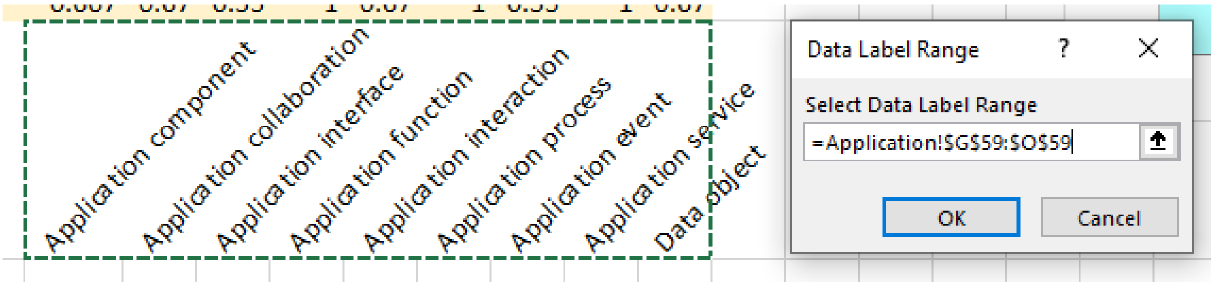
This will initially cause numeric data labels to be displayed, in fact we see the Y-values. We obviously want to change this to be the concept names, and so now we right-click again on one of the data points and this time select Format Data Labels:

Figure 70 Selecting Format Data Labels



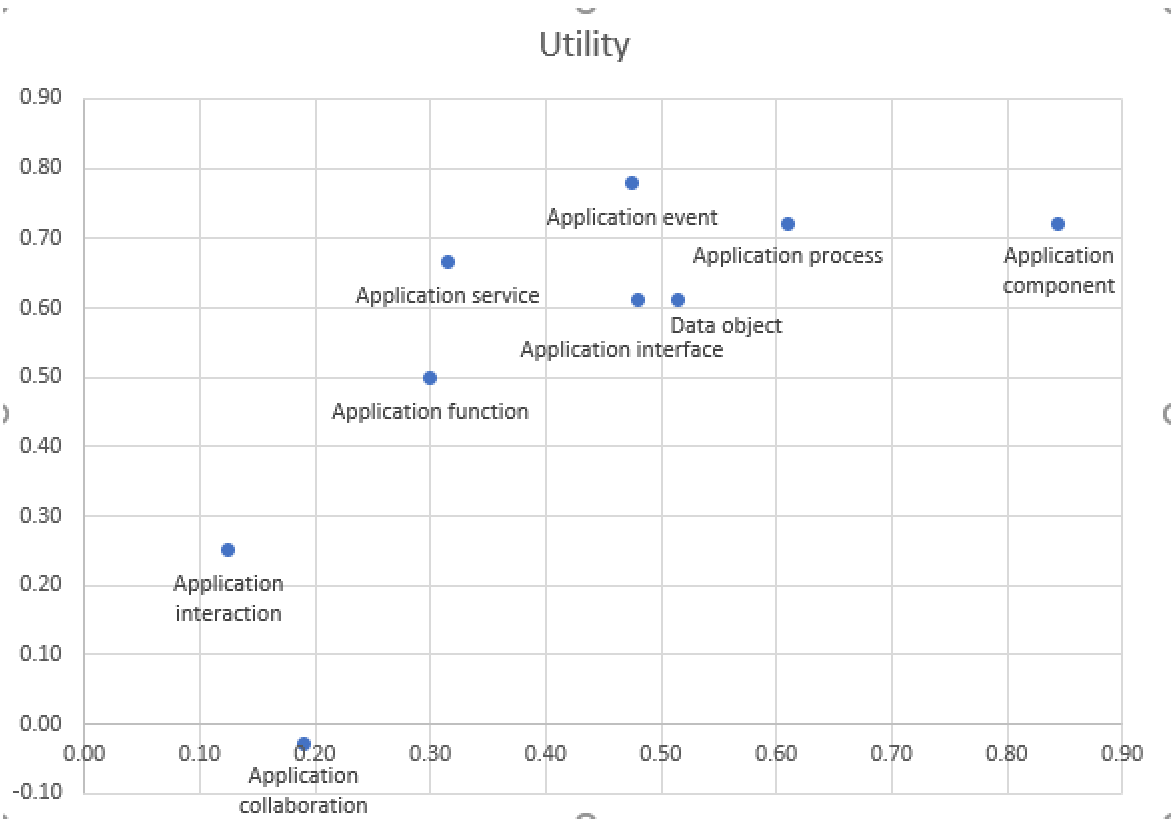
We now have the option to get our labels from cells, using the Value From Cells option. When this check mark is selected, we are prompted to select a Data Label Range, and we then select cells [G59..O59] that contain our labels:

Figure 71 Selecting Cells containing Data Labels



By also ensuring that we have cleared the checkmarks that say Y Value and Show Leader Lines, we finally arrive at correctly labelled data points:

Figure 72 Correctly Labelled Data Points



The remaining changes are cosmetic, and including changing the title from Utility, to (in this case) Overall Application Ratings; changing the chart background colour; changing the colour of the data points; and changing the font size.

All of the charts shown in section 7.7.4 below were created in this manner.

7.7.4 Detailed Results

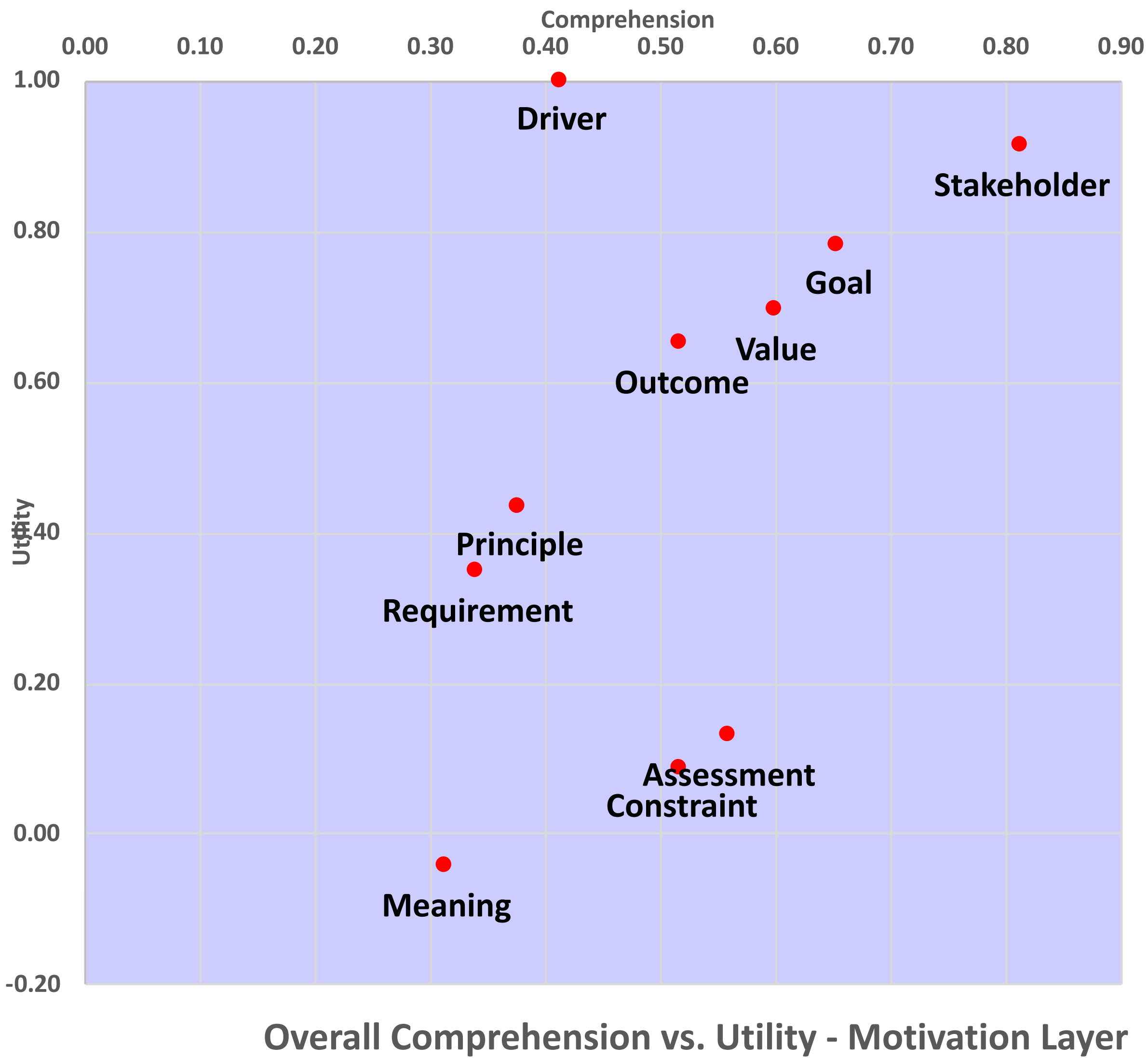
We present here the graphs showing the comprehension and utility scores for all the concepts within the three layers of the ArchiMate language. The results are broken down by layer; and for each layer we show results grouped as follows. We also include a discussion as to the immediate conclusions from each graph; and also any comments included from interviewees elaborating on their scores.

(a) Overall Results – scores averaged across all the interviewees
(b) Effect of ArchiMate Notation – when compared with Neutral Notation
(c) Results by "Role with respect to EA work" - see section 7.6.1 above; we have not included results for category C users as the data volumes are much too low for this category

7.7.4.1 Results for Motivation Layer

7.7.4.1.1 Overall Results

Figure 73 Motivation – Overall Results

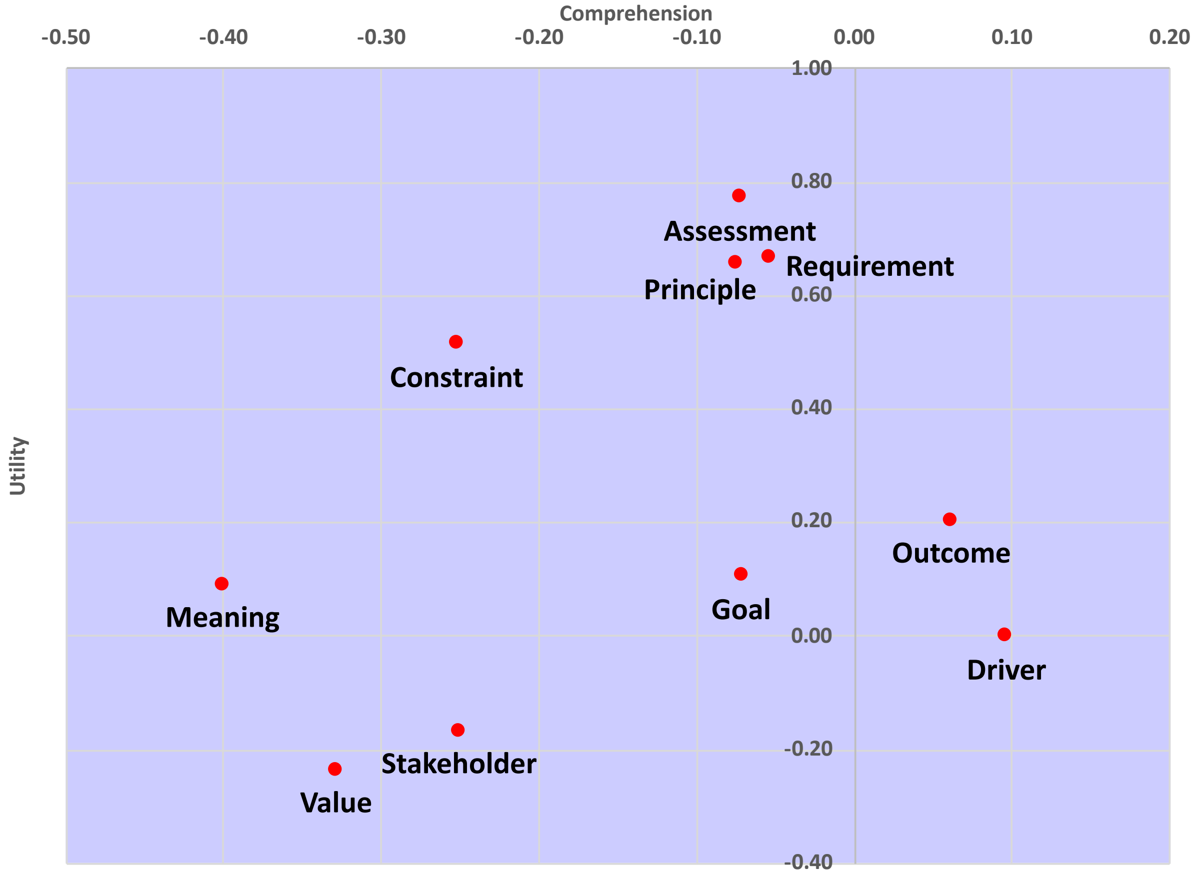


It is clear that the most well-understood, and useful, concept in the Motivation layer is that of a Stakeholder. The concepts of Goal, Value, Outcome and Driver are less well understood, but still seen as relatively useful, in terms of contributing to the value of a motivation diagram (e.g. Driver: *"absolutely essential",* Goal: *"absolutely vital")*.

Principle, Requirement, Assessment and Constraint are slightly less well understood, and are viewed as being significantly less useful; and Meaning is both poorly understood and, with a negative Utility score, overall the respondents felt it should not be included.

7.7.4.1.2 Effect of ArchiMate Notation

Figure 74 Motivation - ArchiMate Notation vs. Neutral



ArchiMate vs. Neutral - Motivation Layer

In terms of overall trends, it is clear that for the majority of the concepts in this layer, the degree to which the respondents were able to articulate the underlying concepts was reduced where ArchiMate notation had been used, compared to where it had not. This can be seen by the fact that the data points are clearly to the left of the central vertical line (which represents "neutral" for the understanding of the concepts). Specifically, of the 10 concepts, 8 (80%) were less well understood when expressed in ArchiMate.

In some cases the respondents made it clear why the ArchiMate notation was not helpful, for example comments like: *"Don't like the cylinder that represents it"* and *"The cloud is not helpful here"*. This does not account however why, given that the respondents understood when the ArchiMate symbols were misleading, the comprehension was still impaired. There may be other psychological factors here that further research may be able to uncover.

The trend is reversed for a slight majority of the utility scores: the concepts (even if poorly understood) have been rated, in most cases, more useful, when included in an ArchiMate diagram. Thus, put in simplistic terms, the respondents felt that the information that they were presented with were important but did not recognise the underlying concepts. This is does not necessarily represent an error: a respondent may feel that a piece of information is important without being able to classify that information into a specific taxonomy (architecture language)

7.7.4.1.3 With Respect to EA Role

The results for the motivation layer for Category A respondents (non-technical stakeholders) are as follows:

Figure 75 Motivation – Category A

The number of data points here is quite low (11 respondents for the Motivation layer in this category), and so these results should be taken merely as suggestive. They do indicate that for these types of people, many of the concepts would not be at all obvious.

For example, if we take the example of Outcome (known as M7 in our survey)– we see that it has a comprehension rating of 0.26 and a utility rating of 0.64. The latter indicates that there was a preference to include it in models because it seemed to add value.

We have included below a sample set of comments from the interviewees in category A concerning this particular concept (Outcome), along with the ratings. Recall that the comprehension ratings are assigned by us; whereas the utility ratings are assigned by the interviewee.

Table 35 Outcome Explanations - Category A

Comprehension		Utility	
Interviewee Explanation of Concept	Interviewer Rating	Interviewee Rating	Interviewee Explanation of Rating
Measurable targets – has a target number	0.3	+1	How we're going to measure the business - KPIs
Positive outcomes that serve everyone – outcome meaning what happens as a consequence of the way you run business processes	1	+1	Really helpful – they call out something regulatory about doing the entire thing
Measures – which help achieve the previous stuff – something you can measure on an ongoing basis	0.3	0	Could be integrated into M6 [=Goal]
Specific measurable targets	0.7	+1	Also pretty key
Things that you should be aiming for, corporately non-negotiable	0.7	+1	They are numeric, firm targets, measurements by which you run the business – the only quants on here

Action points specific to this business -	0	+1	They give specific targets
The specific detail of targets are belong (*sic*) to the customers and some of them have specific number.	0.3	+1	

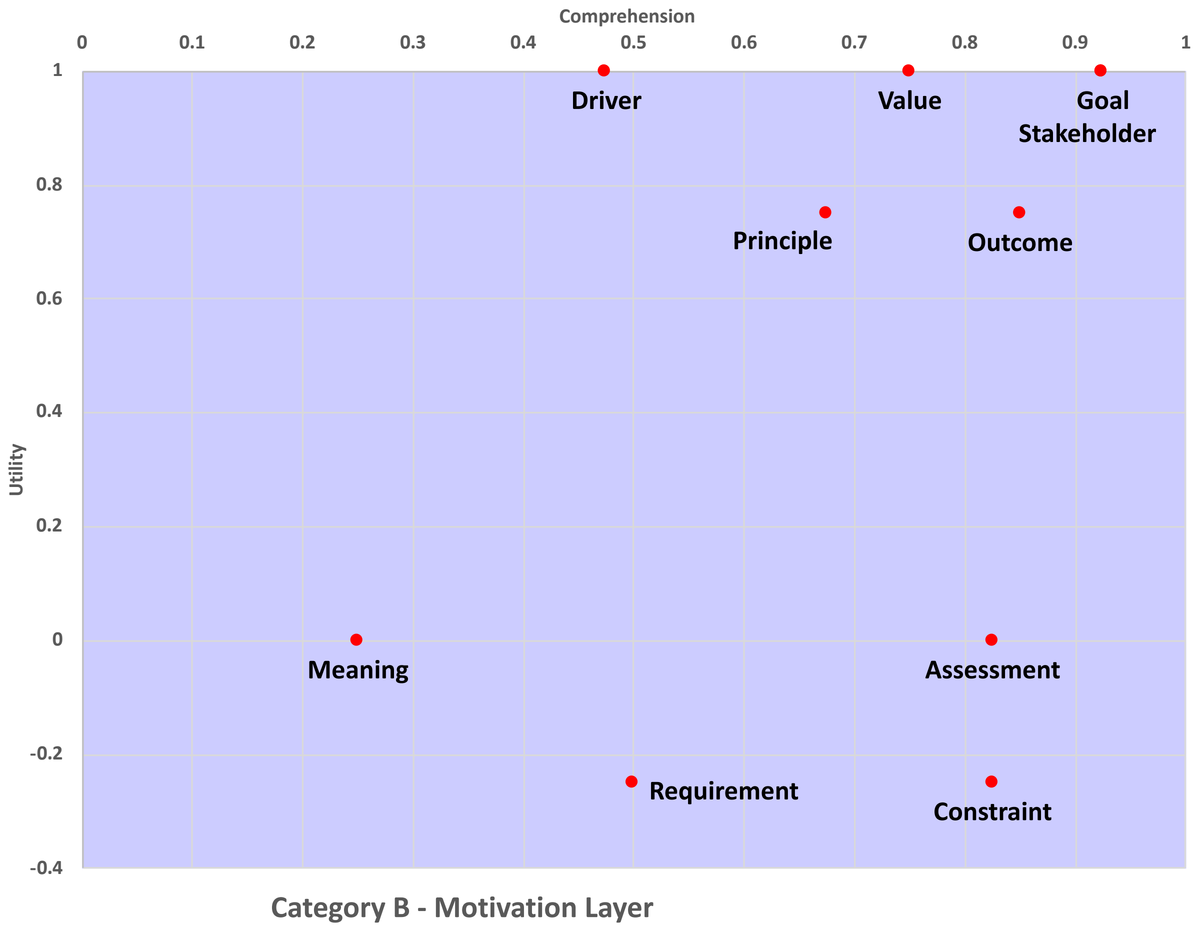
Almost all the interviewees responded with a +1 for utility, in that they thought the concept was helpful. However, only one of them clearly identified the concept as an outcome, defined in ArchiMate 3.0 as “an end result that has been achieved”. Recall that our view of ‘understanding’ is here tied to Kvanvig’s “moderate view of objective understanding’s factivity”, and so we are looking for alignment on the core beliefs of the definition, rather than a perfect understanding of all aspects of the ArchiMate definitions.

It is the past tense in this definition that resulted in the lower scores being awarded by us, because one of the core beliefs, in our view, was that this represents something already achieved. It should be noted that ArchiMate 3.1, released since our material was developed, has modified this definition to remove the past tense: the new definition is “represents an end result”. Thus, if our research had been carried out against ArchiMate 3.1, with its revised definitions, then these comprehension ratings would have been higher.

The most easily understood concepts would be Stakeholder, Value and Goal. These tend to be expressed in ‘familiar’ terms, not necessarily technical, for example a Stakeholder was described by one (category A) respondent as “The two main players; no-one more important to keep happy”. Thus, the concept appeared familiar without knowing the technical term for it.

For the IT architecture practitioners, we obtained the following:

Figure 76 Motivation – Category B



Unsurprisingly, many of the concepts were much more familiar to practicing IT architects; also of note is the fact that the architect group found many of the items much more useful than did the category A respondents (e.g. Principle, Goal); although Value was seen as less important. Assessments and Constraints were both well understood by architects but felt to be neutral or slightly unhelpful in terms of their inclusion in the models.

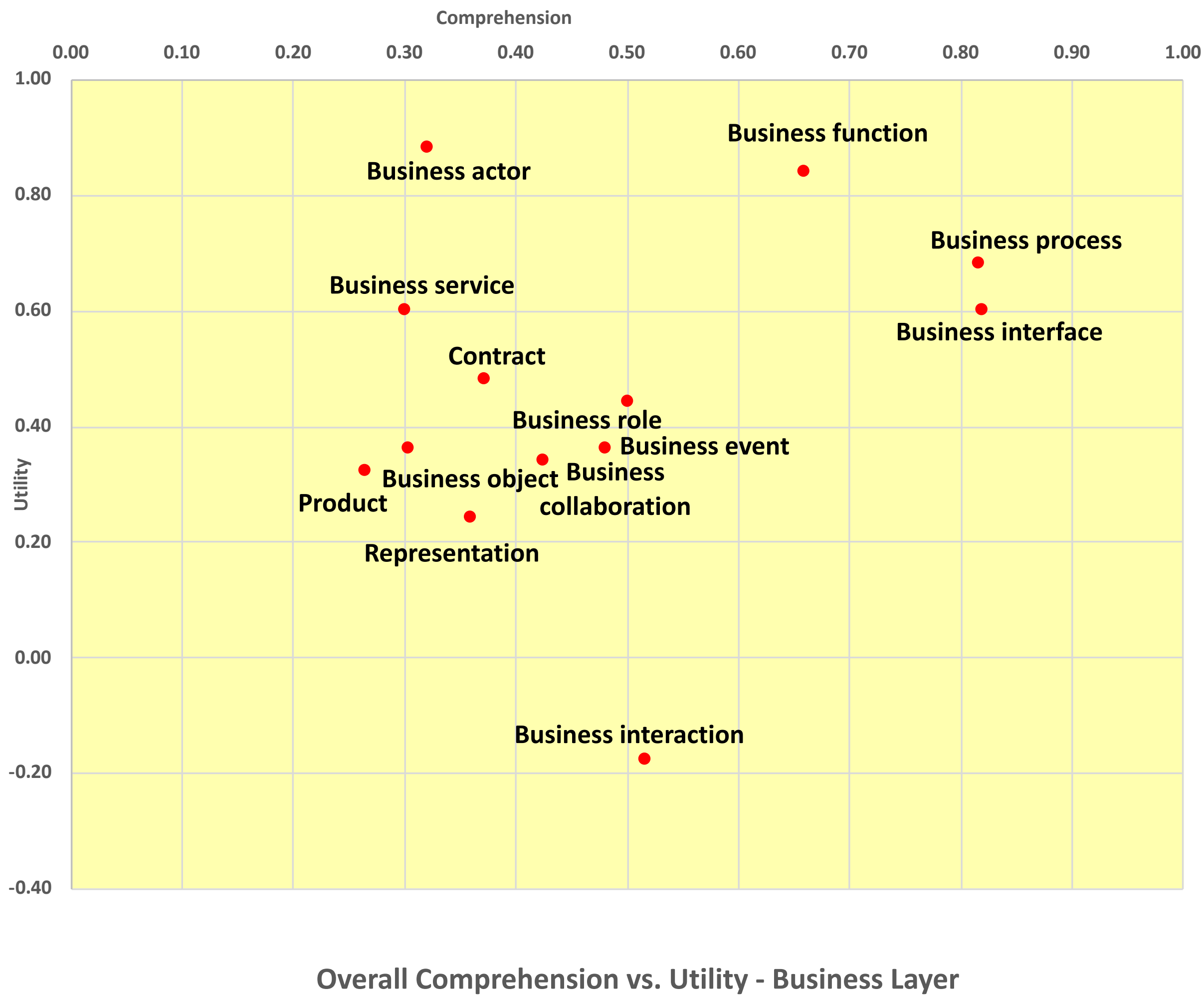
Both groups found the idea of "Meaning" hard to express and were neutral or negative about including items of this time in models.

7.7.4.2 Results for Business Layer

7.7.4.2.1 Overall Results

The results for the business layer across all respondents are:

Figure 77 Business – Overall Results

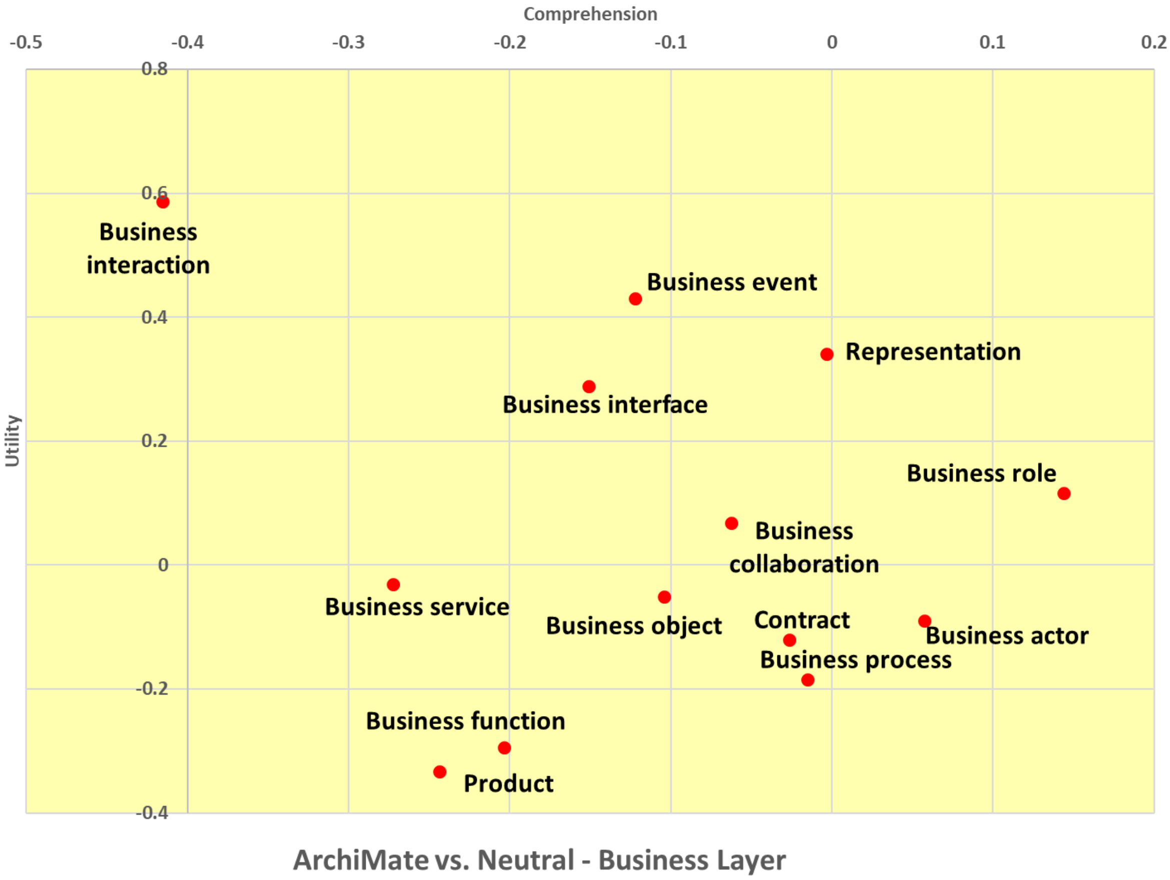


Clearly business processes, functions and interfaces were easily understood. Somewhat surprisingly, the business actor and business service concepts were not well understood, although they were seen as adding significant value, nonetheless. For the actor, this may well be because all the examples given in the ArchiSurance case study used organisational structures for actors, thus overlapping in meaning with business functions. For example, one respondent thought these were *"Departmental areas and the business itself – so relates to business hierarchy and the functions within that"*. By contrast, the actual ArchiMate definition has as a key component the fact that the entity *"is capable of performing behaviour"*.

7.7.4.2.2 Effect of ArchiMate Notation

If we subtract the results for the ArchiMate respondents from that of those using neutral (non-ArchiMate) models, then we obtain the following:

Figure 78 Business – ArchiMate vs. Neutral Notation



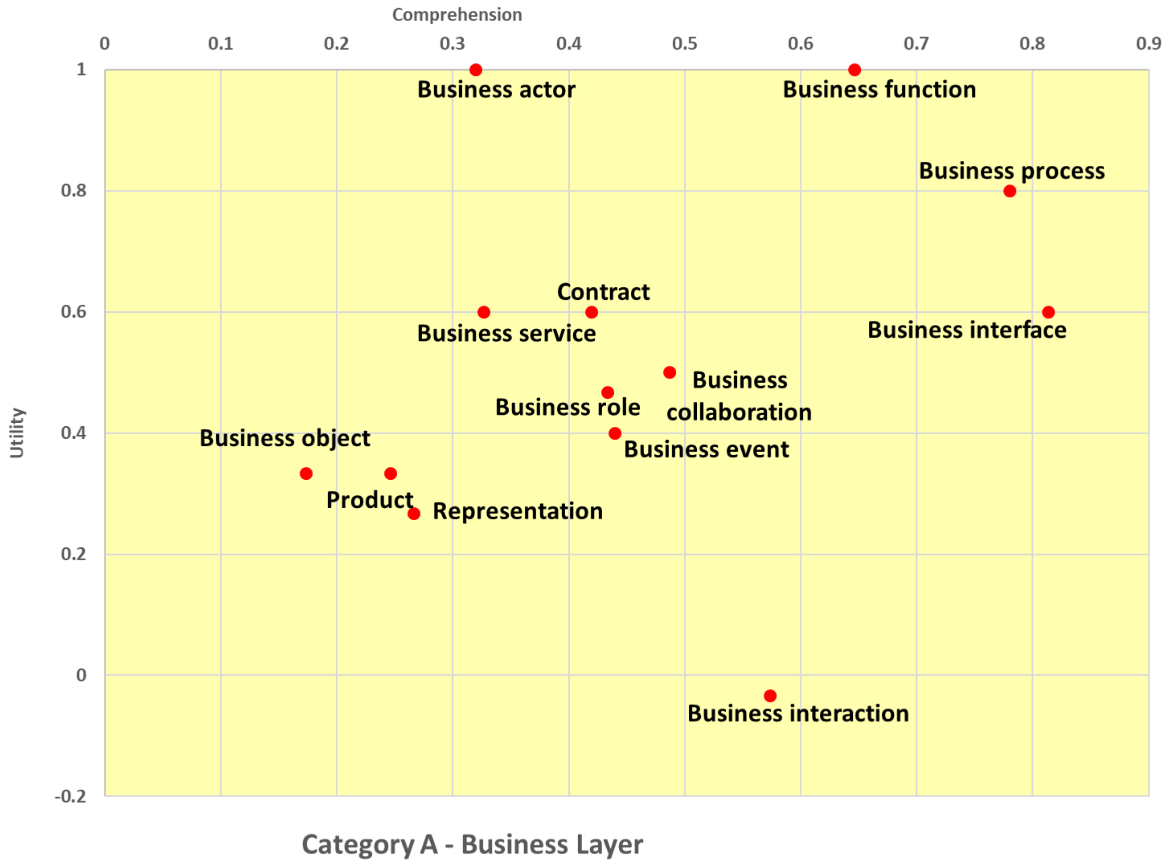
The marked effect whereby using ArchiMate notation clearly reduced the comprehension of the underlying concepts (as seen in Figure 74 above) is also apparent here. Of the 13 concepts, 11 (85%) were less well understood to some extent when expressed in ArchiMate notation.

Unlike the Motivation layer, however, the use of ArchiMate for many of the Business concepts also reduced their perceived usefulness.

7.7.4.2.3 With Respect to EA Role

The results for category A (social actors) are as follows:

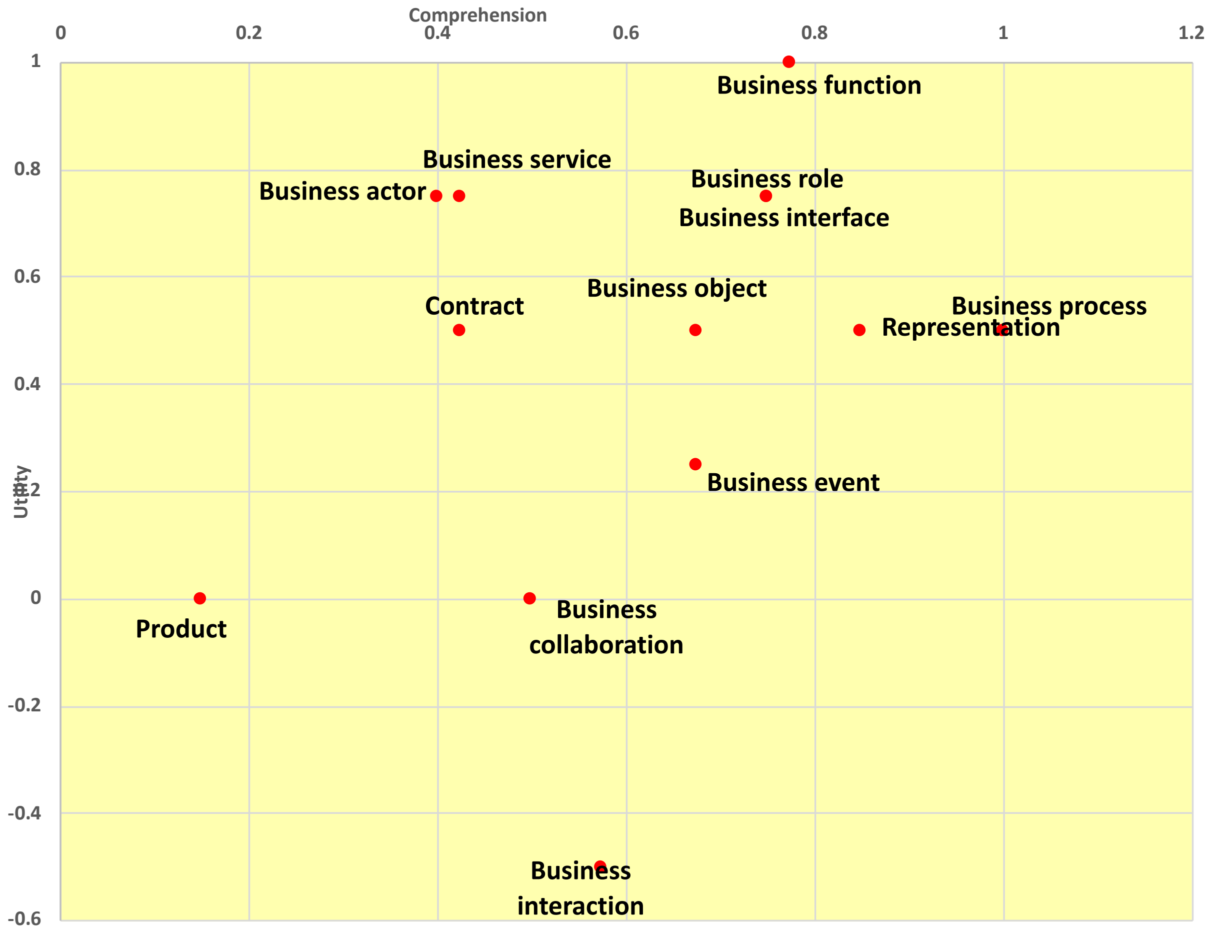
Figure 79 Business – Category A



The most useful concepts (even if misunderstood) to this category of respondents is clearly the business actor, function and process. This might be because the descriptive names seemed to make sense to the respondents, even if the categorisation did not (e.g. for business function, "*Departments within the company*", or "*Functions inside the company*".

For category B respondents we obtained the following results:

Figure 80 Business – Category B



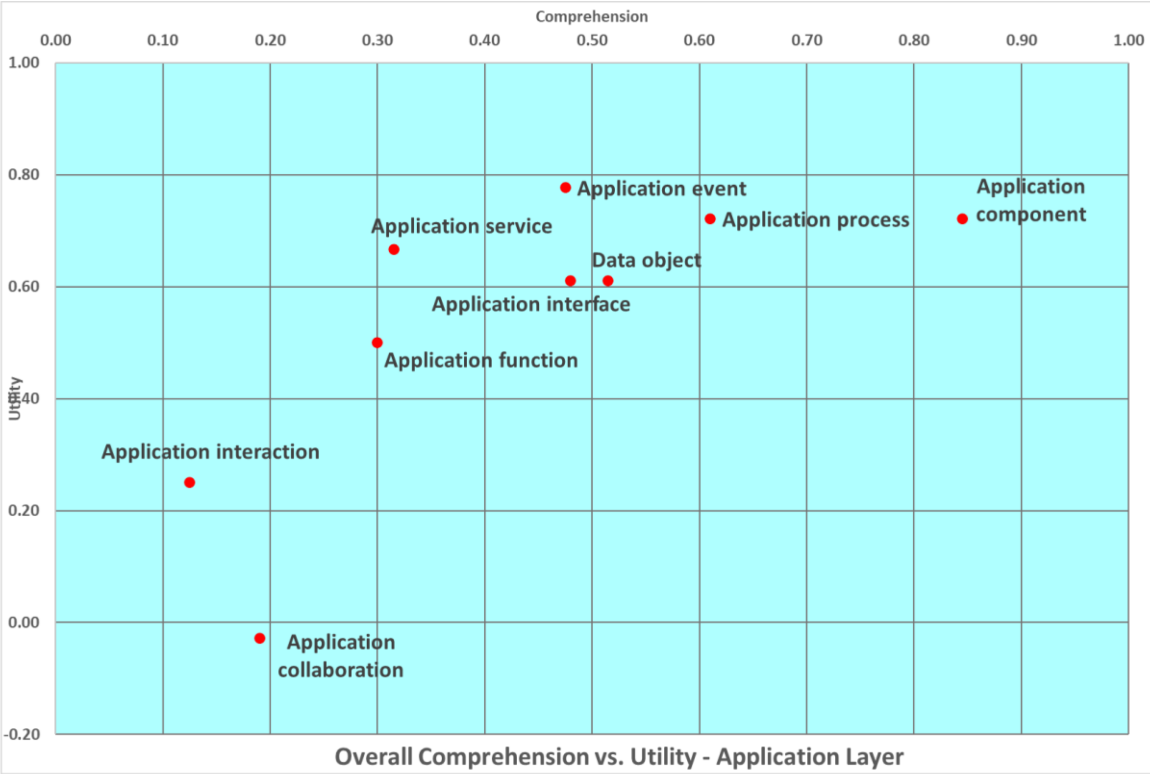
Category B - Business Layer

For the IT architects, we see a small increase in comprehension of the concepts (as compared to the overall population as shown in Figure 77 above); there is also a slight increase in the perceived utility of the concepts.

7.7.4.3 Results for Application Layer

7.7.4.3.1 Overall Results

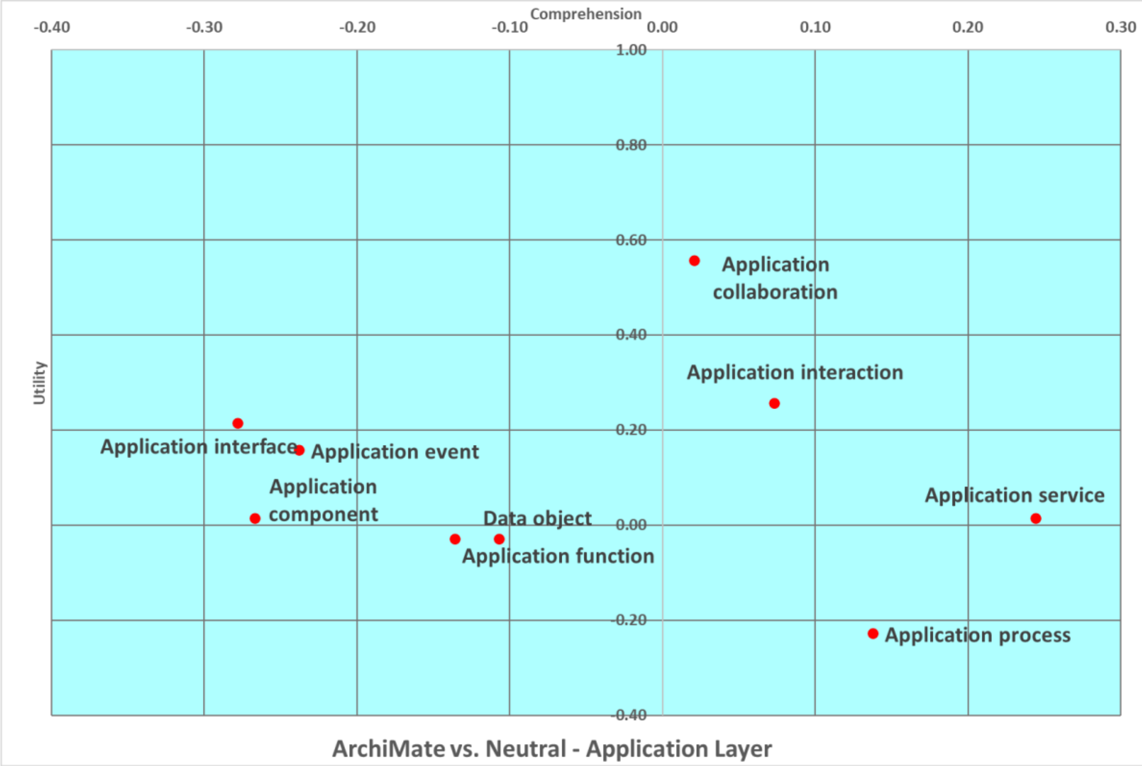
Figure 81 Application – Overall Results



Clearly the most understood concepts relate to application components and processes and the least well understood (and useful) are the application interactions and collaborations, although a limitation in the test design might have contributed to the latter (poor example in our case study of application collaboration and interaction).

7.7.4.3.2 Effect of ArchiMate Notation

Figure 82 Application - ArchiMate Notation vs. Neutral



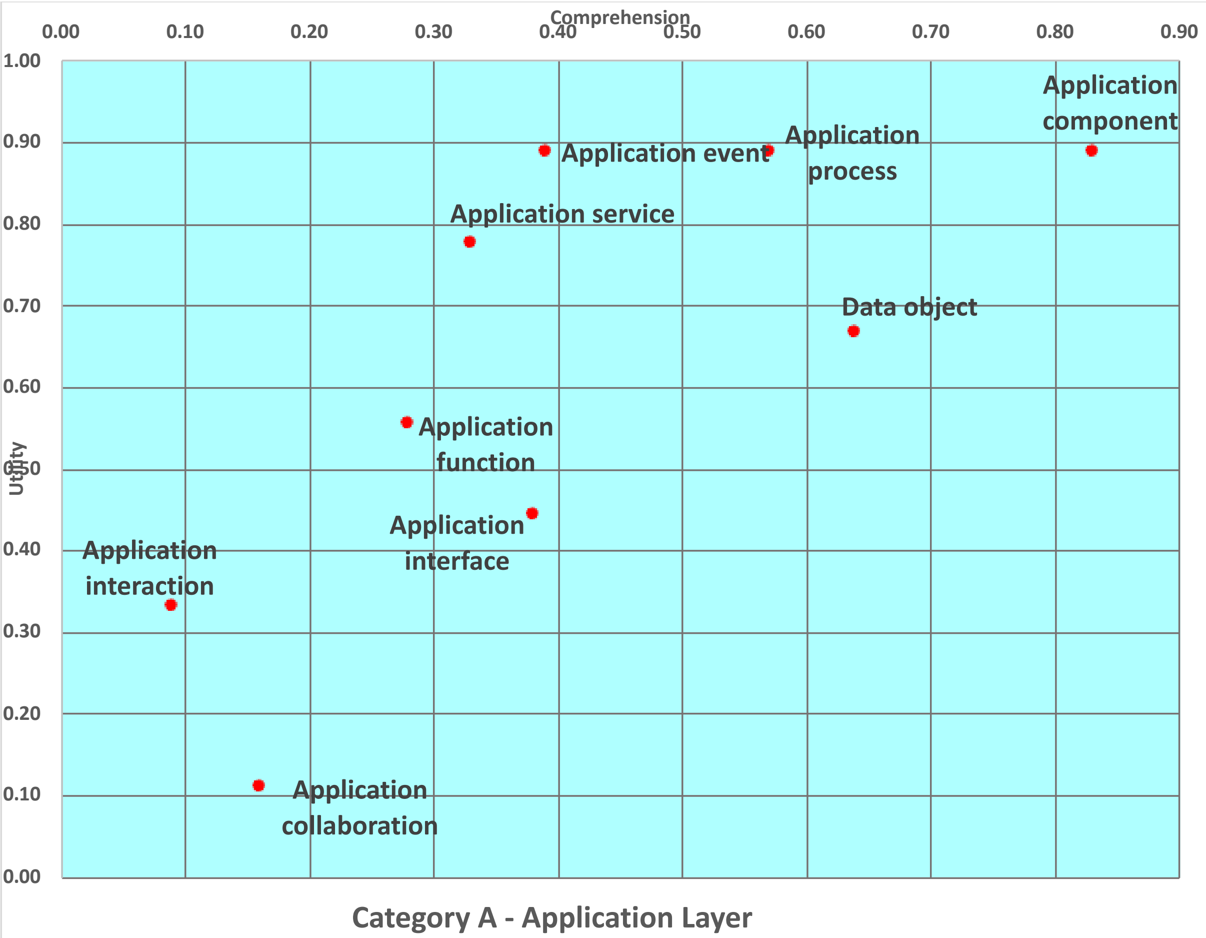
The use of ArchiMate notation seems broadly neutral in terms of overall effect on concept comprehension: Of the 9 concepts, 5 (56%) were less well understood to some extent when expressed in ArchiMate notation. This result is different to that obtained for the result of ArchiMate in the motivation and business layers. One possible explanation for this is that the respondents who felt comfortable answering questions about the application domain were more likely to be comfortable with technical, specialist concepts; whereas non-technical people might have elected not to answer questions in this area (application layer). This possible bias is not, however, notable when examining the job titles captured (see Appendix K below) during the interviews.

In terms of perceived usefulness, the overall most poorly understood concepts (application collaborations and interactions) benefit most by being expressed in ArchiMate notation.

7.7.4.3.3 With Respect to EA Role

For our social actors, in category A, the results are:

Figure 83 Application – Category A

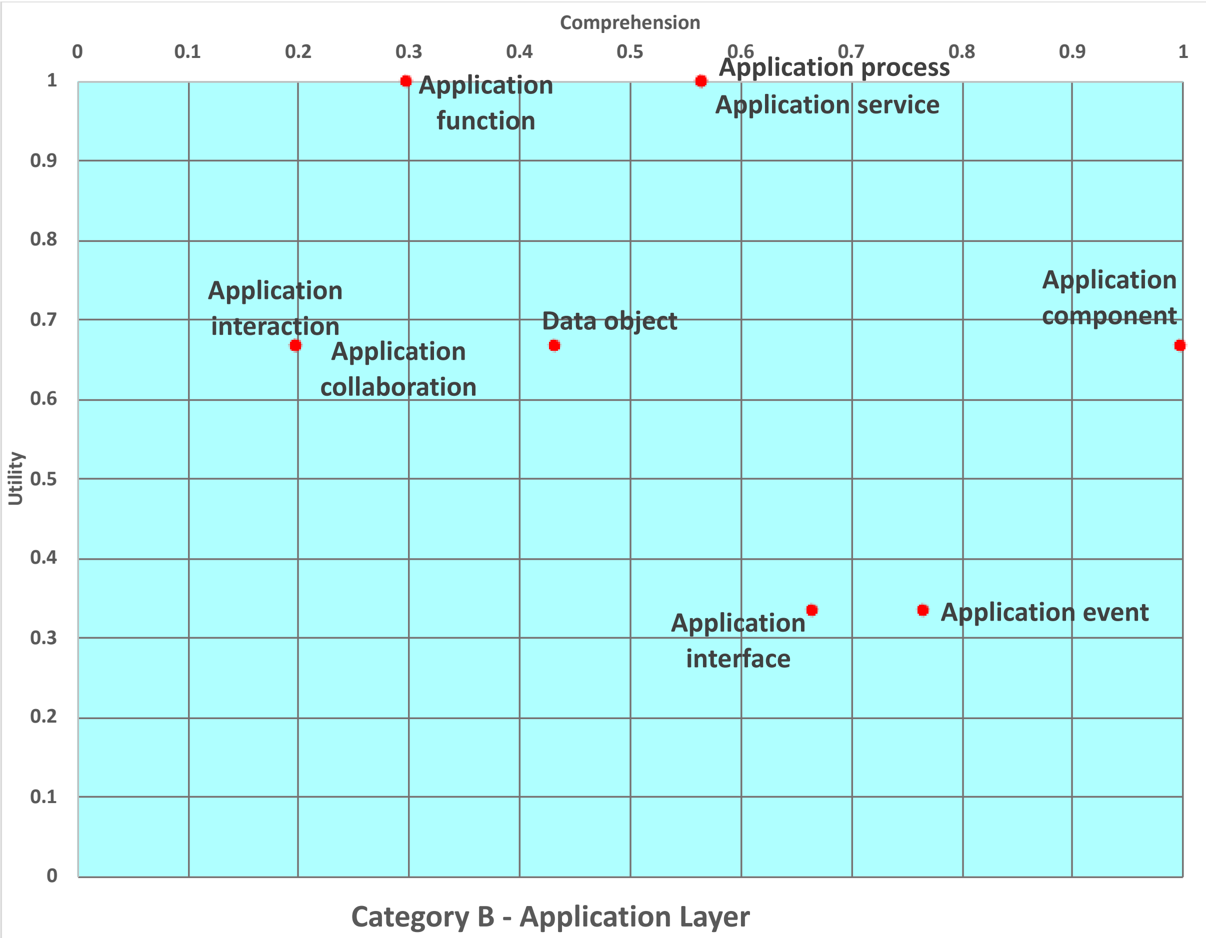


For these category A stakeholders, it is clear that a few of the concepts in this layer - application components, application processes and data objects in particular - are both well understood and useful.

As with the business layer, application interactions and collaborations scored poorly.

For category B – IT architects, practitioners – we obtained:

Figure 84 Application – Category B



When compared with the overall population of interviewees, there was a slightly stronger comprehension for some concepts (application component and application event) and the interaction and collaboration concepts, although still largely not understood, were viewed as more important.

This concludes the analysis and interpretation of the data, and hence the deliverable:

Deliverable 2.3. Analysis and interpretation of results of ArchiMate research

7.7.5 Observations from Analysis

Drawing together the results from the above graphs, the following would appear to be key results from the analysis:

7.7.5.1 Across all respondents

Table 36 Conceptual understanding for all respondents

Layer	Most well understood concepts	Most poorly understood concepts
Motivation	Stakeholder Goal Value	Meaning Requirement Principle
Business	Business Process Business Interface Business Function	Product Business Object Business Actor
Application	Application Component Application Process Data Object	Application Interaction Application Collaboration Application Function

Table 37 Perceived Usefulness for all respondents

Layer	Most useful concepts	Least useful concepts
Motivation	Driver Stakeholder Goal	Meaning Constraint Assessment
Business	Business Function Business Actor Business Process	Business Interaction Representation Product
Application	Application Event Application Component Application Process	Application Collaboration Application Interaction Application Function

Based upon the above results, when combined with the Effectiveness Grid that we presented in Figure 57 on page 211, we can make some provisional recommendations as to what to include and what to leave out of ArchiMate diagrams, if we want the diagrams to be meaningful and helpful to the broadest range of readers.

The items that fall in the lower left quadrant (not understood, not useful) would include Meaning, Product, Application Interaction, Application Collaboration and Application Function; so we would recommend that the use of these be avoided except when communicating with people trained in the use of ArchiMate.

The items that fall in the upper right quadrant (understood, useful) would include Stakeholder, Goal, Business Function, Business Process, Application Component and Application Process and so we would recommend that these be included in ArchiMate models.

7.7.5.2 For Category A respondents (non-technical stakeholders)

If we now repeat the previous analysis and only include those respondents that identified as category A (non-technical, more likely to be sponsoring architecture work than doing or following it), then we obtain the following results:

Table 38 Conceptual understanding for category A respondents

Layer	Most well understood concepts	Most poorly understood concepts
Motivation	Stakeholder Goal Value	Outcome Constraint Principle
Business	Business Process Business Interface Business Function	Product Business Object Representation
Application	Application Component Application Process Data Object	Application Interaction Application Collaboration Application Function

Table 39 Perceived Usefulness for category A respondents

Layer	Most useful concepts	Least useful concepts
Motivation	Driver Stakeholder Goal	Meaning Principle Requirement / Constraint (joint 3rd)
Business	Business Function Business Actor Business Process	Business Interaction Representation Product / Business Object (joint 3rd)
Application	Application Event Application Component Application Process	Application Collaboration Application Interaction Application Function

Using the same Effectiveness Grid (Figure 57), we can now make provisional recommendations as before for ArchiMate models presented to category A users, which includes those that are likely to be sponsoring our work, and who are not likely to have experience of Enterprise Architecture frameworks.

Starting again with the items that fall in the lower left quadrant, we find from the three layers that we have Principle and Constraint; Representation, Product and Business Object; and Application Interaction, Application Collaboration and Application Function. These are not recommended therefore for inclusion in models presented to such stakeholders.

Conversely, looking now at the top right quadrant, we have: Stakeholder and Goal; Business Function and Business Process; and Application Component and Application Process. These are recommended for inclusion in such models.

7.7.5.3 For Category B respondents (IT Architects)

Our final category of summary analysis will be for those in IT Architecture roles such as Enterprise Architects. The respondents have had no training in ArchiMate (indeed, the majority have had no training in *any* architecture framework – see section 7.7.1 above) , and so these results will reflect their general experience of IT architecture work. Caution should be taken here because of the low volume of respondents in category B.

Table 40 Conceptual understanding for category B respondents

Layer	Most well understood concepts	Most poorly understood concepts
Motivation	Stakeholder Goal Outcome	Meaning Driver Requirement
Business	Business Process Representation Business Function	Product Business Actor Business Service / Contract (joint 3rd)
Application	Application Component Application Event Application Interface	Application Interaction Application Collaboration Application Function

Table 41 Perceived Usefulness for category B respondents

Layer	Most useful concepts	Least useful concepts
Motivation	Driver Stakeholder Goal / Value	Constraint Requirement Meaning / Assessment (joint 3rd)
Business	Business Function Business Interface / Business Service / Business Role / Business Actor (joint 2nd)	Business Interaction Product Business Collaboration
Application	Application Function / Application Service / Application Process (joint 1st)	Application Event Application Interface Application Interaction / Application Collaboration / Data Object / Application Component (joint 3rd)

Our category B respondents are of course the ones that we expect to be creating these architecture models. Thus, as well as giving an indication as to how best to communicate with IT architects, if we look at the absolute values of the comprehension ratings in particular (rather than the top or lowest three), we gain more insight. It is perhaps surprising that although the concept of Meaning (which is specific to ArchiMate – no other architecture framework has this concept) is poorly understood (comprehension of 0.25), the concepts of Driver and Requirement also score relatively poorly, having comprehension ratings of approximately 0.5. These are common in architecture frameworks, and used commonly when describing solutions, and so this is perhaps a surprising result. This might point to a training need amongst this community.

Again using our Effectiveness Grid, we can see that the concepts that are best both understood and valued by this subset of respondents are: Stakeholder and Goal; and Business Function and Business Process. None of the top three understood concepts in the Application, for category B respondents, overlap with any of the top-three useful concepts for the same respondents. Thus, we can specifically recommend unambiguously only the above four concepts for communicating with this type of respondent.

In terms of the lower-left quadrant, we see that the concepts to unambiguously avoid using in our models would be: Meaning and Requirement; Product; and Application Interaction and Application Collaboration.

7.7.5.4 Absolute vs. Relative Ratings

The above analysis that takes only the top three and bottom three concepts is somewhat simplistic, as it ignores the possibility of many concepts being well or poorly understood. Nevertheless, it gives an indication of where communication is likely to work well (or not). A fruitful area for further research would be an extension of this analysis whereby one sets two thresholds (say, 0.3 and 0.7), and puts anything above 0.7 in the 'high' category, anything below 0.3 in the 'low' category, and omits things in between. This is however outside the scope of this particular work.

7.7.5.5 Impact of ArchiMate notation

In some cases the ArchiMate notation is misleading to some respondents (e.g. business roles), reducing their comprehension scores significantly when compared to respondents that were given alternative notation (basically, colours instead of icons); this applies in particular to Meaning, Value, Stakeholder and Constraint from the Motivation layer; to Business Interaction and Business Service from the Business layer, and to Application Interface from the Application layer.

If we do a statistical analysis of the differences in scores between ArchiMate and non-ArchiMate notation, calculating the average and standard deviations in each case (see section 3.3.5 above) using the Excel AVERAGE and STDEV.S functions, then we see the following:

Table 42 Statistical Analysis of Impact of using ArchiMate Notation

ArchiMate Layer	Effect on Comprehension		Effect on Utility	
	Average	Standard Deviation	Average	Standard Deviation
Motivation	-0.13	0.17	0.26	0.37
Business	-0.11	0.15	0.06	0.28
Application	-0.06	0.19	0.18	0.30
Overall	-0.10	0.16	0.16	0.32

Without any further statistical treatment, or probabilistic interpretation, then the above figures suggest a trend along the following lines:

(a) Across all ArchiMate layers, there is a 10% reduction in understanding of the underlying ArchiMate concepts where the ArchiMate notation was used, compared to where it was not used. This effect is most noticeable in the Motivation layer and least noticeable in the Application layer.

(b) Across all ArchiMate layers, there is a 16% increase in perceived utility of the underlying ArchiMate concepts where the ArchiMate notation was used, compared to where it was not used. This effect is most noticeable in the Motivation layer and least noticeable in the Business layer.

(c) For the effect on comprehension, the standard deviation is slightly greater than the average, and so there will be a few exceptions where comprehension is increased with the use of ArchiMate notation, on average however the effect is deleterious.

(d) For the effect on utility, the standard deviation is significantly larger than the average, which may call into question the significance of the effect whereby utility seems to be increased using ArchiMate notation (the “error bars” are so wide).

To better understand the statistical significance of these figures, it is necessary to employ additional techniques, and in particular to to evaluate the possibility that these results are unlikely to have occurred under the null hypothesis (i.e., they are samples derived from the same population). The detailed statistical evaluation can be found in Appendix M below. It is customary to state the "p-values" giving these possibilities, and to use a threshold p-value of 0.05.

The p-values obtained for the likelihood that our difference in comprehension and utility scores between ArchiMate and non-ArchiMate views (diagrams) is down to chance are summarised below:

Table 43 p-values for Differential Ratings

Comprehension p-value	Utility p-value
0.04612	0.105594

The comprehension p-value is under the 0.05 threshold; the utility p-value is not. We can therefore legitimately infer that that the difference in Comprehension between the ArchiMate and non-ArchiMate is not down to chance: the result is statistically significant.

The same does not apply to the Utility ratings; the p-value is > 0.05, and so we cannot infer that the use of ArchiMate notation has a real effect on the perceived usefulness of the concepts.

7.8 Research Results

7.8.1 Theoretical Framework

We have successfully constructed a mathematical framework and methodology that enables us to assess empirically the effectiveness and usefulness of the constituent parts of an entity metamodel forming part of an architecture framework. We have not however so far found a way of working into our framework any kind of relationship with individual backgrounds (training, job role etc.).

7.8.2 Practically Useful

We demonstrated, by using the framework and methodology upon a particular architecture framework (ArchiMate 3.0 [39]), that the framework and methodology can be used in practice to gain insights as to which parts of the modelling language are more effective than others.

7.8.3 Use of ArchiMate Notation

As we can see from the results of the ArchiMate vs. Neutral graphs (Figure 74, Figure 78, Figure 82), for two of the three layers, there was a decrease in the ability of the respondents to express the meaning of concepts where ArchiMate notation was used, when compared to the use of colours on plain boxes (what we have referred to as 'neutral' notation). Overall, when the results were taken across all ArchiMate layers, there was a statistically significant result showing a reduction in the Comprehension scores when ArchiMate notation was used (see 7.7.5.5 above); and for the perceived utility (usefulness) of concepts, there was conversely a slight increase in utility scores, although the results were not strong enough to be statistically significant.

We have already mentioned some of the comments (in the qualitative data captured) about some of the icons being misleading, however that does not necessarily account for this difficulty, and thus this would be a useful area for further research. Another pair of related questions that we would like to examine in a future survey, is to ask respondents whether they would prefer to be given a model in ArchiMate or using something else like colours, and the reason why.

Anecdotally, we have discussed this with non-technical colleagues, and the responses so far indicate that it is the amount of time taken to discern which icon is being used is off-putting, whereas the use of colours (or position on the page) is much more immediately obvious, with less effort required for the reader. Obviously more rigorous research would be required to be more definitive about this.

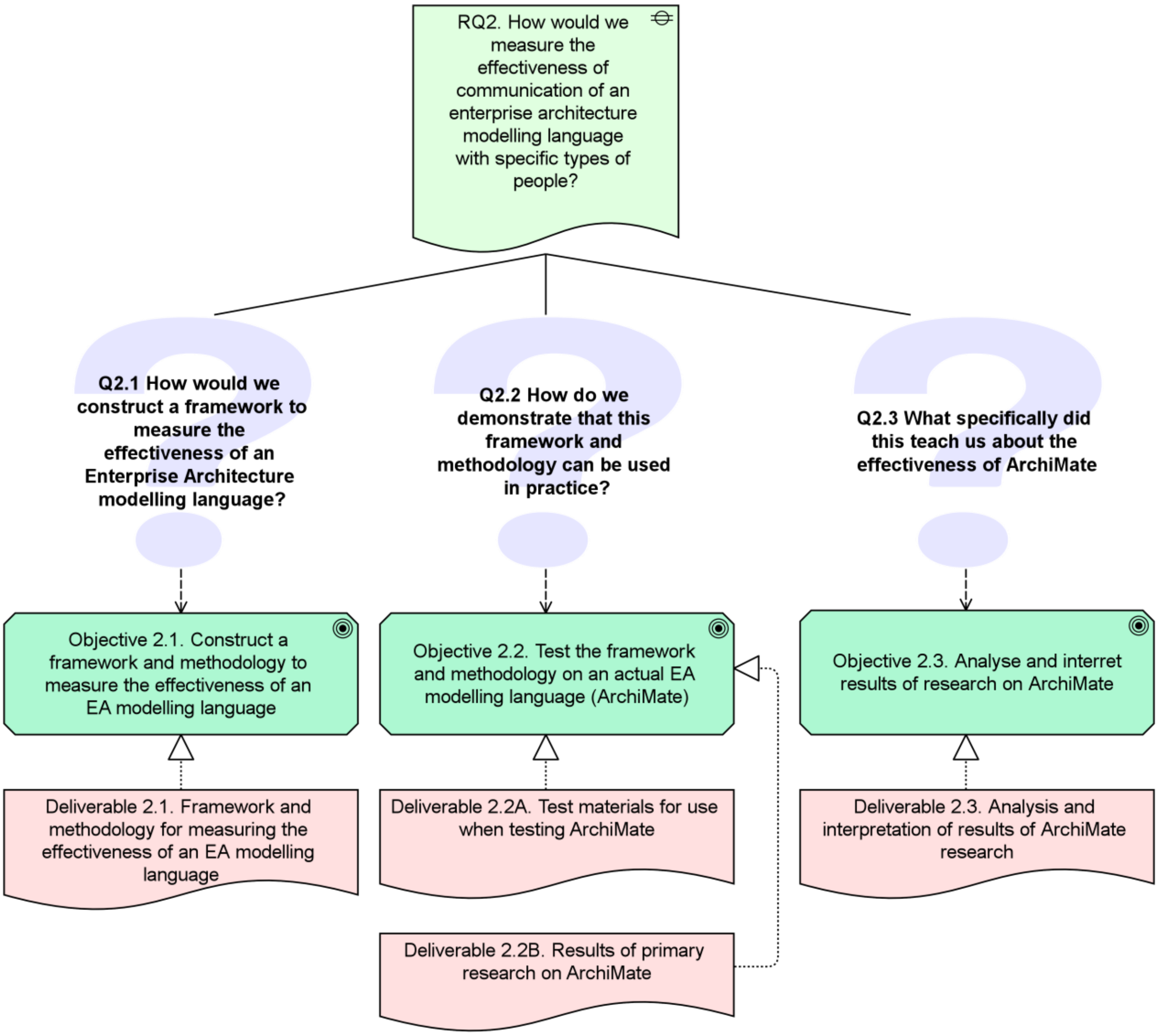
7.8.4 Recapitulation of Research Question

Our research question was RQ2, *How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?*

We have demonstrated this by constructing a theoretical framework with two separate dimensions (comprehension and utility), and then demonstrating that both of these dimensions work in practice. In the course of doing this, we have highlighted some parts of the ArchiMate language that are not well understood by those not trained in the language, as well as some potential issues with the use of ArchiMate notation for those not trained in its use.

We have thus completed the following:

Figure 85 Completion of RQ2



7.9 Reflections

7.9.1 No mixed-layer models

The mathematical framework we constructed in 6.4 above was designed to handle the same concept appearing on multiple models. The ArchiMate language encourages models spanning multiple layers. The results might have been different had we used more models that spanned layers, for example:

(a) A motivation model

(b) A motivation + business model (for example, showing which goals affect which processes)

(c) A business model

(d) A business + application model (for example, showing which processes require which applications)

(e) An application model

This might have yielded more representative results, however the interviews would have taken longer to complete, too long in our judgment for a PhD project.

A useful piece of future research to be carried out would be to build on what we have achieved by modifying our test material to (a) base it upon ArchiMate 3.1 instead of ArchiMate 3.0 (a new version of the standard was released whilst this research was in progress, with altered definitions). (b) add additional test models that mixed elements between ArchiMate layers, (c) bring into scope more of the ArchiMate layers (for example, adding the Strategy and Technology layers), and (d) introduce more examples of some of the rarer concepts such as collaborations and interactions (as discussed in 7.4.1 above).

7.9.2 No analysis of job role clusters

For lack of time, and suitable background in the literature, we did not attempt to group results by clusters of similar job roles. If we had been able to do this, we would have been able to be more specific as to what kind of concepts could be used when communicating with what kinds of stakeholders.

7.9.3 Poor examples of some concepts

As previously discussed in 7.4.1 above, we struggled to create multiple examples of business and application interactions and collaborations, and this may have affected the comprehension and utility scores for those concepts. In hindsight we might have sought help from others more familiar with their use to construct more examples of these.

7.9.4 Rigour in scoring comprehension

In hindsight, when scoring the comprehension responses from the interviewees, it could have been improved by laying out in advance, what we felt the *central beliefs* were regarding each of the concepts (using the moderate view of objectual understanding’s factivity). This might have enabled additional rigour in the assignment of the ratings.

7.9.5 No analysis of relation symbols

Although this research has focused on the entities (nodes on the models) within the ArchiMate framework, as we have made clear, the relations (edges on the models) are also symbols with specific meaning. The mathematical framework that we constructed for entities would work just as well for the relations of various types, and this perhaps an area for future research, to help us understand which (if any) of the ‘customised arrows’ provided by ArchiMate should be used when communicating with our stakeholders.

7.9.6 Use of ArchiMate Notation

Depending on the layer chosen (Motivation, Business or Application), we found that overall, the perceived comprehension of the underlying concepts was poorer (or at best, similar) when using ArchiMate notation as compared to different coloured rectangles. This effect was most marked in the Motivation layer. Overall, there was a statistically significant reduction in comprehension when ArchiMate notation was used.

Conversely, again depending on the layer chosen, the perceived utility (usefulness) of the underlying concepts was better (or similar) when using ArchiMate notation as compared to different coloured rectangles, although unlike the result for comprehension, the utility result is not statistically significant.

Putting these together, the use of ArchiMate notation may (not statistically significant) slightly increase people’s perception of the importance of some of the ideas on the views, and at the same time is likely to (statistically significant) reduce their understanding of those same ideas – or at least, their ability to articulate their understanding of those ideas. In other words, the data may seem to be more important, but they cannot articulate the meaning of the data.

Anecdotally, business stakeholders with whom we have had a discussion on the subject appear to have a strong aversion to “technical diagrams”, stronger than would be accounted for merely by the 10% reduction in apparent understanding of the ArchiMate concepts. Thus, it is common practice now with IT architects to avoid using technical (specialised) modelling languages when communicating with non-technical stakeholders, as we discussed in the introduction.

The reasons why would be a suitable area for future research but for now, we would suggest that the case for avoiding the use of ArchiMate notation with some stakeholders (as we discuss in our final research question, specifically in sections 8.3.3 below and again in section 9.2.2) is stronger than it might first appear.

Chapter 8. EA Model Evolution

The research contained in sections Chapter 4 and Chapter 6 covered two related but distinct areas: how do we measure the effectiveness of an EA modelling language (RQ2) and how do we measure the quality of a set of EA models (RQ1).

We now move to consider the tailoring of EA models to meet the needs of various stakeholders.

Our final question, RQ3, is *Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?*

The results from the previous two chapters will have a bearing on this, so we start with a summary of the key learnings from those chapters.

8.1 Summary of learning from previous chapters

From Chapter 4 we learnt that we can measure the quality of a set of models by considering three metrics applied across those models:

(a) Are they in the correct syntax (comply with our agreed language)?
(b) Do they actually exist?
(c) Do they correspond to reality (are they telling the truth)?

From Chapter 6 we learnt that some of ArchiMate concepts are poorly understand by people not familiar with the language. We also learnt that the use of ArchiMate notation can at times be a hindrance to understanding the meaning of a diagram.

8.2 Rethinking our language definition

In section 4.3.3 above we define a quality metric – syntactical quality for our set of architectural models thus:

$$Q_S = \frac{|\, M \cap L \,|}{|\, M \,|} \qquad (1)$$

Here, M is the set of all our models, and L is the set of models that have the correct syntax, and when modelling using a defined language that includes a metamodel, that would include restricting ourselves to the concepts within that metamodel.

If we have decided to use a modelling language such as ArchiMate, how should our metrics for measuring the quality of those models be adjusted given what we have found about the comprehension and utility of three particular sets of symbols within that language?

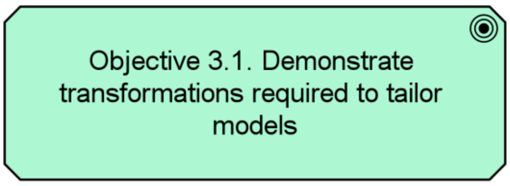
We may find that we need to adjust the definition of our language $\mathcal{L}$. This might be enterprise-wide – across all models. It might also be that we need to adjust the scope of $\mathcal{L}$ just for specific stakeholders who do not wish to see certain concepts on models shown to them.

Our language definition may also involve the graphical notation involved, as is the case with our ArchiMate research, and again this will represent a change to our language for specific stakeholders.

For example, if they prefer diagrams to use colours instead of ArchiMate icons, then if we produce a standard ArchiMate diagram for them, although those diagrams might be inside $\mathcal{L}$ as far as the authors (e.g. Enterprise Architects) are considered, they might be outside $\mathcal{L}$ as far as the (say) business stakeholders are concerned. These ideas are discussed more fully below.

8.3 Model Evolution

Having established that we may wish to change some of our models, to increase both quality and understanding with our stakeholders, we now combine this with our understanding of the purpose of models, to show some of the common transformations that we expect to have to apply to those models. This will therefore address our next objective:



8.3.1 From Complex to Simple

We may now consider whether we actually want to include all of the possible elements from our architecture language(s) in our models. It may, for example, be 'valid ArchiMate' to include the concept of Meaning (from the Motivation layer) in our models, but if we have concluded that this particular concept is neither useful nor understandable to those reading our models, then we may wish to consider not using this concept.

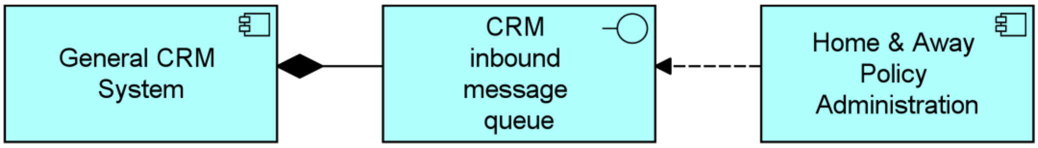
There are of course a number of reasons why we may wish to include, or exclude, particular concepts from our chosen modelling language(s), and these are not limited to a consideration of the ability of a particular group of stakeholders to understand that concept. As an example, consider the ArchiMate concept of an Application Interface, defined in ArchiMate 3.1 thus:

An application interface represents a point of access where application services are made available to a user, another application component, or a node.

Now it is entirely reasonable that a business stakeholder with no specialist knowledge of Information Technology might not be interested in details such as these, and so models produced to satisfy the needs of such stakeholders might be best produced omitting some of these technical concepts. One might question whether the business user *should* be interested in such details but ultimately one of our aims for modelling is communication, and if parts of our models do not address any concerns of the intended readers, then by including them we risk making it harder for those stakeholders to understand the parts that they *should* be concerned with. We want our models to be as simple as possible in order to focus attention on the elements that really need their attention.

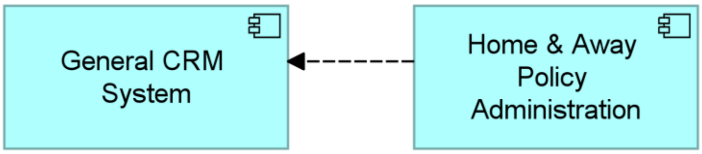
Therefore, rather than showing details of two applications interacting via an application interface, as per the following example:

Figure 86 Two applications interacting via an application interface



… they might be content to see two applications communicating directly, for example as shown below (taken again from the ArchiSurance [62] case study), and leaving out the intervening application interface:

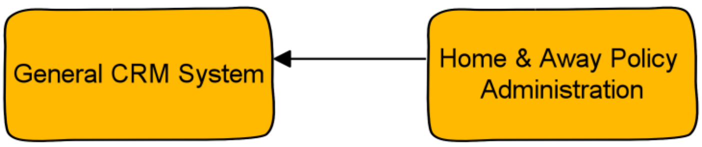
Figure 87 Two applications interacting directly (ArchiMate)



What we have done here is extended the relation originating from the Home & Away Policy Administration system (which happens to be an ArchiMate 'Flow' relation) so that it terminates on the other application (General CRM System). This is an example of an ArchiMate 'derived' relation that we will discuss in more detail in section 8.4.1 below; they are a standard mechanism to allow us to 'leave out' particular elements from our models whilst leaving elements either side of them still connected. In this example, we removed the application interface 'CRM inbound message queue' but left the two systems still connected (by the derived relation).

Indeed, our category A business stakeholders, who find the notation confusing, may prefer not to see this kind of technical notation at all, preferring something that uses colours rather than icons to separate class of objects, and also using simpler line types, such as the following:

Figure 88 Two applications interacting directly (neutral)

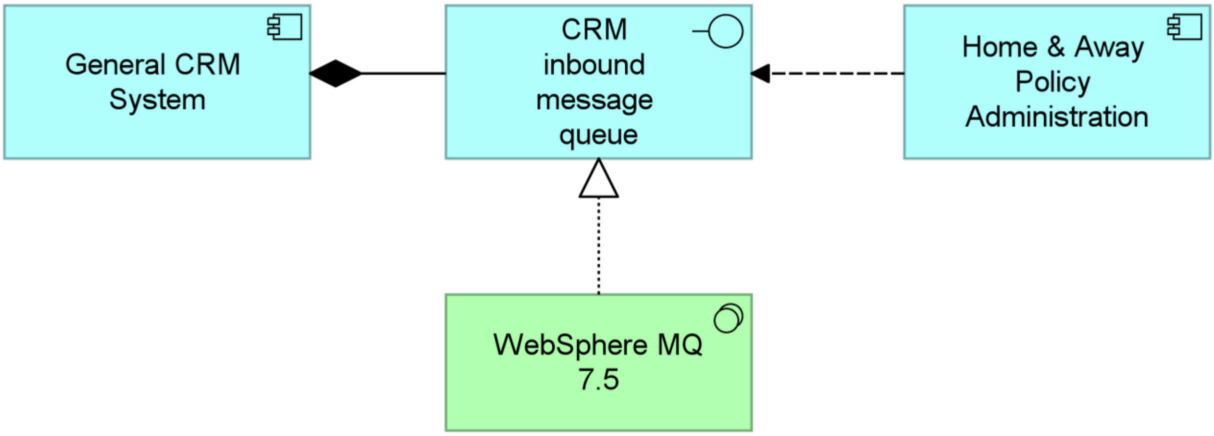


Our test models (see Appendix J below) already contain many examples of these different relations, however we have not drawn attention to them so far as our focus was on the different element types instead.

However, different stakeholders, in particular our category B users (IT architects) may well wish to see details of such interfaces, in order to make design choices both tactical (e.g. what are the ways in which application A could choose to get information from application B?) and strategic (e.g. what percentage of our applications have interfaces based upon a particular kind of middleware, and what licensing and support costs could be avoided if we consolidated onto a different integration technology?).

Thus, models for our IT architects may be more complex than Figure 86, for example:

Figure 89 Two applications interacting via a middleware-based interface



[22]

Both of them are 'correct ArchiMate' but a more appropriate metric may be whether they are appropriate for the stakeholders for whom they are intended. For stakeholders that are interested in changing the applications or middleware, these technical details are essential but for non-technical stakeholders, they are not, and may just serve to confuse.

This has an impact on two of our metrics for the quality of a set of EA models. Recall that our triad of metrics for model quality are broadly as follows:

If we are able to determine how many models should exist, then of all those models, those that actually exist

[22] WebSphere MQ (Message Queue) is a middleware product produced by IBM

> Models in the region $\mathcal{M}$ represent the subset of models that should exist, that do actually exist. Models outside of $\mathcal{M}$ represent those models that should exist, but do not.

We can also separate our present or missing models by whether or not they comply with our language:

> Models in the region $\mathcal{L}$ represent the subset of models that comply with our specified language(s). Models outside of $\mathcal{L}$ represent those models that do not.

Our final metric (domain $\mathcal{D}$) is not directly affected by this discussion.

Now when we consider the models that should exist (say, for a particular business service), and we consider the various views of this business service required to satisfy the needs of our stakeholders, we can see that our definition of language may change in two ways for some of the stakeholders.

Firstly, our vocabulary (e.g. set of nouns) may be restricted for some stakeholders, and so a model that may be acceptable (within the region $\mathcal{L}$) for one kind of stakeholder may be outside that region for a different stakeholder.

Secondly, the representation (visual appearance) of those terms on the diagram will change for different stakeholders and thus, a representation of a specific concept may fall within our language for one stakeholder

There are specific rules included in the ArchiMate framework to enable a more complex diagram to be collapsed down to a simpler diagram; these normative rules rely upon the ‘strength’ of the different relationship types.

The rules for creating these ‘derived relations’ are somewhat simpler in ArchiMate 3.0, being based upon a simple table of seven relationship strengths, from influence (being the weakest) to composition (being the strongest) [214]. The rules are more complex in ArchiMate 3.1, as there are several smaller tables for different groups of relations [215], the subject of an entire appendix in the later specification.

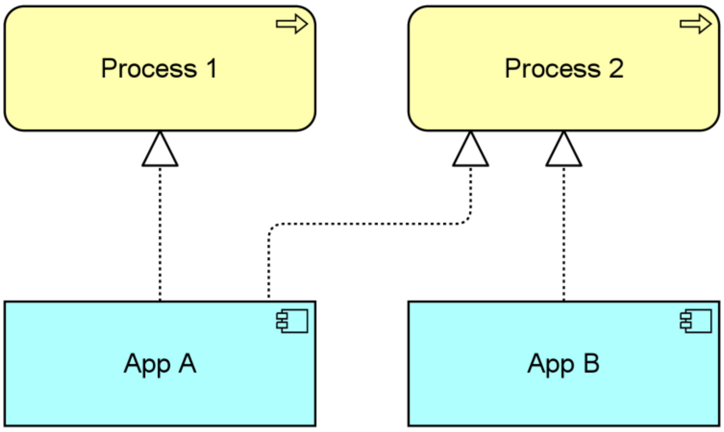
Thus, if we have the diagram shown in Figure 89 above, then there is a deterministic way of reducing the elements shown, for example to make it more suitable for a less technical stakeholder. In this example, the composition link (filled diamond) is “stronger” than the flow relation (dashed line), and so according to the rules of ArchiMate, we can derive the flow relation between “Home and Away Policy Relation” and “General CRM System”, such as we see in Figure 87 above. Obviously, it is not possible to start from the simpler diagram and then derive from that, the more complex diagram. This would suggest therefore that when modelling, we first construct the more complex diagrams (perhaps using elements from our language that are not suitable for all stakeholders), and then derive simpler diagrams from those, perhaps at the same time changing the visual appearance away from the default ArchiMate notation.

8.3.2 From Simple to Complex

In another sense, our models will have to go from simple to complex. This is because different roles involved in producing models may produce complementary aspects that when added together, provide a composite picture that is more complex than any individual piece.

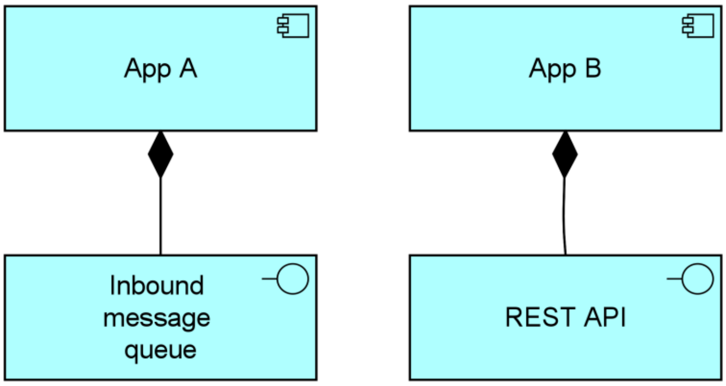
For example, a business analyst may produce a business process model and make reference, in that model, to the applications used at various stages in the process, which in ArchiMate might appear as:

Figure 90 Processes realised using Applications



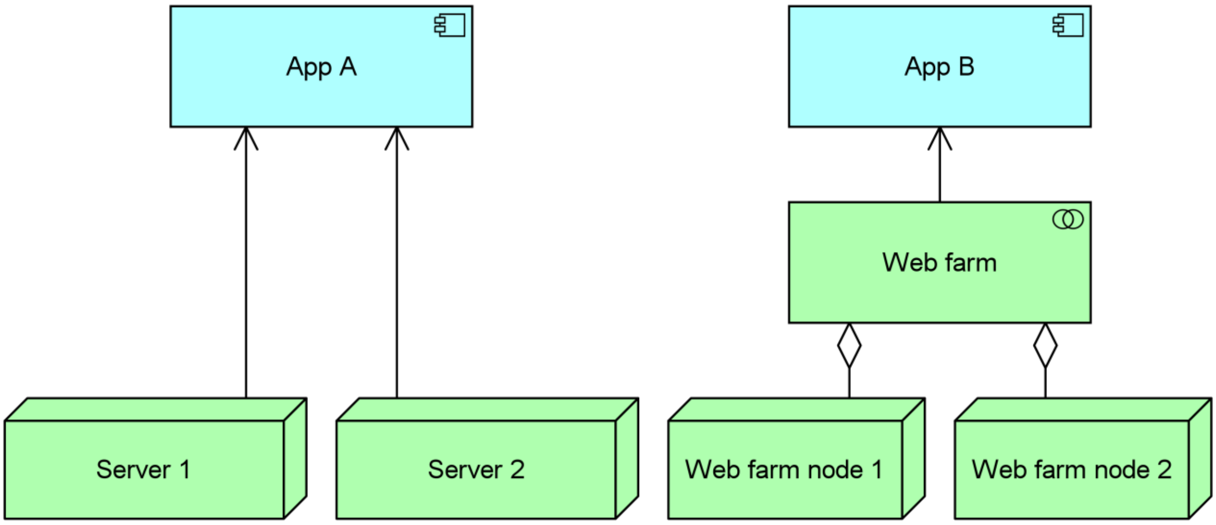
Independently, a technical architect might introduce details of the interfaces provided by the applications, that enable applications to communicate with each other:

Figure 91 Applications and Interfaces



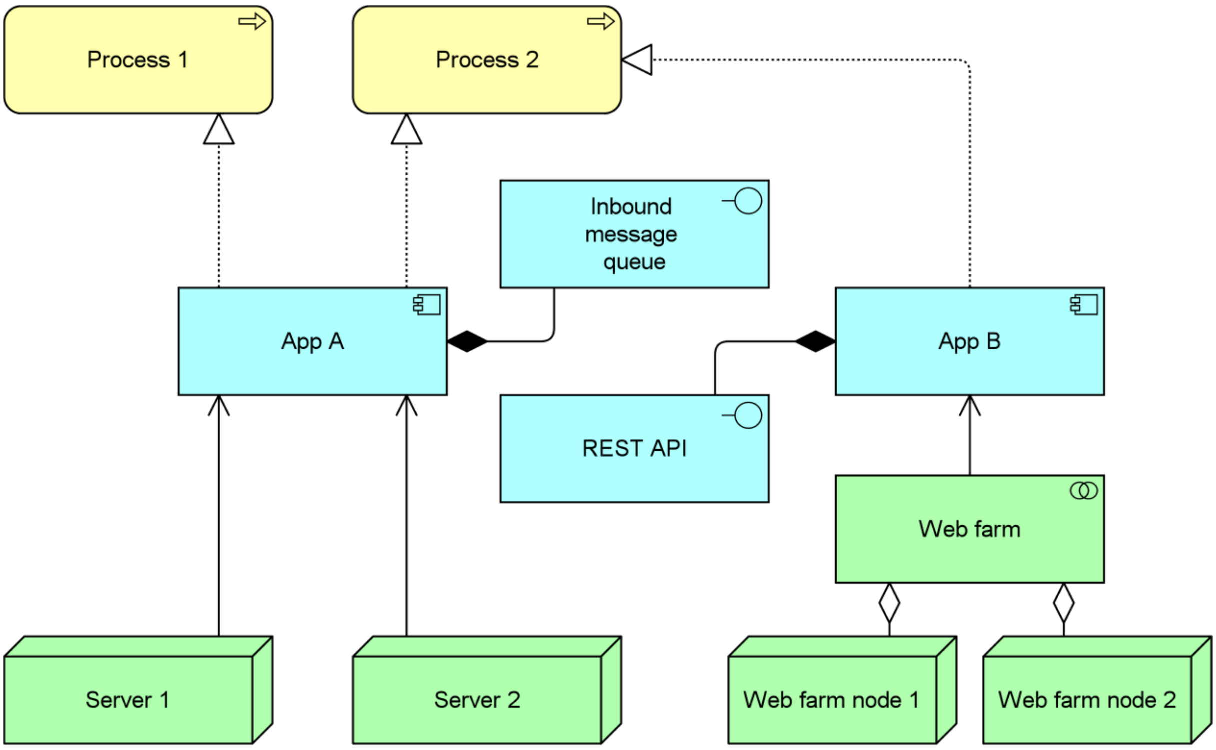
Finally, someone concerned with providing infrastructure on which to run the applications might suggest a design for the infrastructure.

Figure 92 Application Infrastructure



Taken individually these are quite simple models but put together there is obviously more complexity:

Figure 93 Application Usage and Structure



The data shown here is identical to the data in the previous three diagrams. Indeed, in the language employed in our modelling language quality research (see Figure 139 below), there is a correspondence between every node and edge on the overall diagram and the nodes and edges in the partial (constituent) diagrams. Borrowing the term from [148] we might use the term 'coherence' to describe the situation where the information from these various sources is consistent and 'joined-up'

– where every node and edge on the composite diagram (Figure 93 above) has a correspondence to at least one node or edge on a 'source' diagram that provided the information. Trying to maintain this coherence over time, through multiple changes to the architecture, will be a significant challenge, if each change to each diagram must be made independently, with no automation or consistency checking involved.

8.3.3 Change in Notation

From our research, specifically the graphs that show the comprehension of concepts in ArchiMate vs. the neutral notation, we can see that of the overall concepts in the Motivation, Business and Application layers, 24 out of 32 (75%) were less well understood when expressed in ArchiMate (see discussion in 7.8.3 above).

Now one *might* take the view that this understanding is not important, as the perceived utility ratings do not suffer in the same way when ArchiMate notation is used.

For these reasons, we might decide that the default notation (ArchiMate) is unsuitable for a particular set of stakeholders. We might also consider whether we might want to render some elements in 'pure ArchiMate' and some in an alternative notation. This is perhaps something that should be discussed in an organisation as part of the process of establishing an architecture practice, as part of standardising on the way that information is presented to stakeholders. Using a mix of notations might be more confusing; alternatively, it might be a signal that some areas of the model have been subject to a more rigorous analysis than others.

The alternative notations that we select could just be a re-rendering of the ArchiMate concepts in a non-ArchiMate representation (for example, coloured rectangles, as used in some of the interviews). It could also be that we decide to use a different language altogether for some stakeholders, for example for detailed process diagram, using BPMN [205], or for detailed solution design, resorting to UML [49].

8.3.4 Summary of Transformation Types

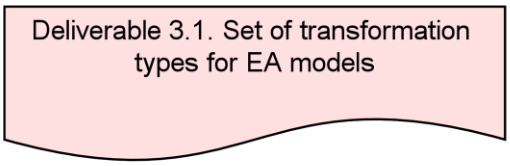
We have seen that a number of transformation types may need to be applied to our EA models:

Table 44 EA Model Transformation Types

Cluster	Description	Type

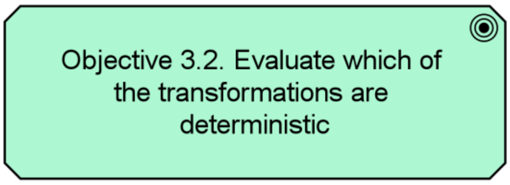
Complex to Simple	Omitting some concept types[23]	Type 1
	Change Notation	Type 2
	Change Language	Type 3
Simple to Complex	Merge fragments to produce a complete picture	Type 4

This summary of transformation types comprises our next deliverable:



8.4 Determinism of Model Evolution

We now consider which of these transformation types is in principle capable of being executed in a deterministic fashion, and thus address our next objective:



We start with a definition of deterministic as used in this research. By deterministic, we mean that for any given set of inputs (in our case, set of data comprising architecture descriptions), we can identify a set of outputs that is the inevitable result of applying the transformations that we have defined. In other words, there is nothing 'random' or 'subjective' in the way the transformations are carried out.

This is an important question to answer because, if the transformations are not deterministic, then they are subjective (subject to challenge) and will have to be done manually (or randomly). The best outcome would be if we could construct an algorithm for each of the transformation types that might lead to the possibility of being able to implement them automatically.

[23] Perhaps because they are poorly understood by stakeholders, or they are not perceived to be useful for those stakeholders

We consider each of the transformation types listed in Table 44 above and consider whether it could be executed in a deterministic fashion.

8.4.1 Type 1 – remove unwanted element types

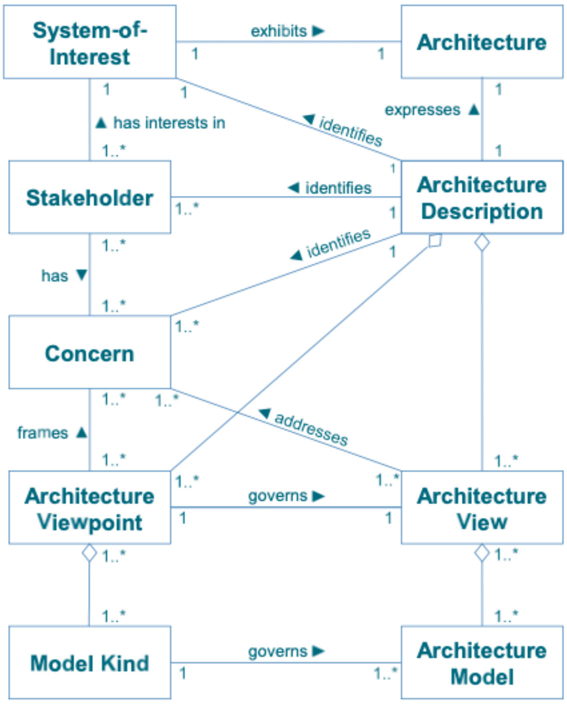
In order to remove unwanted element types, we need to answer the following questions:

(a) What is our scope – which diagrams, and which element types?

(b) Are there rules in our language that describe how this can be done?

Addressing the first question, this is part of tailoring our enterprise architecture framework for our particular organisation, to meet the needs of our particular set of stakeholders. The output of this will ideally be a set of rules, possibly based conceptually upon the following diagram from TOGAF 9.3:

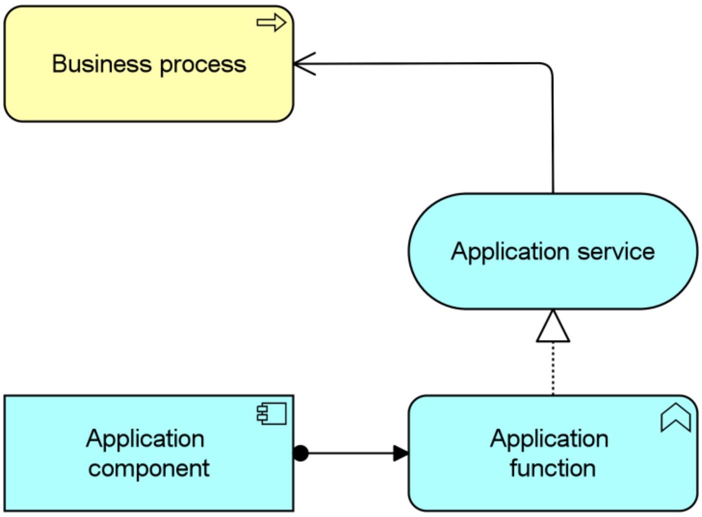
Figure 94 Basic Architectural Concepts {The Open Group, 2018 #709}



However, we would wish to include a number of extensions to the definition of Architecture Viewpoint, to include a number of additional items. For the purpose of removing unwanted element types (for example, some of the concepts highlighted in 7.7.4 above that scored poorly in comprehension or utility), we would wish to include in the definition, the set of elements that should be specifically excluded. The precise algorithm to use to do this will depend on the choice of framework being used.

When it comes to the rules for achieving this, the ArchiMate framework is helpful here in that makes specific reference to techniques that can be used to assist with the removal from specific views of items that are not needed; in particular, the idea of **derived relations**. Take the following example (from the sample BiZZdesign architecture repository provided to the author by Marc Lankhorst of BiZZdesign [216])

Figure 95 Application Component eventually serving Business Process

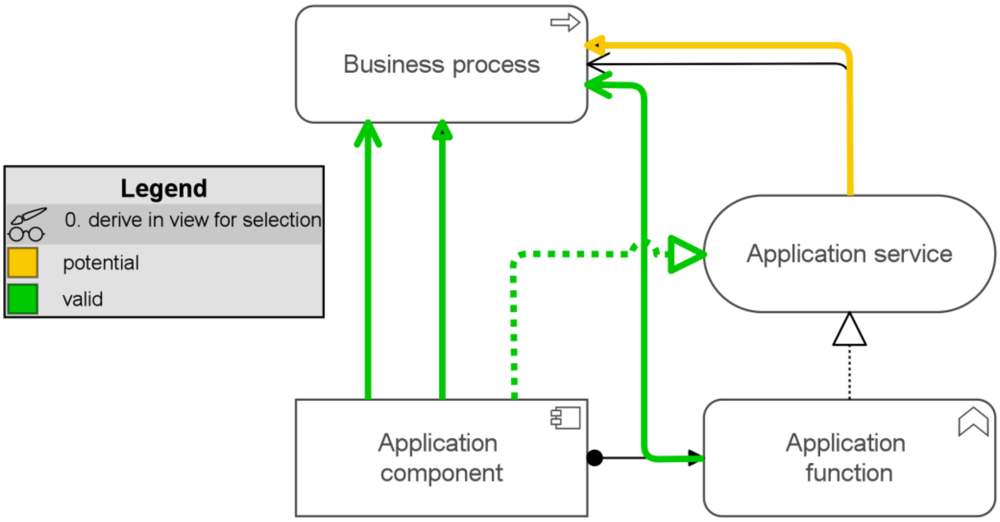


What we see here is a set of concepts than could be translated thus:

The Application Component has been assigned to provide certain functionality which is made available externally (provided as a service) and used in the execution of a Business Process). If one were interested in the intricacies of how the service was being provided and the options that might be available in the future for providing the service via a different mechanism, then this level of detail might be useful. However, for non-technical users, this level of detail would not be helpful. The concept of an application service is quite abstract (you can't point to one and say look, I'm using it), whereas an application [component] is more concrete (you can point to an application on a screen and show what you're doing with it). So, for non-technical users, they would be content with knowing that, in their job, they carry out certain processes and use certain applications as part of those processes. Thus, some of the concepts would be relevant to non-technical users and some would not.

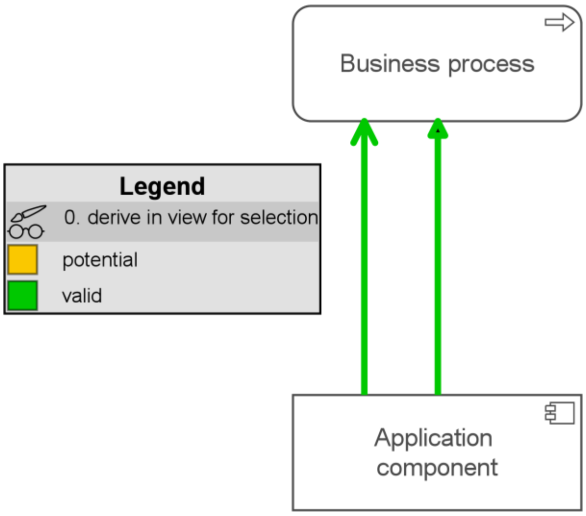
What happens in the situation where some of our stakeholders do not understand the concept of an application function or service, so that we wish to include just the application component and the business that it [indirectly] supports? How should we relate them, when the objects in the middle have been omitted? We still need to relate the Application Component and Business Process in some way. We want to create a relation to do this that is derived from the other relations shown above. These *derived relations*, and the way they are constructed, is described in the ArchiMate specification [214, 215]. If we apply the rules contained in the ArchiMate specification, we calculate that the following relations can be derived from the three given as our starting point in Figure 95 above:

Figure 96 Potential Derived Relations



If we now remove the concepts that the stakeholders will not understand, we get to the following:

Figure 97 Hiding Unwanted Concepts



We are left here with two possible derived relations: one dependency (serving, with an open arrow) and one dynamic (triggering) and the choice of which we choose to retain on this particular diagram would depend on the needs of the stakeholder(s) for whom the view was intended. It is certainly feasible to include in our rules for this kind of view, a choice as to whether we are more interested in dependency or dynamic relations.

Thus, we can confirm that, given a set of rules that define what concepts should be removed from what models, and which classes of relation are most useful to the stakeholders for whom the models are intended, we can deterministically modify the models to remove unwanted element types.

8.4.2 Type 2 – change notation

This again requires us to ask a number of separate questions:

(a) What is our scope – which element types do we wish to change and on which models?

(b) What should be our alternative notation for the element types on those models?

The answers to both questions will be part of customising our Enterprise Architecture methodology for use in an organisation. If we know, for example, that for stakeholder A on models of type B all elements of type C should be represented as red circles, then the algorithm for doing this is self-evident: simply change the notation for all such concepts on all such models intended for the stakeholder.

8.4.3 Type 3 – change language

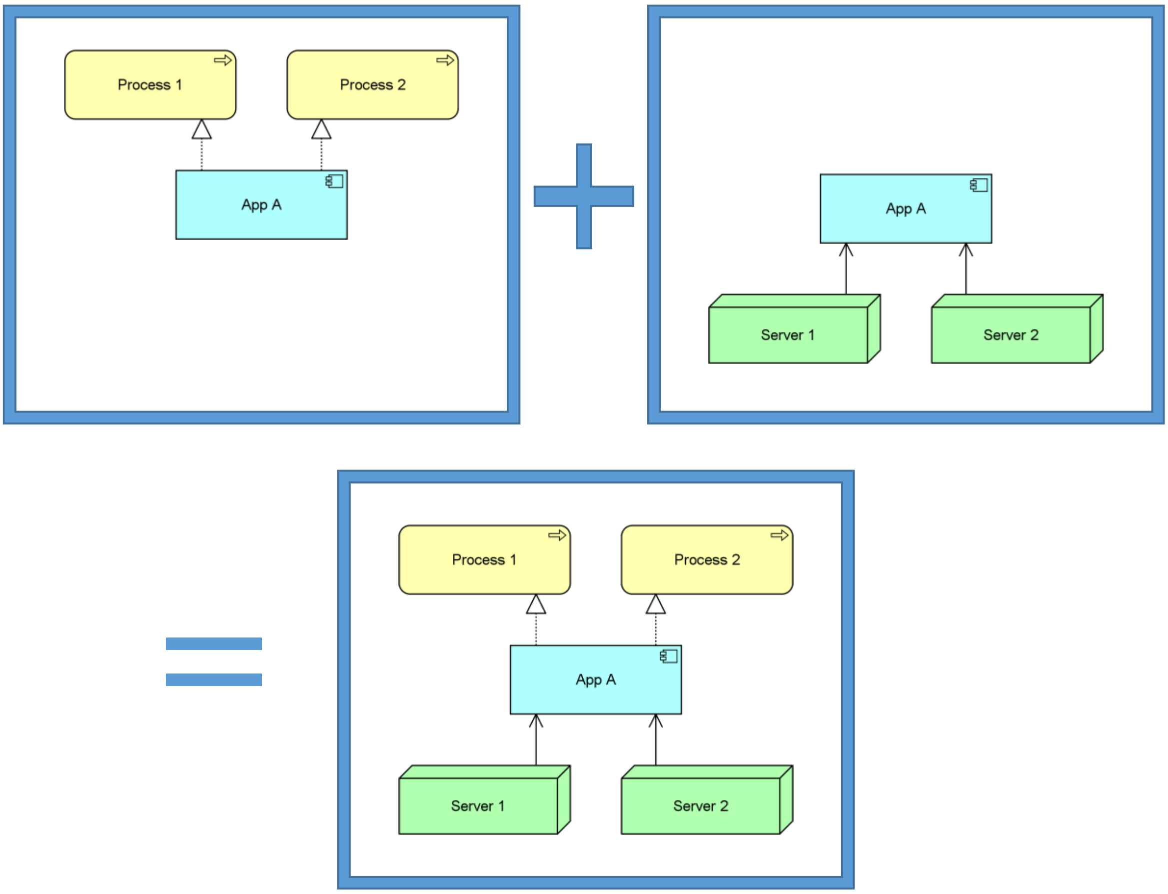
Catalogues (see section 8.5.1 below) hold lists of data of a particular type (concept). In many cases, a concept appears in multiple languages and sometimes spanning multiple classes of language (e.g. spanning both EA and process modelling languages, or EA and solution modelling). Examples of this would include business processes (appear in EA languages such as ArchiMate and TOGAF as well as process modelling notations such as BPMN [126] and EPC [206]). One of the useful features of EA modelling languages is that, although they do not necessarily contain the domain-focused detail of (say) a process or solution modelling language, they cover a wide variety of types of concept at a high level and so it is likely that parts of the more domain-focused languages will overlap at some point with parts of an EA language. This means that there are going to be elements of an EA modelling language that can be mapped directly to elements of other classes of language, and vice versa. Thus, the conceptual task of translating the data is reduced to understanding which parts of the source and target languages have concepts in common that need to be mapped.

8.4.4 Type 4 – merge fragments to produce a complete picture

We are not concerned at this stage with the practicality of this transformation; that will be covered in a later section. For now, we consider the conceptual task. This does however necessitate a somewhat technical discussion about the representation of nodes and edges that maps directly into how EAM tools work.

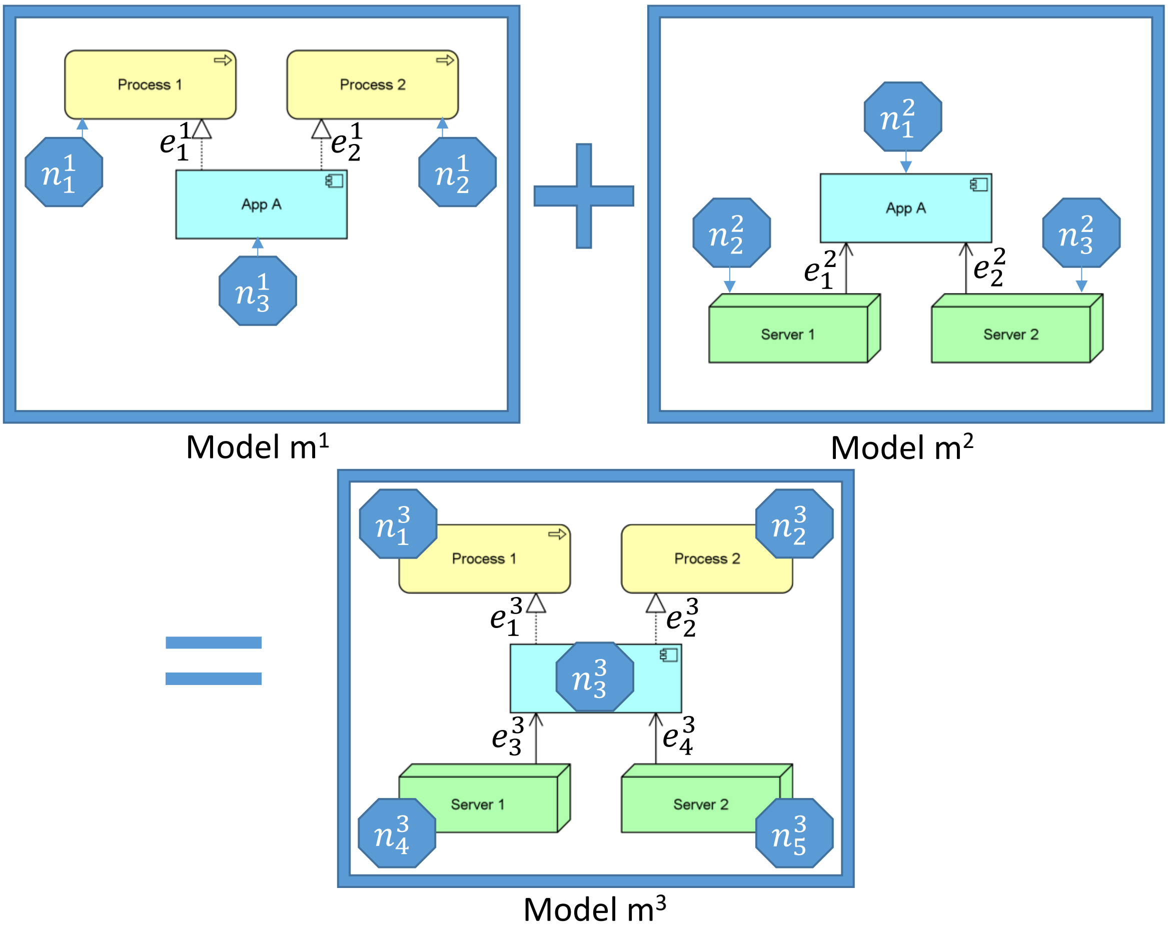
To consider how this could be done, consider one diagram that shows a relation between an application and two business processes and another diagram that shows a relation between the same application (i.e. with the same referent in the repository contained within the tool) and relations to two servers hosting the application. We can see that if the App A in the first picture represents the same App A as the second picture that we ought to be able to combine them, creating a more complex diagram from two simple diagram:

Figure 98 Combining architecture diagrams

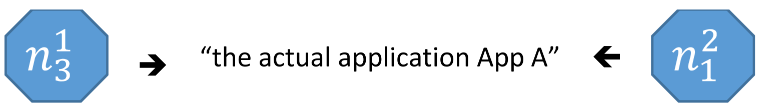


Using the notation developed in Chapter 4, we can illustrate how this works by overlay the edge and node notation on the combined architecture diagrams thus:

Figure 99 Combining architecture diagrams (annotated)



Recall that the upper index refers to the number of the model (diagram), whereas the lower index is used to count the various nodes and edges. What we mean by saying that both App As (models m^1 and m^2) refer to the same thing is that *they have the same referent*. We might notate this as:



Merely by examining the diagrams visually, we might infer that from the fact that they have the same name. This is not however an infallible technique.

The process of creating the combined model, therefore, might start by selecting a particular referent (say, "the actual application App A"), and then executing an algorithm like the following:

Figure 100 Algorithm for generating a context diagram (combined picture)

```
Put a reference to our application (starting) referent on a blank model
   (this would create n₃³)
For each relation referent connected to our application referent
  Find the element referent at the other end of that relation
  Create reference to those other elements on the model
```

(this would create n_1^3, n_2^3, n_4^3 and n_5^3)
Create a reference to the relation joining them
(this would create e_1^3, e_2^3, e_3^3 and e_4^3)

Some further algorithm would be required to lay out the elements and relations on the new model; there are algorithms available to do this, for example the Spring algorithm [217].

Thus, we have established that, in principle, it is possible to merge fragments of a model together to produce a more complete model.

8.4.5 Summary of Deterministic Potential

We have established in the above sections that for the four classes of model evolution we described, their ability to be carried out in a deterministic way is as follows:

Table 45 Determinism of EA Model Transformation Types

Deterministic	Description	Type
Yes	Remove unwanted element types (concepts)	Type 1
Yes	Change Notation	Type 2
Yes	Change Language	Type 3
Yes	Merge fragments to produce a complete picture	Type 4

This completes our penultimate deliverable:

Deliverable 3.2: Evaluation of transformation algorithms with regard to determinism

8.5 Enterprise Architecture Modelling Tools

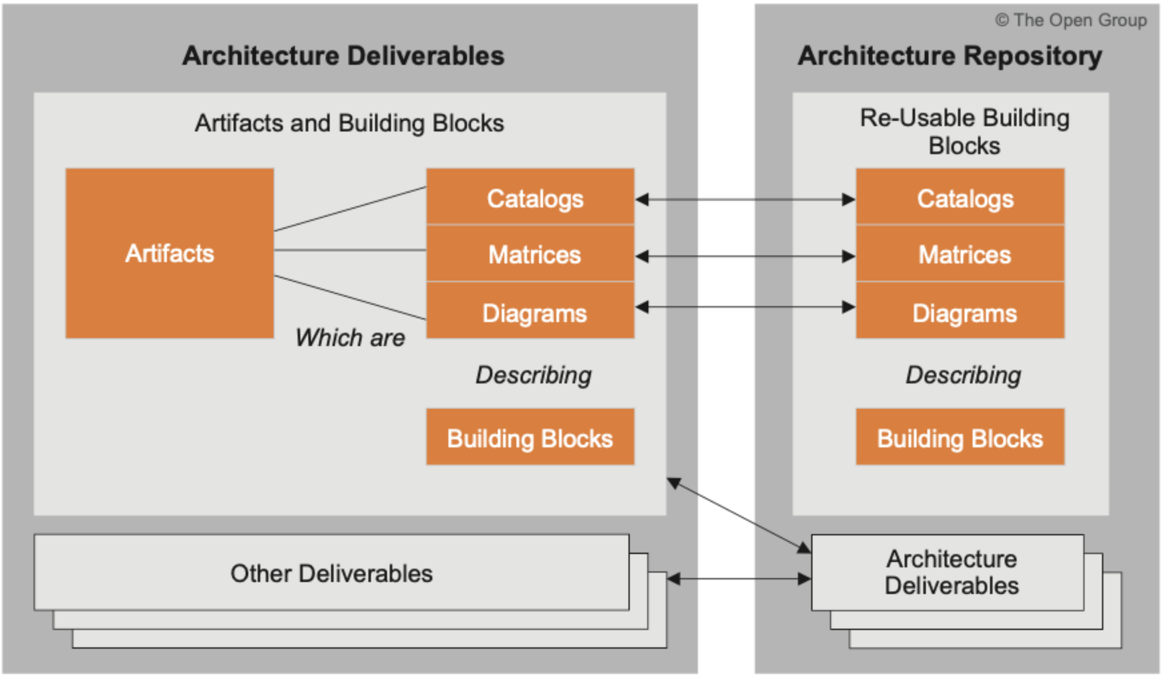
Before proceeding with the final objective, we provide a brief explanation of the way that Enterprise Architecture Modelling Tools (as described in industry reports such as [17]; often abbreviated to "EAM tools") function. We focus on their key differentiating feature from standard Office products such as Microsoft PowerPoint ® and Visio ®, using key concepts from the field of semiotics. This is useful in order to lay the groundwork for discussions about the practicality of model evolution as we will be using an EAM tool to demonstrate the practicality of these algorithms.

EAM tools provide a repository to manage architecture descriptions, along with tools to update, visualise and report on the descriptions.

8.5.1 Architecture Descriptions

To illustrate the concept of architecture descriptions, TOGAF 9.2 provides the following illustration:

Figure 101 Relationship between deliverables, artifacts and building blocks [50]



In the context of the architecture models we have been discussing, we can consider **catalogues** to be lists of elements that *could*[24] appear on diagrams (e.g. a list of business services provided by the organisation), **matrices** are representations of a specific relation between two lists (perhaps catalogues) of elements (e.g. which business services are owned or provided by which parts of an organisation), and **diagrams** are often (but not always) combinations of nodes and edges (see 6.4.1.1 above) which correspond to items from catalogues linked by relations. As discussed in our introduction (section 1.3.2 above), models can be a synonym for diagrams (as in, have a look at this particular architecture model); diagrams are models, but not all models are diagrams.

For example, consider the following three lists (catalogues) of ArchiMate elements:

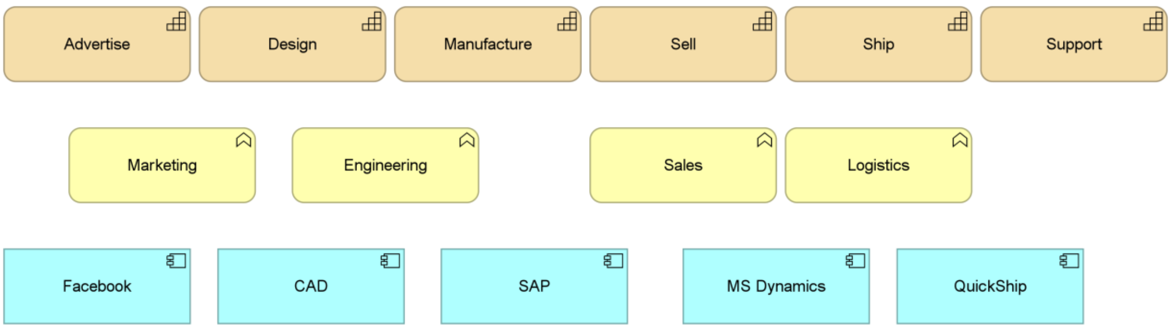
[24] And perhaps *should*, if they are part of the set of models that we expect to see in a complete architecture description – see our description of a completeness metric in section 4.3.5

Table 46 Three ArchiMate Catalogues

Test organisation	Test applications	Test capabilities
Relations	Relations	Advertise
Engineering	CAD	Design
Logistics	Facebook	Manufacture
Marketing	MS Dynamics	Sell
Sales	QuickShip	Ship
	SAP	Support

The items in these catalogues can be placed on a diagram, thus:

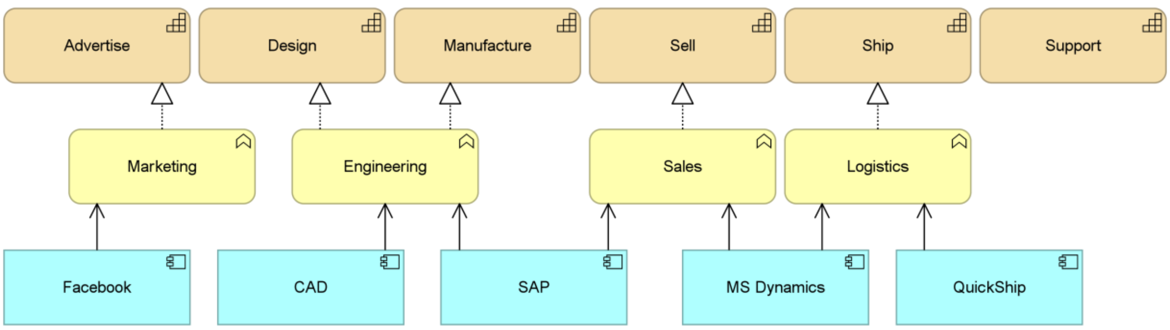
Figure 102 Elements from Catalogues in a Diagram



When placed on a diagram, elements from a catalogue form the nodes, as we can see above.

These can be connected by relations that show which specific elements relate to which other specific elements, which in the terminology adopted in our model quality research (see section 6.4 above), means we add edges to connect these nodes, for example:

Figure 103 Elements from Catalogues Linked by Relations



Matrices can be used to visualise the relations between two specific lists, so for example (using the above data) we might wish to see which business functions rely upon which applications:

Figure 104 Matrix showing relations between catalogue items

	Test organisation	Engineering	Logistics	Marketing	Sales
Test applications					
CAD		X			
Facebook				X	
MS Dynamics			X		X
QuickShip			X		
SAP		X			X

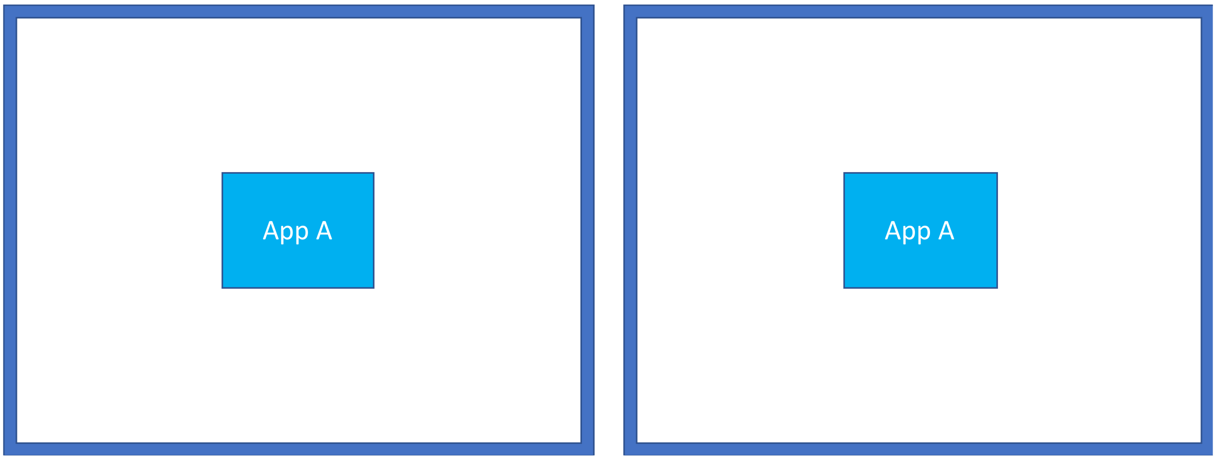
All of these are different ways of representing the aspects of concepts in the catalogues and the relationships between them. They can be produced using any software, or even drawn by hand. However, doing so manually is both time-consuming and error prone because each change to the data must be replicated by hand in many places and representations.

8.5.2 Representing Architecture Descriptions in Software Tools

In section 1.7.2 above we discuss the difference between a sign (representamen) and the object (referent) to which the sign refers (as mediated via an interpretation process). This key semiotic concept is one of the key differentiating factors between EA modelling software and basic Office software such as PowerPoint® or Visio®.

Consider two separate diagrams, perhaps produced using standard Office software (or even drawn on paper) that both contain a single element, both containing the same text:

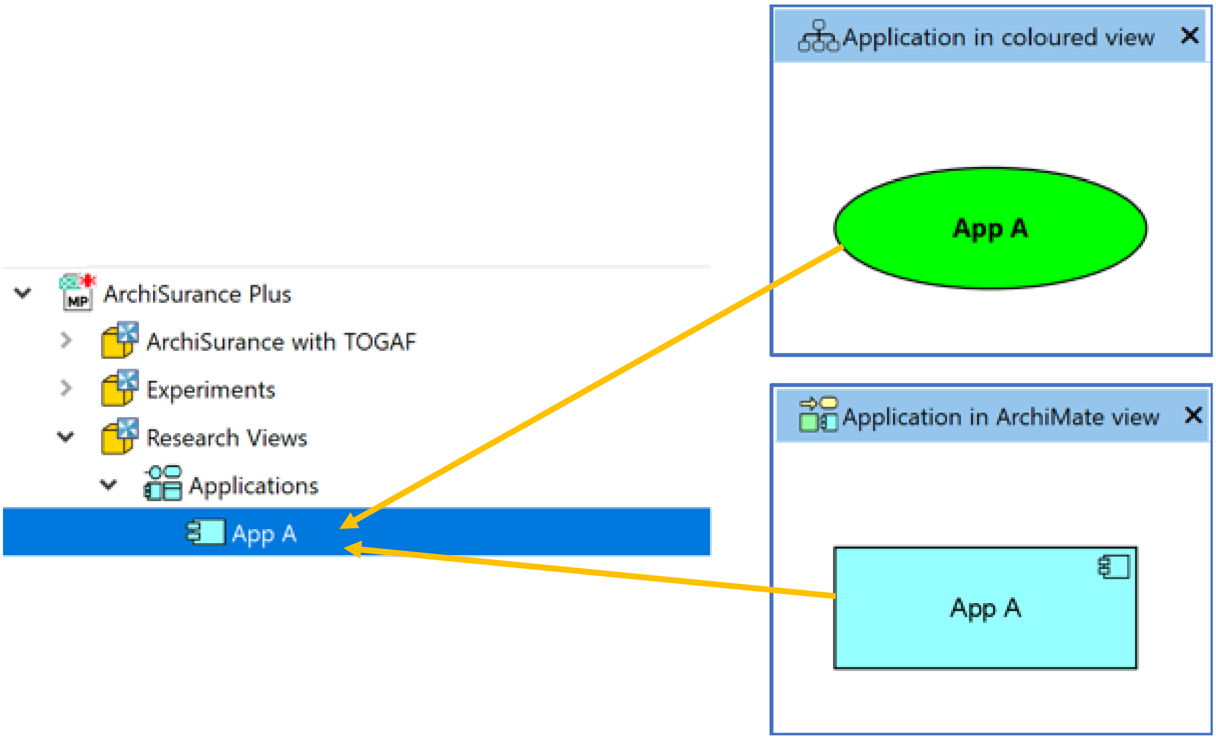
Figure 105 Two diagrams, two similar applications



Apart from the fact that the two boxes share the same name, there is no actual connection between the boxes. When we see such diagrams, we may, in our thinking, conclude (via the interpretation process) that these refer to the same application, however, as far as the tool used to draw them is concerned, they are fundamentally different objects that just happen to share a common property (their name). Thus, strictly within the bounds of this kind of drawing tool, there is no referent – no common object to which these refer. Any referent is purely in our minds. If we alter one of the diagrams to change the name of the box within it, only that box is changed; the box in the other diagram remains unchanged.

By contrast, with EAM tools (such as described in industry reports such as [17], also discussed in the literature for example [218]), **objects on views have explicit referents** within the same data store (or in TOGAF terms, architecture repository) as the objects themselves:

Figure 106 Two diagrams, one application



The above set of three screenshots, taken from one particular EA modelling tool, illustrate that two separate symbols (nodes, in this case, as they are entities) both point to (refer to) a common object, being an Application Component called “App A”, stored within the architecture repository (on the left).

Even though the representations of the application may look quite different (colour, shape), the internal properties (attributes) remain the same, because they both refer to the same object (referent) within the database. Thus, changing the internal properties of one of them (for example, its name, or perhaps other attributes such as functionality, cost etc.) changes the properties within the referent, not the representamen: and so the data changes from wherever it is viewed (from whichever representamen).

8.5.3 Representation within EAM tool

To illustrate the way EA data is stored within an EAM tool, we now show conceptually how the nodes and edges on the models shown in Figure 98 above (and annotated with node and relation numbering in Figure 99) would appear in a tool repository. In this table, the Reference column shows the sign (graphical element) from the figure and the Referent column shows the underlying element that is pointed at (referred to) by the reference. It will be clear here that there are multiple references to (signs pointing at) each of the underlying referents:

Table 47 References and Referents

Model	Reference	Type	Referent
m^1	n_1^1	Element	Process 1[25] (business process)
m^3	n_1^3		
m^1	n_2^1	Element	Process 2 (business process)
m^3	n_2^3		
m^1	n_3^1	Element	App A (application component)
m^2	n_1^2		
m^3	n_3^3		
m^2	n_2^2	Element	Server 1 (node)
m^3	n_4^3		

[25] Using the ArchiMate notation in the Referent column would be misleading in this table because the notation (visual appearance) is an attribute of the reference, not the referent. It is perfectly feasible (indeed, often desirable) to have two or more elements on the same or different diagrams that look different, but that have a common referent, given our discussion on model evolution.

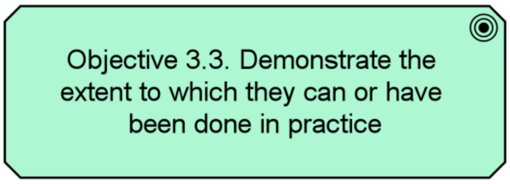
m^2	n_3^2	Element	Server 2 (node)
m^3	n_5^3		
m^1	e_1^1	Relation	Relation joining Process 1 and App A (realization)
m^3	e_1^3		
m^1	e_2^1	Relation	Relation joining Process 2 and App A (realization)
m^3	e_2^3		
m^2	e_1^2	Relation	Relation joining App A to Server 1 (serving)
m^3	e_3^3		
m^2	e_2^2	Relation	Relation joining App A to Server 2
m^3	e_4^3		

Each referent is a single item in the repository and attributes of the element or relation are stored against this item, not against the references on the various diagrams, so that updates made to attributes only have to be made in a single location.

Thus, for EAM tools, we have a separation between the catalogues of elements (and the relations between them) and the visualisation of those elements and relations on views. There are many other kinds of features provided by the vendors of such tools however, this is perhaps the key distinguishing feature.

8.6 Practicality of Model Transformation

We now consider each of the transformation types and consider whether in practice it is feasible to implement, thus addressing the objective:



8.6.1 Type 1 – remove unwanted element types

The theoretical part of this was discussed in 8.4.2 above. We now consider the practicality of removing unwanted element types. Drawing a model that contains certain element types and leaves out others is not complex; it involves just a slight modification to the algorithm given in Figure 100 above. We will show two examples of how this has been done, one focusing on the ‘hiding’ of certain element types and one focusing on the creation of derived relations.

8.6.1.1 Automated View Creation

Considering the example of Figure 98 above, we can see that the diagram contains information from three different ArchiMate layers: Business, Application and Technology. It is entirely likely that particular stakeholder groups will only be interested in a subset of the layers, so for example business stakeholders are not likely to be interested in details of the underlying platforms and infrastructure.

During the course of our current employment we have produced scripts that automatically produce tailored views, centred around key catalogued items (for example, a diagram per application that shows the business use of the application, a diagram per application that shows the platforms on which it relies, a diagram per risk (specialisation of ArchiMate assessment) that shows the applications and technologies affected by it; and so on).

These diagrams are created automatically, thus saving time and effort as well as ensuring consistency and accuracy. A sample of the scripts is shown below:

Figure 107 Code to generate technical and project context view for applications

```
ElementTypes = List ("ArchiMate:Application”,
                     "ArchiMate:Technology",
                     "ArchiMate:IM",
                     "ArchiMate:MotivationAssessment");

UpdateCatalogueViews ("Auto-generated views",
                            "ArchiMate:AbstractFolder",
                            "Application Implementation Views",
```

```
                                    "ArchiMate:AbstractFolder",
                                    "ArchiMate:AllView",
                                    "implementation",
                                    ElementTypes,
                                    undefined,
                                    "Application Views (implementation)",
                                    viewStyles,
                                    "Applications/EA",
                                    "ArchiMate:ApplicationComponent",
                                    undefined,
                                    enumExclusions,
                                    undefined);
```

The above code generates a set of views, one per application, in our application catalogue. Each view holds the application itself plus:

- Any related items in the Application layer;
- Any related items in the Technology layer;
- Any related items in the Implementation and Migration layer;
- Any related assessments from the Motivation layer (we use these for modelling corporate risks)

The common method that we implemented in software, to implement the above instruction, accepts as one of its parameters, a list of ArchiMate types (encoded in a way that is specific to by this particular EAM tool). Because this tool provides type names that always start with the layer name, followed by the particular concept name (e.g. BusinessActor, ApplicationCollaboration, TechnologyService), we were able, in effect, to implement an inferred ‘wildcard’. Thus, in the example above, where we specify the various types that we want to allow on the model as …

List (ArchiMate:Application”,   "ArchiMate:Technology",

"ArchiMate:IM",   "ArchiMate:MotivationAssessment");

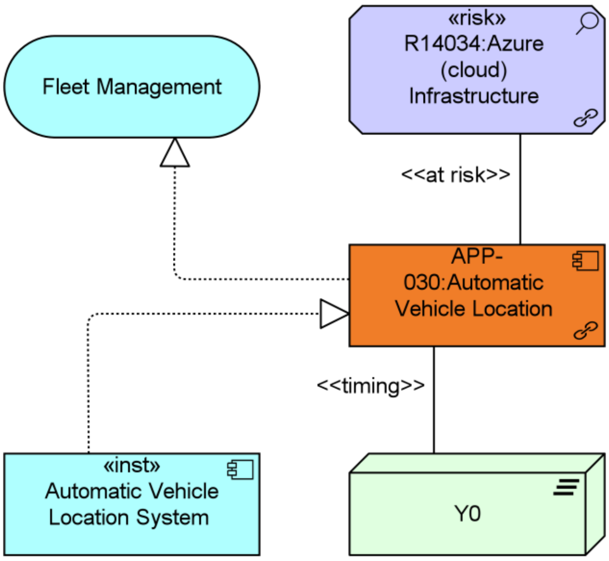
… this is interpreted as:

“Allow any concepts from the Application layer, any concepts from the Technology layer, any concepts from the IM (Implementation & Migration layer), plus specifically the Assessments[26] from the Motivation layer”.

The above code uses a standard view type provided by this particular EAM tool (ArchiMate:AllView). A sample diagram created by these scripts is shown below:

[26] In our current organisation, we use a specialisation of Assessments to represent Risks.

Figure 108 Example Autogenerated Technology/Project Context View



This view only contains items from the application layer (application components and services), technology layer (none in this particular example), implementation and migration layer (a plateau corresponding to a particular year) and risks (ArchiMate 'assessments', of which there is one). This was produced using a more complex version of the algorithm outlined in Figure 100 above. Thus, we have established that the exclusion of particular element types is indeed practical, as we have already automated it.

8.6.1.2 Automated Derivation of Relations

The most complicated algorithm here is the construction of the derived relations that will take the place of our excluded concept types.

Consider again the example ArchiMate fragment used in our theoretical discussion, shown in Figure 95 above, and how it was transformed to a much simpler model (Figure 97). The construction of the derived relations was actually carried out automatically, using the scripting language built into a particular EAM tool (BiZZdesign Enterprise Studio [22]), implementing the rules contained within the ArchiMate specification. Thus, the derivation of relations can be automated. A few samples of the code from this script (kindly provided by BiZZdesign) illustrates how the concepts from the ArchiMate specification appear in practice:

Figure 109 Sample Code for Deriving Relations

```
 (relation strengths from ArchiMate 3.0)

function strength(r) {
    if(r is "CompositionRelation")       return 8; else
    if(r is "AggregationRelation")       return 7; else
    if(r is "AssignmentRelation")        return 6; else
```

```
      if(r is "RealisationRelation")      return 5; else
      if(r is "UseRelation")              return 4; else
      if(r is "InfluenceRelation")  return 3; else
      if(r is "AccessRelation")           return 2; else
      if(r is "AssociationRelation")      return 1; else
      return -1;
   }

   (checking if a particular relation is allowed)

   function isAllowed(a, b, c, tp) {
      if( a.type() == b.type() &&
            (tp == "AggregationRelation" || tp == "CompositionRelation" ||
             tp == "SpecializationRelation") ) {
            return true; // these are always allowed
      }
      if( ( (isIM(a) || isCore(a) || isStrategy(a)) && isMotivation(b) )
&&
            ( tp != "RealisationRelation" && tp != "InfluenceRelation" ) )
{
            return false;
      }
      if( (isIM(b) || isCore(b) || isStrategy(b)) && isMotivation(a) ) {
            return false;
      }
      if( ( (isIM(a) || isCore(a)) && isStrategy(b) ) &&
            ( tp != "RealisationRelation" ) ) {
            return false;
      }
      if( ( isIM(b) || isCore(b)) && isStrategy(a) ) {
            return false;
      }
      if( ( isIM(a) &&  isCore(b) ) &&
            ( tp != "RealisationRelation" ) ) {
            return false;
      }
      if( isIM(b) && ( isCore(a) || isStrategy(a) || isMotivation(a) ) )
{
            return  false;  //  added  Strategy  and  Motivation  here  as
forbidden sources
      }
      if(   !isPassive(b) &&
            ( tp == "AccessRelation" ) ) {
            return false;
      }
      if( isPassive(b) &&
            ( tp != "AccessRelation" && tp != "AssignmentRelation" &&
            // added Realisation here compared to the standard
              tp != "RealisationRelation" ) ) {
            return false;
      }
      if( isPassive(a) &&
            ( tp != "RealisationRelation" && tp != "InfluenceRelation" ) )
{
            return false;
      }

      return true;
   }
```

This is by no means the complete code; we have just included enough to show some examples of how the ArchiMate rules have been implemented in practice.

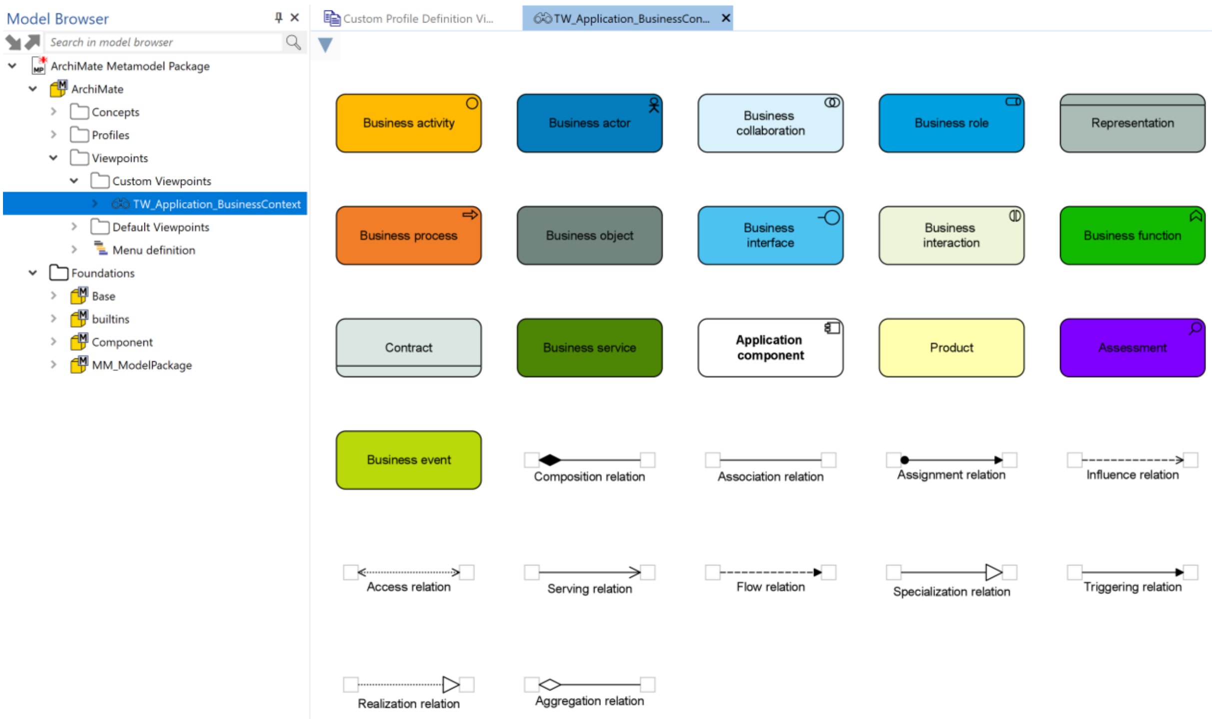
These are the main two techniques required in order to programmatically exclude elements from view types.

8.6.2 Type 2 – change notation

To demonstrate the practicality of this, we will use again the example discussed in section 8.6.1.1 above, but this time focus on the notation used on the views.

Our particular EAM tool allows for the creation of custom view types, which allow us to select alternative visual representations. Having established that using colours to distinguish different element types is, for many stakeholders, preferable to the use of ArchiMate notation, we created a new type of view, specifically for business stakeholders, that used colours in this way. This is done using the built-in metamodeller:

Figure 110 Custom View Specification in Metamodeller



This enables us to select whatever visual representation we like for each of the allowable concepts on our custom view.

Our script to regenerate the set of views is then almost identical to Figure 107 above, the significant change here being the specification, in the call to our *UpdateCatalogueViews* method, of our new custom view type:

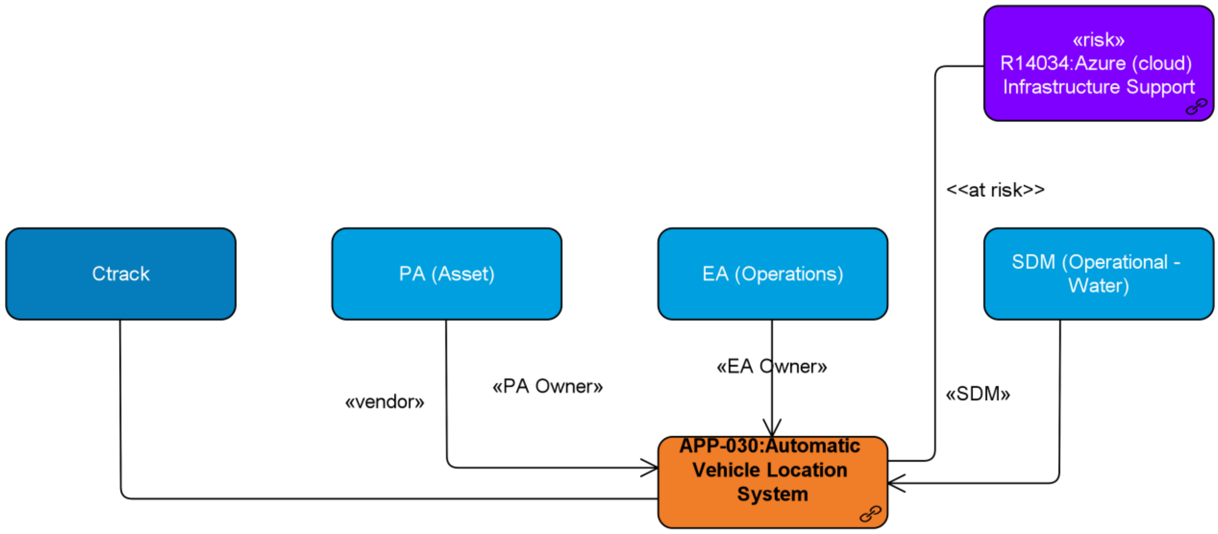
Figure 111 Code to generate business context view for applications

```
ElementTypes = List ("ArchiMate:Business",
                     "ArchiMate:MotivationAssessment");

UpdateCatalogueViews ("Auto-generated views",
                            "ArchiMate:AbstractFolder",
                            "Application Views (Business Context)",
                            "ArchiMate:AbstractFolder",
                      "ArchiMate:TW_Application_BusinessContext",
                            "business context",
                            ElementTypes,
                            undefined,
                            "Application Views (business)",
                            viewStyles,
                            "Applications/EA",
                            "ArchiMate:ApplicationComponent",
                            undefined,
                            enumExclusions,
                            undefined);
```

In this case, the restriction of the other elements to the business layer, plus risks (assessments), and the selection of a custom view type that uses colours to distinguish between elements (instead of ArchiMate shapes and icons), results in the following diagram being created:

Figure 112 Example Autogenerated Business Context View



This is another simplified view, based upon the same object as Figure 108 above, but intended for different stakeholders, and using a different notation. In this view, apart from the application itself, we have a combination of business actors and roles, plus again a related risk.

The process of actually adding references to a view is the reverse of the normal semiotic process of send a sign – interpret the sign – find the object (referent). In this case, we are starting from a referent within the architecture repository and wanting to create a sign (reference) that points (refers) to it. The code fragment that adds these references to existing referents can be seen below:

Figure 113 Code to add reference to existing referent

```
if (target in ObjectReferences.keys())    // if already on diagram
{
  refTarget = ObjectReferences.valueFor (target); // use existing reference
}
else                                // if it's not already on the diagram
{
  refTarget = view.addNewReference (target); // then add it to the diagram
  ObjectReferences.add (target, refTarget); // remember it
}
refRelation = view.addNewReference (r); // reference to relation as well
```

8.6.3 Type 3 – change language

Many organisations, including our own, produce architecture descriptions in multiple languages. For example, our master application catalogue is in the ArchiMate language. However, we wish to have those available in other languages, for example BPMN for business process modelling. Our particular EAM tool does not allow the same object to appear in views from different languages (e.g. an ArchiMate object can only appear in ArchiMate views, not BPMN views). Therefore, we use an automated (scripted) process to make our catalogues available in multiple languages by duplicating them in each required language. Thus, we are not so much changing the language of the elements, but rather making them available for use in multiple languages.

A sample of the script that implements this, showing clearly the mapping between modelling languages, are shown below:

Figure 114 Sample scripts to synchronise catalogues across languages

```
  //
  //  Synchronise applications from ArchiMate to BPMN
  //
  source = "Applications/EA";       // Source catalogue
  target = "Applications (BPMN)";   // Target catalogue

  mappings = Index();
  mappings.add ("nm", "nm");   // Copy name attribute from source to target
  mappings.add ("doc", "doc"); // Copy documentation attribute as well

  synchroniseCatalogues (source,
            "ArchiMate:ApplicationComponent",   // ArchiMate type
                        target,
                        "BPMN:BPMN_DataStore",  // BPMN type
                        equalsXMR,        // relates the master and copy
                        undefined,
                        mappings,
                        exclusions,
                        true);
```

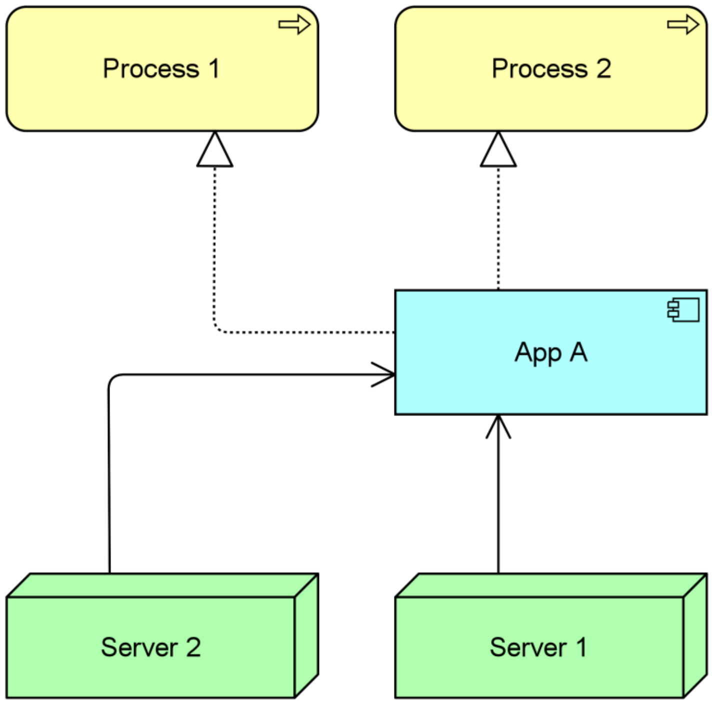
This maintains a parallel set of applications in a BPMN catalogue, thus enabling the applications to be used in both ArchiMate and BPMN models (crossing the language boundary), while still maintaining a link for traceability purposes between the master and copy elements.

Thus, using elements across multiple languages can be achieved programmatically.

8.6.4 Type 4 – merge fragments to produce a complete picture

It is a common feature of many tools to automate the production of a model showing all the elements that relate to a particular element. The diagrams already shown above were produced using this technique. Another example can be given to demonstrate this, involving the two fragments at the top of Figure 98 above. Each of the two fragments presents part of a picture. The particular EAM tool we are currently using has a “Generate View” function designed to do precisely this: to show all elements that relate to a particular element. The following diagram was created in this way, with no requirement for us to draw any elements or relations manually

Figure 115 Automatically generated merging of fragments



We have thus demonstrated, through these examples, that the merging of fragments together to create bigger, more complex models is feasible and practical, and so we can expand our table dealing with the determinism of model transformations to show the demonstrated practicality of each transformation type:

Table 48 Determinism and Practicality of EA Model Transformation Types

Deterministic	Demonstrated Practically	Description	Type
Yes	Yes	Remove unwanted element types (concepts)	Type 1
Yes	Yes	Change Notation	Type 2
Yes	Yes	Change Language	Type 3
Yes	Yes	Merge fragments to produce a complete picture	Type 4

This concludes our final deliverable:

Deliverable 3.3. Evaluation of the practicality of implementing the transformation algorithms

8.7 Practical Reflections

We can see that our diagrams can go from simple to complex (like assembling a jigsaw puzzle), and then back to simple when it comes to selecting a particular subset of our 'jigsaw puzzle' that is relevant to a particular set of stakeholders and in the process, the appearance of any one consistent element (e.g. an application or an interface) may change drastically across different diagrams.

Maintaining these diagrams over time, preserving their consistency and coherence, can be a challenge (as noted in [151]). The need to alter the representation of certain key technical ideas to cater for non-technical audiences can lead to the duplication of work, for example it is not uncommon for IT architects to use modelling tools for technical work, and then to redraw them in PowerPoint, much simplified and using completely different shapes and colours, for presentation to non-technical stakeholders.

Moreover, trying to draw conclusions from sets of unconnected and dissimilar models can be difficult. These considerations have led us to a clearer understanding of the value that EAM tools bring.

8.8 Research Results

8.8.1 Transformation Types

We have derived a number of transformation types, leading out of our other research questions, that we would expect to use when modelling Enterprise Architecture in practice.

We have seen that not only are these transformations possible in theory, we have demonstrated by real examples that they have actually been achieved in practice.

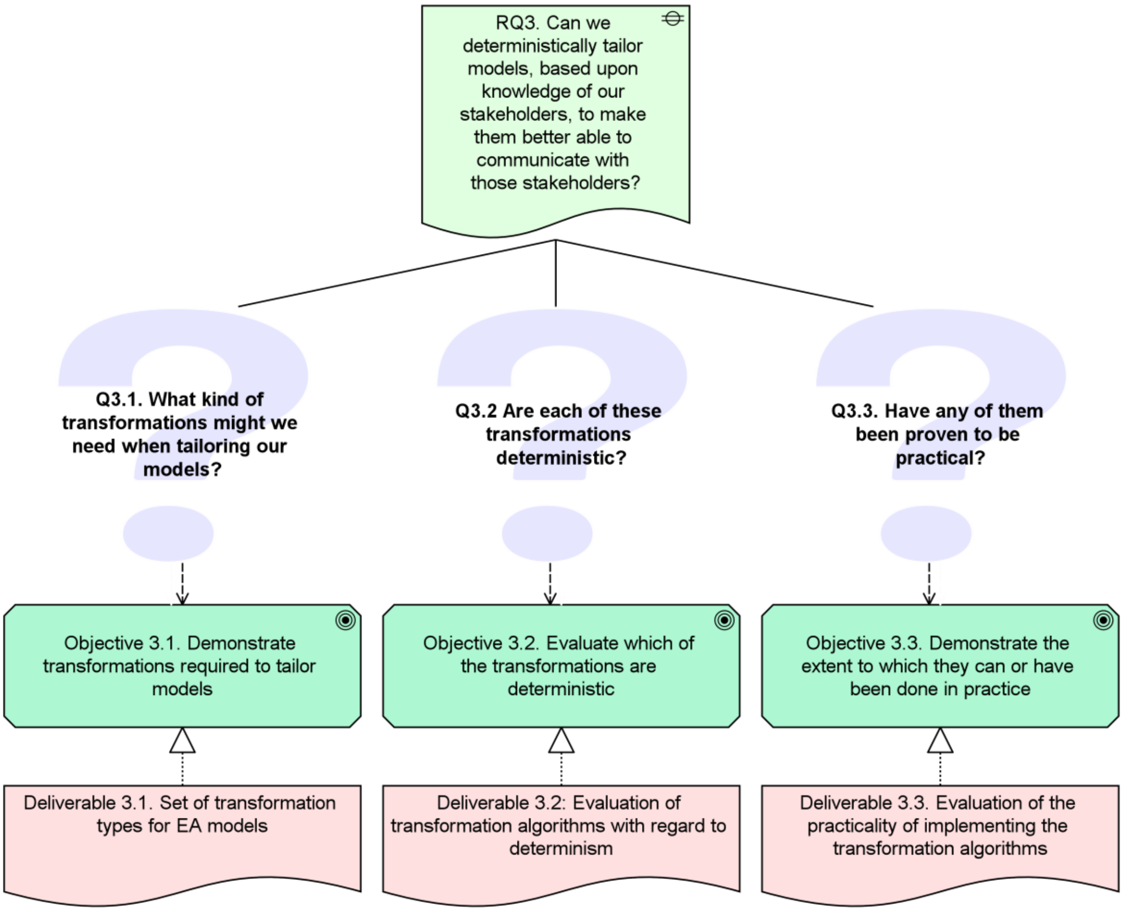
8.8.2 Recapitulation of Research Question

Our research question was RQ3, *Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?*

We have demonstrated this by categorising (in our 'transformation types') the different techniques that we would use to tailor our models, and then demonstrating that they are both conceptually possible and entirely practical.

We have thus completed the following:

Figure 116 Completion of RQ3



Chapter 9. Summary

9.1 Research Questions

The three questions that we set out to answer in this research were:

Table 49 Summary of Research Questions

RQ1	*How can we measure the quality of a set of Enterprise Architecture models*
RQ2	*How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?*
RQ3	*Can we deterministically tailor models, based upon knowledge of our stakeholders, to make them better able to communicate with those stakeholders?*

9.2 Research Review

In this section we provide a summary of our research: what we did, what we have created, our contribution in terms of methodology and practice, as well as a springboard for further research.

We start by summarising our research objectives and deliverables, related to the above questions:

Table 50 Summary of Research Objectives

ID	Objective
1.1	Construct a framework to measure the quality of a set of Enterprise Architecture models
1.2	Test the framework with a case study
2.1	Construct a framework and methodology to measure the effectiveness of an EA modelling language
2.2	Test the framework and methodology on an actual EA modelling language (ArchiMate)
2.3	Analyse and interpret results of research on ArchiMate
3.1	Demonstrate transformations required to tailor models
3.2	Evaluate which of the transformations are deterministic
3.3	Demonstrate the extent to which they can or have been done in practice

Table 51 Research Deliverables

ID	Deliverable	Section
1.1.	Framework for measuring the quality of a set of Enterprise Architecture models	4.3
1.2.	Case study showing results of applying framework to a specific real-life scenario	5.4

2.1	Framework and methodology for measuring the effectiveness of an EA modelling language	6.4
2.2A	Test materials for use when testing ArchiMate	7.3
2.2B	Results of primary research on ArchiMate	7.7.1
2.3	Analysis and interpretation of results of ArchiMate research	7.7.2
3.1	Set of transformation types for EA models	8.3
3.2	Evaluation of transformation algorithms with regard to determinism	8.4
3.3	Evaluation of the practicality of implementing the transformation algorithms	8.6

We now review in detail what has been achieved in our research with respect to each of our three main research questions. In the discussion below, our specific contributions are highlighted using the following notation:

Con/T nn indicates a theoretical contribution – where we have extended theory in some way

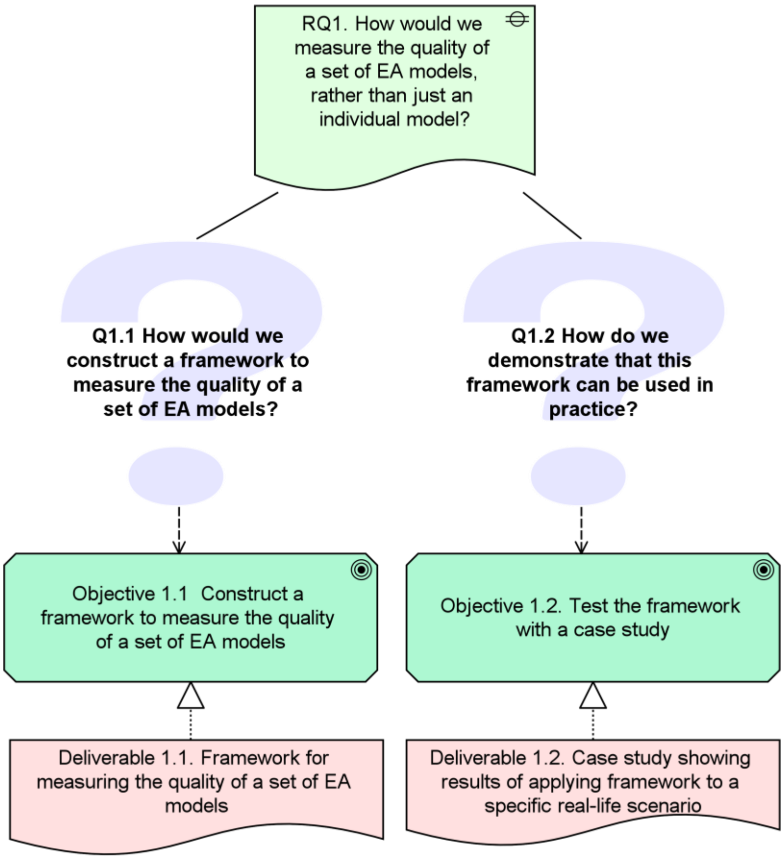
Con/M nn indicates a methodological contribution – where we have found a new way of doing something

Con/P nn indicates a practical contribution – something that we have done

9.2.1 Review of Research related to RQ1

In narrative form, to answer RQ1, we set out to construct a framework, mathematical and thus hopefully objective, that we could use to measure this kind of quality; our subsequent aim was to test that framework with a case study. As discussed previously, this was structured around the following linked objectives and deliverables (see Figure 55 on page 206):

Figure 117 Completed Objectives and Deliverables related to RQ1



To do this, in our first chapter we introduced a number of key concepts, such as Enterprise Architecture (section 1.1), its use in Information Technology (section 1.2), the concept of a model and its relevance to Information Technology (section 1.3), including the notion of model quality. We then examined existing academic work on model quality (section 1.4), identified gaps in the literature in this area (section 1.5) and, relevant to this first research question, considered the concept of what models *should* exist (for the sake of completeness).

We then considered in particular how we might address the issue of model quality for Enterprise Architecture. We then established the first of our main research questions (*How can we measure the quality of a set of Enterprise Architecture models?*), in response to this, in section 1.6.2, which was subdivided into two further questions.

In our literature survey, we covered the introductory topics in more detail, in particular Enterprise Architecture (section 2.2.1) and other related types of IT Architecture; in section 2.2.2 we examined Enterprise Architecture Frameworks (in particular their provision of a standard 'language' for modelling); sections 2.2.3, 2.2.4 and 2.2.5 respectively engage more deeply with the literature on models, quality and tools. Also in this section is a systematic review of those Information Systems (IS) journals in the "Senior Scholar's Basket of Journals" [63], looking for literature relevant to EA model quality.

We then reviewed in detail the literature closely related to our research question, and so look in detail at model quality in section 2.3, where we see an existing conceptual framework for model quality, and how it was extended into a particular kind of (less abstract) model called a Software Requirement Specification. This extension from a more abstract to a more concrete use of model quality is an example of what we sought to do, but into the area of Enterprise Architecture Model Quality. Our survey also includes a search of two specific journals in which we have previously published, as they are known to us to contain research in our own area.

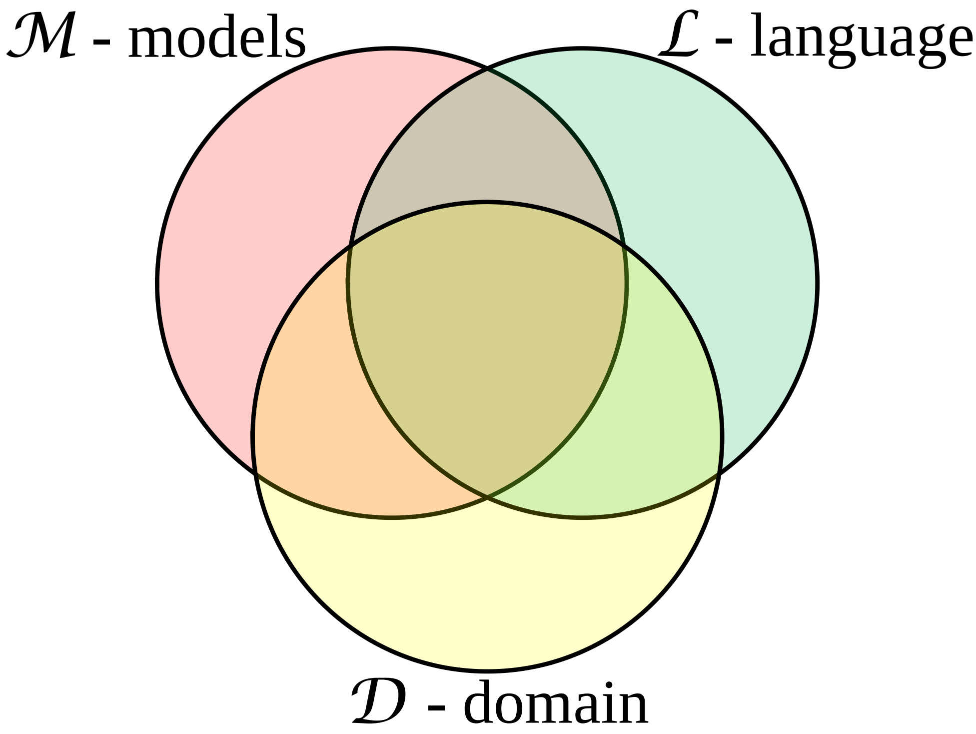
In our Methodology chapter, we started with our Epistemology (section 3.1) where we reviewed existing research theories and methods. We cover quantitative, qualitative and mixed methods, although for the first research question the quantitative method is the one we selected. Included in this literature is a five-step process to executing quantitative research, which we then used in our case study and survey, both of which contain quantitative data.

We covered some approaches to collecting data and looked at the difference between three different approaches (case studies, experiments and surveys). In section 3.2.1 we explained how we collected data relevant for this first research question, and justified our choice.

We then moved on to cover some background information about the mathematical techniques that we will used to analyse the data captured for RQ1 and RQ2. We closed our methodology chapter with a discussion of various kinds of scientific approaches and focus, in particular, on Design Science which has guided how we have carried out our research, although not exclusively so.

In Chapter 4 we reviewed (in section 4.2) some of the concepts found in the literature relevant to model quality, and also the gaps that we have identified. We then constructed a mathematical framework, based loosely upon the work of Lindland and Krogstie but extended specifically into the area of Enterprise Architecture Modelling; this is contained in section 4.3. This framework addresses the gap previously identified, that of the concept of models that should exist (previously missing from the literature). This included the following conceptual model, showing how the existence of models, their conformance to the specific language, and correspondence to truth, interact (original in Figure 43 above) in the form of a Venn diagram:

Figure 118 Venn Diagram for Conceptual Model Quality showing Models, Domain and Language



Con(T) 01. Extensions of conceptual model quality framework for use with sets of EA models, in particular dealing with the idea of models which should exist but do not

The framework defines three metrics to measure the quality of a set of models:

Q_S, syntactical quality, tells us what fraction of the existing models have the correct syntax;

Q_A, semantical quality, tells us what fraction of the existing models are factually accurate (truthful);

Q_M, completeness, tells us how many of the models that should exist, actually do exist.

These metrics, as derived in section 4.3 above, are defined as:

$$Q_S = \frac{|M \cap L|}{|M|} \qquad (1)$$

$$Q_A = \frac{|M \cap D|}{|M|} \qquad (2)$$

$$Q_C = \frac{|M \cap D|}{|D|} \qquad (3)$$

Con(T) 02. Mathematically precise formulation of three model quality metrics

We also derived equations (section 4.3.6) for the rate of change of these metrics, and used those to determine subsequent conditions that had to be satisfied for the quality of each of our metrics to continue to increase (in other words, for our models to 'get better'). These equations are as follows:

$$\frac{d}{dt}Q_S = \frac{|M \cap L|'.|M| - |M \cap L|.|M|'}{|M|^2} \qquad (4)$$

$$|M \cap L|'.|M| - |M \cap L|.|M|' > 0 \qquad (5)$$

$$\frac{d}{dt}Q_A = \frac{|M \cap D|'.|M| - |M \cap D|.|M|'}{|M|^2} \qquad (7)$$

$$|M \cap D|'.|M| - |M \cap D|.|M|' > 0 \qquad (8)$$

$$\frac{d}{dt}Q_C = \frac{|M \cap D|'.|M| - |M \cap D|.|M|'}{|D|^2} \qquad (10)$$

$$|M \cap D|'.|M| - |M \cap D|.|M|' > 0 \qquad (11)$$

Con(T) 03. Mathematically precise formulation of conditions required in order for model quality to increase over time

We extended our framework also to take into account organisations with evolving maturity, something that also had not been present in the literature. This resulted in modified versions of the definitions of the regions in Figure 118 above, dividing region $\mathcal{M}$ into two mutually exclusive regions $\mathcal{M}_S$ and $\mathcal{M}_O$. This in turn led to modified versions of the syntactical quality formulae:

$$Q_S = \frac{|M \cap L|}{|M_S|} \qquad (12)$$

$$\frac{d}{dt}Q_S = \frac{|M \cap L|'.|M_S| - |M \cap L|.|M_S|'}{|M_S|^2} \qquad (13)$$

Con(T) 04. Extension of mathematically precise quality metrics to take into account evolving language definitions

We finished this theoretical chapter by reflecting in section 4.4 on the difficulty in determining when a model can be deemed to be "true", as well as highlighting an inconsistency that had appeared in one of our previously- published papers [61], which contains an abridged version of Chapter 4 and Chapter 5. We also highlighted some areas for further research, including a more rigorous treatment of some other aspects of EA content found in our literature survey.

Moving on now to Chapter 5, we set out to test our theoretical framework in a small case study. After demonstrating how we used the quantitative five-step process found earlier (section 5.1), we then gave some context for the case study (section 5.2), and then in section 5.3 discussed our approach to data collection, building upon what was covered in our literature survey. Thus, we provided a method for capturing and analysing the data.

Con(M) 01. Method for measuring quality of set of EA models, based upon a simple quantitative case study

Having defined the mathematical framework, and a methodology for capturing the data, we then applied it in a case study, analysing data from a particular public sector organisation. On this occasion, the data that we captured is of itself not of interest outside of that particular case; the aim was to prove that that framework was practical to use, and we did that for two of the three quality metrics, Q_S and Q_C, as shown in section 5.4 above.

Section 5.5 lists the main results obtained from testing our framework. We found that the framework was indeed practically useful and we include a discussion on why it can be generalised and used elsewhere. We were disappointed that we were only able to demonstrate the practicality of two of the three quality metrics (Q_S and Q_M) that we proposed during this particular case study, which although was driven primarily by a lack of available time, also would have entailed some theoretical difficulties that still need to be addressed further.

Con(P) 01. Demonstration that the method for measuring quality of a set of EA models can be used in practice for the two metrics Q_S and Q_C

We have also shown that the above two quality metrics can be calculated automatically, thus opening the possibility of providing some kind of dashboard or other reporting mechanism, demonstrating the progress of EA modelling initiatives.

Con(P) 02. Demonstrate the practicality of automating the measurement of the two metrics Q_S and Q_C

We close the chapter, and indeed our attention to RQ1, with some reflections about future research that could extend our own in this area.

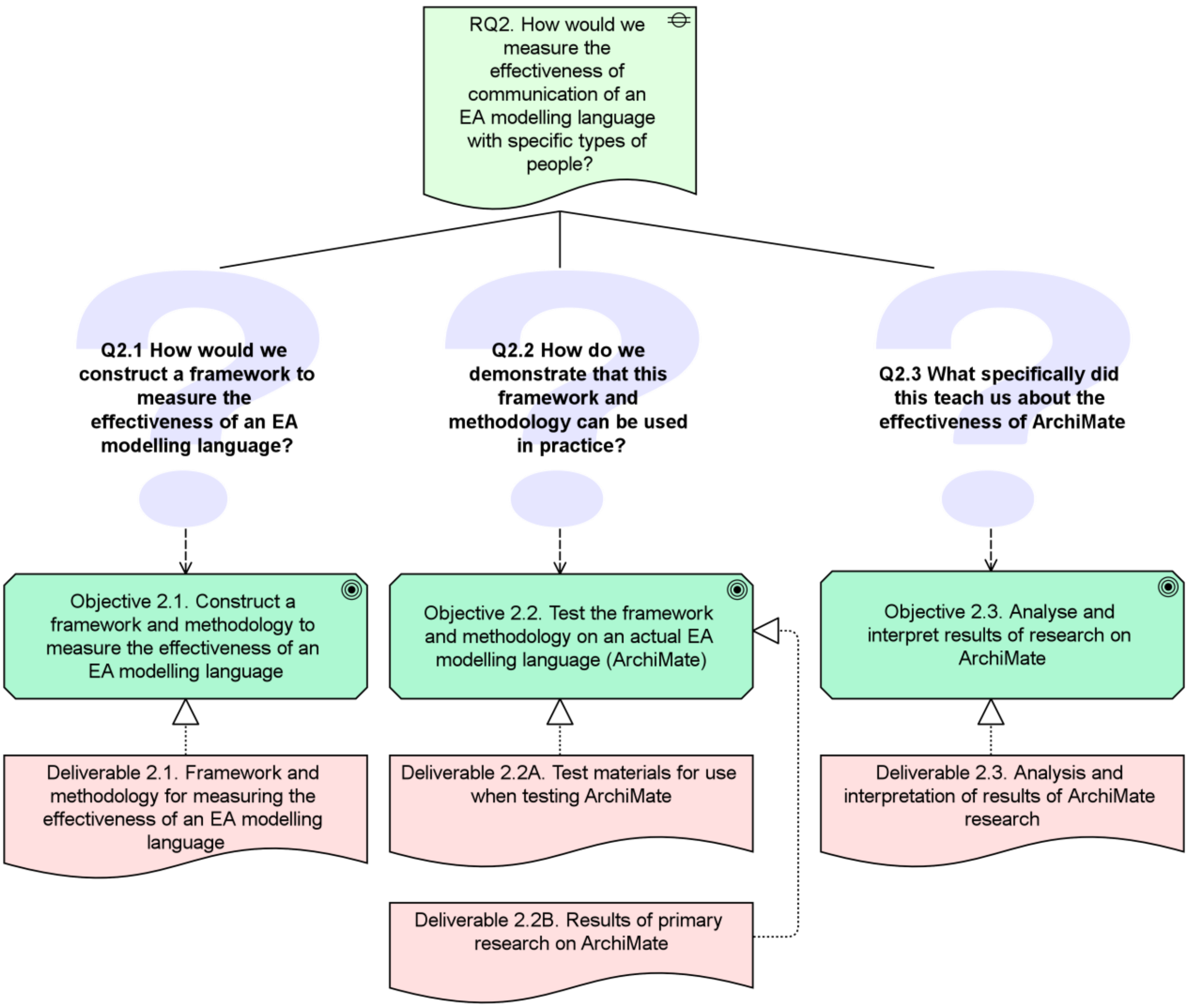
As can be seen in section 5.6 above, one area resulting from our own research would be a treatment of Bean's EA Content Aspects, giving them a formal mathematical footing and perhaps also a practical, automated way of measuring them, on similar lines that we have followed for our existing quality metrics. Further research also could help us formulate a flexible approach to determining the truth/correspondence values for models and also some practical research into automating quality measurement. When an EAM tool is being used, there are undoubtedly additional quality-related metrics that could be proposed, as discussed in 5.6.3, and these are known to be practical because we have in fact implemented them in an EAM tool in our previous employment. These could include other kinds of 'coherence' which would be relevant to the metric that we have not yet demonstrated in practice.

9.2.2 Review of Research related to RQ2

To answer RQ2, our intention was again to create a mathematical framework for measuring language effectiveness, and then to use a case study, testing the framework on ArchiMate, to do two things: (a) show that the framework produces useful results; and (b) use the results from the case study as applied to ArchiMate to learn more about how to make models that are based upon the ArchiMate framework effective in communicating with others.

This completed the following linked objectives and deliverables (see Figure 85 on page 286):

Figure 119 Completed Objectives and Deliverables related to RQ2



We started to lay the foundations for this research in the introduction where we talk about the purposes of IT models, in particular the role in communication (section 1.3.2 above). We introduced the idea of an 'entity metamodel', using a fictitious example of pets, objects and food (Figure 3), to explain that the metamodel provides a standard language in which we can express a description of an IT architecture. We discussed the common practice of practitioners producing drawings at least twice (once in an EAM tool; once in PowerPoint® or similar) and asked ourselves if part of the reason for this was that the concepts used in these frameworks had evolved to something too complicated for our non-technical stakeholders to understand. This naturally leads us to question which parts of these standard languages are actually comprehensible to our stakeholders, and thus to our second research question (*How would we measure the effectiveness of communication of an Enterprise Architecture modelling language?*).

In the literature survey we covered the topic of architecture frameworks more extensively, examining their purpose and content. We specifically took a longitudinal view of the entity metamodels from two common architecture frameworks (ArchiMate and TOGAF) to demonstrate that these frameworks were getting more complicated over time, ArchiMate markedly so (related to our introductory point), as seen in Figure 22 on page 97.

Con(P) 03. A demonstration of the increasing complexity over time of the entity metamodels from TOGAF and ArchiMate

We then discussed the impact of semiotics (the science of signs) on Enterprise Architecture, relating the notation and concepts in our EA models to signs that are interpreted according to the background (culture, norms etc.) of the reader. We surveyed some relevant literature, including some specifically related to the comprehension of IT (not specifically architectural) models, and saw that they do not provide a way of measuring comprehension or usefulness (although they do contain some interesting and relevant ideas). Given that we are looking to measure understanding of architectural models, we included some literature on the concept of understanding (section 2.3.16 above); we saw that there are three types and establish the most appropriate choice of definition from the literature (moderate factivity of objectual understanding) to use in our methodology.

In the methodology chapter, in section 3.2.2, we looked at questionnaire design for the survey that we were intending to use and concluded that it is of analytic design, with a single experimental variable (whether or not we use ArchiMate notation in the test models). We covered stages in questionnaire design, as well as the potential disadvantages of self-completion questionnaires, one of which we mitigated for the group interviews. We also discussed the idea of triangulation – capturing multiple pieces of data to get a better quality measurement – using an aeronautical example, and explained how we planned to use triangulation to get a better measurement of understanding in the survey data collection (see Figure 39 in particular)

Con(M) 02. Creation of triangulation method to measure understanding of concepts in an EA model

Moving on now to Chapter 6, where we created the theoretical framework to measure the effectiveness of concepts from an EA framework, we considered (in section 6.3.1) what we mean by effectiveness and broke this into two separate concepts: comprehension (or understanding) – how well the concepts are understood by the readers of the models – and usefulness (or utility) – whether or not the readers of the models find the concepts useful. We reasoned that these two factors should determine which concepts from our models are included for certain stakeholders.

We then looked in detail (section 6.3.2) at the semiotic processes involved in two people communicating with each other, and in section 6.3.3 we explained, using semiotic processes again, our approach to test whether or not another person understands a concept, given examples of that concept. In section 6.3.4 we showed how this applies when we have adopted a specific (restricted) language for communication, using again our 'pet entity metamodel' as an analogy.

Con(T) 05. A theoretical explanation for how we can test for comprehension of certain concepts through a two-way communication between two people

We also explained our motivation for choosing to test our framework on one particular language (ArchiMate) and, given the specific notation provided by ArchiMate, explained a differential approach that we can use to determine whether or not the ArchiMate notation is helpful.

Con(M) 03. A methodology for testing the impact of using the ArchiMate notation

We then constructed, in section 6.4, our mathematical model that we can use, in conjunction with the data that we will collect, to provide two metrics for comprehension and utility of a particular EA concept, starting from nodes and edges commonly found in EA models, and ending in the formulae below for the metrics of comprehension and utility of a particular concept. Starting from a notation of the symbols (nodes and edges) on a model as $L^d = |\ S^d\ |$ $\quad L_n^d = |\ N^d\ |$ $\quad L_e^d = |\ E^d\ |$

we then derived formulae for these metrics; firstly, for comprehension, from section 6.4.2:

$$\mathrm{H_c} = \frac{1}{\mathrm{P}}\sum_{\mathrm{p}=1}^{\mathrm{P}} \mathrm{H_c^p} \qquad (10)$$

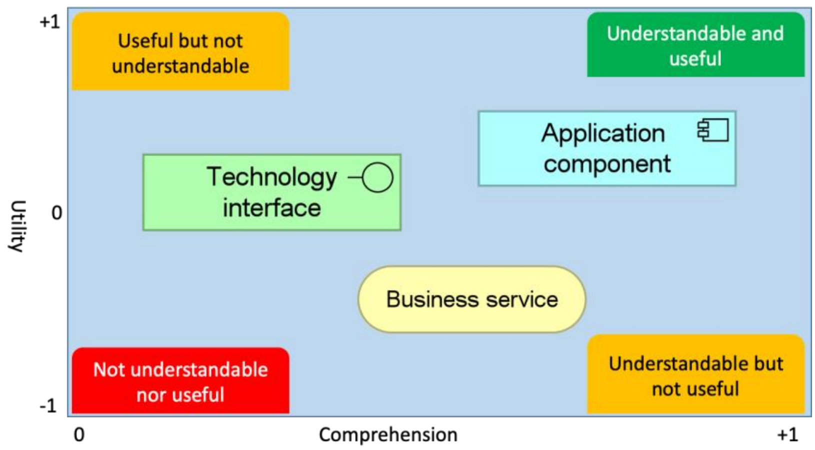
From section 6.4.3, we derive a formula for the metric of utility:

$$\mathrm{U_c} = \frac{1}{\mathrm{P}}\sum_{\mathrm{p}=1}^{\mathrm{P}} \mathrm{U_c^p} \qquad (15)$$

Con(T) 06. Construction of rigorous metrics to measure comprehension and utility of EA concepts

In section 6.4.4 we constructed a theoretical visualisation space that allows us to compare these metrics for various EA concepts (reproduced below)

Figure 120 Conceptual Visualisation Space



Con(T) 07. Construction of visualisation space to compare comprehension and utility of EA concepts

The theoretical chapter closed with an explanation, in section 6.4.5, of our method for determining the effect on comprehension and utility of the effect of ArchiMate notation, summarised in the following ordered pair of values:

$$(H_C^A - H_C^{\neg A}, U_C^A - U_C^{\neg A})$$

Con(M) 04. Method for determining the effect on comprehension and utility of using the ArchiMate notation

As with our previous research question, we now explored the practicality of our theoretical approach by trying it in practice, this time using a survey. We started with existing official ArchiMate teaching material (known by the title "ArchiSurance") and, on checking against the language specification, realised that it was missing some of the language elements. We therefore extended the material with examples of the missing concepts, as well as condensing the material into a number of test models in each of three layers of the language. We also created versions of the test models that used colours instead ArchiMate icons and shapes to differentiate between the different concepts. This process was summarised in section 7.3, and the precise details of exactly how the models were constructed and extended are given in Appendix I below. The actual test materials are shown in Appendix J below.

Con(M) 05. A method for creating test models that exercise the whole of a given subset of a language

Con(P) 04. Extended test models based upon ArchiSurance covering every language element in the Motivation, Business and Application layers,

We described in section 7.5 how we carried out a controlled experiment to pilot this method of testing language effectiveness with a group of masters-level students, prior to commencing the formal interview process.

In section 7.6 we described the data collection process, including our rationale for our choice of people to interview and the data to collect from them. We carried out some one-to-one interviews and some group interviews. Moving into section 7.7, we see the breakdown of the data captured in Table 34 on page 251; in summary, we carried out a total of 68 interviews, split across the three ArchiMate layers in scope, with roughly 60% of those on an individual basis.

Con(P) 05. Carried out 68 interviews (mixed method) to collect comprehension and utility ratings and related qualitative data based upon the three test models

We show in detail the Excel® spreadsheet used to manage and analyse the captured ratings. As we move into the analysis, we show the calculations in detail (specifically in section 7.7.3), showing exactly where every result came from, to the extent that although we have not provided the spreadsheet in electronic form, any future researcher could easily reproduce the spreadsheet given the detail that we have shown.

Con(P) 06. Complete implementation details of an analysis tool, based upon a spreadsheet, for producing various ratings charts from captured data

We also demonstrated the effectiveness of our visualisation space in showing with clarity which concepts are understood and/or perceived to be useful.

Con(P) 07. Demonstration of effectiveness of visualisation space technique

We then moved on to show the actual results of testing the ArchiMate language, in section 7.7.4. For each of the three layers selected in the ArchiMate language (motivation, business and application) we presented a populated visualisation space diagram showing the comprehension and utility ratings for each of the concepts in those layers. For each layer, we subdivided the data to show the following variations:

- Overall results (across all respondents)
- Differential results (ArchiMate vs. non-ArchiMate notation)
- Results specifically for category A respondents (non-technical stakeholders)
- Results specifically for category B respondents (IT architects)

Thus, we produced a total of 12 charts, the four above for each of the three layers.

Con(P) 08. Populated 12 visualisation spaces showing comprehension and utility of various groups and layers

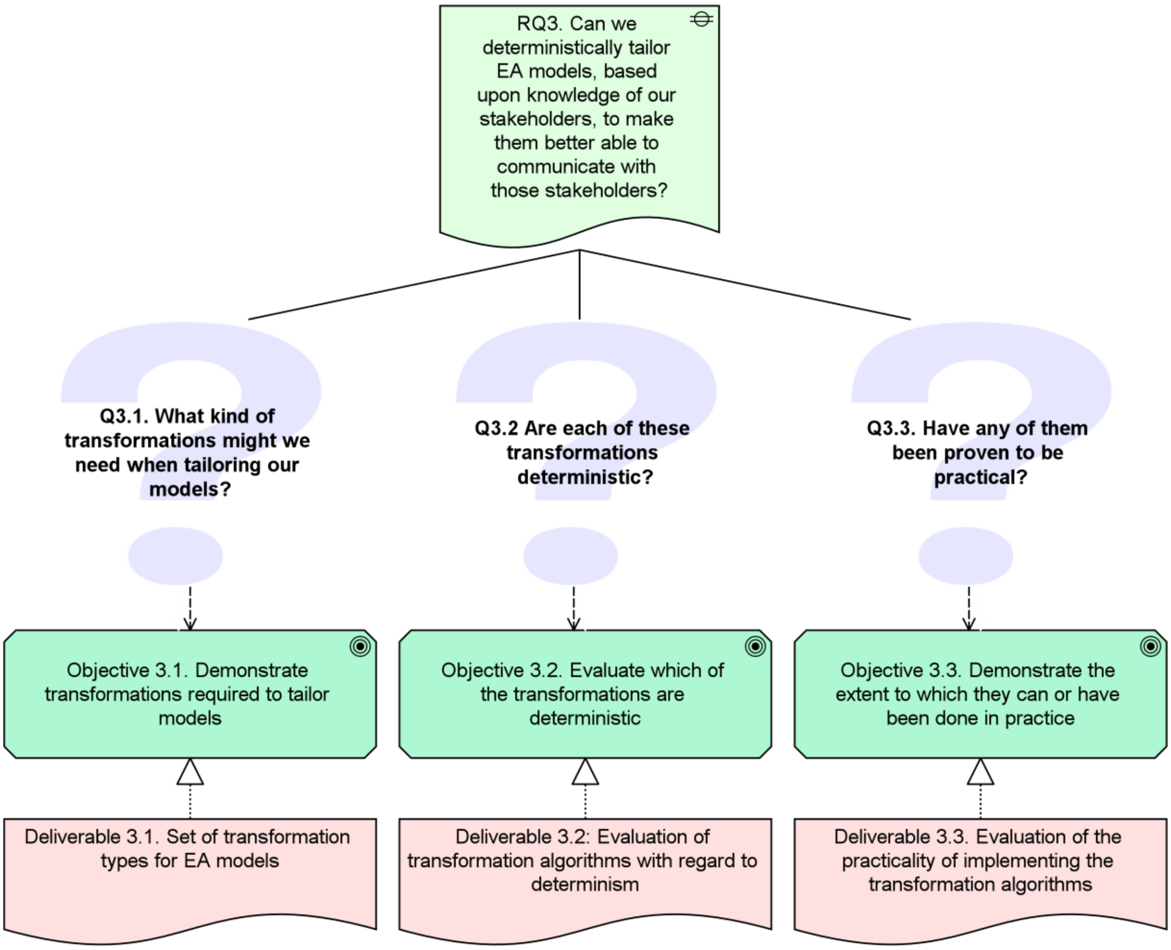
We were able to make some interesting observations from this; they can be found in section 7.7.5 above. Overall the use of ArchiMate notation appeared to have a slight (10%) negative affect on the understanding of the concepts being used (the data is strong enough to be statistically significant), but at the same time a slight (and not statistically significant) increase in the perceived usefulness of those concepts. We might paraphrase this phenomenon as "*I don't know what they are, but they look important*".

Con(P) 09. Demonstrated the effect on comprehension and utility of using the ArchiMate notation

9.2.3 Review of Research related to RQ3

Finally, to address RQ3, we first of all laid out some of the key model transformations that our research suggests are required, and then set out to demonstrate by practical means that the transformations are both deterministic and entirely achievable, based upon state-of-the-art technology. The objectives and deliverables are shown below:

Figure 121 Completed Objectives and Deliverables related to RQ3



We started in the introduction (section 1.6.4) to lay the foundation by discussing the fact that our models are likely to need to be adjusted for different stakeholders and ask the question about whether those adjustments could be automated (thus saving time and money as well as increasing accuracy). In order to automate something, it is obviously necessary, first of all, to have some kind of transformation mapped out that we wish to follow – some kind of specification or algorithm – which can then be implemented in some fashion, perhaps using a tool, and review the possible benefits of this in section 1.9.3.

In the literature survey (section 2.2.5), we discussed the idea of a tool, looking at Heidegger's work, and saw that their purpose was derived from the things around them. Also relevant, when we come to looking at Enterprise Architecture modelling tools, is the concept of semiotics again – the idea that the sign is different from the thing that the sign refers to (as discussed for example in section 2.3.4).

In our methodology section we discussed different types of research and suggest that our third research question can best be answered by means of an experiment.

In Chapter 8, we built upon the results already established in previous chapters to discuss different kinds of changes ("evolutions") that we might wish to make to our Enterprise Architecture models, putting these changes in classes of like kind, for example simplifications to omit concepts that would not be helpful to particular types of stakeholders (section 8.3.1), combining smaller fragments of models, like pieces of a jigsaw, to create a more complete picture (section 8.3.2), also perhaps a change of notation (section 8.3.3) and then summarising these different types of transformations.

Con(T) 08. Provide a categorisation of model transformation types

We then went on to examine (in section 8.4) from a theoretical perspective whether or not these different types of transformations could be deterministic and thus, *in principle,* capable of being automated. The answer we obtained, partly by reference to the ArchiMate specification and its notation of derived relations, and partly by reference to our previous mathematical model of model content, was yes, in theory all these transformations could be automated.

We then examined some of the theory behind Enterprise Architecture Modelling (EAM) tools and the key differentiating factor that separates these from standard office software; that unlike tools such as PowerPoint® and Visio®, EAM tools have explicit referents within them connected to the signs that comprise the models (diagrams). This leads directly to the ability of these tools to assemble and disassemble models (diagrams) as required.

Con(M) 06. Identification of key semiotic concept underlying EAM tools leading directly to methods for deterministic model transformation

Finally, we set out to demonstrate that the required transformation types are not just possible in theory, but in practice. We did this by giving practical examples of how these tools have in fact automated each type of transformation.

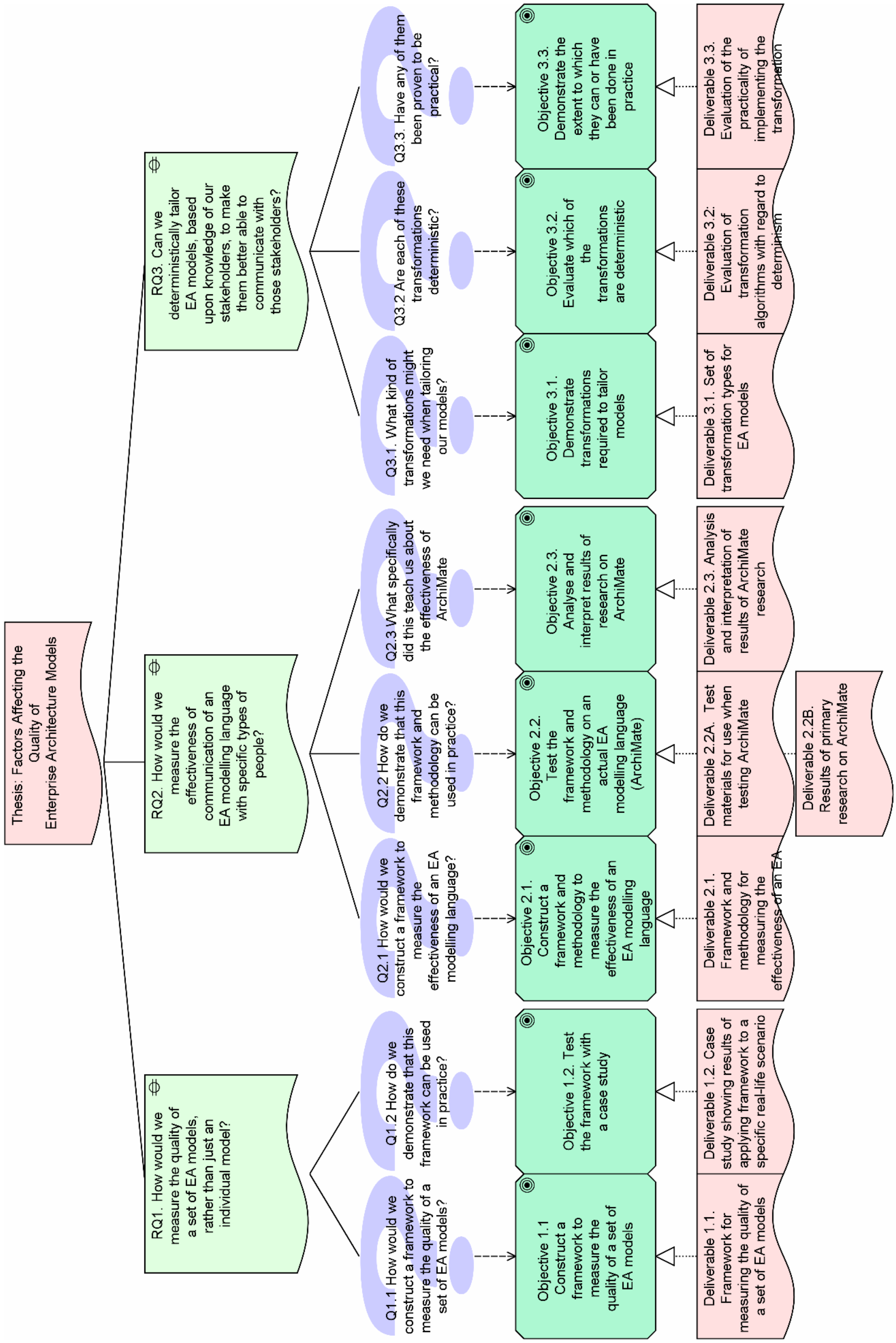
Con(P) 10. Demonstration that all required model transformation types can be automated in practice

This concludes the review of the three key pieces of research that we have carried out within this thesis.

9.3 Drawing it Together

The complete set of research questions, objectives and deliverables are related as follows:

Figure 122 Research Questions, Objectives and Deliverables



Our theoretical, methodological and practical contributions, discussed above, are summarised below:

Table 52 Summary of Research Contributions

Theoretical Contributions	Con(T) 01. Extensions of conceptual model quality framework for use with sets of EA models, in particular dealing with the idea of models which should exist but do not Con(T) 02. Mathematically precise formulation of three model quality metrics Con(T) 03. Mathematically precise formulation of conditions required in order for model quality to increase over time Con(T) 04. Extension of mathematically precise quality metrics to take into account evolving language definitions Con(T) 05. A theoretical explanation for how we can test for comprehension of certain concepts through a two-way communication between two people Con(T) 06. Construction of rigorous metrics to measure comprehension and utility of EA concepts Con(T) 07. Construction of visualisation space to compare comprehension and utility of EA concepts Con(T) 08. Provide a categorisation of model transformation types
Methodological Contributions	Con(M) 01. Method for measuring quality of set of EA models, based upon a simple quantitative case study Con(M) 02. Creation of triangulation method to measure understanding of concepts in an EA model Con(M) 03. A methodology for testing the impact of using the ArchiMate notation

	Con(M) 04. Method for determining the effect on comprehension and utility of using the ArchiMate notation Con(M) 05. A method for creating test models that exercise the whole of a given subset of a language Con(M) 06. Identification of key semiotic concept underlying EAM tools leading directly to methods for deterministic model transformation
Practical Contributions	Con(P) 01. Demonstration that the method for measuring quality of a set of EA models can be used in practice for the two metrics QS and QC Con(P) 02. Demonstrate the practicality of automating the measurement of the two metrics QS and QC Con(P) 03. A demonstration of the increasing complexity over time of the entity metamodels from TOGAF and ArchiMate Con(P) 04. Extended test models based upon ArchiSurance covering every language element in the Motivation, Business and Application layers, Con(P) 05. Carried out 68 interviews (mixed method) to collect comprehension and utility ratings and related qualitative data based upon the three test models Con(P) 06. Complete implementation details of an analysis tool, based upon a spreadsheet, for producing various ratings charts from captured data Con(P) 07. Demonstration of effectiveness of visualisation space technique Con(P) 08. Populated 12 visualisation spaces showing comprehension and utility of various groups and layers Con(P) 09. Demonstrated the effect on comprehension and utility of using the ArchiMate notation Con(P) 10. Demonstration that all required model transformation types can be automated in practice

9.4 Limitations

9.4.1 Difficulty testing correspondence to domain

When answering the question *How can we measure the quality of a set of Enterprise Architecture models?*, one of the metrics in our model was correspondence to the domain or, put another way, how truthful each model was. This is inherently more difficult to measure than the other two metrics, as it requires the examination of data outside of the model; in some cases, the data does not exist yet (because the model is describing a possible future state). The techniques one might use will differ depending on the information being modelled.

9.4.2 Limited number of respondents

We carried out 68 interviews: 23 in the Motivation layer, 25 in the Business layer and 20 in the Application layer. This volume of data is certainly suggestive of trends but cannot be taken to be definitive. Many more interviews would be needed, perhaps using a survey that, for the sake of scalability, did not rely upon interpretation and judgment by the interviewer; perhaps moving to a multiple-choice technique for data collection.

9.4.3 Limited examples of ArchiMate concepts

We found it difficult to construct examples of some of the concepts in the ArchiMate language that aligned well with the definitions, which means that for these concepts (collaborations and interactions), we were not able to offer interviewees multiple instances of each. This may (or may not) be responsible for the low ratings of these concepts.

9.4.4 Limited context through single layer test models

Using more test models, in particular, having each concept appearing in more than just a single test model, and also having concepts from different ArchiMate ‘layers’ appearing in a single test model, might give better context to the use of the ArchiMate concepts and thus lead to slightly different, perhaps more significant results.

9.4.5 Difficulty in scoring responses

As already discussed, the ratings for comprehension were assigned by the interviewer in each case, based upon verbal or written responses from the interviewees. The interviewer had at hand the official definitions for the concepts being tested and so was looking for correspondence to the central beliefs about those concepts. However, in hindsight we should perhaps have made a list of central beliefs about the various concepts prior to the interviews; this would have lessened the possibility of one kind of error in our scoring.

9.5 Further Research

The following areas for further research flow directly out of the research documented above.

9.5.1 Correlation of Stakeholders, Comprehension and Utility

Although initially we were aiming to see how stakeholder comprehension and utility varied by job role, experience and training, we were not successful in doing this, as discussed in 7.4.2 above. Possibly the volume of data collected so far would be insufficient in any case and we lack a framework for grouping the data by job role, experience and training. Having such a framework, and results broken down by such categories, would reduce the need for detailed research in each organisation to determine which concepts should be used when modelling.

9.5.2 Framework for Categorising Stakeholders, Views and Notation

Although TOGAF does discuss an approach for deciding what artefacts (including views/diagrams) may be required in general, bearing in mind stakeholder concerns (as discussed in [209]), we are lacking such a framework for ArchiMate. This would be an fruitful area for further research, especially if it was able also to incorporate the idea of notational changes, bearing in mind the results obtained regarding the effect of using ArchiMate notation for those not trained in its use (see section 7.9.6 above)

9.5.3 Use of Alternative Notation to ArchiMate

All of the research in this chapter was carried out with interviewees that had not been trained in ArchiMate. The results for comprehension and utility may well be better for people trained in ArchiMate, however many (if not most) important stakeholders will not have (or want) that training. Our research showed overall slightly better comprehension when using just colours as the distinguishing factor between concepts, rather than ArchiMate's shapes and icons. We do not know however if this is the best we can get, or whether there is some other combination that could be found (for example, different colours combined with different shapes) that would yield better results or indeed using any of the other Gestalt factors discussed in 2.3.14 above. Some of the other literature mentioned above would be helpful in this endeavour, for example the research discussed in 2.3.12.

9.5.4 Additional Quality Metrics for use with EA Modelling Tools

This research has primarily addressed EA models and largely avoided the topic of modelling tools, save as necessary to demonstrate that it is practical to implement some of the techniques for model transformation.

However, if one is using an EA modelling tool, then there are many other quality measures that could be incorporated, to answer the question that is similar to, but subtly different to, our first research question: how good is the information stored in our particular modelling tool? We anticipate that in this case, our concept of 'model' would need to move away from mere diagrams and back towards the broader scope of Krogstie's conceptual model. We foresee a number of metrics that might appear here, for example:

Are the diagrams all using a reference to the same underlying concept instance (good), or have the modellers created their own copies (bad)? Are they using the correct reference catalogues?

Are all the attributes of the underlying concept instances sufficiently populated?

Such research might produce, as a deliverable, a generic set of quality checks that could be used in a number of modelling tools. Based upon our experiences so far, these checks could all be automated, and are likely to make it easier to enhance the quality of our models, and thus the quality of the decisions made that are reliant upon that data.

9.5.5 Actual use of ArchiMate Concepts

We discussed Recker's research, in section 2.3.18, on the actual usage of various BPMN concepts, noting his categorisation of BPMN concepts by their frequency of use in practice. It would be informative to carry out a similar study on ArchiMate models (diagrams) to see which concepts are used most, in practice, and also see how that correlates with the comprehension and utility ratings.

Appendix A Architecture Frameworks

This appendix contains relevant excerpts of technical details of some architecture frameworks referenced from this work. All of the data in this appendix belongs to these other works; none of it is original to this work.

A.1 TOGAF 9.1

The Open Group Architecture Framework [29] is a non-proprietary framework for carrying out architecture work.

A.1.1 TOGAF Entity Metamodel

The following illustration shows the entities used within TOGAF and the suggested relationships between them:

Figure 123 TOGAF Entity Metamodel

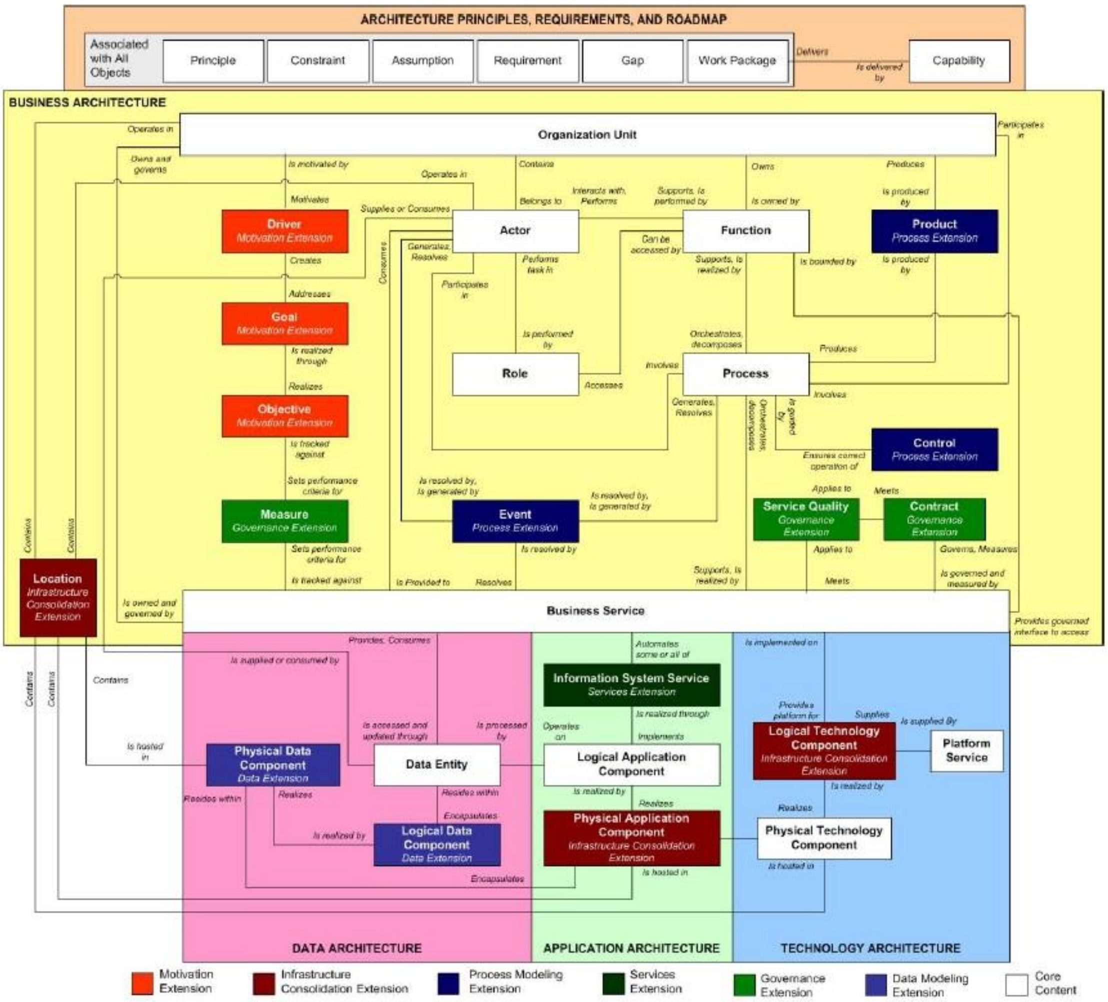
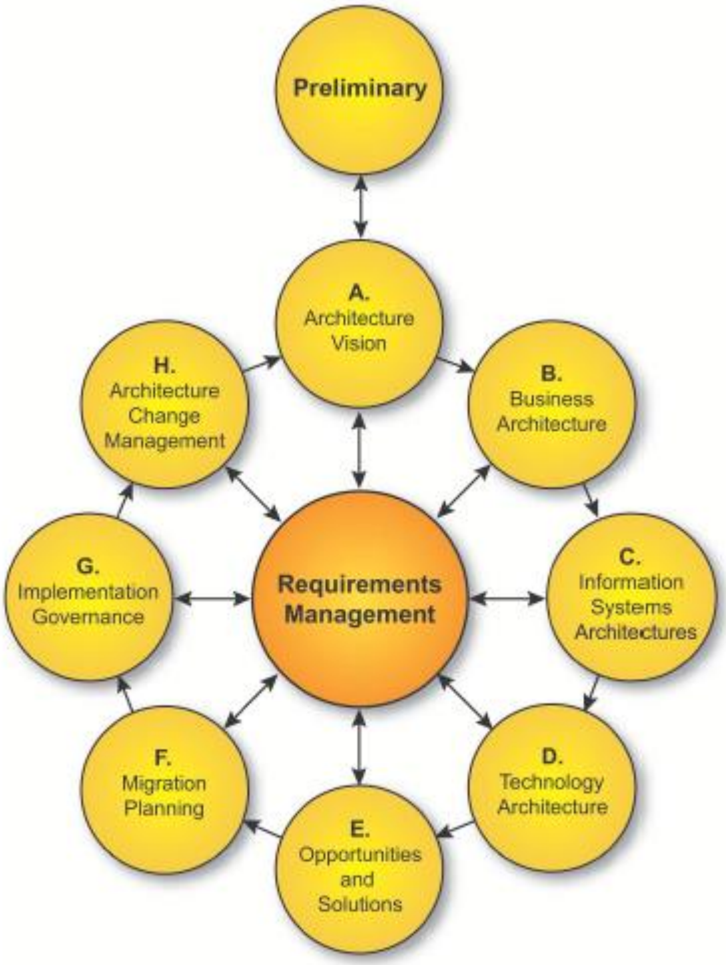


Figure 34-8 Relationships between Entities in the Full Metamodel

A.1.2 TOGAF Architecture Development Method

The following process is suggested for developing architectures using TOGAF:

Figure 124 TOGAF 9.1 Architecture Development Method [29]



A.1.3 TOGAF Technical Reference Model

Figure 125 TOGAF 9.1 High-Level Technical Reference Model [29]

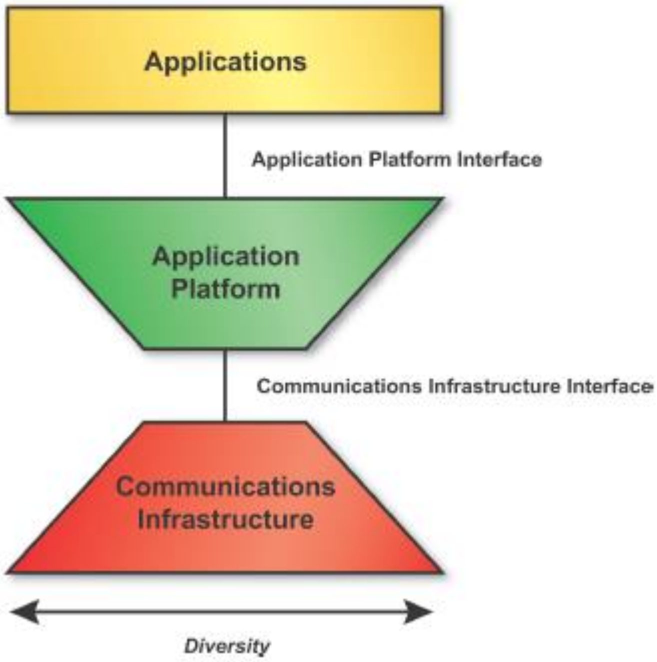
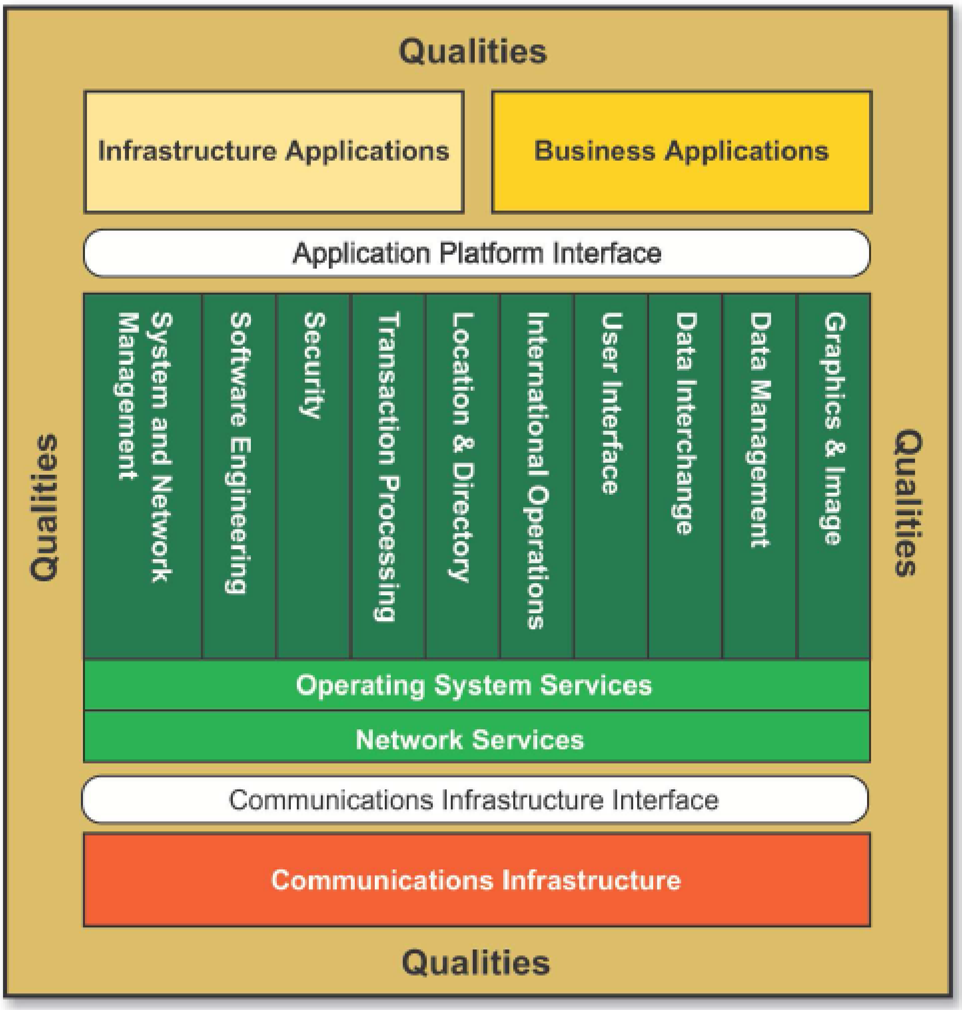
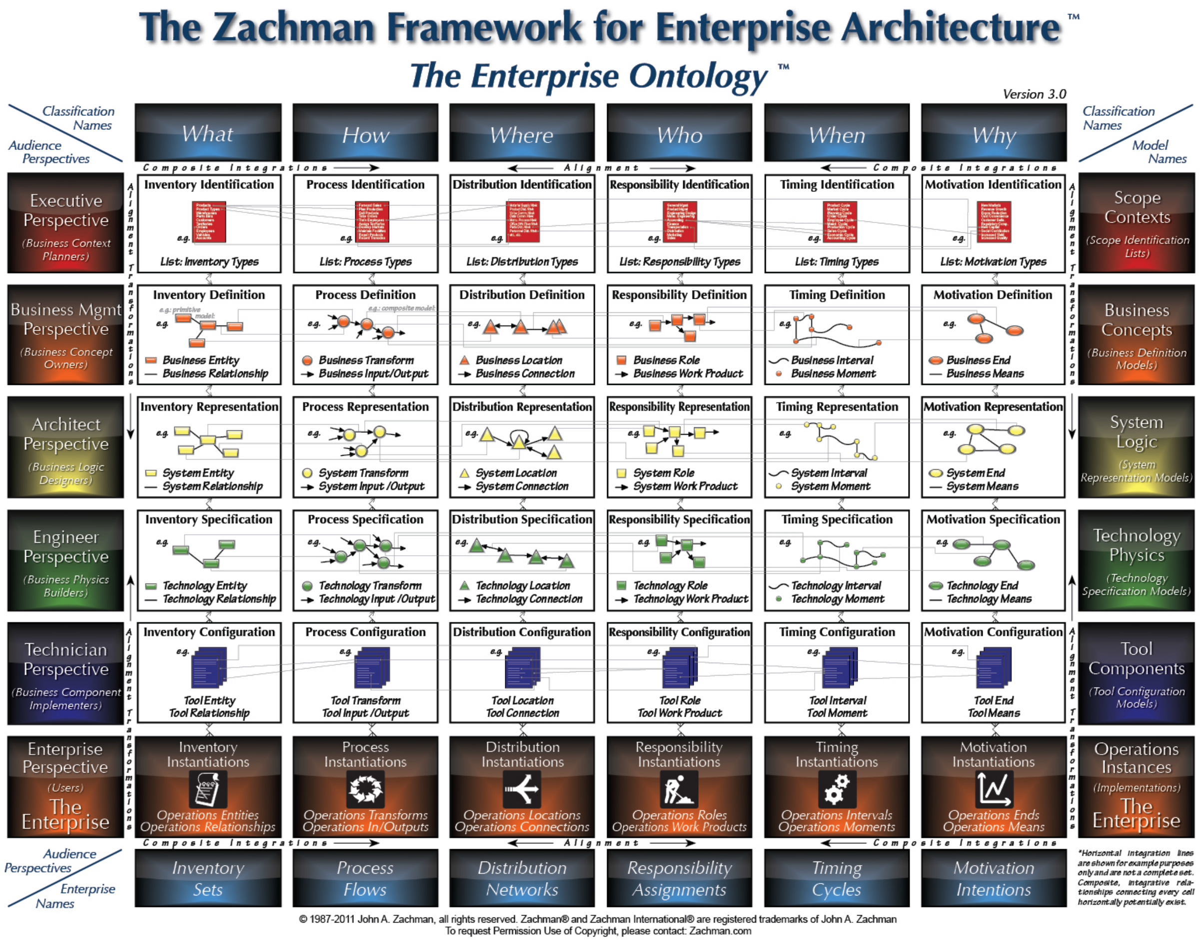


Figure 126 TOGAF 9.1 Detailed TRM for Application Platform Layer [29]



A.2 Zachman Enterprise Architecture Framework

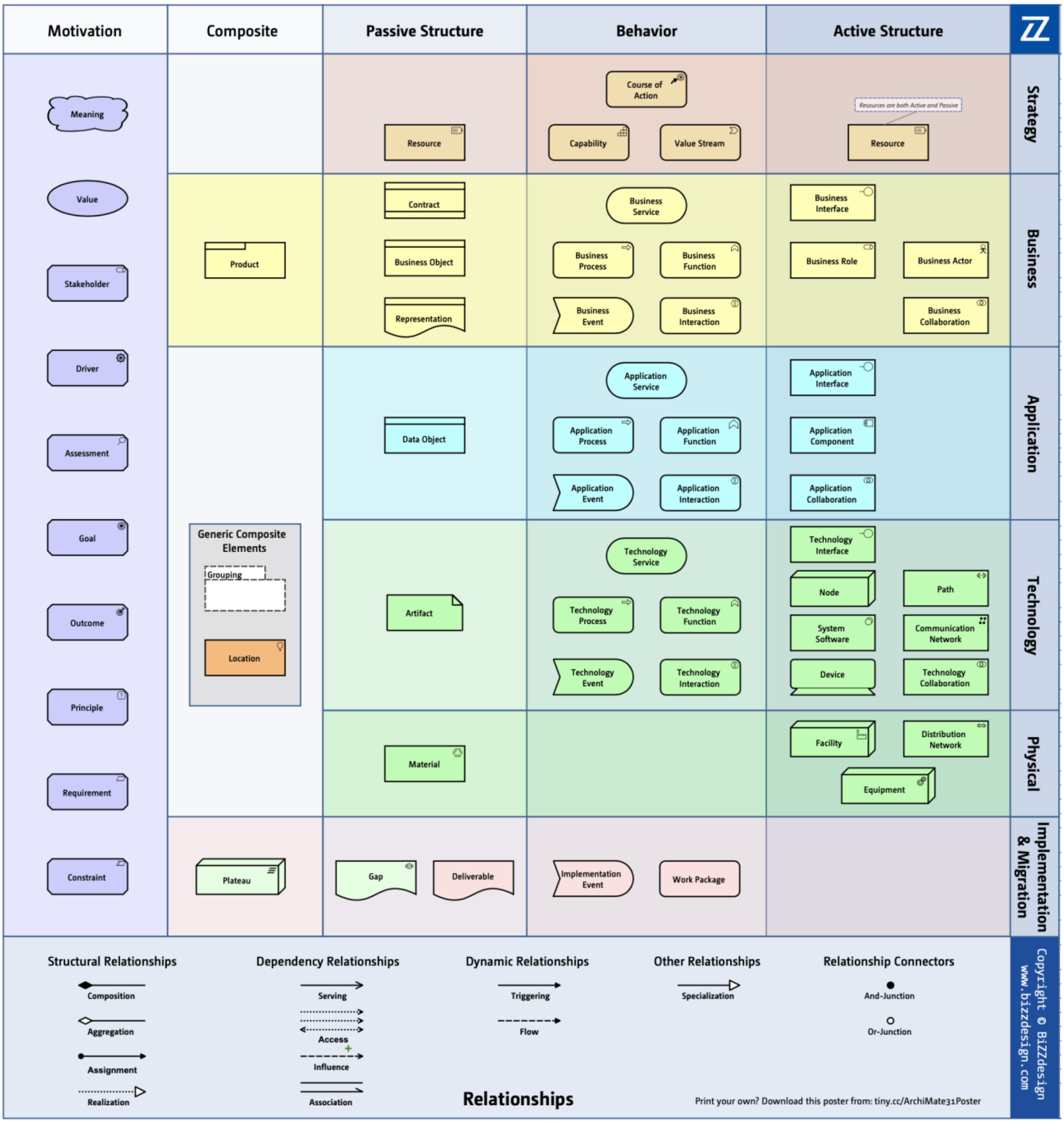
Figure 127 The Enterprise Ontology



A.3 ArchiMate 3.1 Framework

The material in this section, concerning the ArchiMate 3.1 Framework, was downloaded from BiZZdesign's website [219].

Figure 128 ArchiMate 3.1 Metamodel



Concept	Definition
Application Collaboration	An aggregate of two or more application internal active structure elements that work together to perform collective application behavior.
Application Component	An encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.
Application Event	An application state change.
Application Function	Automated behavior that can be performed by an application component.
Application Interaction	A unit of collective application behavior performed by (a collaboration of) two or more application components.
Application Interface	A point of access where application services are made available to a user, another application component, or a node.
Application Process	A sequence of application behaviors that achieves a specific result.
Application Service	An explicitly defined exposed application behavior.
Artifact	A piece of data that is used or produced in a software development process, or by deployment and operation of a system.
Assessment	The result of an analysis of the state of affairs of the enterprise with respect to some driver.
Business Actor	A business entity that is capable of performing behavior.
Business Collaboration	An aggregate of two or more business internal active structure elements that work together to perform collective behavior.
Business Event	An organizational state change.
Business Function	A collection of business behavior based on a chosen set of criteria (typically required business resources and/or competences), closely aligned to an organization, but not necessarily explicitly governed by the organization.
Business Interaction	A unit of collective business behavior performed by (a collaboration of) two or more business actors, business roles, or business collaborations.
Business Interface	A point of access where a business service is made available to the environment.
Business Object	A concept used within a particular business domain.
Business Process	A sequence of business behaviors that achieves a specific outcome such as a defined set of products or business services.
Business Role	The responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.
Business Service	An explicitly defined exposed business behavior.
Capability	An ability that an active structure element, such as an organization, person, or system, possesses.
Communication Network	A set of structures that connects nodes for transmission, routing, and reception of data.
Constraint	A factor that limits the realization of goals.
Contract	A formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.
Course of Action	An approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.
Data Object	Data structured for automated processing.
Deliverable	A precisely-defined result of a work package.
Device	A physical IT resource upon which system software and artifacts may be stored or deployed for execution.
Distribution Network	A physical network used to transport materials or energy.
Driver	An external or internal condition that motivates an organization to define its goals and implement the changes necessary to achieve them.
Equipment	One or more physical machines, tools, or instruments that can create, use, store, move, or transform materials.
Facility	A physical structure or environment.
Gap	A statement of difference between two plateaus.
Goal	A high-level statement of intent, direction, or desired end state for an organization and its stakeholders.
Grouping	The grouping element aggregates or composes concepts that belong together based on some common characteristic.
Implementation Event	A behavior element that denotes a state change related to implementation or migration.
Location	A place or position where structure elements can be located or behavior can be performed.
Material	Tangible physical matter or energy.
Meaning	The knowledge or expertise present in, or the interpretation given to, a core element in a particular context.
Node	A computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.
Outcome	An end result that has been achieved.
Path	A link between two or more nodes, through which these nodes can exchange data, energy, or material.
Plateau	A relatively stable state of the architecture that exists during a limited period of time.
Principle	A statement of intent defining a general property that applies to any system in a certain context in the architecture.
Product	A coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.
Representation	A perceptible form of the information carried by a business object.
Requirement	A statement of need defining a property that applies to a specific system as described by the architecture.
Resource	An asset owned or controlled by an individual or organization.
Stakeholder	The role of an individual, team, or organization (or classes thereof) that their interests in the effects of the architecture.
System Software	Software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.
Technology Collaboration	An aggregate of two or more technology internal active structure elements that work together to perform collective technology behavior.
Technology Event	A technology state change.
Technology Function	A collection of technology behavior that can be performed by a node.
Technology Interaction	A unit of collective technology behavior performed by (a collaboration of) two or more nodes.
Technology Interface	A point of access where technology services offered by a node can be accessed.
Technology Process	A sequence of technology behaviors that achieves a specific result.
Technology Service	An explicitly defined exposed technology behavior.
Value	The relative worth, utility, or importance of a concept.
Value Stream	A sequence of activities that create an overall result for a customer, stakeholder, or end user.
Work Package	A series of actions identified and designed to achieve specific results within specified time and resource constraints.

Appendix B Objectives and Deliverables

Table 53 Objectives

Objective 1.1 Construct a framework to measure the quality of a set of Enterprise Architecture models
Objective 1.2. Test the framework with a case study
Objective 2.1. Construct a framework and methodology to measure the effectiveness of an EA modelling language
Objective 2.2. Test the framework and methodology on an actual EA modelling language (ArchiMate)
Objective 2.3. Analyse and interpret results of research on ArchiMate
Objective 3.1. Demonstrate transformations required to tailor models
Objective 3.2. Evaluate which of the transformations are deterministic
Objective 3.3. Demonstrate the extent to which they can or have been done in practice

Table 54 Deliverables

Deliverable 1.1. Framework for measuring the quality of a set of Enterprise Architecture models
Deliverable 1.2. Case study showing results of applying framework to a specific real-life scenario
Deliverable 2.1. Framework and methodology for measuring the effectiveness of an EA modelling language
Deliverable 2.2A. Test materials for use when testing ArchiMate
Deliverable 2.2B. Results of primary research on ArchiMate
Deliverable 2.3. Analysis and interpretation of results of ArchiMate research
Deliverable 3.1. Set of transformation types for EA models
Deliverable 3.2: Evaluation of transformation algorithms with regard to determinism
Deliverable 3.3. Evaluation of the practicality of implementing the transformation algorithms

Appendix C Ethics Approval Documentation

C.1 Completed Ethics Approval Form

Henley Business School Appendix 1

HENLEY
UNIVERSITY OF READING

School of Management
Research Ethics Committee

Application for Research Project Approval

Introduction

The University Research Ethics Committee allows Schools to operate their own ethical procedures within guidelines laid down by the Committee. The University Research Ethics Committee policies are explained in their Notes for Guidance (see the link to "Guidance Notes (PDF – 299kb)" which can be found at http://www.reading.ac.uk/internal/res/ResearchEthics/reas-REethicshomepage.aspx) .

The School of Management (SoM) has its own Research Ethics Committee and can approve project proposals under the exceptions procedure outlined in the Notes for Guidance. Also note that various professional codes of conduct offer guidance even where investigations do not fall within the definition of research (e.g. Chartered Institute of Marketing, Market Research Society, etc). A diagram of the SoM Research Ethics process is appended to this form.

Guidelines for Completion

- If you believe that your project is suitable for approval by the SoM's Research Ethics Committee you should complete this form and return it to the Chair of the Committee. Note that ethical issues may arise even if the data is in the public domain and/or it refers to deceased persons.
- Committee approval must be obtained before the research project commences.
- There is an obligation on all students and academic staff to observe ethical procedures and practice and actively bring to the attention of the SoM's Research Ethics Committee any concerns or questions of clarification they may have.
- Records will be maintained and progress monitored as required by the University Research Ethics Committee, overseen by the School Ethics Committee
- This form should be completed by the student/member of academic staff as appropriate. **All forms must be signed by a member of the academic staff before submission.**
- This form is designed to conform to the University's requirements with respect to research ethics. Approval under this procedure does not necessarily confirm the academic validity of the proposed project.

- All **five** parts of the form and **all** questions must be completed. Incomplete forms will be returned. Students should submit forms to their supervisor, who together with staff should pass these to the SoMREC.
- **Student research projects** - initial approval may be given by the academic supervisor. **At the completion of the project students should submit a further copy of the form** to confirm that the research was conducted in the approved manner. **The project will not be marked until this form is received**. If in the course of work the nature of the project changes advice should be sought from the academic supervisor.

1. Project details

Date of submission: 2019/2020 Student No (if applicable): **019024084**

Title of Proposed Project:-

Measuring the Effectiveness and Efficiency of EA Models

Responsible Persons

Cameron Spence c.d.spence@pgr.reading.ac.uk, cameron.spence@thameswater.co.uk

Dr Vaughan Ashley Michell v.a.michell@reading.ac.uk

Nature of Project (mark with a 'x' as appropriate)

Staff research		Masters	
Undergraduate		Doctoral	X
MBA		Other	

(Student research projects should be signed off in **section 2. 3** below by the supervisor)

(Staff research projects should be signed off in **section 2. 4** below by the Research Ethics Committee)

Brief Summary of Proposed Project and Research Methods

TITLE OF RESEARCH

Factors Affecting the Quality of Enterprise Architecture Models

Brief Introduce the reason and background for the research

I am seeking to understand how individual's ability to understand architectural diagram in a particular modelling language varies with their personal background, in order to judge the effectiveness of the modelling language in question.

Research methods: explain the approach briefly

- ➔ Mixed (qualitative + quantitative) research:
- ➔ At this stage I will be conducting a study using a combination of open and closed questions.
- ➔ Target Audience: Business professionals from a variety of backgrounds

➔ Interview topics: Interviewees will be asked some questions about themselves to enable them to be characterised in various ways. They will then be given sample diagrams and asked questions about them.

Sample size: The sample size will be of the order of 100.

X	I confirm that where appropriate an information sheet and consent form has been prepared and will be made available to all participants. This contains details of the project, contact details for the principal researcher and advises subjects that their privacy will be protected and that their participation is voluntary and that they may withdraw at any time without reason.
X	I confirm that research instruments (questionnaires, interview guides, etc) have been reviewed against the policies and criteria noted in The University Research Ethics Committee Notes for Guidance. Information obtained will be safeguarded and personal privacy and commercial confidentiality will be strictly observed.
X	I confirm that **any** related documents which would include any questionnaires, interview schedules etc, and, where appropriate, a copy of the **Information Sheet**, **Consent Form** are attached and submitted with this application.

2. School Research Ethics Committee Decision (*delete as appropriate*)

2.1 I have reviewed this application as **APPROVED** and confirm that it is consistent with the requirements of the University Research Ethics Committee procedures

2.2 This proposal is **NOT APPROVED** and is returned to the applicant for further consideration and/or submission to the University Research Ethics Committee

2. 3. For student and programme member projects

SUPERVISOR – AT START OF PROJECT	STUDENT – ON COMPLETION OF PROJECT
Date:-	Date:-
Signed (Supervisor)	Signed (programme member or student)
& Print Name	& Print Name
(before start of project)	(on completion of project)

2. 4. For staff research projects

Signed:

(School Research Ethics Committee Chair or member)

3. Please reply to **all** of the following questions concerning your proposed research

If these questions cannot be confirmed please contact your supervisor.

		Yes	No
1.	Are the participants and subjects of the study patients and clients of the NHS or social services to the best of your knowledge?		X
2.	Are the participants and subjects of the study subject to the Mental Capacity Act 2005 to the best of your knowledge (and therefore unable to give free and informed consent)?		X
3.	Are you asking questions that are likely to be considered impertinent or to cause distress to any of the participants?		X
4.	Are any of the subjects in a special relationship with the researcher?		X
5.	Is your project funded by a Research Council or other external source (excluding research conducted by postgraduate students)?		X

If you have answered **YES** to **any** of these questions, refer to the University's Research Ethics Committee. If you are unsure about whether any of these conditions apply, please contact the secretary of the University Research Ethics Committee, Nathan Helsby (n.e.helsby@reading.ac.uk), for further advice.

4. Please respond to **all** the following questions concerning your proposed research project

		Yes	No
1.	The research only involves archival research, access of company documents/records, access of publicly available data, questionnaires, surveys, focus groups and/or other interview techniques.	X	
2.	Arrangements for expenses and other payments to participants, if any, have been considered.	X	
3.	Participants will be/have been advised that they may withdraw at any stage if they so wish.	X	
4.	Issues of confidentiality and arrangements for the storage and security of material during and after the project and for the disposal of material have been considered.	X	
5.	Arrangements for providing subjects with research results if they wish to have them have been considered.	X	
6.	The arrangements for publishing the research results and, if confidentiality might be affected, for obtaining written consent of this have been considered.	X	
7.	Information Sheets and Consent Forms had been prepared in line with University guidelines for distribution to participants.	X	
8.	Arrangements for the completed consent forms to be retained upon completion of the project have been made.	X	

If you have answered **NO** to **any** of these questions, contact your supervisor if applicable, staff members should refer to the SoM Research Ethics Committee.

If the research is to be conducted outside of an office environment or normal place of work and/or outside normal working hours please note the details below and comment on how the personal safety and security of the researcher(s) has been safeguarded.

Interviews would be scheduled and located strictly within normal working hours and in an office environment to ensure the subjects of the interview are not subjected any additional stress on this account whatsoever.

Please confirm that at the conclusion of the project primary data will be:-

Destroyed		Submitted to the Research Ethics Committee	X

For SoM Research Ethics Committee use Comments

Henley Business School Appendix 2

School of Management
Research Ethics Committee

HENLEY
UNIVERSITY OF READING

Consent Form

1. I have read and had explained to me by

……………………………………………..…

the accompanying Information Sheet relating to the project on:

Assessing the effectiveness and efficiency of EA modelling Languages

2. I have had explained to me the purposes of the project and what will be required of me, and any questions I had have been answered to my satisfaction. I agree to the arrangements described in the Information Sheet in so far as they relate to my participation.

3. I understand that participation is entirely voluntary and that I have the right to withdraw from the project at any time, and that this will be without detriment.

4. This application has been reviewed by the School of Management Research Ethics Committee and has been given a favourable ethical opinion for conduct.

5. I have received a copy of this Consent Form and of the accompanying Information Sheet.

Name: ………………………………………………………………………………

Date of birth: ……………………………………………………………………

Signed: ………………………………………...………………………………

Date: ……………………………………………………………………………….

Henley Business School Appendix 3

School of Management
Research Ethics Committee

Please find below an indicative list of information for the interview subject based on the guidance provided by the Research Ethics Committee, School of Management, Henley Business School:

Contact Details

Cameron Spence (doctoral student)

Mobile:

E-mail: cameron.spence@thameswater.co.uk; c.d.spence@pgr.reading.ac.uk

Dr Vaughan Ashley Michell (Supervisor)

Tel:

E-mail: v.a.michell@reading.ac.uk

Project brief

I (Cameron) am seeking to understand the quality of enterprise architecture models, and the languages used in the models. As part of this, I am seeking to understand the effectiveness and efficiency of elements of one particular modelling language (ArchiMate 3.0).

This particular piece of research is designed to answer two questions:

- Which elements of this language are useful when communicating with certain types of stakeholder?
- Which elements of this language are understandable when communicating with certain types of stakeholder?

To help me do this, I need to understand some information about you (the interviewee), to help me characterise you and thus draw conclusions about how the language works for "people like you". I will be asking you to look at one or more diagrams in conjunction with a narrative description that accompanies each diagram. I will then ask you questions to understand how well certain information was received by you.

You yourself are not being tested or evaluated by this survey. The survey is designed to evaluate the suitability of elements of a modelling language for use when communicating with people like you. Please be completely honest about which things you understand and which you do not. My hope is that understanding what bits of the ArchiMate language is useful with certain types of people will enable people producing such models to better tailor the models for use when communicating with people like yourself.

Expenses and payments

This is entirely a voluntary initiative and there would be no financial benefit or re-imbursement of expenses incurred on participating in the research.

Withdrawal

The interview participants can withdraw at any stage.

Confidentiality Agreement

The material collected would be submitted to the School of Management and an authorised signatory would sign on the confidentiality agreement with the subject of the interview. Further, if any of the research results are published and if confidentiality of any information shared by the subject is affected, a written consent would be obtained from the subject. I am planning to anonymise the data so that confidentially is not affected.

Research results

The research results would be provided if any interview subject might want to.

Ethical review

The entire research process undertaken for this project has undergone a thorough review and approval process by the School of Management Research Ethics committee. The project has been given a favourable ethical opinion for conduct.

C.2 Compliance with Ethics Agreement

In the ethics form completed before starting the surveys (shown above), we laid out how we thought the data gathering would proceed, and in practice, there are two areas in which we have diverged from what we expected, for good reasons.

Firstly, the form we completed suggested that at the conclusion of the research, our primary data would be submitted to the Research Ethics Committee. Given that the data collected consists purely of Word documents containing views by interviewees of what was meant by diagrams constructed by myself, in hindsight this would appear to be unnecessary, and so we no longer intend to do this; we shall just delete the questionnaires.

Secondly, the ethics form contains the following statement:

> *"If the research is to be conducted outside of an office environment or normal place of work and/or outside normal working hours please note the details below and comment on how the personal safety and security of the researcher(s) has been safeguarded."*

Our original response was:

> *Interviews would be scheduled and located strictly within normal working hours and in an office environment to ensure the subjects of the interview are not subjected any additional stress on this account whatsoever.*

When we wrote this we had intended to interview people purely within ONE organisation, however as the research progressed it became clear that we would need to interview people outside our current organisation, and we there chose to interview people that were already known to us in a professional or personal capacity in other organisations. Given that we, as well as many of those we were interviewing, were in full-time employment, this necessitated meeting some of them outside office hours. These were generally carried out in public places. In a few cases, the interviewees were not face-to-face; these interviews were carried out remotely.

Appendix D Publication: EA Model Quality

Published originally in 2016 [51]; numbering scheme and layout adjusted here to match the overall thesis structure and style. Reprinted with permission from the Journal of Enterprise Architecture.

D.1 Abstract

In this paper we consider how to measure the quality of a set of Enterprise Architecture (EA) models. We review some relevant literature, focusing in particular on conceptual model quality, and adapt a conceptual model for use specifically with sets of EA models. We develop three objective metrics for this purpose, and also consider the conditions necessary for these metrics to converge towards increasing model quality. We conclude with a partial case study where two of these metrics were used in practice.

D.2 Introduction

D.2.1 Context

Enterprise Architecture is widely used to model and analyze businesses, to a greater or lesser degree depending on geography [2, 3] and its practitioners wield a significant amount of influence (either "final decision maker" or "great deal of influence") on over \$ 10^{12} of IT-related spend, according to Gartner [4].

Enterprise Architecture requires the management of complex data sets to satisfy the needs of its various clients, both in the business and technical domains [16]. For example, in the business domain we would include information about services provided and the actors involved; in the technical domain we would normally include applications, platforms and technical infrastructure. This data is assembled and presented in different ways to suit the needs of the stakeholders. For example, some stakeholders will be interested in a financial view; others will be interested in security, or in data replication and duplication. A variety of tools are used to manage the underlying data and build these models, ranging from very rudimentary tools such as basic office productivity software, up to sophisticated software products specifically designed to handle this kind of data (such as those researched by Gartner [17]), for example Sparx Enterprise Architect [18] and Troux [20].

Together, the set of models built and managed by this kind of tool provide a visualization of the data and relationships comprising our 'body of knowledge' of the architecture of the enterprise in question.

D.2.2 The Problem

This set of models has to be built up over a period of time by a number of people with differing experiences and perspectives, who may not necessarily share the same ideas about how best to represent certain concepts within models, especially in the early stages of the development where it is not entirely clear what should actually be in the set of models representing the enterprise. The model is a simplification of the actual business and supporting ICT. The models are partly used to make business decisions, for example, between possible solution alternatives, on the basis of best fit to the organization's strategic and tactical goals. Therefore, if the information presented in those models is inaccurate, then the decisions made upon the basis of those models are made on the basis of inaccurate information, and are thus more likely to be sub-optimum. Therefore, in order to avoid making poor decisions based upon inaccurate models, the models need to be accurate.

Measuring and ensuring the overall quality of the set of models is therefore important – but how? For example, someone sponsoring the production of the enterprise models may well be interested in knowing when the initial set of models is complete. How could such a question be answered? This paper addresses the question of how we can measure the quality of sets of architectural models.

D.3 Existing Literature

D.3.1 What is a Model?

Models are generally defined as explicit representations of some portions of reality as perceived by some actor [92]. Three "features" of a model can be distinguished, according to Stachowiak, translated from the original German by [91]:

Mapping: Models are always models <u>of something</u>, i.e. mappings from, representations of natural or artificial originals, that can be models themselves.

Reduction: Models in general capture <u>not all</u> attributes of the original represented by them, but rather only those seeming relevant to their model creators and/ or model users.

Pragmatism: Models are not uniquely assigned to their originals per se. They fulfil their replacement function a) for <u>particular</u> – cognitive and/ or acting, model using <u>subjects</u>, b) within <u>particular time intervals</u> and c) restricted to <u>particular mental or actual operations</u>. [36]

This suggests a model contains a reduced and hence partial representation of the information available, designed for a particular use at particular times, perhaps to help make particular decisions.

D.3.2 Conceptual Model Quality

Quality attributes of Enterprise Architecture are examined in a paper from 2013 [104] that seeks to define the attributes of high-quality Enterprise Architecture products and services. The paper defines the quality of EA products and services as the extent to which the products and services meet the needs of the EA stakeholders. See also Bernus [105] which discusses quality again in terms of outcomes, for example *efficiency* being defined in terms of conveying the intended meaning; and *completeness* being defined in terms of how the model can be used to create the intended interpretation. These outcomes could perhaps be used to help shape the choice of language used for the modelling products, and the set of viewpoints that sufficiently represent the interests of our particular set of stakeholders, but will not help us in defining the quality of a set of models, *given* a particular choice of modelling language and viewpoints.

We are interested in creating a measure that enables us to calculate, in an objective manner, how close a particular set of EA models is to an ideal state, assuming that the particular choice of modelling language and viewpoints has already been made for us.

Shanks [58] also suggest criteria for validating conceptual models: semantic accuracy, semantic completeness relative to the focal domain, no semantic conflict in model parts, no redundant semantics. In this paper, we are assuming that a set of criteria such as these has already been agreed upon as part of the definition of our 'modelling language', and asking how we can measure the quality of a set of conceptual (EA) models as a whole.

Semantic conflict is a major issue for EA analysis as stakeholders and participants frequently use terms in conflicting ways, for example using different terms to describe what is essentially the same concept (e.g. service and function) as far as those using the terms are concerned.

A framework to measure general conceptual model completeness was developed by Lindland [56] and subsequently extended by Krogstie [106]. The original paper considered model quality from three dimensions, and the subsequent model extended this to six dimensions taken from the field of semiotics. Although Lindland distinguishes between explicit and implicit statements ($\mathcal{M}_{\mathcal{E}}$ and $\mathcal{M}_{\mathcal{I}}$), we will not be making that distinction in this work; we will be dealing purely with explicit statements (models) whose conformance to the language $\mathcal{L}$ can be explicitly tested.

Using these sets, Lindland defines *Syntactic quality* as the degree of correspondence between model $\mathcal{M}$ and $\mathcal{L}$. The set of syntactic errors is the set difference:

M \ L

This is the set of all statements in the model $\mathcal{M}$ that are not part of the language $\mathcal{L}$. In other words, how much of the model is using the wrong language (syntax).

Semantic quality is the degree of correspondence between model and domain.

If the set difference: $\mathcal{M} \setminus \mathcal{D} \neq \emptyset$ then the model contains invalid statements (i.e. the model makes statements that are incorrect, i.e. not in the domain).

If $\mathcal{D} \setminus \mathcal{M} \neq \emptyset$ then the model is incomplete (i.e. there are elements in our Enterprise Architecture domain $\mathcal{D}$ that do not appear in the model $\mathcal{M}$).

Pragmatic quality is the degree of correspondence between model and audience interpretation (i.e., the degree to which the model has been understood).

If $\mathcal{I} \neq \mathcal{M}$ then the model has been misunderstood.

D.3.3 Models in Enterprise Architecture

Enterprise Architecture is perhaps an over-specified phrase having many definitions in the literature, however deals ultimately with the structure and evolution over time of a business and its supporting technology [33].

The Open Group's Architecture Framework, TOGAF [29] explains how models, stored in an architecture repository, use elements from a metamodel to represent the "real-world" enterprise, or perhaps a possible future state of it; and that these models, organized according to stakeholder views, present information designed to help specific stakeholders.

D.4 Application of Conceptual Model Quality to Enterprise Architecture Models

D.4.1 Sets of Models

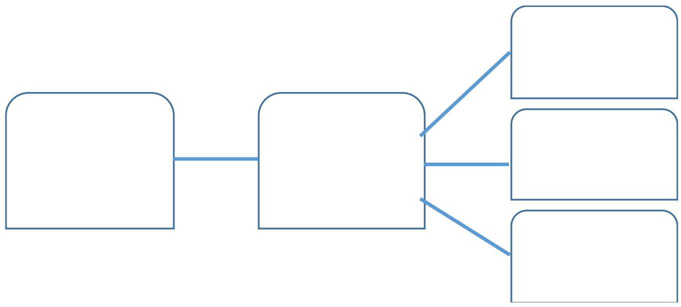
To apply Lindland's theory of conceptual model quality to Enterprise Architecture, we are going to define the scope of our conceptual model to be the set of all EA models in our control, stored within an architecture repository, so relating the terms from Lindland's original theory to the terms in our paper thus:

Table 55 Adaptation of Terminology for Model Quality

Original Theory	Equivalent in this Research
Statement within the Model	One of the models within the repository
Scope of Conceptual Model	The set of all models within the repository

Using the TOGAF enumeration, we expect our models to conform to some kind of agreed language, and to consist of diagrams, lists or matrices:

Figure 129 Types of Models



Syntactic errors can be associated with any diagram that does not correspond to our expected language. The language here we will take to be the model types and contents suggested by our architecture framework. Thus, any model in the scope of that framework, that does not use the style of diagrams suggested by that framework, we can define to be syntactically incorrect. That modelling language can be defined as appropriate; it might be a formally defined language such as UML [220] or ArchiMate [221]; or we can extend it to cover any type of less formal model that we consider appropriate for our enterprise (perhaps defined in Phase 0 of the TOGAF ADM).

Similarly, we can take each model that we have, and if it does not reflect 'reality', either current or planned future, then we define that to be semantically incorrect.

We can therefore adapt Lindland's definitions to this new context:

$\mathcal{M}$ is the set of all the EA *models* within our architecture repository (typically each will be a diagram, catalogue or matrix, as per the TOGAF definition), irrespective of their content (subject) or format; each of the models being a simplified, tailored view of (a possible) reality designed to meet the needs of a particular stakeholder;

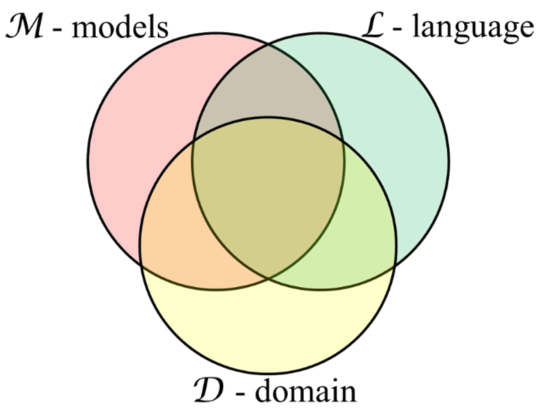
$\mathcal{L}$ is the language, i.e., the set of all statements which are possible to make according to the vocabulary and grammar of the EA language(s) that we have agreed to use in our repository, for example use case diagrams from UML or structural or behavioral diagrams from ArchiMate;

$\mathcal{D}$ is the domain, the boundary of our enterprise architecture, both the current state of the business and its supporting ICT, and possible future states (options or alternatives) that we are investigating. More precisely, this is the full set of models that we would expect to see in order to fully and accurately describe the domain of interest. In TOGAF terms, this means the set of models that fully populates the required views and viewpoints for all the stakeholders in scope. An example of a model in $\mathcal{D}$ would be a diagram that showed an existing business service (e.g. Intelligence Management, in the policing sector) relating to a new Intelligence system that is being acquired, because this represents a possible (indeed planned) reality;

We are focused on the syntactic (relating to tokens and language) and semantic (relating to meaning) views. The pragmatic elements are out of scope at this stage

We can visualize the three sets $\mathcal{M}$, $\mathcal{L}$ and $\mathcal{D}$ thus:

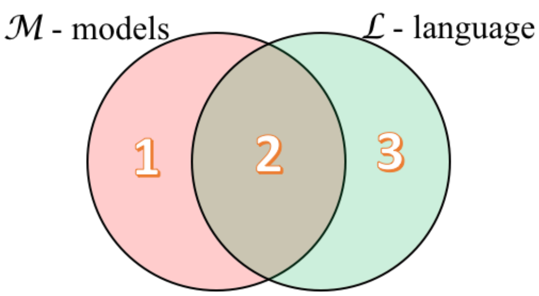
Figure 130 Intersection of Models, Domain and Language



An assumption we are making at this stage is that anything in the domain $\mathcal{D}$ is in theory capable of being modelled in language $\mathcal{L}$. That is, $\mathcal{L}$ is sufficiently comprehensive to be capable in principle of modelling the whole of $\mathcal{D}$.

Each individual model m within the overall set $\mathcal{M}$ can be considered as a point within one or more of the circles. It is valuable to consider the precise meaning of each of these regions, as they will have a bearing on our quality metrics defined later on.

Figure 131 M and L Regions

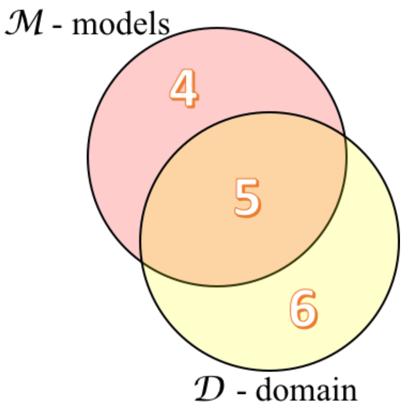


Region 1 represents existing models (diagrams, matrices or catalogues) that are not compliant or written within the agreed modelling notation (language) for our enterprise, or for a particular architecture repository.

Region 2 represents existing models that are compliant and written as (syntactically) correct statements in the modelling language.

Region 3 represents models, or types of models, that would be compliant with our language $\mathcal{L}$, but which have not been created (perhaps because they are not required). An example of this might be UML interaction diagrams, if our agreed notation is UML but we have only hitherto used (say) sequence and use case diagrams. We cannot in this diagram indicate relevance to the real world because the domain $\mathcal{D}$ does not appear in it.

Figure 132 M and D Regions



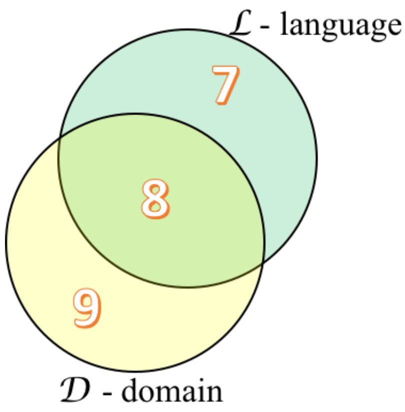
Region 4 represents existing models that do not reflect the domain, either its current state or a possible future state. An example of this might be a model that was created some time ago, reflecting what was deployed previously, but which is now out of date because the environment has since changed. Models in this region (or subregions thereof) we classify as inaccurate, so count against our accuracy metric Q_A, defined later.

The modelling language or syntax irrelevant to regions 4, 5 and 6 because the modelling language $\mathcal{L}$ is not included in this diagram.

Region 5 represents existing models that do accurately reflect what currently exists, or a possible option, which may or may not be compliant with our modelling language.

Region 6 represents models that should exist (i.e. would be required to fully populate the required views and viewpoints), but do not. For example, if we have decided that as a standard, we should have a particular kind of model for every business service, explaining the purpose of the models, its various stakeholders, and the resources that it requires, then every model that exists, we would put in region 5 (assuming that it is accurate), and every business service model that is missing, we would put in region 6.

Figure 133 L and D Regions



Region 7 represents existing or possible models (note that the set of models that actually exists, $\mathcal{M}$, is excluded from this diagram) that are compliant with our agreed language $\mathcal{L}$, but which are outside our domain of interest.

Region 8 represents models (existing or possible) within our domain of interest (actual or possible reality) that are also compliant with $\mathcal{L}$.

Region 9 represents models (existing or possible) within our domain of interest that are in theory capable of being represented in a form compliant with $\mathcal{L}$, but which are not in fact compliant with $\mathcal{L}$. This assumes again that all relevant truth about $\mathcal{D}$ can be modelled in $\mathcal{L}$; we consider how to deal with exceptions to this in a later section.

Further meaning can be elaborated for the smaller regions (for example the intersection of $\mathcal{M}$, $\mathcal{L}$ and $\mathcal{D}$), but they are not necessary for our analysis here.

Models in these regions are summarized mathematically as:

Table 56 Mathematical Definitions of Model Regions

Region	Definition	Region	Definition
1	$m \in M \setminus L$	6	$m \in D \setminus M$
2	$m \in M \cap L$	7	$m \in L \setminus D$
3	$m \in L \setminus M$	8	$m \in L \cap D$

Region	Definition	Region	Definition
4	$m \in M \setminus D$	9	$m \in D \setminus L$
5	$m \in M \cap D$		

D.4.2 Defining Quality Metrics

We can use this approach to define three quality metrics for sets of architecture models, adapting Lindland's method:

D.4.2.1 Syntactical Quality

We define the syntactical quality of the overall set of EA models by comparing the total number of correctly formatted models that exist (i.e. models conformant with language $\mathcal{L}$) that exist with the total number of models (irrespective of format or syntax), considering therefore the ratio of the populations of region 2 and $\mathcal{M}$:

$$Q_S = \frac{|M \cap L|}{|M|} \qquad (1)$$

In other words, this is the number of EA models corresponding to our defined language $\mathcal{L}$, divided by the number of models. Obviously,

$$0 \leq Q_S \leq 1$$

If $Q_S = 1$ then all of the models are as defined by our modelling language. We discuss later a possible extension to this definition.

D.4.2.2 Semantic Quality

Similarly, we can consider a different intersection to determine whether our models are truthful, by defining the semantic quality, or Accuracy, of the overall set of EA models. This would use as the numerator the population of region 5 thus:

$$Q_A = \frac{|M \cap D|}{|M|} \qquad (2)$$

In other words, this is the number of models corresponding to our defined domain $\mathcal{D}$, divided by the number of models. Obviously,

$$0 \leq Q_A \leq 1$$

If $Q_A = 1$ then all of the models accurately reflect our domain, that is, they all reflect currently reality, or a possible reality (may make sense in the Opportunities and Solutions phase of the ADM when we consider a variety of possible implementations).

D.4.2.3 Completeness

This metric is used when considering whether the set of models completely reflects the domain in question. In other words, have we done enough modelling to satisfy all of our stakeholders, with their different viewpoints? Do we have all the models that we would expect (or need) to have? We define the completeness of the overall set of EA models by comparing the population of region thus:

$$Q_C = \frac{|\, M \cap D \,|}{|\, D \,|} \qquad (3)$$

In other words, this is the number of models corresponding to our defined domain $\mathcal{D}$, divided by the number of models that we would expect to find in our domain. Obviously,

$$0 \leq Q_C \leq 1$$

If $Q_C = 1$ then we have completely modelled our domain.

These three measures – Q_S, Q_A and Q_C – provide quantitative metrics that enable us to measure the completeness and quality of a set of EA models with a reasonable degree of objectivity.

Of course, an absolutely ideal set of EA models is one where:

$Q_S = Q_A = Q_C = 1$

D.4.3 Quality Convergence

Also of interest is the set of conditions that need to be satisfied in order to make positive progress in increasing the quality of our set of models. If we differentiate these definitions with respect to time, using the quotient rule:

$$\frac{d}{dt}\left(\frac{f(t)}{g(t)}\right) = \frac{f'(t).g(t) - f(t).g'(t)}{g(t)^2}$$

then we obtain the following equations showing how these quality measures change over time:

D.4.3.1 Convergence of Syntactical Quality

Applying the quotient rule to (1) we obtain:

$$\frac{d}{dt}Q_S = \frac{|\, M \cap L \,|'.|\, M \,| - |\, M \cap L \,|.|\, M \,|'}{|\, M \,|^2} \qquad (4)$$

But in order for this to be positive (i.e. for Q_S to increase over time), knowing that $|\, M \,|^2$ will always be positive, we require the numerator also to be positive, thus:

$$| M \cap L |'.| M | - | M \cap L |.| M |' > 0 \qquad (5)$$

Now if no new models are being created, and all effort is directed for a time into modifying the existing models to correct their syntax, then $| M |'$ will be zero, and so from inspection, if any models are being modified to comply with the language L then $| M \cap L |'$ will be positive, and thus Q_S will inevitably increase over time. However, if people continue to create new models with the incorrect syntax, then it will be much harder for the overall syntactical quality to increase. We should then require, rearranging (5):

$$\frac{| M \cap L |'}{| M |'} > \frac{| M \cap L |}{| M |} \qquad (6)$$

Put into words, the ratio of the rate of change of correctly formatted models to the rate of change of all models needs to be greater than the ratio of the number of correctly formatted models to the number of all models. If there are a low percentage of correctly formatted models, then it will be relatively easy to achieve this inequality. For example, if only 1 in every 10 models is correctly formatted, then we only require

$$\frac{| M \cap L |'}{| M |'} > 0.1$$

in order to make progress, which means that the ratio of new correctly-formatted models to the number of new incorrectly-formatted models just needs to be more than this (relatively low) target. We can see however that as our overall model quality increases, it would be relatively easy for the overall syntactic quality to slip backwards as we require the above ratio to be higher and higher, approaching 1.

D.4.3.2 Convergence of Semantical Quality

Applying the quotient rule to (2) we obtain:

$$\frac{d}{dt} Q_A = \frac{| M \cap D |'.| M | - | M \cap D |.| M |'}{| M |^2} \qquad (7)$$

But in order for this to be positive (i.e. for Q_S to increase over time), knowing that $| M |^2$ will always be positive, we require:

$$| M \cap D |'.| M | - | M \cap D |.| M |' > 0 \qquad (8)$$

Now if no new models are being created, and all effort is directed for a time into modifying the existing models to make them accurate (i.e. describe what actually exists, or might (a solution option)), then as with syntactical quality, $| M |'$ will be zero, and so from inspection, if any models are being modified to make them accurate, then $| M \cap D |'$ will be positive, and thus Q_A will inevitably increase over time. However, if people continue to create new models that do not reflect [a possible] reality, then it will be much harder for the overall semantical quality to increase. We should then require, rearranging (7):

$$\frac{|\, M \cap D \,|'}{|\, M \,|'} > \frac{|\, M \cap D \,|}{|\, M \,|} \qquad (9)$$

Put into words, the ratio of the rate of change of accurate models to the rate of change of all models needs to be greater than the ratio of the number of accurate models to the number of all models.

As discussed in the previous section on syntactical quality, this task may get harder as the overall semantical quality of the model [set] increases.

D.4.3.3 Convergence of Completeness

Applying the quotient rule to (3) we obtain:

$$\frac{d}{dt} Q_C = \frac{|\, M \cap D \,|' . |\, M \,| - |\, M \cap D \,| . |\, M \,|'}{|\, D \,|^2} \qquad (10)$$

But in order for this to be positive (i.e. Q_S should increase over time), knowing that $|\, D \,|^2$ will always be positive, we require:

$$|\, M \ \cap \ D \,|' . |\, M \,| - |\, M \ \cap \ D \,| . |\, M \,|' > 0 \qquad (11)$$

This is of course exactly the same condition that we require for the semantical quality to increase; the choice of denominator (number of models that exist, |M|, or number of models that should exist, |D|) is not determinative of the direction of travel because it is squared and thus always positive. The denominator just affects the magnitude of the rate of change of the quality measures. The more models that <u>actually</u> exist, the slower it will be, for a given modelling effort, to increase Q_A, whereas the more models that <u>should</u> exist, the slower it will be to change Q_C.

D.5 Extension for Evolving Maturity

For organizations that are relatively new to architecture frameworks, one further extension of our calculations may be helpful. There may be occasions when modelling efforts start without having a clear scope or framework in mind. Later on, when the scope has become clearer, our understanding of the appropriate language for our models may become clearer.

However, in the early stages of modelling, our language may not be complete. In other words, there may be statements of truth in domain $\mathcal{D}$ which are not actually expressible in language $\mathcal{L}$, which corresponds to regions in Figure 130 above, which are inside $\mathcal{D}$ but outside $\mathcal{L}$.

This might apply in particular to free-format models that are designed in particular for business users for whom formal languages (such as UML) would be inappropriate. Another example might be a desire to incorporate process models with swimlanes in our models, where we have restricted ourselves at the current time to using only ArchiMate which does not support these kind of diagrams.

Now we could just extend our language definition $\mathcal{L}$ to include such free-format models. However, we feel that it is useful to recognize this potential disconnect between what we want to be able to model (domain $\mathcal{D}$) and our understanding of the best way of modelling this (modelling language $\mathcal{L}$). Although the original concept as discussed in Lindland did not allow for this, we feel this is a useful extension for situations where the modelling language is still evolving to take into account more and more of an existing EA model content. The EAF[2] [86] may be useful when comparing EA frameworks to see which of them are able to model what kind of concepts.

Thus, when we count the number of models in $\mathcal{M}$, when calculating the syntactical quality of the set of models, we may wish to exclude those that are not currently expressible (whose format is not prescribed) in $\mathcal{L}$. This will have a direct effect on how we actually assess the quality of an EA model.

However, with our current definition of Region 9, there is no way of making this distinction (between models that should exist, but are outside the bounds of our language, and those that should exist, and are not). Recall that our definition of this region is that it contains all model that should exist (in our domain $\mathcal{D}$) that are expressible in language $\mathcal{L}$, but are not in fact conformant to $\mathcal{L}$. We can in principle consider a subset of all models $\mathcal{M}$ that are currently outside the scope of our formal language $\mathcal{L}$. That is, they cannot be expressed in $\mathcal{L}$ even if we wanted to, and even though they describe part of our domain $\mathcal{D}$.

We can say that:

Models falling inside the scope of language $\mathcal{L}$ comprise set $\mathcal{M}_S$

Models falling outside the scope of language $\mathcal{L}$ comprise set $\mathcal{M}_O$

The whole set of models, $\mathcal{M} = \mathcal{M}_S \cup \mathcal{M}_O$ (the union of the two sets).

A model is either inside or outside the language, so $\mathcal{M}_S \cap \mathcal{M}_O = \emptyset$ (none are in both)

Thus, for syntactical quality, we are only concerned about models that <u>should</u> conform to the language but <u>don't</u>, so we need to revise definition (1) above to give:

$$Q_S = \frac{|\, M \cap L \,|}{|\, M_S \,|} \qquad (12)$$

The time derivative of this is given by:

$$\frac{d}{dt}Q_S = \frac{|M \cap L|'.|M_S| - |M \cap L|.|M_S|'}{|M_S|^2} \qquad (13)$$

and in a similar manner to before, we can see that for this to converge towards 1, we should require:

$$\frac{|M \cap L|'}{|M_S|'} > \frac{|M \cap L|}{|M_S|}$$

D.6 Summary of Quality Metrics

We have constructed three normalised measures of the quality and completeness of an overall EA model, related to syntax, truthfulness (accuracy) and completeness, and also looked at how they change over time:

Table 57 Summary of Quality Metrics

Metric	Rate of Change of Metric
$Q_S = \frac{\mid M \cap L \mid}{\mid M_S \mid}$ (12)	$Q_S' = \frac{\mid M \cap L\mid'.\mid M_S\mid - \mid M \cap L\mid.\mid M_S\mid'}{\mid M_S\mid^2}$ (13)
$Q_A = \frac{\mid M \cap D \mid}{\mid M \mid}$ (2)	$Q_A' = \frac{\mid M \cap D\mid'.\mid M\mid - \mid M \cap D\mid.\mid M\mid'}{\mid M\mid^2}$
$Q_C = \frac{\mid M \cap D \mid}{\mid D \mid}$ (3)	$Q_C' = \frac{\mid M \cap D\mid'.\mid M\mid - \mid M \cap D\mid.\mid M\mid'}{\mid D\mid^2}$ (10)

D.7 Partial Case Study

D.7.1 Context for Case Study

The study focused on a UK public sector organization where development teams were engaged in a number of delivery projects, documenting their design work in a single enterprise-context syntactical Wiki.

The lead author had been asked to recommend a new structure to be used by these teams, in effect a new architecture framework and a corresponding standard layout for Wiki pages, one page per architectural entity. The question was asked: “how close are we to having the pages complete and in the new layout?”

This provided the motivation to construct an objective methodology for measuring the completeness of a set of models which could then be used in practice in this particular organization to answer the question posed. A 'model' in this context we chose to relate to a complete Wiki page. The intention was that each Wiki page should correspond to a single architectural entity (e.g. a particular business service or solution building block). Thus, our definition of the approved language $\mathcal{L}$ in this scenario included all standard elements that we would expect to see on a standard page for (say) a business service, including a diagram giving it context as well as standard header and footer information giving related information (for example, any information objects used by it – so relationships to other entities).

D.7.2 Counting the Number of Models

The number of models is one of the key figures used to calculate our metrics derived above. Recall that for the syntactical measure Q_S, we are interested only in models that are within the scope of our modelling language $\mathcal{L}$ (our new standard layout, using new architectural terms) whereas for the other quality measures we are interested in all models. When we carried out this exercise, the total number of pages (available from a database report) in our Wiki was 6,598; this is the value we used for $|\ \mathcal{M}\ |$.

For Qs, though, we counted how many pages held content that should comply with $\mathcal{L}$ which we will interpret to mean, how many pages hold content in the scope of our new framework ($\mathcal{M}_S$, not $\mathcal{M}$). Given that we were changing the architectural framework, and in particular introducing some new entities not previously used in this organization, we interpreted this to mean "how many pages hold information describing entities that appear in the new entity metamodel?". Taking a sum of the pages that held such information, we arrived at a total of 2,738 for $|\ \mathcal{M}_S\ |$. (14)

D.7.3 Counting Pages with the Correct Syntax

Only 328 pages were in the correct format. This is a low percentage of the total number of pages, but was to be expected as we were in the very early stages of rolling out a change to the way that we described our enterprise architecture.

Thus, we used a value of 328 for $|\ M\ \cap\ L\ |$. (15)

We were able to do this, although the figures involved are very approximate, due to the uncertainty over the expected population of region $\mathcal{D}$ which requires us to consider all models that we would expect to find. In our situation, we did this by means of the following algorithm:

```
SET expected page count to 0

FOR each entity in the new metamodel
```

```
    IF a complete set of instances exists
  for that entity
    THEN
    Add the number of instances to our expected page count
    OR
    Approximate the number of instances by reference to another
entity
  Add that approximation to our expected page count
    ENDIF
  END LOOP
```

We ran reports on the Wiki, using the above algorithm manually to calculate an approximation to our expected page count. By way of example, leaving aside the contextual layer which we used for overarching governance information such as principles, we assigned entities in our metamodel within one of three layers: conceptual, logical and physical layer; and for Information Architecture in particular, we had one entity in each: a conceptual information entity (which we called a "Business Object"), a logical information entity ("Logical Entity") and a physical information entity ("Physical Entity").

When it came to counting how many models we expected in the Information domain, we already had a complete set of Business Objects, and so we just used the total number of Business Objects to add to our total of expected pages (one page per entity instance). However, for the next layer down, Logical Entities, we did not have a complete set. Therefore, we used the examples where we had populated this layer to estimate how many Logical Entities there were likely to be, on average, for each Business Object. This enabled us to estimate a likely total number of Logical Entity pages that we expected. Obviously, the more complete sets of entities exist, the more accurate this kind of estimation would be.

Across the complete set of entities, we found that whereas we would have expected to see 1,524 models (one for each entity instance, if fully populated), in fact we could only see 674:

$| \mathcal{D} | = 1524$ (16)

$| M \cap D | = 674$ (17)

D.7.4 Assessment of Current Quality

We were then in a position to calculate two of our three quality measures for the current set of architecture models.

D.7.4.1 Syntactical Quality

Using equation (12), and inserting values from (14) and (15) for the numerator and denominator respectively, we obtain:

$Q_S = \frac{328}{2738} =$ 0.12, or 12%.

So we can say that in terms of syntax, our overall set of models are 12% of the way to being correct. It should not be difficult to increase this quality measure, because the condition that needs to be satisfied for the time derivative to be positive, from (6), is

$$\frac{|\, M \;\cap\; L\,|'}{|\, M\,|'} > \; 0.12$$

In other words, if the number of correct models being produced per unit time is more than 12% of the total number of models being produced (that is, if more than 1 out of every 8 new models is correctly formatted), then we will make progress in increasing the overall model quality with regard to its syntax. We will obviously make progress much faster if all new models created are of the correct syntax – that is, if our development teams stop creating models that use the old syntax. This

D.7.4.2 Completeness

Using equation (3), and inserting values from (17) and (16) for the numerator and denominator, we obtain:

$Q_C = \frac{674}{1524} =$ 0.44, or 44%.

So in terms of completeness, we can describe our overall set of models as being 44% complete.

We will need to work a bit harder now to maintain this completeness figure, because from (11), we can see that we require:

$$\frac{|\, M \;\cap\; D\,|'}{|\, M\,|'} > \; 0.44$$

So if we are expanding the set of models that should exist, then we need to ensure that we actually produce over 44% of what is being asked of us in order for us not to fall behind. As with all these quality figures, of course, we will make progress much faster if we stop producing models with incorrect syntax (to help Q_S) and make concerted efforts to finish the required modelling before increasing the need for more models (to help Q_C).

D.8 Conclusions

We have demonstrated that it is possible to create objective metrics to enable us to measure different aspects of the quality of a set of Enterprise Architecture models. We can measure the quality of their syntax; the quality of their meaning; and their completeness.

D.9 Reflections

We were unable to test the accuracy metric Q_A for practical reasons, although in theory there is no reason why this could not be done given more time. It will be harder to test that “model represents reality” where either:

(a) reality is abstract and/or subjective (e.g. “what is our set of business services” is a more subjective question than “what is the structure of that database”), or

(b) reality is a possible reality – one of the options being explored as part of the Solutions and Opportunities phase.

Two other concepts from the Lindland paper have not been incorporated so far. Firstly, that paper has the concept of ‘feasible completeness’, recognizing that it may not be worth the effort completing 100% of the model. The incorporation of this concept would affect our target of 1.0 for one or more of our metrics.

Secondly, the Lindland paper considers the pragmatic level – how content from the model is interpreted by those perceiving it. This is surely relevant to EA models which are intended to communicate to EA stakeholders.

There are perhaps ways that these difficulties could be addressed. External industry-specific standards can provide reference sets and catalogues of some business and technical artefacts (the TOGAF Technical Reference Model is an example of one of these), against which our models can be compared.

The set representation seems to be self-consistent as developed so far, although it would be preferable perhaps to be able to incorporate the concept of domain knowledge outside the scope of the language in our sets somewhere (the distinction between $\mathcal{M}$ and $\mathcal{M}_I$).

D.10 Further Work

Further research would be useful to explore a number of ideas, some of which are discussed above:

(c) How can we be more certain about the accuracy of models corresponding to either abstract concepts, or realities that do not yet exist?

(d) How do we better incorporate domain knowledge outside the scope of our modelling language into our formal description?

(e) How can we incorporate the idea of 'feasible completeness' in some objective manner, so we can determine when it is not worth pursuing any further increase in our quality metrics?

(f) Can we make some kind of relationship between the quality of our EA models and the corresponding quality of decisions made upon the basis of those EA models?

This latter question in particular seems particularly relevant, because if we knew that better models led to better decisions, then that gives us a stronger motivation to produce better models.

Appendix E Pending Publication: EA Language Quality

Submitted to journal but not yet published, so no reference currently available; numbering scheme and layout adjusted here to match the overall thesis.

E.1Abstract

A method is described for evaluating the effectiveness of Enterprise Architecture languages with different kinds of stakeholders. The method is then used in a survey to evaluate the effectiveness of the ArchiMate language. The results indicate that some parts of ArchiMate are of limited use with some kinds of stakeholders and in addition that the use of ArchiMate notation is detrimental to the understanding of the models when presented in that language.

E.2Introduction

E.2.1 Context

A system can be considered a set of artefacts or an organisation serving a common purpose [67]. The Architecture of a system describes the constituent parts of a system, the way those parts are used together, and the principles governing the ongoing design and evolution of that system [33]. Enterprise Architecture applies this to a system that is a complete Enterprise, in other words something that consists of people, information and technologies; performs business functions, and a number of characteristics summarised in an early definition of Enterprise Architecture [68].It therefore deals with the totality of the business and technical infrastructures, as well as the business and technical strategy, and the alignment between them as described in the Strategic Alignment Model [66] which is critical for business success. The term Solution Architecture deals with the same layers (i.e. business and technical information) but in the context of a single programme or project rather than the whole enterprise.

Enterprise architecture is more often concerned about which initiatives should be carried out, for what reason, and so can be categorised as specifying the WHY and WHAT in terms of those initiatives. Solution architecture is concerned with implementing the initiatives, and thus deals with HOW and WITH WHAT (for example the choice of a specific solution component).

One of the key purposes of enterprise and solution architecture is that of communication, passing information about existing systems as well as planned changes to those systems. The CHAOS report identified that only 16% of software projects complete on time and budget, and met the original specification [34]. One of the key success factors identified was executive management support. Senior managers are key stakeholders in both enterprise and solution architecture, and a key purpose of architecture work is communication of intent in a manner that can be clearly understood by the relevant stakeholders in order to ensure project success.

This communication occurs in a number of directions. In some cases, a project may wish to demonstrate that its plans adhere to the guidelines and constraints imposed by the enterprise, thus demonstrating compliance or conformance. This can be viewed as communicating 'up' to those that have the power to help or hinder a project. In other cases, this communication forms part of the hand-off to other teams (perhaps including specifications, patterns or context for further work), such as envisaged by Model Driven Architecture [35], where the communication might be said to be 'down' to implementation teams.

In both cases (up and down), it is important that the communication is successful, that is, that the information intended to be conveyed, is in fact conveyed and understood accurately. In other words, the meaning (semantics) that the author of the model intended to convey is accurately received by the reader of the model, as envisaged in the definition of a model being a simplification of reality constructed for a particular person with a purpose in mind [36]. This relies upon the sender and receiver knowing the same language, having the same norms, as discussed for example in [37].

However, the stakeholders who ultimately 'consume' the information are very different. When communicating 'up' to perhaps non-technical stakeholders (e.g. CxO executives, or to programme and project managers), the kind of language that we can use is very different to the language that we can use when communicating to technical staff, or indeed to actual technology, in the cases where our models are actually consumed by software (for example, to auto-generate program code or database schemas).

There are a number of different means by which our languages (architecture frameworks) enable us to communicate, including:

- They give us a standard set of words, a language (taxonomy), along with a standard (semantic) interpretation to apply to those words, i.e. an ontology. These words can refer to entities (nouns) in our description of the architecture (e.g. an application, a server); they can also refer to relationships between them (this application is composed of those components). These sets of entities and relations are sometimes referred to as the "entity metamodel" within an architecture framework (the set of allowable nouns and relations between them). This way of communicating works both via the written medium (using words) and also verbally. Thus, we can use the word "capability" or "principle" and expect that both the sender and receiver of the communication both know what is meant by these words, if they are all using the same framework;

- Some (not all) frameworks also specify a graphical notation that enable the reader (if they know the notation) to recognise the type of entity or relation being discussed. For example, TOGAF [29] does not specify any particular notation to use in diagrams, and so without additional context or labels, it would not be possible to know, merely by inspection, whether a particular symbol represented for example business service or a logical application component. By contrast, ArchiMate [39] and UML [40] both have specified symbolic notation that enables one to identify the type of an element, e.g. recognising that something is a capability just by the shape of the element (the stack of small squares in the top right are the key to recognising this particular symbol):

Figure 134 Example ArchiMate Symbol

Both of these (underlying terminology plus graphical notation) are used when communicating with stakeholders. The former is used when giving written reports or giving information verbally. The latter is used typically when presenting diagrams. This paper will use both.

E.2.2 The Problem

EA modelling languages change and evolve regularly. One of the latest such languages, ArchiMate [10], has at the time of writing had four revisions, growing in scope to accommodate additional symbols arising from a more diverse set of ideas to be represented. Undoubtedly the language is getting richer. But is it getting “better”? For example, is it able to communicate more effectively (i.e. actually convey the correct message) with the various stakeholders? How do we even measure this? Anecdotally, architects in some organisations use an EA modelling tool to do their “architecture work”, and then redraw those models in Powerpoint in order to make them comprehensible to business as users. Why is it that the original models are incomprehensible to some stakeholders? Is there something wrong with the tool, or with the language expressed in the tool (UML, ArchiMate etc.)? If we understood what it is about some modelling languages that make them hard to understand for some people, then we might be able to either adapt our modelling languages, or perhaps enable our tools to automatically adjust our models depending on the target audience, by understanding if there is a deterministic way in which models (diagrams) can be altered to suit the needs of different stakeholders.

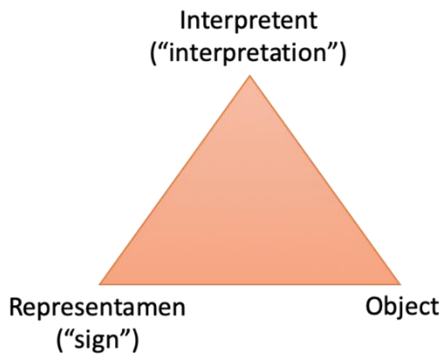
E.3Existing Literature

E.3.1 Model as Communication

As summarised in a related paper [51], models contain a reduced and hence partial representation of the information available, designed for a particular use at particular times, for specific reasons.

Models act as a communication medium between the creator of the model and the consumer of the model with encoding and decoding processes inherent in communicating ideas. Semiotics, the "science of signs" [53] contains a triad commonly depicted as follows:

Figure 135 The Semiotic Triangle



The model contains only signs that we hope convey a correct interpretation that points to the correct real-world objects.

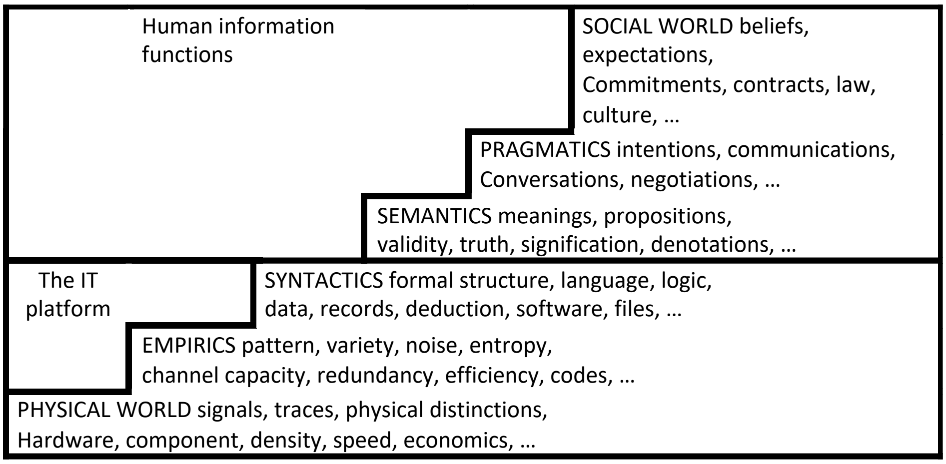
Different individuals with different experiences will "read" different signs as referring to different objects. Two trivial examples are:

- The words that we use (both spoken and written) are obviously culturally relative, and the same word may be interpreted differently in different cultures (even those words that are common to more than one culture), for example the word "pants" mean quite different things in American and English [103]. A another example is the different meaning assigned to colours by various cultures around the world, which has significance for the way that organisations manage their brands [54].
- Exactly the same principle concerns how we communicate concepts related to Enterprise and Solution architecture. The lead author once worked on a portfolio management project with another colleague; and there were misunderstandings between us because one of us had a TOGAF [29] background and one an IAF [55] background, and so when one of us discussed the idea (sign) of a business service, the other got the wrong idea (had a different object in mind).

Understanding the interpretation process is crucial to the successful communication of ideas. The interpretation that we apply comes often from our surrounding culture, perhaps our family or society (in the case of colours or words), or in the case of enterprise architecture terminology, the prevailing nomenclature in our organisation, or the particular framework or notation that we have been taught.

The “semiotic ladder” from [100] identifies the different levels of communication:

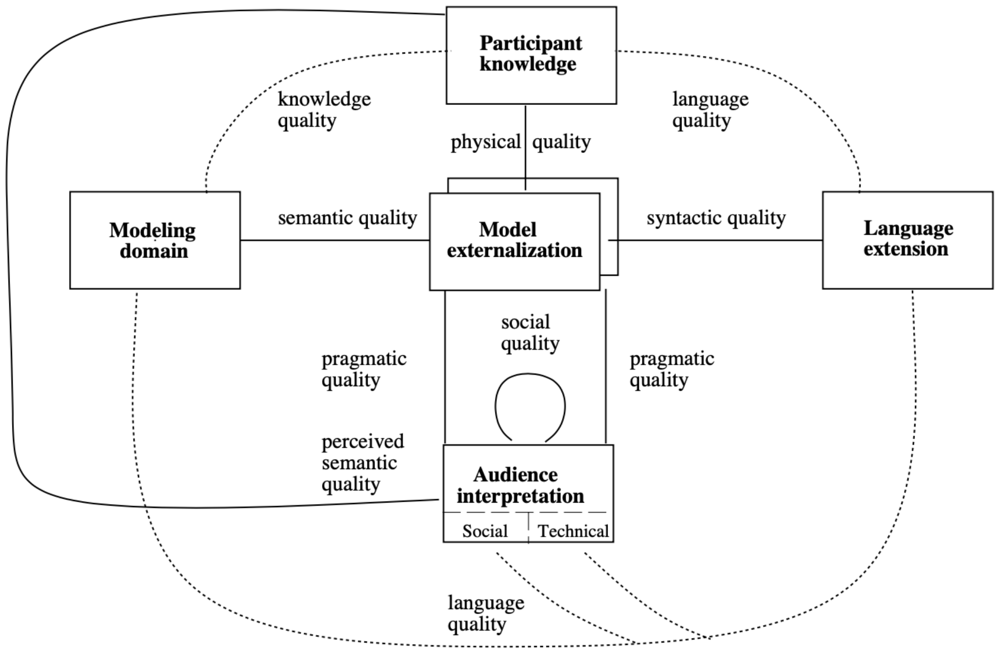
Figure 136 Semiotic ladder between the Physical and the Social World



The definition of our formal modelling languages is encapsulated in the SYNTACTICS layer, which defines the syntax (order and relations) of our modelling languages. The SEMANTICS layer is where we have used our modelling language to represent something that we wish to communicate. However, the mechanics and efficiency of that communication – of that language – relates to the PRAGMATICS layer. It is here that we would wish to test whether or not the ideas that we wished to communicate to our stakeholders, and thus expressed in a particular ‘sentence’ in our language, have actually been properly received and understood correctly.

A framework for measuring general conceptual model quality was developed by Lindland [56] and later extended by Krogstie [48]. The later paper considered model quality from these six dimensions taken from the semiotic ladder. The following illustration is taken from that paper:

Figure 137 Framework for discussing the quality of models, from [48]



Of interest here, in terms of comprehensibility of communication, are the two actors shown explicitly: a social actor and a technical actor. In the original paper, a technical actor is taken to be software that interprets the model; and as previously discussed, a language would have to be rigorous in order for that to be feasible (after all, a programming language is just a specific type of modelling language that describes algorithms). For this reason, no "Participant Knowledge" is required of the technical actor; it is assumed that there "is only one" interpretation of the model, from a technical perspective, assuming that the model has valid syntax (complies with language $\mathcal{L}$). However, for our purposes we will extend this to mean any human actor that takes the model as input for their own work, and thus is expected to understand the language very accurately.

Key to our discussion is the concept of pragmatic quality, defined as is the degree of correspondence between model and reader interpretation [48], the goal of which therefore is accurate comprehension by the reader. More specifically, we are interested in this work in the comprehension of the human (social) actors rather than technical (automated) actors, at least at an enterprise level.

The Krogstie paper suggests that "formal syntax and formal semantics are means for achieving pragmatic quality" [48]. Formal semantics in this context includes techniques such as mathematical descriptions and constraints on the models, treating them as graphs containing nodes and edges, as illustrated for example in [222].

Part of the goal of this work is to question that assumption. Are formal syntax and semantics really the best way of ensuring that all of the stakeholders, for whom our EA models are produced, are able to understand our models? There is perhaps, in the words of the Krogstie paper, a need to balance "expressive economy" (not using too many symbols or words in our modelling language) with a need to maintain sufficient structure so as not to lose the validity of the message. Is it possible that being less formal can, in some cases, actually *aid* comprehension for some readers of our models?

E.3.2 Designing Modelling Notation that Users Can Understand

Popescu [44] examines the effectiveness of two IT-related (but not EA) modelling languages in detail (SEAM and i*), and recommends ways in which the effectiveness of these languages in communicating with non-IT stakeholders can be improved. He highlights confusion amongst his subjects (readers of models) by some of the concepts implicit in the two languages being studied, for example the ideas of system, service and process (section 4.4.1). The purpose of our research is to determine if such confusion also occurs with EA modelling languages such as ArchiMate.

Many of his recommendations deal with <u>sets</u> of models (for example, by using four story phases); his first key recommendation is "the relation with reality, by focusing on the readers' conceptualizations", by for example aligning the models to the readers' own views of reality, on what is visible to them, in terms that are natural to them.

There is thus a need in modelling for accommodations, defined as "conflicting interests that appear in situations that do not fully satisfy every stakeholder but are sufficient to enable actions" [115]. In this case the accommodation that needs to be reached is between the modellers, who may have their ideas about what makes a 'pure' model, and the readers (in particular the social actors) whose interests may be better served by it being comprehensible than being completely accurate and unambiguous.

This answers a related question: "how can we improve modelling languages", but does not help us understand "how good are our existing languages at communicating". An explicit aim of this paper is to measure empirically and mathematically the effectiveness of communication of modelling languages, and hence suggestions for tailoring those languages with particular classes of readers.

E.3.3 The problem with some visual architecture diagrams

Some practitioners [116] suggest the following six ways in which 'architecture visuals' can be made more effective in communicating messages:

(1) Catching the eye using visual clues that highlight instantly key areas to focus on
(2) Using Gestalt principles (see next section), for example grouping or similar colours to highlight similar elements or relations
(3) Application of colour theory

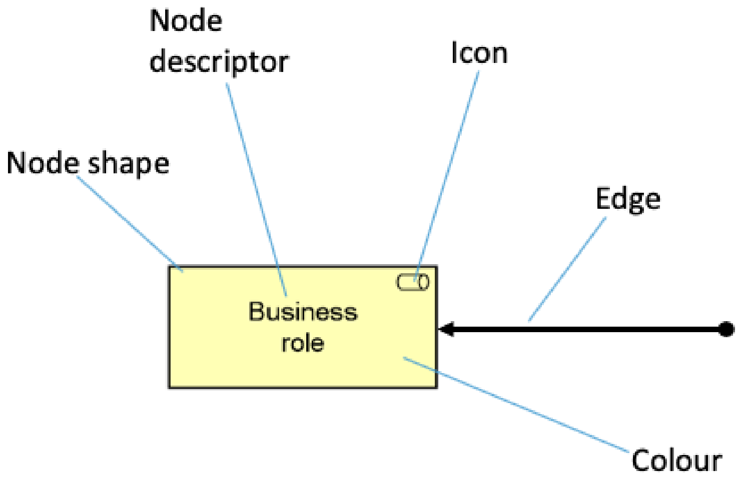
(4) Use of appropriate icons, logos and images (the authors make a good point about the large number of abstract symbols used in ArchiMate)

(5) Use of design thinking and graphic design

(6) Use of informal sketches (hand-drawn)

Just as colours do not always convey messages across cultures, as previously noted, abstract symbols do not either. A particular example of this is the ArchiMate symbol for a business role:

Figure 138 ArchiMate symbol for Business Role



This appears to rely upon an English homonym, whereby a cylindrical shape that could suggest the word "roll" is meant to suggest the completely different concept of a "role" as bearing responsibility for performing specific behaviour. This particular symbol is therefore specific to the English language (although it also works in Dutch), and is therefore is arguably a poor choice of symbol for what is meant to be a global standard. The inappropriateness of this symbol is highlighted as an example in the results section below, where it is mistakenly taken to refer to information (possibly because a confusion with the vertical cylinder sometimes used to denote a database in information systems).

Thus, use of 'appropriate' icons, logos and images that are relevant to the culture and experience of those reading the models is likely to increase comprehension of the models and/or recognition of some of the symbols.

E.3.4 Elements of a Model

Architecture modelling languages work with the components comprising a system as well as relationships between them, and so both need to be represented in the set of symbols used in our language.

For example, the ArchiMate 3.0 language specifies a large number of symbols (icons, shapes and colours) to denote the various language concepts. The symbols describe both a set of entities or concepts (e.g. business services, application components) and also a set of allowable relationships between entities (e.g. composition, aggregation).

As well as the explicit symbols, we also need to bear in mind the implicit messages (e.g. arising from spatial relationships) suggested from the general layout of the symbols, for example arising from the Gestalt "Gestalt Laws of Perceptual Organisation" [117]. These laws show how we experience larger scale constructs by combining simpler constructs according to certain laws or factors. For example, showing several elements in a similar visual style (e.g. colour) or location suggests a grouping of those elements, directly analogous to the 'group' concept in ArchiMate.

Thus, the elements in our model are both explicit and implicit.

Alongside the explicit and implicit symbols, we also consider models to have a *purpose* (what are they trying to convey) [36], which in an architecture context will depend on the role and specific concerns (interests) of the reader. The graphical symbols may well include several visual elements including shape, colour, icons, line types and endings), all of which have a particular significance in ArchiMate [39] as an example.

E.4 Constructing the Measurement Framework

E.4.1 Defining our model content

We want to know if there are symbols on models that, for a certain group of stakeholders, either hold no meaning (were not understood), or do not add to the meaning of the model as a whole (were understood but were not felt to be relevant to the purpose of the model, or even worked against achieving the specified purpose). To do this, we need first to make some definitions that will be useful in our measurements. Let us consider:

(a) a set of models **M** with D members m^d ($1 \leq d \leq D$)
(b) a set of actors **A** with P members a^p ($1 \leq p \leq P$)

The explicit symbols on an architecture diagram will typically consist of **nodes** (often, but not always, closed shapes) **n** (denoting elements – the nouns in our language) joined with **edges** (normally lines with possible extra shapes at either end) **e** (denoting relations between the elements), forming a directed graph. There are other implicit symbols (or signs), for example relative position on the page, colour etc., however for the purpose of our analysis we are just focusing on nodes, and on edges to give context to the nodes. In an ArchiMate context, for example, the nodes might represent applications and services provided by the applications; and the edges might represent the relations between the two (indicating which applications provide which services).

We can refer to these as:

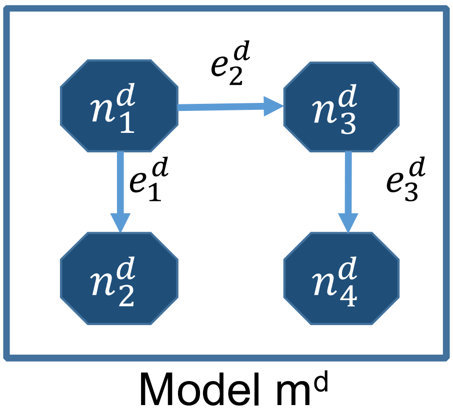
$n_x^d \in N^d$	describes the set of nodes on model M^d
$e_y^d \in E^d$	describes the set of edges on model M^d

We can combine the two to give us our overall set of concepts: elements + relations:

$s_z^d \in S^d = N^d \cup E^d$	the set of all (explicit) symbols on m^d

Here, x, y and z are just index variables that denote the range of the set of nodes and edges. To illustrate this notation, consider the following diagram (or model) consisting of four nodes connected by three edges:

Figure 139 Model, Nodes and Edges



We refer to the number of symbols on a particular model m^d by L^d where *L* can be taken to be the length of the set of symbols on this particular model; we can break this down into $L^d = L_n^d + L_e^d$, where L_n^d is the number of nodes on model m^d and L_e^d the number of edges. Thus we have, for any one model m^d:

$L^d = | S^d |$ $\quad L_n^d = | N^d |$ $\quad L_e^d = | E^d |$

EA modelling languages like ArchiMate have a specific set of elements (e.g. application functions, business processes) and relations (e.g. realises) already defined; and so each of the symbols on our model we would expect to map to one of these predefined concepts: the elements to nodes, and the relations to edges. We thus have mapping functions, which we denote with a lowercase t, with the domain being the set of all elements and relations on our models, and the range being the set of element and relation types defined by our particular choice of language. Thus:

$\forall\, n_x^d \;\exists\, t_N :\; n_x^d \rightarrow T_{EL}$, where T_{EL} is the set of element types;

$\forall\, e_y^d \;\exists\, t_E :\; e_y^d \rightarrow T_{RE}$, where T_{RE} is the set of relationship types; or combining these;

$\forall\, s_z^d \;\exists\, t_S :\; s_z^d \rightarrow T_{CO}$, where T_{CO} is the set of concept (element + relationship) types.

These are necessary (but not necessarily sufficient) conditions for the particular model to be considered conformant to the modelling language being used.

If we define I_C^d to mean the set of instances of a particular concept (element or relation) **c** on model m^d, then we have:

$$I_c^d = \{\, s_z^d \in S^d : t_S(\, s_z^d) = c \,\} \qquad (1)$$

In other words, this is the set of all concept instances that happen to map to a specific concept **c** on this specific model m^d. In other words, the complete set of all concepts used is the set of all element types plus all relationship types on the specified model m^d.

If we sum this over all the models then we obtain the set of all instances of a particular concept **c** on our set of D models:

$$I_c = \bigcup_{d=1}^{D} I_c^d \;= I_c^1 \cup I_c^2 \cup I_c^3 \dots I_c^D \qquad (2)$$

The length of this set of concept instances across all models is therefore:

$$L_c = \sum_{d=1}^{D} \mid I_c^d \mid = \mid I_c \mid \qquad (3)$$

This is the total number of occurrences of a particular concept (element or relation) across all models in our set.

E.4.2 Measuring Comprehension

We are then able to ask a particular actor a^p, for each type of element (concept c) on a model m^d, do they understand what is meant by symbols of the type s_z^d? More specifically, do they understand the meaning behind this particular kind of symbol?

This is of course a subjective question; and so will require triangulation to ensure that we catch ‘false positives’: people who claim to understand which concept is being represented when they do not in fact understand it. We are less concerned about ‘false negatives’, as people are less likely to say that they do not understand a concept when they in fact do.

Enterprise Architecture diagrams do not purely contain graphical symbols: the elements (and sometimes relations) are given descriptive names to indicate the specific instance of a concept (for example, the name of a particular business process or capability, as shown in Figure 134 above.

In the survey used to test this, in our questionnaires we focused only on the elements (nodes), and ignored the relationships (edges), because the elements represented the core of the language and (in our view) the relationships have no real meaning outside of the elements that they connect (for example, the term ‘booking’, as a business process element, has some meaning in its own right, whereas the relational term ‘comprises’ has no inherent meaning unless attached to something else). Thus, in the subsequent analysis, although we will still refer to concepts, they can be understood specifically to refer to elements, rather than relations.

We do not do this purely by asking the interviewees to look at a diagram; we also give them a printed narrative, that describes the content of the test model(s) that they are given. This narrative refers to each of the elements by something close to their descriptive name (that is also on the diagram), and gives some context to their use, such that anyone already familiar with such constructs (the underlying concepts) would recognise them. For example, corresponding to Figure 140, the narrative contains the following text:

The policy administration system must manage *both* access to detailed customer data as well as information on their claims and policies.

There is a close correlation, hopefully close enough for the respondents to recognize, between the following words in the narrative and words on the actual nodes in the diagram:

Table 58 Correspondence between Narrative and Descriptive Text in Diagram

Narrative Fragment	**Node with Text**
policy administration system	
detailed customer data	

information on their claims and policies	

It can be seen that there are nodes on the diagram corresponding to the policy administration system as well as two things that the system does.

So, for model m^d, which contains a number of different element types (denoted t_c^d), we have:

$$\forall\, n_x^d \;\exists\, t_N :\; n_x^d \rightarrow t_c^d \qquad \text{where } t_c^d \in T_{EL} \quad (4)$$

In other words, for every node n_x^d on model m^d, we are able to map it to a specific type t_c^d which is one of the terms in our vocabulary of elements T_{EL}.

Our question therefore relates to each type of element t_c^d on model m^d, as read by the specific actor a^p. We ask the actor a^p if they understand the purpose or meaning of the kind of elements with a given appearance (to enable them to distinguish items of the same type visually), to help us derive a score to assign to our comprehensibility score denoted by h_c^{dp}. We offer the actor (interviewee) a choice of levels of understanding to choose from:

- No clear understanding (score 0)
- A partial understanding of how the items relate (score 0.3)
- Can articulate the commonality but are not familiar with the concept (score 0.7)
- Can articulate the commonality and are familiar with it (score 1) (*Q07, Q10, Q13*)[27]

If the actor claims to understand the concept in part or in whole (i.e. selects any option apart from the first one), then a further question is asked to validate this. The actor is asked to try and characterise this concept: to describe the purpose of the elements of this type, what they have in common. This is obviously more complex than a binary answer, and we search in their narrative response for synonyms or key phrases that correlate to the correct underlying concept. The user is not told if they have it 'right' at the time, as this risks frustrating the user or making them feel that they have made a mistake. If their response indicates that they do understand the concept, in part or whole, then we assign an appropriate score to h_c^{dp}: 0.3 if there is a weak understanding, 0.7 for a stronger understanding, and 1.0 when it is clear that the concept has been understood.

[27] The numbers in red (e..g Q07) indicate which of the questions in the questionnaire (reproduced in Appendix G) are used to capture this specific piece of data.

We wish to weight the scores by the number of instances of a particular concept on each model, which we denote:

$$\mathrm{w}_{\mathrm{c}}^{\mathrm{dp}} = |\ I_c^d\ | \qquad (5)$$

This is so that we may more attention to models that contain many instances of a particular concept than to models that contain few. In practice, for the tests models used in this particular research, each concept only appeared on a single model, and so the weighting factors were not required when calculating the results.

For any models m^d that do not contain concept t_c, we will of course obtain a weight of zero for that concept, thus:

$$\mathrm{w}_{\mathrm{c}}^{\mathrm{dp}} = 0 \text{ if } I_c^d = \emptyset \qquad (6)$$

This is repeated for each model m^d, and for each actor a^p.

We can see that for any concept t_c that the highest possible comprehension score for actor a^p would be the sum of all the weighting values, which we denote thus:

$$\mathrm{H}_{\mathrm{max}}^{\mathrm{cp}} = \sum_{\mathrm{d}=1}^{\mathrm{D}} \mathrm{w}_{\mathrm{c}}^{\mathrm{dp}} \qquad (7)$$

The actual score will be the sum of the responses, each multiplied by the weight for that model, and so is given by:

$$\mathrm{H}_{\mathrm{actual}}^{\mathrm{cp}} = \sum_{\mathrm{d}=1}^{\mathrm{D}} \mathrm{h}_{\mathrm{c}}^{\mathrm{dp}} \,.\, \mathrm{w}_{\mathrm{c}}^{\mathrm{dp}} \qquad (8)$$

In order to normalise the ‘score’ for this concept, for this actor, we then divide the actual score by the maximum possible score, to arrive at a normalised rating for this particular concept by this particular actor, in the range [0, 1]:

$$\mathrm{H}_{\mathrm{c}}^{\mathrm{p}} = \frac{\mathrm{H}_{\mathrm{actual}}^{\mathrm{cp}}}{\mathrm{H}_{\mathrm{max}}^{\mathrm{cp}}} \qquad (9)$$

We can sum this over the comprehension scores from all actors a^p, and then divide by the number of actors *P*, in order to get an overall average figure for the comprehension of a particular type of concept t_c; this would be:

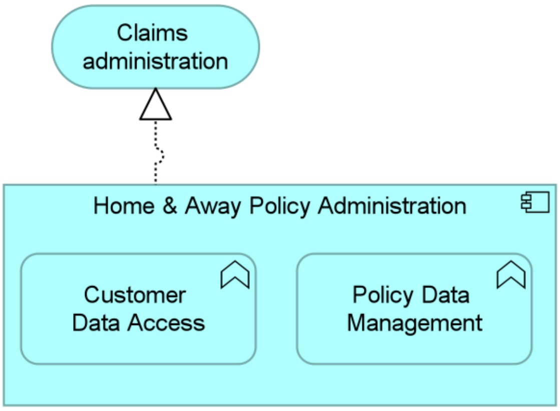
$$\mathrm{H}_{\mathrm{c}} = \frac{1}{\mathrm{P}} \sum_{\mathrm{p}=1}^{\mathrm{P}} \mathrm{H}_{\mathrm{c}}^{\mathrm{p}} \qquad (10)$$

Scores close to 1 would indicate concept types that are understandable and familiar to most users interviewed; scores close to 0 would indicate concept types that are not understandable to most users interviewed. Of particular interest, however, would be correlations between particular types of actors a^p and particular concepts t_c; as we wish to investigate whether particular kinds of actors find certain concepts more comprehensible than other actors. This understanding may come about through two kinds of analysis. Quantitative techniques may give correlation; however the additional comments captured (as to WHY the concept was understood) will yield more immediately useful knowledge.

9.5.5.1 Causes of Comprehension

As well as measuring the level of comprehension, we are also interested in why an actor was able to understand a concept. We are more interested in the underlying concept, however it is possible that an actor may gain meaning about the underlying concept from a number of sources. Consider the following fragment of an ArchiMate model:

Figure 140 Fragment of application Model (ArchiMate notation)



When we are asking for the meaning underlying the concepts that appear above, for example the ones labelled Customer Data Access and Policy Data Management, there are a number of classes of symbols (or signs) as to their referent:

(g) Their *colour*: anyone who is familiar with this particular modelling language knows that blue colouring is used to refer to concepts in the application layer;

(h) Their *shape* (a rounded rectangle): this is probably not very useful as there are a number of elements in the ArchiMate language that all have the same shape;

(i) Their *icon* (the small drawing at the top right – in this case a vertical chevron); anyone knowing ArchiMate will know that this always refers to a kind of 'function' (for example, an application function);

(j) The *descriptive names* ('node descriptor') assigned to the element (e.g. Customer Data Access)

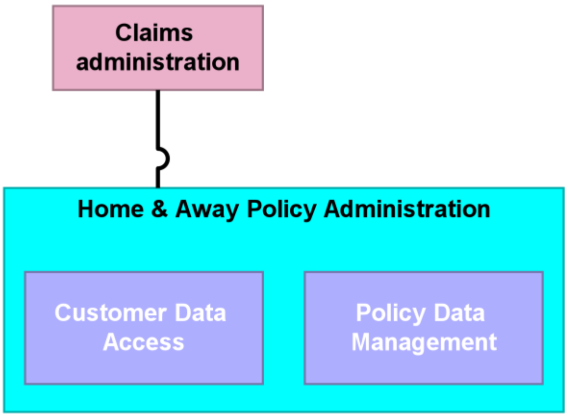
The comprehension of the concept underlying these may come from their personal background and experience, or alternatively it might come from a visual interpretation linked to previous exposure to the ArchiMate language (or perhaps an alternative language with a different, possibly conflicting graphical notation such as UML [39] or BPMN [205]).

We might illustrate this as:

Experience + Colour + Shape + Icon + Node Descriptor = Meaning

To understand how much an understanding of ArchiMate notation contributes to the comprehension of the concepts, half the interviews were carried out using ArchiMate notation, and half with a different notation as can be seen below:

Figure 141 Fragment of Application Model (neutral notation)[28]



Here, drastic changes have been made to the four classes of signs:

(e) The *colours* are now different for each underlying class; this is necessary as for these models, it is now the only visually distinguishing feature. The previous ArchiMate standard colour has been removed;

(f) The *shapes* are now identical for each symbol type;

(g) The *icons* have been removed;

(h) The *descriptive names* have been left unchanged.

If the colour, shapes or icons have had a major impact on the comprehension of the concepts then we would expect to see a change in comprehension scores with the ArchiMate views as compared to the neutral views. If, however there is no marked difference, then this would indicate that these three elements of themselves are not providing extra meaning. Thus, these 'neutral' or 'non-ArchiMate' views provide the equivalent of a control group.

[28] By 'neutral' we mean something other than ArchiMate; in fact making no use of any standard notations from any framework or language

In other words, if by removing Colour, Shape and Icon from our illustrative equation above, we don't alter the meaning conveyed to the interviewees, then we can infer logically that these three classes of symbol were not contributing to the comprehension process (either positively or negatively).

E.5Measuring Utility

Let us denote the purpose of model m^d (probably expressed as a single sentence) as r^d. To understand how useful a given concept type is for an actor, we start by asking the actor if they believe, for each model m^d, whether that model as a whole fulfils the given purpose r^d. (*Q08, Q11, Q14*). This purpose will be given to the actor as part of the interview process, in writing, along with the narrative mentioned previously. If they do not believe that the model fulfils the purpose as currently shown, then no utility scores are recorded against this model for this actor. This is because the methodology we are adopting for the utility of symbols is to examine whether a model is improved or degraded by the removal of certain symbols; and if the model starts out in a state (as far as the actor is concerned) where it substantially fails to achieve the stated purpose, then adding or removing specific symbols is unlikely to improve the situation.

So if the actor affirms that the model m^d does fulfil the given purpose r^d, then for each type of concept on the model (again denoted by t_c^d), we ask them to imagine the model without that type of concept included, and ask the question each time: would the extent to which this model fulfils the stated purpose be reduced, unchanged or increased by the omission of the specific concept (element) type (*Q09, Q12, Q15*)? This will yield one of three answers; we score +1 for reduced (losing the concept type meant the model was not as useful), 0 for unchanged (losing the concept type had no effect) and -1 for increased (losing the concept type actually increased the usefulness of the diagram, perhaps by making it easier to understand).

The mathematics flows much as before; we denote here the score for the utility of a particular concept t_c on model m^d for actor a^p as u_c^{dp} (analogous to h_c^{dp} but considering utility rather than comprehension). Given the three possible responses, we can see that $-1 \leq u_c^{dp} \leq 1$. Applying weighting in just the same way as for comprehension, then we see that the highest possible utility scores for actor a^p for a particular concept t^c would be given by the sum of all the weighting factors taken across all models, which we have already calculated in equation (7) above:

$$U_{max}^{cp} = \sum_{d=1}^{D} w_c^{dp} \qquad (11)$$

However, the lowest possible utility score is not 0; it is in fact given by:

$$U_{min}^{cp} = -\sum_{d=1}^{D} w_c^{dp} \qquad (12)$$

This would be obtained when an actor a^p says that whenever concept t_c is removed on a diagram, it increases the degree to which the diagram achieves what it was intended to (in other words, it just got in the way of the message).

The actual score would be the sum of the trinary values, each multiplied by the weighting factor, thus:

$$U^{cp}_{actual} = \sum_{d=1}^{D} h^{dp}_{c} \cdot w^{dp}_{c} \qquad (13)$$

We normalise this to get an answer for the utility of concept t_c for actor a^p, in the range [-1, 1]:

$$U^{p}_{c} = \frac{U^{cp}_{actual}}{U^{cp}_{max}} \qquad (14)$$

We can also average this out across all actors to get an overall average figure for the utility of a particular concept t_c, thus:

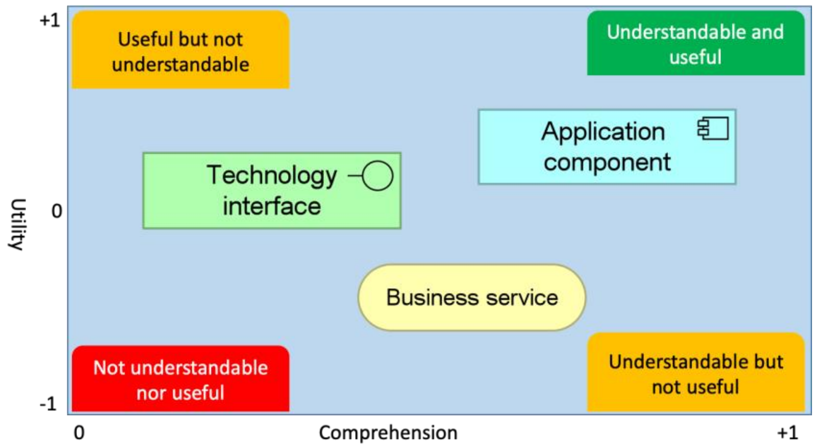
$$U_c = \frac{1}{P} \sum_{p=1}^{P} U^{p}_{c} \qquad (15)$$

Concepts with a utility value of zero or below can be considered not useful when communicating with stakeholders, and for negative scores, could be considered worse than useless, as they actually hinder communication.

E.5.1 Association of Scores and Concepts

Each concept type t_c has a tuple, a pair of scores (H_c, U_c), where $0 \leq H_C \leq 1$ and $-1 \leq U_C \leq 1$. These can be visualised in a two-dimensional bounded space thus:

Figure 142 Conceptual Visualisation Space for Scoring Element Types



We can think of each concept as fitting on such a diagram, with the ‘best’ concepts sitting in the top-right corner, being both understandable and useful.

E.5.2 Constructing the test models

The scenario examined in the test models is based upon a fictitious insurance company, and makes heavy use of the materials provided by the Open Group in teaching ArchiMate. In particular, the models are modified versions and extensions of those provided in the ArchiSurance Case Study [62]. This was done to ensure that we were measuring the concepts as intended by the authors and maintainers of the language.

The following principles were used when constructing the set of test models:

(c) The sets of models covered the complete set of symbols contained within ArchiMate 3.0 (completeness);

(d) The definitions used (which affect how the symbols are used) are intended to adhere as close as possible to the standard (one reason why the "ArchiSurance" models were chosen as a starting point).

ArchiMate has a wide-ranging scope, broken into a number of layers, for example strategic, business, application and so on. Trying to test all the elements in the language would be too time-consuming, and so a decision was made to focus on three particular layers: Motivation, Business and Application. These were chosen because the first two are more likely to be accessible to people with little IT background, which is part of the motivation for this research. The latter was chosen to provide a more familiar domain to people working directly in IT.

Each of the test models was constructed by forming a composite of all the ArchiMate concepts belonging to a particular layer from the ArchiSurance [62] material. The ArchiSurance material was notably missing some of the concepts from the ArchiMate language, and so where that was the case, examples of these concepts were added to the test models so that each test model contained at least one example of each concept from the layer being studied.

Specifically, the following example concepts were missing from the ArchiSurance sample material and so were added to the composite in order to complete the test models: to the Motivation layer were added examples of Value, Meaning, Assessment, Outcome and Constraint; to the Business layer were added examples of Business Collaboration, Business interface, Business interaction, Business service, Business object, Contract, Representation and Product; and to the Application layer were added examples of Application collaboration and Application interface.

Thus, the test models consisted of three views: a complete set of examples of Motivation concepts (elements and relations); a complete set of examples of Business concepts; and a complete set of examples of Application concepts. Each one was accompanied by a single sentence that described the purpose of the model, and then a narrative description that tried to say the same thing as the picture, but in words. It was a challenge finding significant practical uses for some of the concepts (which may be why they do not feature in the training material), and so some of the concepts (in particular, the collaborations and interactions that apply in both the Business and Application layers) only had a single example in the test models.

E.5.3 Categorising actors

We are interested in measuring in particular characteristics of actors that may have an influence on the way that they perceive architecture models. We therefore included the following:

(4) Role with respect to EA work (*Q03*)

We put people into one of three categories: (a) those people who are informed by models and make decisions based upon them (often non-technical stakeholders); (b) those people who produce such models (practitioners) and (c) those people who use these models as a basis for their own work (e.g. designers and developers). This scale would be self-selecting, tested directly during interview. In particular, the category A respondents would correspond to the ‘social audience’ in Figure 137 above; and the category B and C respondents would correspond to the ‘technical audience’.

(5) Formal education and training (*Q04, Q05, Q06*)

Based upon early pilot measurements, we are interested to see if people who have had certain kinds of training are more or less likely to understand certain symbols. Given that students specialise significantly in their educational choices, typically, from the age of 16, we will therefore ask for the equivalent of British ‘A’-level subjects as well as degree choices and subsequent professional training.

(6) Job title (*Q02*)

This is more a reflection of their current role and interest than their previous experience.

E.5.4 Capturing the Data

Two techniques were used: one-on-one interviews, and group questionnaires. For the interviews, the interviewees were shown the test material and then asked questions verbally, with the interviewer capturing their responses in questionnaires. All the marking of the comprehension ratings was carried out by the interviewer, during the interview, making a judgement as to how well the responses indicated an understanding of the underlying concept. For the group questionnaires, the respondents filled in the questionnaires themselves, after a briefing as a group. The interviewer then scored the comprehension parts of the questionnaires by reading the description that had been filled in by the respondents. This method had the advantage of capturing data quicker, however was more subject to misunderstandings as there was less scope for intervention and discussion between the interviewer and respondents.

E.6Results

E.6.1 Volume of Data

Each of the interviews covered a single layer (one test model) and took between 30-60 minutes to complete. The volume of data captured was as follows:

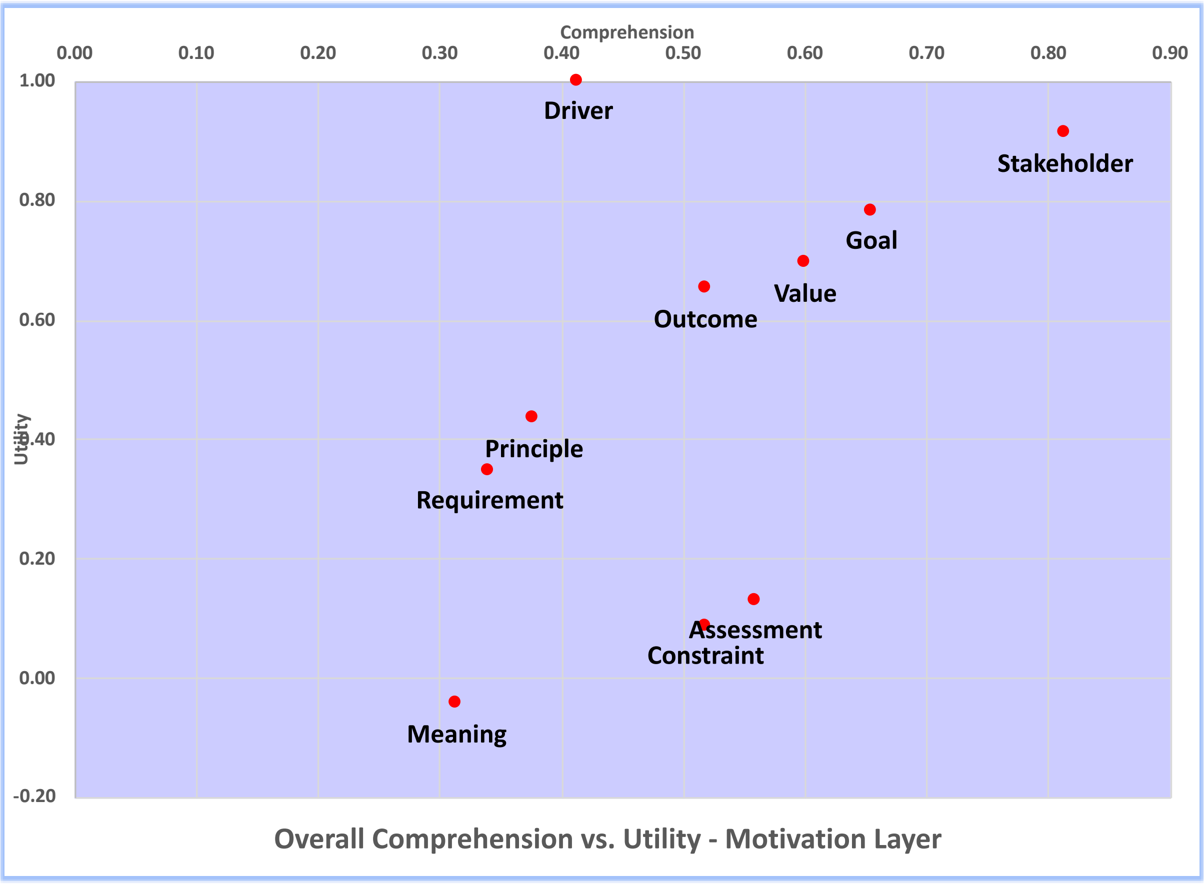
Table 59 Data Volume Collected

	Total	Archi Mate	Neutra l	Interview	Group
M	23	12	11	14	9
B	25	12	13	16	9
A	19	9	10	11	9
	67	33	34	41	27

E.6.2 Quantitative Analysis – Motivation Layer

Applying the mathematics to produce average scores for Comprehension and Utility across the Motivation layer yielded the following overall results:

Figure 143 Motivation – Overall Results

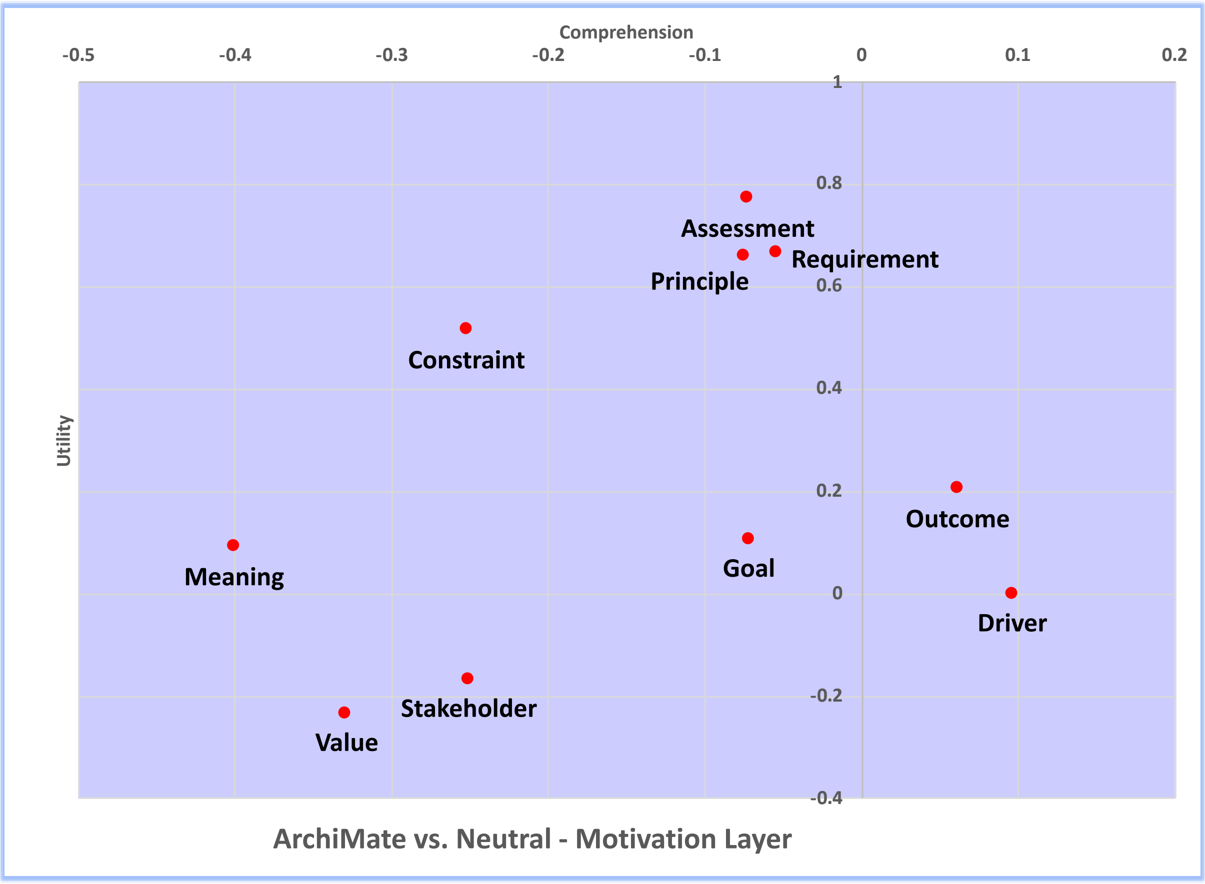


It is clear that the most well-understood and useful concept in the Motivation layer is that of a Stakeholder. The concepts of Goal, Value, Outcome and Driver are less well understood, but still seen as relatively useful, in terms of contributing to the value of a motivation diagram (e.g. Driver: *"absolutely essential",* Goal: *"absolutely vital")*.

Principle, Requirement, Assessment and Constraint are slightly less well understood, and are viewed as being significantly less useful; and Meaning is both poorly understood and, with a negative Utility score, overall the respondents felt it should not be included.

We now show the results of subtracting the scores for the Neutral models from the ArchiMate models. This is to answer the question: How much does it help to use ArchiMate notation when communicating with others. The results are as follows:

Figure 144 ArchiMate Notation vs. Neutral

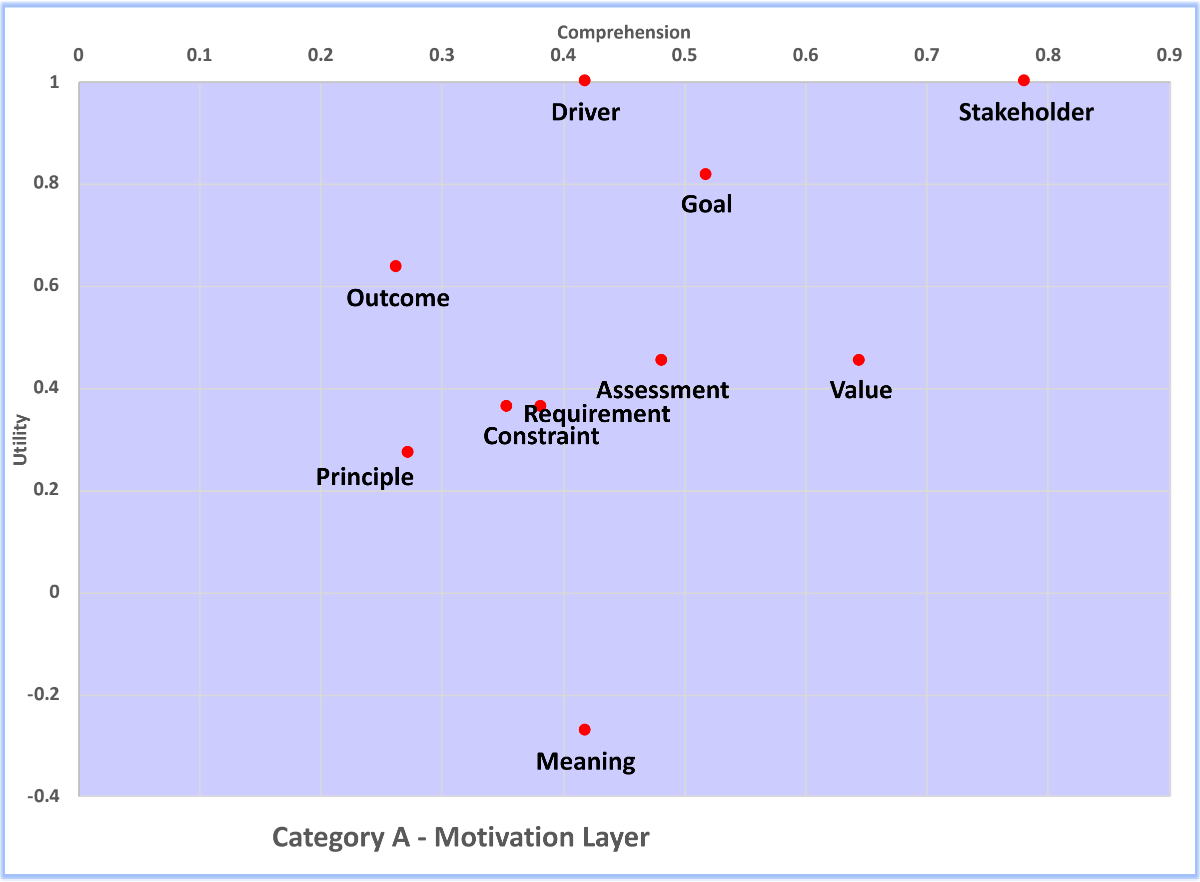


In terms of overall trends, it is clear that for the majority of the concepts in this layer, the degree to which the respondents were able to articulate the underlying concepts was reduced where ArchiMate notation had been used, compared to where it had not. In some cases the respondents made it clear that the ArchiMate notation was not helpful, for example comments like: *"Don't like the cylinder that represents it"* and *"The cloud is not helpful here"*. This does not account however why, given that the respondents understood when the ArchiMate symbols were misleading, the comprehension was still impaired. There may be other psychological factors here that further research may be able to uncover.

The trend is reversed for the utility scores: the concepts (even if poorly understood) have been rated, in most cases, more useful, when included in an ArchiMate diagram. Thus, put in simplistic terms, the respondents felt that the information that they were presented with were important but did not recognise the underlying concepts. This is not necessarily an anomaly; a respondent may feel that a piece of information is important without being able to classify that information into a specific taxonomy (architecture language).

Part of the motivation for this research is to understand how better to communicate with sponsors of architecture work – those in category 'A' – and so we finally present, for the motivation layer, the comprehension and utility ratings just for this category:

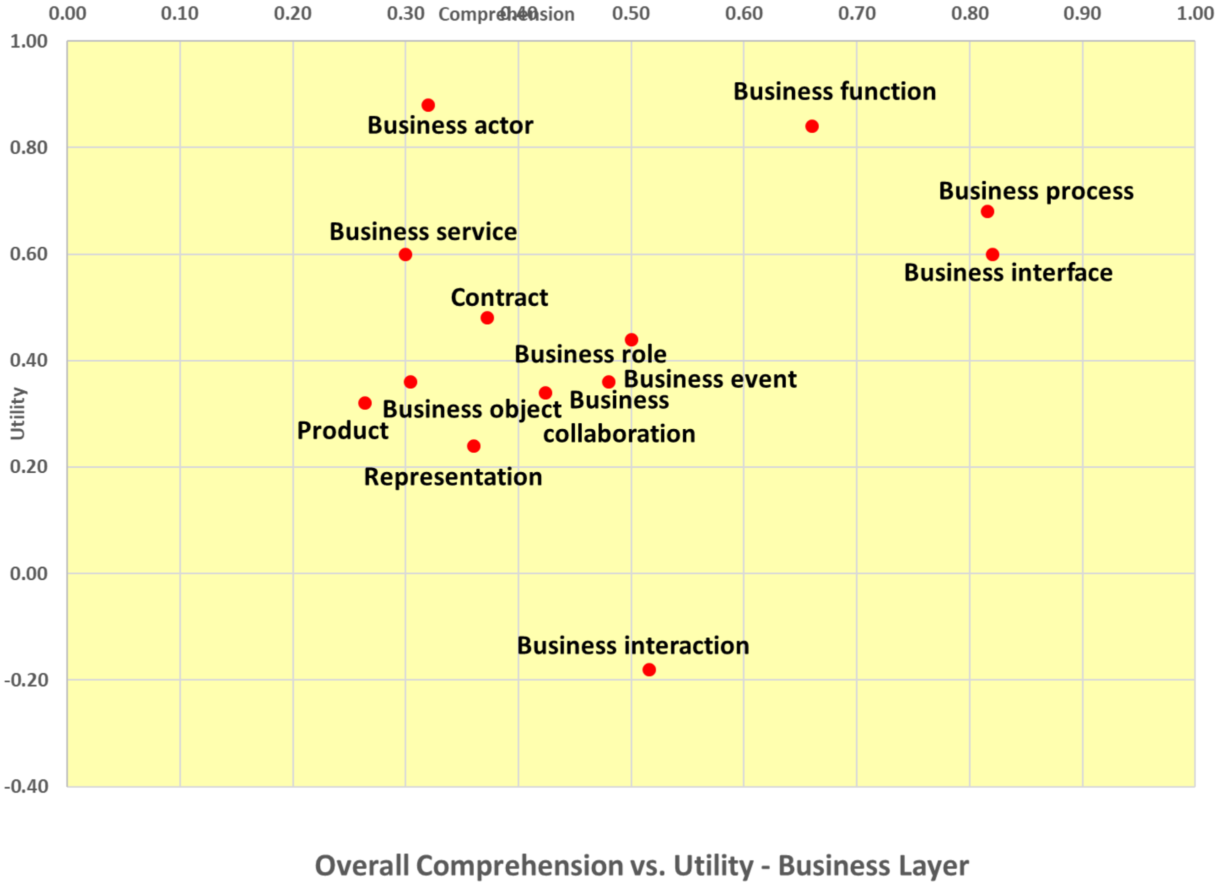
Figure 145 Motivation – Category A



The number of data points here is quite low (11 respondents for the Motivation layer in this category), and so these results should be taken merely as suggestive. They do indicate that for these types of people, many of the concepts would not be at all obvious. The most easily understood concepts would be Stakeholder, Value and Goal. These tend to be expressed in 'familiar' terms, not necessarily technical, for example a Stakeholder was described by one (category A) respondent as "The two main players' no-one more important to keep happy". Thus, the concept appeared familiar without knowing the technical term for it.

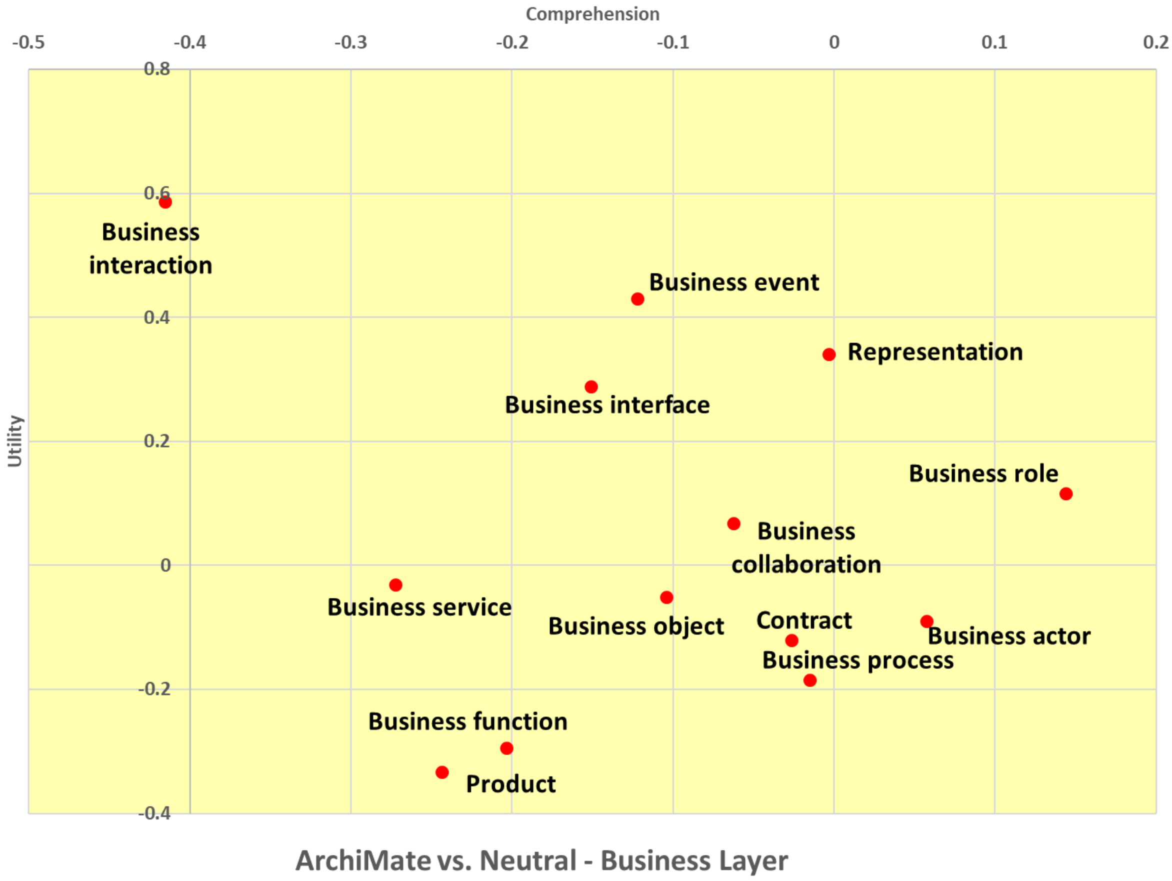
E.6.3 Quantitative Analysis – Business Layer

Figure 146 Business – Overall Results



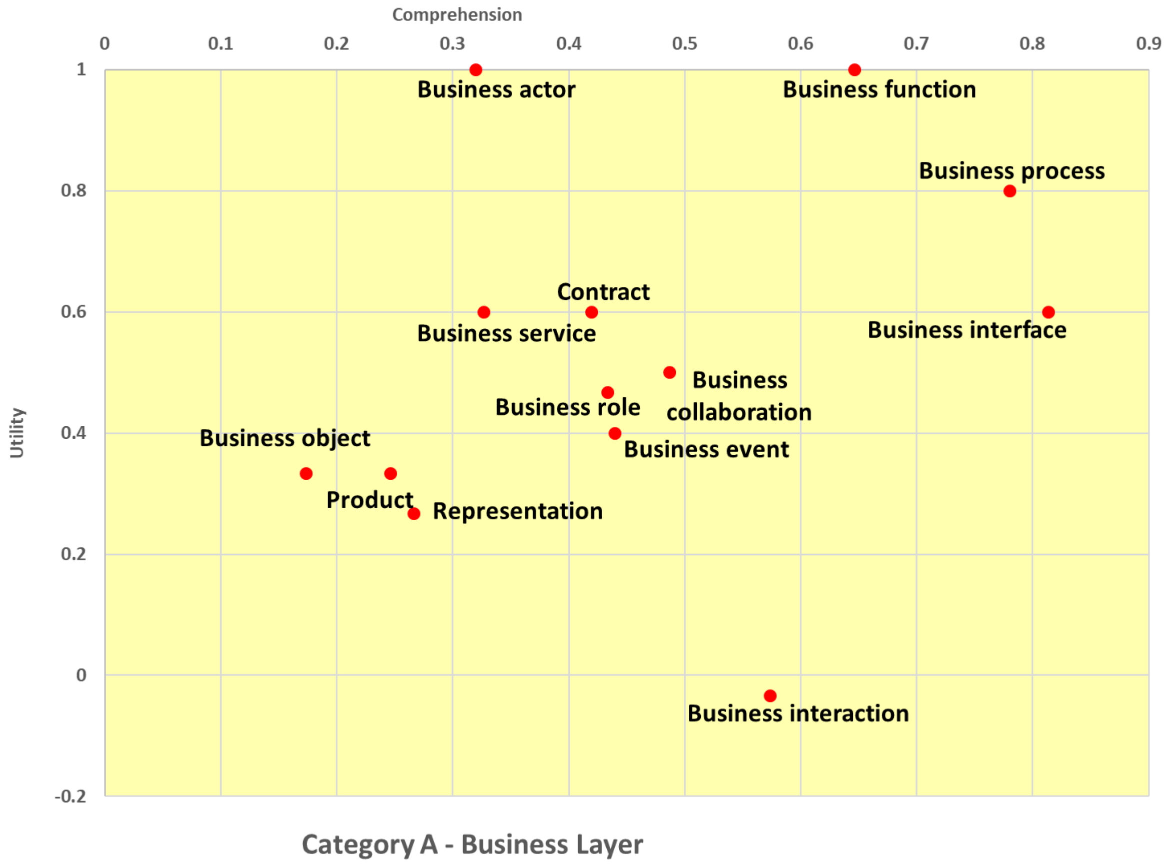
Clearly business processes, functions and interfaces were easily understood. Somewhat surprisingly, the business actor and *business* service concepts were not well understood, although they were seen as adding value nonetheless. For the actor, this may well be because all the examples given in the ArchiSurance case study used organisational structures for actors, thus overlapping in meaning with business functions. For example, one respondent thought these were *"Departmental areas and the business itself – so relates to business hierarchy and the functions within that"*. By contrast, the actual ArchiMate definition has as a key component the fact that the entity *"is capable of performing behaviour".*

Figure 147 Business – ArchiMate Notation vs. Neutral



The marked effect whereby using ArchiMate notation clearly reduced the comprehension of the underlying concepts is also apparent here. Unlike the Motivation layer, however, the use of ArchiMate for many of the Business concepts also reduced their perceived usefulness.

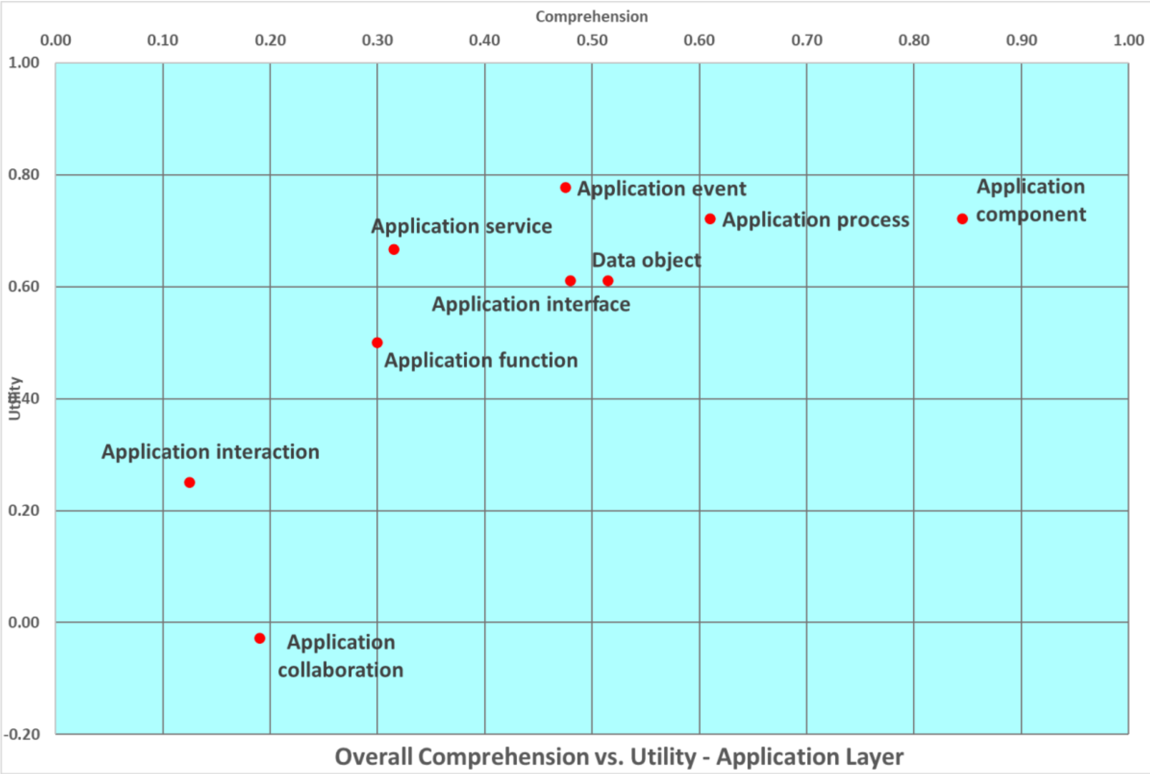
Figure 148 Business – Category A



The most useful concepts (even if misunderstood) to this category of respondents is clearly the business actor, function and process. This is because the descriptive names seemed to make sense to the respondents, even if the categorisation did not (e.g. for business function, "*Departments within the company*", or "*Functions inside the company*".

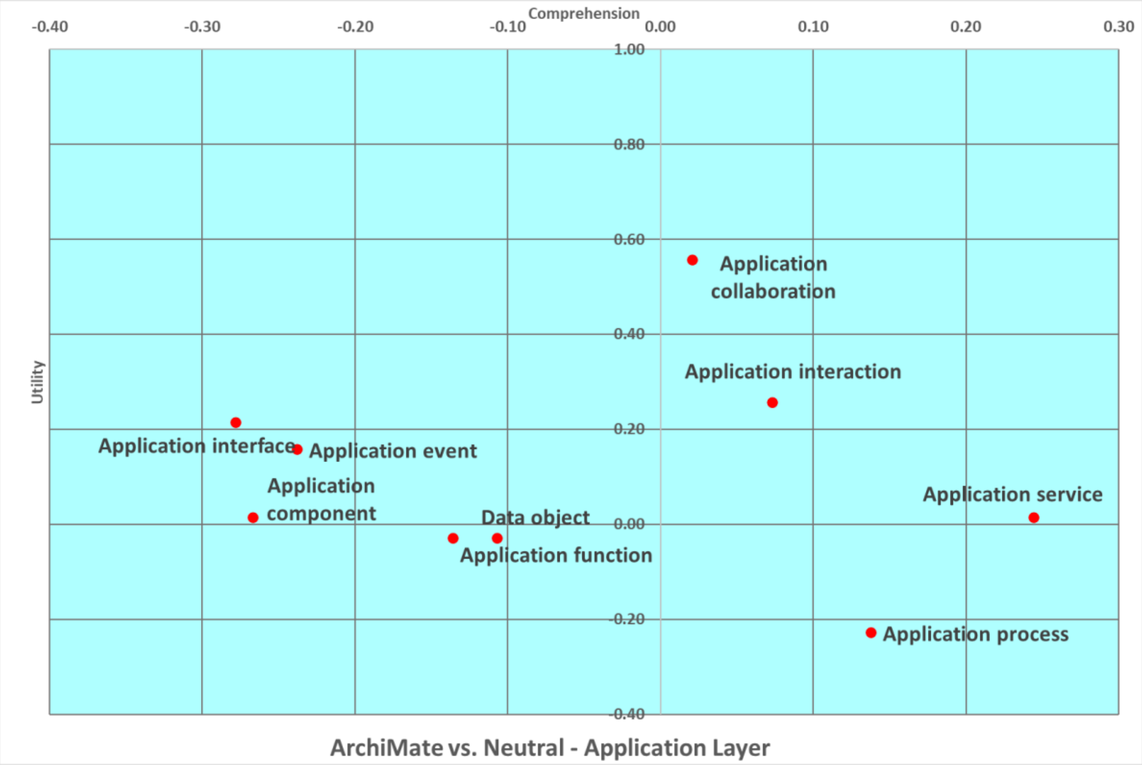
E.6.4 Quantitative Analysis – Application Layer

Figure 149 Application – Overall Results



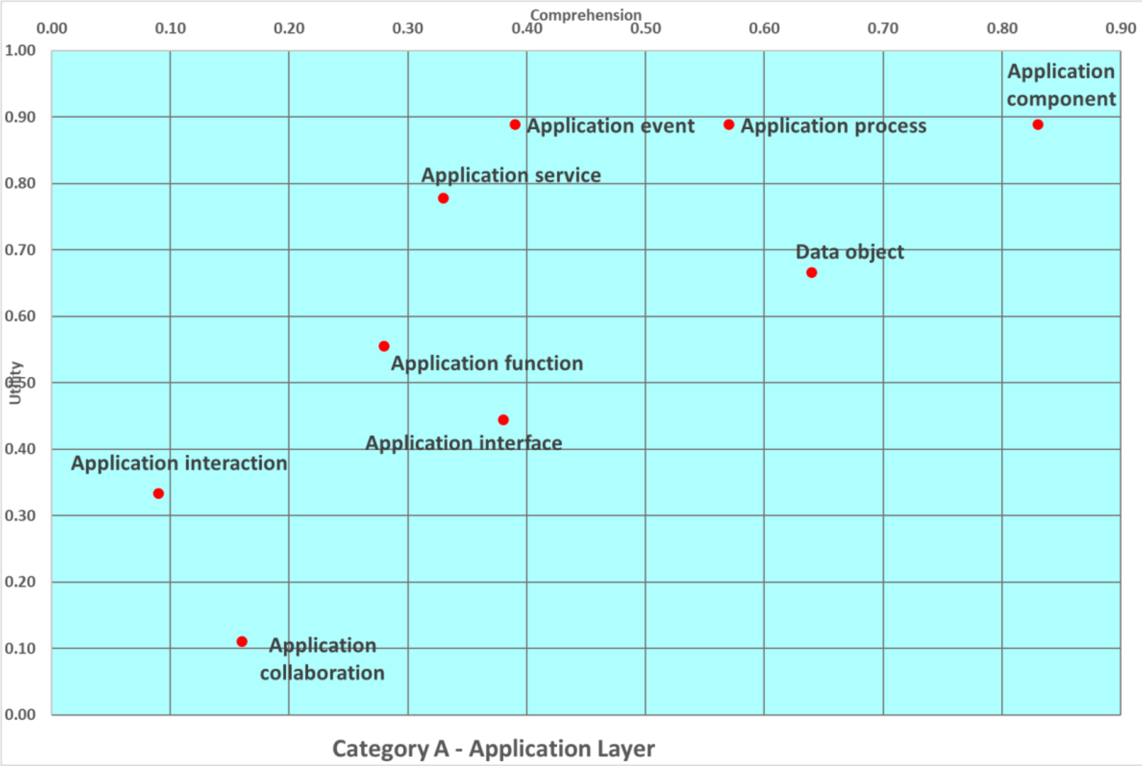
Clearly the most understood concepts relate to application components and processes; and the least well understood (and useful) are the application interactions and collaborations; although a limitation in the test design might have contributed to the latter.

Figure 150 ArchiMate Notation vs. Neutral



The use of ArchiMate notation seems neutral in terms of overall effect on concept comprehension: a broadly equal number of concepts are either better or less understood when ArchiMate is used. In terms of perceived usefulness, the overall most poorly understood concepts (application collaborations and interactions) benefit most by being expressed in ArchiMate notation.

Figure 151 Application – Category A



For these category A stakeholders, it is clear that application components and processes are both well understood and useful.

E.6.5 Summary of Quantitative Results

For the Motivation layer, we see that: six concepts – Driver, Principle, Value, Outcome, Goal and Stakeholder – stand out as being both understood and valuable when explaining the motivation of organisations.

Three others – Requirement, Constraint and Assessment – are slightly less well understood, and significantly less valuable for the purposes of explaining motivation.

One – Meaning – is poorly understood, and on average is not seen as useful (a neutral score close to 0).

There is less of a distinct clustering of results in the Business layer. Three concepts – Business Function, Business Process and Business Interface – appear to be well understood and useful. It is interesting to note that both Business Actor and Business Service are seen as useful but not well understood.

Business Interaction, as the other extreme, has a negative score for Utility, which means that the respondents felt on average that it should be left out of the diagrams.

Within the Application layer, six concepts – Driver, Principle, Value, Outcome, Goal and Stakeholder – stand out as being both understood and valuable when explaining the motivation of organisations.

Three others – Requirement, Constraint and Assessment – are slightly less well understood, although significantly less valuable for motivation purposes.

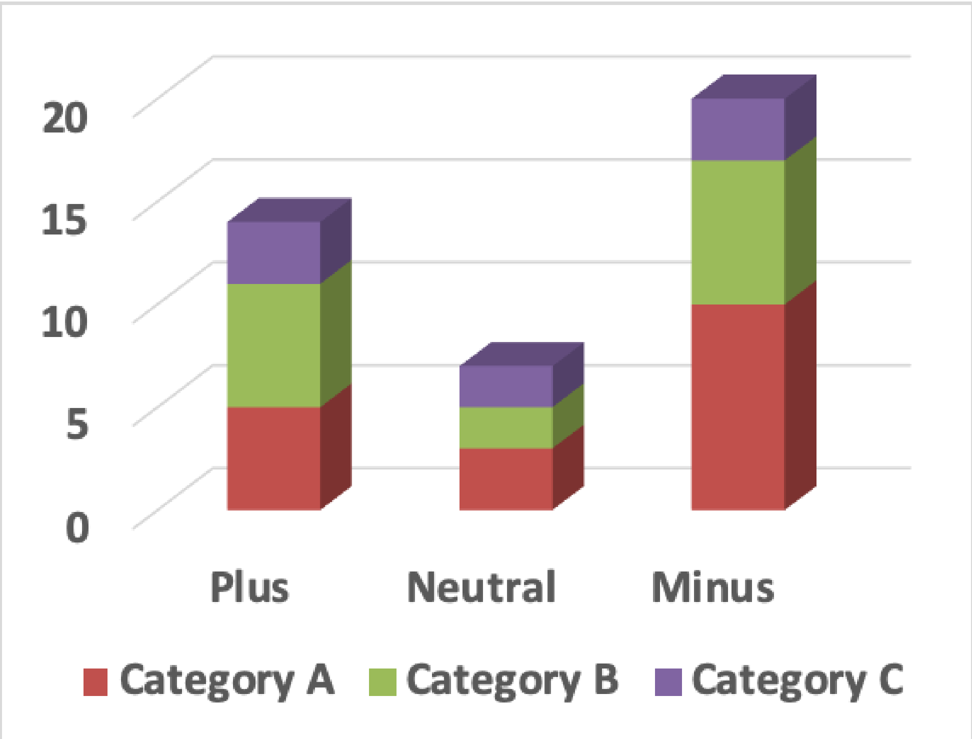
One – Meaning – is poorly understood, and on average is not seen as useful (a neutral score close to 0).

E.6.6 Qualitative Analysis

The respondents presented with ArchiMate models were encouraged to comment where the symbols used were helpful or otherwise. These were captured and then analysed subsequently by the lead author to determine the overall sentiment. The icons that were seen as most unhelpful or misleading were Stakeholder, Meaning, Business Role (same icon as Stakeholder), Actor and Business Object. In some cases this was because the icon suggested by ArchiMate meant something different in an alternative discipline (for example, a cylinder, used in both Stakeholder and Business Role, is also used in other contexts to denote a data store or database).

These results were also analysed, to determine, for each category of user (their role with respect to EA work), whether the use of ArchiMate notation was helpful. The results are summarised below. The volume of data is not very large, however is at least suggestive of an interesting trend:

Figure 152 Impact of Use of ArchiMate Notation



Overall, the use of ArchiMate notation is seen by more respondents as unhelpful than helpful. This result is most marked in category A users, who are representative of those that are non-technical, including sponsors of architecture work. This would suggest that the use of ArchiMate notation in diagrams intended for use by non-architects, especially influential stakeholders, should be avoided.

E.7 Conclusion

E.7.1 Use of ArchiMate Concepts

None of the respondents (even those in category B – the architects) interviewed for this research had had training in ArchiMate prior to taking the survey, although two had other IT architecture training (one TOGAF [29] and one Zachman [74]). It is clear that those not trained in ArchiMate – even architects - may struggle to understand many of the concepts used in the language. The most understandable concepts are highlighted in the Summary of Quantitative Results above; and use of ArchiMate should where possible be restricted to those concepts when communicating with those not familiar with the language.

E.7.2 Use of ArchiMate Notation

For those not trained in the ArchiMate language, then the use of ArchiMate notation in some cases clearly detracts overall from the understanding of architecture diagrams, and so the use of ArchiMate notation in communicating with such users should be avoided.

This is not to say that ArchiMate notation should never be used. For those that understand it, it provides a well understood notation that conveys well defined meaning. The issues is that most people reading architecture diagrams will not understand it correctly.

E.8 Reflections

Some of the ArchiMate concepts have complicated definitions – meaning that there are multiple parts to the definition – and it is not surprising that non-architects struggle to recognize these concepts. Simple, well-known concepts are more likely to be recognized by non-specialists. For example, a business process would be familiar to business analysts and other business roles, and can be summarized very simply (an ordered set of activities), although the ArchiMate definition is much more complex. By contrast, some of the less well known concepts, that had correspondingly low comprehension scores, have complex definitions that would be hard to summarise so simply. An example would be a business service (defined by ArchiMate as "An explicitly defined exposed business behavior."). There are two essential elements here ("*explicitly defined*" and "*exposed business behavior*"), and the distinction between this and other similar ArchiMate concepts (for example, a capability: "*An ability that an active structure element, such as an organization, person, or system,* *possesses*.”, which itself has three key ideas) requires a considerable amount of thought even amongst those trained in IT architecture to determine the most appropriate concept for a particular idea. If we choose to use concepts that even IT architects need to think about (note the low comprehension scores for Business Service, for example), then then it may not be surprising that these concepts are not understood by those not trained in their use.

The result that most surprised me was the difference in comprehension of the underlying concepts for ArchiMate vs. neutral notation. This is a very interesting effect which is worth further research to understand what is causing this.

E.9Limitations

The volume of data is sufficiently low as to render some of the conclusions suggestive rather than definitive, in particular the results for ‘category A’ respondents (corresponding to sponsors of architecture work, not necessarily familiar with the details of IT architecture). Further research, which might be specific to various organisations and their different cultures, would be valuable in helping to be clearer as to which concepts should and should not be included when communicating architecture information to senior stakeholders.

The interactions and collaborations in the business and application layers I found problematic, because the ArchiSurance case study made no use of them and I struggled to find sensible real-world examples of where they might be useful. A fairer test of the data for these would be obtained by creating multiple examples of each.

Appendix F ArchiMate Viewpoints

The following table illustrates the possible content of each of the suggested ArchiMate viewpoints:

Table 60 ArchiMate Suggested Viewpoints and Contents

		Category	Composition					Support			Cooperation		Realisation		Motivation					Strategy			Implementation	
Layer	Symbol	Viewpoint	Organization	Information Structure	Technology	Layered	Physical	Product	Application Usage	Technology Usage	Business Process Cooperation	Application Cooperation	Service Realization	Implementation and Deployment	Stakeholder	Goal Realization	Requirements Realization	Motivation	Capability Map	Strategy	Outcome Realisation	Resource Map	Project	Implementation & Migration
Motivation	Stakeholder					✓									✓			✓						
	Value					✓		✓									✓	✓						
	Meaning			✓		✓											✓	✓						

	Driver					✓									✓			✓						
	Assessment					✓									✓			✓						
	Goal					✓									✓	✓	✓	✓					✓	✓
	Outcome					✓									✓	✓	✓	✓	✓	✓	✓			
	Principle					✓										✓		✓						
	Requirement					✓										✓	✓	✓					✓	✓
	Constraint					✓										✓	✓	✓					✓	✓
Strategy	Capability					✓													✓	✓	✓	✓		
	Course of Action					✓														✓	✓			
	Resource					✓													✓	✓	✓	✓		
Composite	Grouping					✓																		
	Location		✓		✓	✓					✓	✓												✓

	Business role		☑			☑	☑	☑	☑		☑		☑				☑				☑		☑	☑
	Business actor		☑			☑		☑	☑		☑		☑				☑				☑		☑	☑
	Business collaboratio n		☑			☑		☑	☑		☑		☑				☑				☑			☑
	Business interface		☑			☑		☑			☑		☑				☑				☑			☑
	Business process					☑		☑	☑		☑		☑				☑				☑			☑
	Business function					☑		☑	☑		☑		☑				☑				☑			☑
	Business interaction					☑		☑	☑		☑		☑				☑				☑			☑
Business	Business event					☑		☑			☑		☑				☑				☑			☑

	Business service					✓		✓			✓		✓				✓				✓			✓
	Business object				✓	✓		✓			✓		✓				✓				✓			✓
	Contract					✓		✓									✓				✓			✓
	Representation				✓	✓					✓		✓				✓				✓			✓
	Product					✓		✓									✓				✓			✓
Application	Application component					✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
	Application collaboration					✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
	Application interface					✓		✓	✓		✓	✓	✓	✓			✓				✓			✓

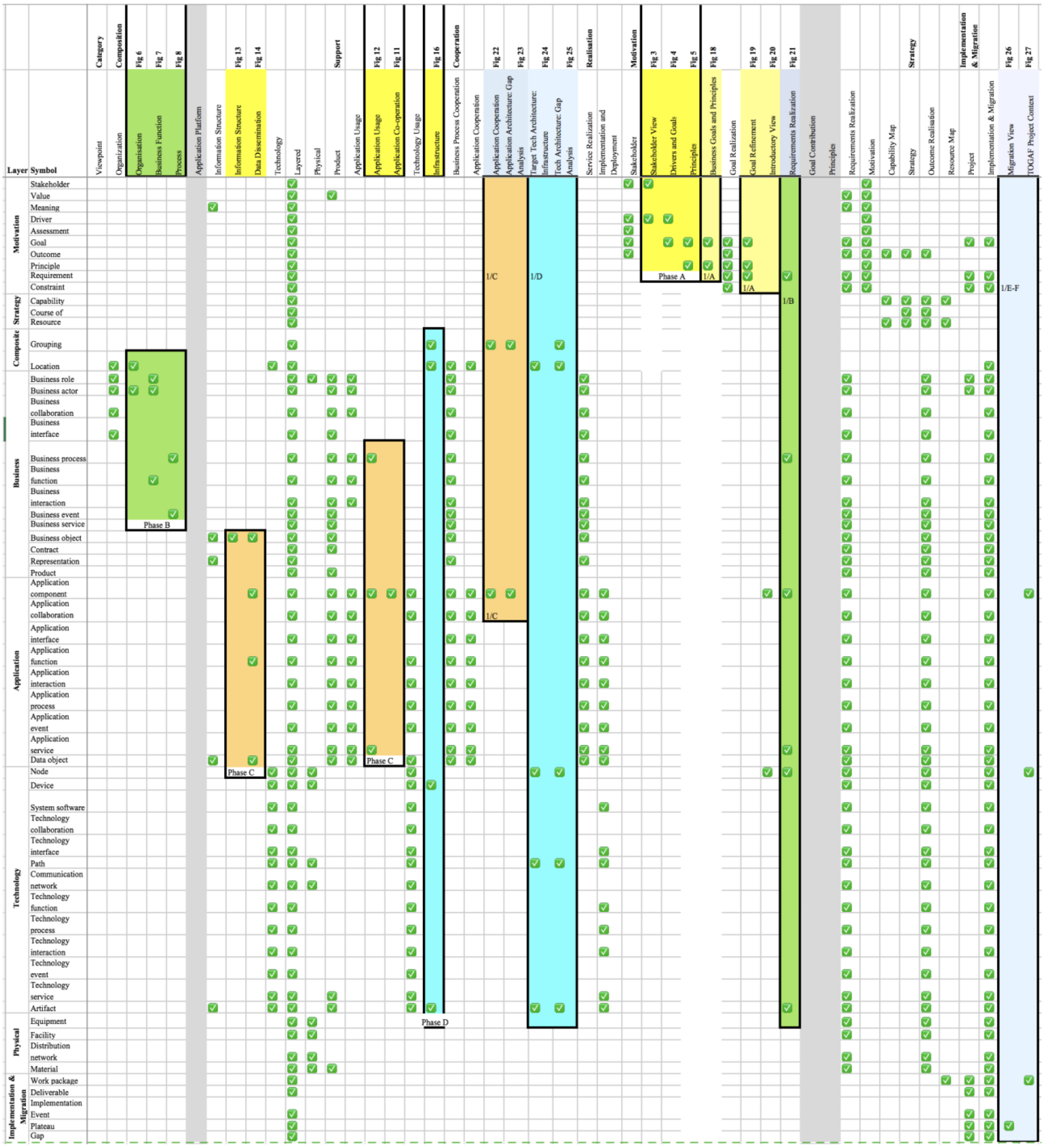
	Application function					✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
	Application interaction					✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
	Application process					✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
	Application event					✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
	Application service					✓		✓	✓		✓	✓	✓	✓			✓				✓			✓
	Data object			✓		✓		✓	✓	✓	✓	✓	✓	✓			✓				✓			✓
Technology	Node				✓	✓	✓			✓							✓				✓			✓
	Device				✓	✓	✓			✓							✓				✓			✓
	System software				✓	✓				✓				✓			✓				✓			✓

	Technology collaboration				☑	☑				☑							☑				☑			☑
	Technology interface				☑	☑				☑				☑			☑				☑			☑
	Path				☑	☑	☑			☑				☑			☑				☑			☑
	Communication network				☑	☑	☑			☑							☑				☑			☑
	Technology function				☑	☑				☑				☑			☑				☑			☑
	Technology process				☑	☑				☑				☑			☑				☑			☑
	Technology interaction				☑	☑				☑				☑			☑				☑			☑
	Technology event				☑	☑				☑							☑				☑			☑

	Technology service				✓	✓		✓		✓				✓			✓				✓			✓
	Artifact			✓	✓	✓		✓		✓				✓			✓				✓			✓
Physical	Equipment					✓	✓										✓				✓			✓
	Facility					✓	✓										✓				✓			✓
	Distribution network					✓	✓										✓				✓			✓
	Material					✓	✓	✓									✓				✓			✓
Implementation & Migration	Work package					✓																✓	✓	✓
	Deliverable					✓																	✓	✓
	Implementation Event					✓																	✓	✓
	Plateau					✓																	✓	✓
	Gap					✓																	✓	✓

The following table shows the ArchiMate suggested viewpoints and contents supplemented by the actual examples contained within the ArchiSurance case study [62]. This was used as a basis for constructing the test models.

Table 61 Viewpoints Supplemented by ArchiSurance Case Study



Appendix G Questionnaire Used with EA Modelling Language Effectiveness Survey

G.1 Metadata

Interviewer	
Interviewee	
Interviewee Number	
Date/Time	
Location	
Used ArchiMate diagrams?	

G.2 Interviewee Background

G.2.1 About the Interviewee

Q01. Where do you work?	
Q02. What is your job title?	

G.2.2 Role with Respect to EA Work

Q03. Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c)

G.2.3 Formal Education and Training

Q04. What did you study at A-level?	
Q05. If you went to university, what did you study there?	
Q06. What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	

G.3 Motivation Layer

Concept ratings

0	Don’t understand what these are
0.3	Have some understanding of what these are
0.7	Understand what they are but not familiar with this kind of thing
1	Understand what they are and are familiar with them

Q07. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1		
M2		
M3		
M4		
M5		
M6		

M7		
M8		
M9		
M10		

Q08. Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

Q09. For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1		
M2		
M3		
M4		
M5		
M6		
M7		
M8		
M9		

M10		

G.4 Business Layer

Concept ratings

0	Don't understand what these are
0.3	Have some understanding of what these are
0.7	Understand what they are but not familiar with this kind of thing
1	Understand what they are and are familiar with them

Q10. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1		
B2		
B3		
B4		
B5		
B6		

B7		
B8		
B9		
B10		
B11		
B12		
B13		

Q11. Does the business model fulfil the purpose specified in the reference material (Y/N)?	

Q12. For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1		
B2		
B3		

B4		
B5		
B6		
B7		
B8		
B9		
B10		
B11		
B12		
B13		

G.5 Application Layer

Concept ratings

0	Don't understand what these are
0.3	Have some understanding of what these are
0.7	Understand what they are but not familiar with this kind of thing
1	Understand what they are and are familiar with them

Q13. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1		

A2		
A3		
A4		
A5		
A6		
A7		
A8		
A9		

Q14. Does the application model fulfil the purpose specified in the reference material (Y/N)?	

Q15. For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1		

A2		
A3		
A4		
A5		
A6		
A7		
A8		
A9		

Appendix H Completed Questionnaires

This appendix contains all the questionnaires completed by the interviewees (group scenario) or completed by the interviewer on their behalf (one-on-one scenario). Some of the information has been redacted as it is personally identifiable and the answers to these interviews are intended to be anonymous. Each section contains one or more questionnaires completed for the same interviewee (as some only did one level; some did two or three). The interviewee codes are numbered sequentially and start with either G for a group interviewee or S for a single interviewee (i.e. one-on-one).

H.1 Interviewee: S1

H.1.1 Metadata

Interviewer	Cameron
Date/Time	16th Oct 2019
Used ArchiMate diagrams?	Yes

H.1.2 About the Interviewee

Where do you work?	Retired – was at Microsoft
What is your job title?	Was an account manager – business development role – for third party accounts
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Maths, Physics, Chemistry
If you went to university, what did you study there?	Electrical Engineering with Electronics
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	none

H.1.3 Motivation Results

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	Struggling to put it into words – they both have objectives Shareholders or customers

		Feels like interviewee has the idea but doesn't know the right word to use – but in phase 2 interviewee said 'stakeholder' so changed score from 0.3 to 1
M2	1	Objectives - value
		Picked the word out of thin air!
M3	0	An idea? A target? An aspiration?
M4	0.3	High level – what we're aiming at
		Not far off a driver
M5	1	A measure of where you're at relative to where you're aiming for
M6	0.7	A target implementation – how you're going to achieve the M4s
M7	0.7	Specific measurable targets
M8	0.3	These are similar to M6s – more detailed breakdown
		Exclamation is normally a warning or a hazard – increased from 0 to 0.3 because interviewee specified in phase 2 how you're going to achieve the targets
M9	0.3	How we're going to do something – quite similar to M8s
M10	0.3	A warning – something critical to the business

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	Prime stakeholders
M2	0	Because they're common sense
M3	-1	redundant
M4	+1	
M5	+1	Measure of what you're at
M6	+1	They say what you're trying to achieve
M7	+1	Also pretty key
M8	+1	Quite useful – how you're going to achieve the targets
M9	+1	

M10	+1	Just a bit of a warning

H.1.4 Application Results

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1	The main systems
A2	0.3	This is the program – underneath it things that fill this task – top high level view of what's going on
A3	0.3	Are these user input? Workstations?
		The arrows confuse – shouldn't they be outputs? There is something about an interface here – but didn't get the API reference
A4	0.3	Is this some kind of database?
		The way they are used implies something active, so I think there is some understanding here
A5	0	Not really sure what this is
A6	1	A flow of something – a sequence of things that happen
A7	0	They are both gathering information
A8	1	It's what they're doing
A9	1	The information contained within the database

H.2 Interviewee: S2

H.2.1 Metadata

Interviewer	Cameron
Date/Time	30/10/19
Used ArchiMate diagrams?	No for Business/Motivation, Yes for Application

H.2.2 About the Interviewee

Where do you work?	Own consulting company – training and development
What is your job title?	Teacher, Consultant
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Philosophy, Art History, Classics, Photograph, Fine Art
If you went to university, what did you study there?	Fine Art and Contemporary Critical Theory
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	

H.2.3 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0	Are they all managers?
B2	0	Physical offices / infrastructure – physical space
B3	0	Review
B4	1.0	Methods of communication
B5	0.7	Workflow – policies and practices
B6	0.7	Departments within the company
B7	0.3	What happens after review
		Some idea of output here
B8	0.3	Decision making
		Does kind of relate to events?
B9	0.3	Administration
B10	0.3	An action – a claim is being made

B11	0	Rules and regulations, policies, procedures
B12	0.7	Summary of procedures, rules & regs
B13	0.7	The subject area

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	0	Its kind of obvious
B2	1	
B3	0	Not that interested in it
B4	0	They're obvious, extraneous
B5	1	
B6	1	
B7	0	
B8	-1	It's partial, adding no value, is context-related
B9	1	
B10	1	
B11	1	
B12	-1	A bit confusing, as only partial information

H.2.4 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
A1	0.3	Software and hardware
A2	0	Purchasing
A3	0.3	Screens
		True but not the point
A4	0.3	Links between different data – how one feeds into the other
		An element of processing there
A5	0	Sale
		Again, I think this is very hard because I only have one example
A6	0.7	Data processing
A7	0.7	Data acquisition
A8	0	Process administration
A9	0.7	Organisation of filing details, organisation of data

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	1	
A2	1	
A3	0	
A4	0	
A5	1	
A6	1	

A7	1	
A8	1	
A9	1	

H.3 Interviewee: S3

H.3.1 Metadata

Interviewer	Cameron
Date/Time	17th June 14:30
Used ArchiMate diagrams?	Yes

H.3.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Head of IT Strategy
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (b)
What did you study at A-level?	Actuarial – pensions – before uni
If you went to university, what did you study there?	Informatics – IT then MBA
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Programming Functional Design Technical Design Architectural Design

H.3.3 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1	A system – a series of programs with a defined input and output
		Combination of the shapes and arrows going in and out
A2	0	
A3	0.3	Something like an input – a connector
		Note: struggled with difference between screens and API -

A4	0.3	Look like process steps –
		they are internal – but not process steps – recognise that they do something – noting verbs - actions
A5	0.3	Looks like a function
		Note this is what an interaction is – but not getting that it's done with multiple
A6	0.7	Process as well – still a bit uncertain
		Clear order – left to right
A7	0.3	Input and start of a process –
		… which they can do …
A8	0	Process steps
A9	0.3	Storage - database
		Yes they do relate to information…

IQ 1. Does the application model fulfil the purpose specified in the reference material (Y/N)?	

IQ 2. For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1		
A2		
A3		
A4		
A5		
A6		
A7		
A8		
A9		

H.4 Interviewee: S4

H.4.1 Metadata

Interviewer	Cameron
Date/Time	30/10/19

Used ArchiMate diagrams?	No for Business, Yes for Application

H.4.2 About the Interviewee

IQ 3. Where do you work?	(technology transfer attached to a university)
IQ 4. What is your job title?	Initiative Assistant?
IQ 5. Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
IQ 6. What did you study at A-level?	Arts: history, Spanish, English, Latin, Ancient Latin, History of Art, Philosophy
IQ 7. If you went to university, what did you study there?	BA in Geography and History
IQ 8. What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Programming Principles for non-developers XML Schemas XSL 6 Sigma Leading Innovation Change Management Project Management Post Graduate Diploma in Education Information and Library Studies

H.4.3 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	External stakeholders
B2	0.7	The action people – without them nothing happens… - the heart of the company
		Feels like a bit of a lucky strike – “action” people – didn’t feel like it should score a 1, although the answer was correct, because it felt like a fluke
B3	1	Something between the two of them
B4	1	Channels: how customers contact the company
B5	0	Outsourcing – tasks that have been outsourced from the main company

B6	1	Functions inside the company
B7	1	The task carried out by B3
B8	0.7	Inputs and outputs
		Yes but not what I was looking for
B9	1	What they do directly with the customer Strategic part of the insurance
B10	0	It's a trigger for a process
B11	0	Implementation of the strategic
B12	0.3	Information – something that you enter
B13	0	Something strategic – a thought, non-operational

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

IQ 9. For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	1	
B2	1	
B3	1	
B4	1	
B5	1	It feels like it's more process, more detailed than the rest of the diagram – keep but maybe not on this diagram (am I being consistent in scoring this as a 1? For others, I've put -1 when it was too detailed for this specific diagram?)
B6	1	
B7	-1	Not important
B8	1	Keep these if you keep the B5s
B9	-1	confusing
B10	1	

B11	-1	Some confusion – why do we need B11 and B12
B12	-1	Again – what is this for?
B13	-1	Not clear what this is for

H.4.4 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1	Systems - software
A2	0.7	Combining information
A3	0.3	A web page? Connecting to another system?
A4	0	A document -
A5	0	It's going out [to the customer] the sales part
A6	1	process
A7	0.7	Trigger – start somewhere
A8	1	Functionality - actions
A9	1	Data

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	1	
A2	-1	As doesn't understand it
A3	1	
A4	1	
A5	-1	As doesn't understand it

A6	1	
A7	1	
A8	1	
A9	1	

H.5 Interviewee: S5

H.5.1 Metadata

Interviewer	Cameron
Date/Time	20th March 2019
Used ArchiMate diagrams?	yes

H.5.2 About the Interviewee

Where do you work?	UK Defence organisation
What is your job title?	Principal engineering manager
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (A)
What did you study at A-level?	Maths, Physics, Biology
If you went to university, what did you study there?	Mechanical Engineering (BSc) MBA
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Project Management in MBA Contracts Management

H.5.3 Business Layer

Before starting, the interviewee made the comments: "As a process savvy person – assumed the whole thing is a process model – where's the start? Also expected inputs on the left, outputs on the right?"

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.7	Stakeholders? Decision makers? People that do stuff? Distinct titles?
		Looks like a pipe – does something flow through here?

		More used to flowcharts and the symbology used there (e.g. decisions, start/stop), also failure modes / event analysis Said the word "role"
B2	0.3	People / departments / functions, might be separated geographically
B3	0.3	Governance? Approvals? – is this because we don't have multiple examples?
		Where (interviewee) works we have various boards that review things from a safety point of view, or production financial management, with terms of reference. A gatekeeper?
B4	1	Means of communication
B5	1	Process -steps of doing something
		Engineering – the process of doing the design work (a business need, requirements, concept design, "optioneering", construction etc.); also the process of how a machine works (coal/gas in one end, electricity out the other)
B6	0.7	They're functions – a part of the business that has an involvement in a process
B7	0	
		The line down the middle of the ArchiMate symbol kind of implies stop/go – moat – gatekeeper?
B8	0.7	Inputs and outputs – to do with processes
B9	0	A human computer? You put the info in there and you get a yes/no answer
B10	0	Record keeping? Filing system? Inbox?
		The ArchiMate notation for B10 looks like something you store things in – an in-tray?
B11	1	Contractual agreement
		Interviewee deals with managing contracts
B12	0	
		Wavy line normally means, on an engineering drawing, that you've curtailed / truncated something.
B13	0	No idea

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	+1	Roles are absolutely useful
B2	+1	Identifying the department etc.
B3	0.5	I explained what the two boxes meant, now we got to phase 2, to assist, as I only had one example each of B3 and B7
B4	+1	
B5	+1	Really important
B6	+1	At a level – it depends on what kind of detail you're looking
B7	0.5	See notes for B3
B8	+1	It's the whole purpose of what this is showing
B9	0	Not fussed either way
B10	-1	Easier if removed (doesn't understand what it is)
B11	+1	Contractual thing (familiar with these)
B12	-1	(told him what it means) – rather have it removed!
B13	-1	Half of a B10 – don't know what it means

H.6 Interviewee: S6

H.6.1 Metadata

Interviewer	Cameron
Date	7th Feb 2019
Used ArchiMate diagrams?	No

H.6.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Enterprise Business Architect
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) B
What did you study at A-level?	Accounting Geography

	Physical Science English Afrikaans Maths
If you went to university, what did you study there?	BSc Commerce/Economics PDip ? Commercial
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Information Systems Business Analysis Zachman, BPMN, ITIL, Prince 2, Change Management

H.6.3 Motivation Layer

IQ 10. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1	1	A business analyst – has done stakeholder analysis
M2	1	What stakeholders what to get out of the business
		Bus Arch – name propositions – value
M3	0.7	Meaning to different people
		Broad background, including marketing
M4	0.3	Focal areas
		Done so many different things….
M5	1	All negative
M6	1	Clarification – desired 'outcome'
M7	0.7	Very specific results, results, desired outcome

		Language in the boxes helped, stated a fact – no ‘adjectives’ – very direct
M8	0.7	Tech solution / tactic / type of behaviour
		Not really [interviewee’s] area
M9	0.3	Policy / technical goal / direction
M10	1	Constraint – boundary to affect what you can and can’t do
		Working in the real world – political - personal

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	Subject of the model – core to the diagram
M2	+1	They’re the motivation
M3	-1	Unnecessary and confusing
M4	+1	Marketing background - useful
M5	-1	Superfluous - nothing to do with motivation
M6	+1	As a Biz Arch – has to construct motivational models
M7	0	
M8	+1	As a Biz arch – has to know these
M9	-1	
M10	-1	Too much on [the diagram]

H.7 Interviewee: S7

H.7.1 Metadata

Interviewer	Cameron

Date/Time	2nd March 19:45
Used ArchiMate diagrams?	No

H.7.2 About the Interviewee

Where do you work?	(charity)
What is your job title?	Chief Operating Officer
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (A)
What did you study at A-level?	Maths, Further Maths, Physics, Music
If you went to university, what did you study there?	Music, Business Administration
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Prince 2 Managing Successful Programmes Environment Law Corporate & Social Responsibility & Business Ethics

H.7.3 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1	1	A job description that carries out identified activities within an organisation
		Has worked in a number of organisations which have job titles like that – calling on previous experience
B2	0.3	Names of clusters of activity that have been deemed to make sense to be put together – name is how it is referred to by people – entities - organisation
		The word ‘organisation’ appears in the text
B3	0.3	Decision making function – governance structure? – they don’t do it themselves, they have oversight of what other people do, and approve it
		This is what it says that it does – the name matches what it does
B4	1	Means of communication
		Interviewee uses email and phone herself regularly to communicate
B5	1	Processes/procedures
		Spent 6 months working as a business analyst in a bank and then wrote up processes like this
B6	1	Functions / teams within ArchiSurance – something that is done there

		Text – "handle five key types of task" – having worked in a bank for six years, which sold insurance, you recognise these structures
B7	1	It's their outputs – an action that they take – what they do
		Correlation of text between B3 and B7
B8	1	Things that trigger the start or end of a process – or key milestones
		6 months working as a business analyst
B9	0.3	Customer facing functions - teams
		Brain exploding ☺ inconsistent with rest of diagram
B10		A piece of paper with a request on it, that has to be processed A piece of modified documentation – a form
	0.3	
B11	1	A contract
		Experience of legal contracts
B12	0.7	Documents which attach to other key documents – subdocuments associated with key document
		Has previous experience with policy summaries and claim submission forms
B13	0.7	A type of business

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	-1	Because we have two types of blue boxes (a customer is critical to the whole process; the titles within the organisation are less relevant)
B2	+1	They represent geographical locations; and they give a higher level structure of the business
B3	+1	Depends on your perspective
B4	+1	Without them there would be no way of contacting the business
B5	-1	Interesting but not critical
B6	+1	Gives you the next layer down of the organisational structure
B7	-1	Just clutters it up – no new words in here
B8	-1	At this high level (that of a COO) they just clutter it up – also confusing because incomplete examples
B9	-1	Inconsistent, confusing
B10	-1	Remove – inconsistent – it's an output

B11	-1	Not useful – it's just an output
B12	-1	Basic forms, key documents – again not relevant
B13	-1	Doesn't add – inconsistent duplication

H.8 Interviewee: S8

H.8.1 Metadata

Interviewer	Cameron
Date/Time	4th March 2019
Used ArchiMate diagrams?	Y

H.8.2 About the Interviewee

Where do you work?	(software vendor)
What is your job title?	IT Manager
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) Doesn't easily fit any category? A lot of the work is reactive; not really guided by the work of others
What did you study at A-level?	Maths, Further Maths, Chemistry
If you went to university, what did you study there?	Maths
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	No

H.8.3 Application Layer

In this questionnaire, where there are two values, the first one is the interviewee's confidence in their understanding; the second one (in parenthesis) reflects the interviewer's assessment.

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1 (=1)	Things that store information in various forms – information systems
		They are things that data is sat in – they contain data
A2	0.3	Two separate parts of a system talking to each other (should have scored 0.7 or 1 really?)
		Note – this conflicts with the API stuff – this why the confidence is only 0.3

A3	0	
		They don't have anything in common really – because an API is not showing the customer anything (unlike the other A3s)
A4	0	
A5	0	
		Just looks superfluous – things in a box in a box
A6	1	Sequence of steps you go through to come out with a result, based upon data previously input
		Sequence implied by the arrows on the diagram that represent a flow of information "Obtain travel insurance" (A6) doesn't follow the same logic as the other ones (bottom left) – so has raised doubts about A6
A7	0.7 (=0.3)	Data being put into the system
		Recognised by the words on the diagram
A8	0.7 (=0.3?)	A transfer of information to something the customer is going to see
		From the wording
A9	0.7 (=0.3?)	This is where the data is sat - databases
		Based upon guessing what the diagram is getting at!

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	+1	It would remove too much of the diagram
A2	-0.5	Please lose either this or A5 (duplicate)
A3	+1	The A8 symbols wouldn't make sense without them
A4	+1	They clearly relate some separate stuff within the larger subsystems so they are important (but interviewee doesn't know what they are)
A5	-0.5	Please lose either this or A2 (duplicate)
A6	0	It depends on how in-depth you want the picture to be
A7	+1	They are the start to the process
A8	0	Don't really care about these
A9	+1	It depends on how-level the diagram is

The interviewee said afterwards that [it would be] *far easier to spot things of the same type if you were to use colours rather than having to recognise combinations of shapes and icons*.

H.9 Interviewee: S9

H.9.1 Metadata

Interviewer	Cameron
Date/Time	30th October 2019
Used ArchiMate diagrams?	

H.9.2 About the Interviewee

Where do you work?	Own AI-related business
What is your job title?	Technical co-founder, head of production development, doing coding
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) C
What did you study at A-level?	History, Business Studies, Theatre Studies
If you went to university, what did you study there?	Foundation in art and design Film school – uni of creative arts – BA in Film Foundation course for Neuroscience Complex systems (biology, physics) Systems and synthetic biology
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Online courses – R, data analysis, Python, quantitative biology, statistics

H.9.3 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
A1	0.3	Features of the system
		They ARE the system – hence the low score
A2	0	Another dependent process
A3	0.7	The user interface of the platform

		The API doesn't fit this…
A4	0	Something the user would do
A5	0.7	The overall process
		It is what is done by the things inside
A6	0.7	The back end of the system … data processing pipelines
		The word pipelines is why I upped the score here – idea of flow
A7	1	Inputs to the system - a process which has happened
		1 – something has happened - this is what an event is
A8	0.3	Processes – things that are being done
		Has the idea of something that is being done by the system
A9	1	Data sets

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	1	
A2	-1	confusing
A3	1	
A4	-1	Adding another layer of information we don't need
A5	1	
A6	1	
A7	1	
A8	0	
A9	1	

H.10 Interviewee: S10

H.10.1 Metadata

Interviewer	Cameron
Date/Time	16th July 2019, 5th September 2019
Used ArchiMate diagrams?	Yes

H.10.2 About the Interviewee

Where do you work?	(aircraft manufacturer)
What is your job title?	Travel Buyer
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Economics Engineering Drawing General Studies
If you went to university, what did you study there?	HNC in Business Studies
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Office Software Training on specific flight booking systems

H.10.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	The two main players
		Didn't understand the tube symbol – vague about terminology – should this be a one? No-one more important to keep happy
M2	0.3	Measure – something that you can in some way measure -a result – for example a customer, what helps them carry on using the service
		There is an element of these being important – but the wrong part of the taxonomy
M3	0.3	A thought how satisfied the board or customer are with what they're receiving
		Cloud makes interviewee think it's a thought Again, bears some relation…
M4	0.7	Must-haves – essentials – what must be done
M5	0.7	Results – you can specifically measure and track them – you can inspect the numbers and track them
		Still a bit abstract – yes you can measure them – you can measure other things as well – but pretty close
M6	0.3	Tasks – to help improve the previous two – the results
		They're not work to do – they are things to reach

M7	0.3	Measures – which help achieve the previous stuff – something you can measure on an ongoing basis
		Similar in some way to outcomes
M8	0.3	Describing something coming together – a consolidation of data into one form -
		Exclamation mark => warning that you need to do these tasks
M9	0	Held elsewhere – being input into this process
M10	0.7	Warning – could be exposed
		Sign indicates an area to look at Feels like he understands

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	
M2	0	They are subjective (on customer side)
M3	-1	A repeat
M4	+1	Important measure
M5	+1	
M6	+1	Important tasks
M7	0	Could be integrated into M6
M8	+1	
M9	0	
M10	0	Could be integrated into M6

H.10.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	People – real humans
		True, but not the essence of it
B2	0.7	Teams/groups/offices of people
		Closer…
B3	0.7	Outside bodies working for the benefit of the company

		An intertwining of two groups
B4	1	Contact points – how people can get in touch
B5	1	Process flow
		Have a lot of these in his employer
B6	0.7	Departments within the company
		Pretty close, especially given the examples
B7	1	The task of B3
B8	0.3	Inputs and outputs to the processes to or from the customer
		True but that's not the essence
B9	0.3	Tasks performed to complete the process within a department
B10	0	One of their key tasks – a claim is received
B11	1	A legal document that goes to the client
B12	0	Inputs into process
		Symbol suggests a piece of paper?
B13	0	Is it to do with this being a legal process?

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	
B2	+1	
B3	+1	
B4	+1	
B5	+1	
B6	+1	
B7	+1	

B8	+1	
B9	+1	
B10	-1	
B11	+1	
B12	+1	
B13	0	

H.11 Interviewee: S11

H.11.1 Metadata

Interviewer	Cameron
Date/Time	18th June 10:00
Used ArchiMate diagrams?	No for Motivation, No for Business

H.11.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Enterprise Architect (Customer)
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (b)
What did you study at A-level?	Maths, Chemistry, Physics
If you went to university, what did you study there?	Bsc Chemical Physics PGCE Secondary Science
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Financial Process Programming Languages Systems Analysis Certified Business Analyst Prince 2 Programme Management TOGAF COBIT Consultancy SaaS courses, technical Marketing, Management, Leadership

H.11.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1	1	Stakeholder, stakeholder group, people
M2	0.7	What the people want – motivational drivers – what drives their behaviour
M3	0	Top level goal of what they want to achieve – goal or outcome – measurable- - top level driver
		Missed the idea of meaning specific to an individual or group
M4	0.3	Measurable outcome – something that you could measure through a KPI of some kind
		In the same ball park
M5	0.3	Levers – rationale – risks or concerns – dependencies – things you have to get over to reach the objectives
		Some basis in reality?
M6	1	Things you would do to reach the desired state
M7	1	The contribution of the things that you're doing – outcomes at a particular level
M8	1	Principles
M9	0.7	Constraints … the mechanism by which you'll achieve the M7 things …
		Fairly close to a requirement
M10	1	Constraint

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	Everything links back to the stakeholder groups
M2	+1	Different stakeholder groups have different wants
M3	+1	They don't to be different to the M4s

M4	+1	They're the key things that the stakeholders need
M5	-1	They don't add value; they could be incorporated in the KPI measures
M6	+1	Needed as they are what you're going to do
M7	+1	How you measure it
M8	0	Could put them somewhere else?
M9	0	Put in a separate diagram at the next level down, maybe?
M10	0	Not useful at this level

H.11.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1	0.7	An actor, a person, a physical being
		They relate to people, but what the people do rather than people themselves
B2	0.3	Organisational units – a group of departments
		Some concept of people in there – misled by the names used in ArchiSurance
B3	1	Agreement between different stakeholders to do something
B4	1	Channels – communication media
B5	1	Key processes and process steps
B6	1	Team groupings – things that you do within an organisational structure
		Fairly close to the definition
B7	1	What the stakeholders do – relates to B3
B8		You receive something, you send something, an information flow – an event
	1	Not very close to the definition – but a true statement
B9	0.7	Almost business capabilities that these link into – a thing the business does carried out through a process – the grouping of the processes

		Pretty close!
B10	1	An object within the business, pushed through a process
B11	0.7	An agreement, a document
		Hesitating a bit, a document has the wrong idea, agreement has the right idea
B12	0.7	Physical documents that act as an interface?
		Yes they are documents, but missed link to business objects
B13	0.3	Group of capabilities – the things the business does -
		Missed the essence of what we are after – the whole point of the business – the product it is selling

Does the business model fulfil the purpose specified in the reference material (Y/N)?	Y

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	+1	The language needs them, although not necessarily on this particular diagram
B2	+1	They are useful
B3	+1	
B4	+1	
B5	+1	At least the high level ones
B6	+1	Very useful – especially mapped to processes
B7	-1	Duplicates what's in B3
B8	+1	Yes they are useful
B9	+1	Depends on the level
B10	+1	Yes you need to know how it's pushed through the process
B11	+1	You need to know what contracts/agreements you have
B12	+1	Yes you need to have an idea of how the info is coming in and going out
B13	0	Don't really know what it is

H.12 Interviewee: S12

H.12.1 Metadata

Interviewer	Cameron

Date/Time	5th October 2019
Used ArchiMate diagrams?	Yes for Motivation, No for Application

H.12.2 About the Interviewee

Where do you work?	(electronic design automation software company)
What is your job title?	Applications Engineer
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (C)
What did you study at A-level?	Maths, Science, Biology, Latin, English, Afrikaans,
If you went to university, what did you study there?	Physics Electronic Engineering
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Chip Design – verification All training is on new product (very technical) Supervising people Time management

H.12.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	0.3	An entity that is affected by the process, or that drives the process - a piece in the game
		Listening to the UTILITY part, there is some understanding coming through, so I’ve changed this from a 0 to a 0.3
M2	0	Intangibles – nouns – concepts – can’t directly interact with them (like you can with M1s)
M3	0	Like M2s but less tangible- more abstract
M4	0.3	No clear message from these – thresholds, measurables, obligatory / mandatory things
		The steering hints at something… - some way towards a driver
M5	0.7	Can be measured in some way – measurable inputs to the process
		Magnifying glass – inspecting something – the FACT that share price is falling – this is very close to an assessment

M6	1	A target of some sort – some state that you want to achieve – an outcome that you want to establish
M7	1	A target that has been hit – something we've already done that has been achieved
		Meaning for this came from the bullseye and arrow
M8	0.7	Stop and pay attention to this – more than just a target – an essential item, a non-negotiable – very important – has to be achieved
		! emphasising something important
M9	0	An element in the whole picture – in the process
M10	0.3	Something in the environment – a de-facto input, can't change it, need to work with it or work around it

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	Everything else is driving towards the requirements of these groups
M2	+1	That tells you what these two major players are concerned about
M3	+1	
M4	+1	Because of their content
M5	-1	Because the current state isn't as important – declutters the diagram
M6	+1	
M7	+1	Although detailed, they can have an impact on the business
M8	+1	As they are apparently indispensable
M9	+1	
M10	+1	

H.12.4 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
A1	1	Application systems
A2	0.7	A module that encapsulates/encloses these detailed modules (A4) –

		When you get to a certain stage of activity or action, you then need to call up these separate blocks – moving through layers or boundaries, where the actual business of doing a task is buried inside these modules – some kind of communicating / coordination function
		Kind of like collaboration
A3	1	A communication layer with the employee – like an interface
A4	1	Detailed data processing blocks
A5	0	Just application layers – a container for the lower level activities
A6	0.7	Specific data processing and preparing steps – algorithms within the system
		Pretty close to processes – does keep using the term 'function'
A7	0.3	RFQ – an input from a customer Data Acquired – input again So an interface to the external world
		Got the external aspect – kind of in the right direction
A8	0	Interfaces between the system and the human visible part
A9	0.7	Customer information
		Got the right idea

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	+1	
A2	-1	It's not adding much information – adds more complexity
A3	+1	
A4	+1	Explains what you're doing within the admin system
A5	-1	Without this (already lost A2) you'd have Get Quotation going straight to Prepare Quotation – makes sense
A6	+1	Key elements
A7	+1	For example if you didn't know there was an RFQ coming in then the stuff on the right wouldn't make any sense
A8	+1	
A9	0	

H.13 Interviewee: S13

H.13.1 Metadata

Interviewer	Cameron
Date/Time	23rd July
Used ArchiMate diagrams?	Yes

H.13.2 About the Interviewee

Where do you work?	(a church), before that – originally internal audit manager, eventually IT director (with no training or experience); got sold again; set up call centre; then worked for consultant in Siemens contact centre team – then freelance – then again back for Sabio (there a business consultant)
What is your job title?	Pastor now; previously freelance IT consultant
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (c) - takes guidance from architects
What did you study at A-level?	French and Spanish
If you went to university, what did you study there?	2 postgraduate diplomas – both in theology – one in Christian Studies, one in applied theology, specialising in Christian Leadership
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Prince 2 – was a practitioner MSP Some ITIL stuff Six sigma ISO 9000 stuff as well Did these when freelance, to get more credibility

H.13.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	Groups of people, categories of people – stakeholders (but not complete list)
		Database symbol is a bit confusing! (referring to the 'role' symbol)
M2	0.7	High level objectives (as opposed to tactical) Perhaps desirables, rather than actual goals Stakeholder desires (this latter one changed the score for M1 to 1)

M3	0	These are almost superfluous – conceptual?
		This was implied by the cloud – again a misleading symbol
M4	0.7	Goals – areas for improvement, areas for action
		Scored higher as these relate to action – hence 'driving'
M5	1	Presenting issues – describing what needs to be acted on in the goals
M6	1	Targets being set to achieve the M4s
		Image of a target
M7	0.7	Measurables, deliverables, something that has been achieved?
M8	1	Organisation rules – in this context, become constraints on projects
M9	1	Combination of different requirements – org inputs that constrain the type of solution
M10	1	Commercial constraint

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	
M2	+1	
M3	0	They don't really add anything
M4	+1	Absolutely essential
M5	+1	Add information without substance ; could be presented better
M6	+1	Give focus to the whole diagram – absolutely vital (with his project manager hat on)
M7	+1	They provide a clear focus
M8	+1	They have to be there, more from an architecture point of view than a project one
M9	+1	Need them in there
M10	+1	Could manage without it for this diagram, but you'd need it later on

H.13.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.7	Actors in a process – people who have a specific role in the business' processes - active stakeholders
		Said the word 'role'…
B2	0.3	Org unit that executes processes – internal or external
		Misled here by the example… which uses all org units as the actors…
B3	0	Reviews rather than initiates – a different kind of actor
B4	1	Types of communication (inbound in this example)
B5	1	Processes and process steps
B6	1	Functions within org units
B7	0	Review process
		Needed more examples…
B8	1	An event – a milestone
B9	0.3	Departments ? almost interfaces between processes and org units
		Very confusing… some element there of the right meaning, with the interfaces
B10	0.7	An event? Process? that has various inputs to it – "doesn't become a claim until you have … " – getting towards the right idea here Something that is created that has a limited lifespan – a transient event?
B11	0.7	A business object – that is not a short term thing – it is created and has a life which may be short or long
		Didn't get the legal bit, but pretty close
B12	0.3	Inputs to processes?
B13	0.3	A higher level business function
		Sort of incorporates other stuff

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	
B2	+1	
B3	-1	Confusing – better if they were gone
B4	+1	
B5	+1	
B6	0	They don't really serve much process, not much connection to the rest of the diagram
B7	-1	confusing
B8	+1	
B9	0	
B10	+1	
B11	+1	
B12	+1	
B13	0	

H.14 Interviewee: S14

H.14.1 Metadata

Interviewer	Cameron
Date/Time	18th June 2019
Used ArchiMate diagrams?	Yes for Business

H.14.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Data Governance Architect
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (b)
What did you study at A-level?	Pure & Applied Maths

	Physics Geography Art
If you went to university, what did you study there?	1 year of a degree in Land Survey
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	ARIS (process, CRUD) Casewise (EA, process modelling centric, also data modelling) InfoSphere (info architecture, data quality stage, info analyser) Prince 2 foundation, practitioner Sewerage and Sludge Processes NVQ level 2

H.14.3 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	1	Roles (party roles in data modelling)
		Data modelling is what helped here
B2	0.7	Organisational bodies
		Note: Pretty close to an actor (could be confused with functions)
B3	0.7	A decision making body – could be accountable for the process
		Note: within the process – scored 0.7 as the relationship to the process is established
B4	1	Communication channels
		Note: this is the same as an interface
B5	1	Business processes
		Their titles and the fact that the diagram shows a flow of sequential activities
B6	0.7	Teams or organisational substructures, departments
		Note: yes they are teams, according to the diagram, however this isn't quite the meaning within ArchiMate
B7	0.3	A process – a series of activities / tasks / decision points to take triggers to outcomes
		Note: it's something that happens, but nothing like a process
B8	0.7	Process triggers and outputs – starts and ends of processes
		Note: pretty close – unable to abstract out to 'events'
B9	0.7	High level processes or capabilities
		Note: similar in meaning to a business service, missing the semantic of externally exposed

B10	1	External trigger – the data entity
B11	0.7	Control, governance
		Has the kind of legal implications that we're looking for in a contract
B12	0.7	Documents
		Note: Yes they are kinds of representation
B13	0.3	The high level service

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	Did say you could lose some, but you couldn't lose customer (note: purpose of the diagram – the narrative – is perhaps too restrictive?)
B2	+1	Fairly fundamental to the business structure
B3	+1	Interviewee likes detail – wants to know who is accountable
B4	+1	Understanding interaction with the customer
B5	-1	If we're looking for structure and services – these describe the how, not the what (again, down to the purpose of the diagram)
B6	+1	Fairly fundamental
B7	+1	Yes for a detailed level
B8	-1	Too much detail here, they go along with the process
B9	+1	The placement of the services
B10	-1	Can be inferred from the fact that we have 'claims administration'
B11	-1	Take it off
B12	-1	Children of other things I've already removed
B13	+1	The services we're providing

H.15 Interviewee: S15

H.15.1 Metadata

Interviewer	Cameron
Date/Time	6 May
Used ArchiMate diagrams?	No

H.15.2 About the Interviewee

Where do you work?	(global food manufacturer)
What is your job title?	Programme Lead - Governance
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (A)
What did you study at A-level?	Biology, Chemistry, Physics
If you went to university, what did you study there?	BSc Biology Msc Applied Biomolecular Technology PhD Food Structure
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Project Management Marketing

H.15.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1	1	Sort of like stakeholders – in the sense that two parties for whom it is important that the business is successful – but on different sides of the wall
M2	1	The things that matter to stakeholder groups – the desired outcomes for the stakeholders
M3	1	A thing called “Customer Satisfaction” – two different definitions of this – both of these are sort of like types of engagement or fulfilment
M4	0.7	These are like the inputs – the things that make the stakeholder engagement possible – if you get these right then other things will succeed – they lead to fulfilment of the satisfaction part for the stakeholders
M5	0.3	A bit like “IF you fail to do X then” – so they are like negative variables that could determine whether the M4s come true or not – like tests one could apply *Note – missed the real nature of these – assessments not tests*
M6	0.3	Also variables that you could put in in order to achieve customer satisfaction – they affect M4s

M7	1	Positive outcomes that serve everyone – outcome meaning what happens as a consequence of the way you run business processes
M8	0.7	Simplification, standardising, harmonising – having the same protocol everywhere
		Met these before, e.g. "single version of the truth" -
M9	0.3	Ultimate goals – what success would look like – also allow compliance with regulatory stuff *Note – not clear distinction from principles or objectives etc.*
M10	1	Something that impedes your capability to comply – a hindrance

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	1	They represent two distinct groups that don't have completely matching ends
M2	1	It shows the commonality and difference in goals of stakeholder groups
M3	0	Kind of duplicating what we can see already
M4	1	They highlight different needs that the stakeholder groups has
M5	1	Like them in that the ones on the LH side the consequence of things not going right (not so useful on the customer side)
M6	1	They do show what happens if you make various positive or less positive inputs to the process
M7	1	Really helpful – they call out something regulatory about doing the entire thing
M8	1	Useful because it speaks to global harmonisation of processes which usually reduces cost or makes for efficiencies
M9	1	They show the absolute consequences if you don't have breaches of data etc. – without them you might miss the point of compliance – so makes the message stronger
M10	1	What's the impact of lack of suppliers

H.15.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1	0.3	Stakeholder – parties who have a vested interest in the output of the organisation
B2	0.3	Functions – an organisational grouping that is responsible for carrying out certain tasks to achieve the organisational goals
		Has just worked in a certain type of place *Note – want to score it higher – think the models are misleading as all the actors given are organisational*
B3	0.7	Committee – a board of senior leaders who take input from different functions and have to make a recommendation that determines output actions
		The data that J presides over feeds into those people that make decisions on strategic direction based upon this data – an oversight function
B4	1	Communication Media
B5	1	Business processes
		They are sequential steps that have to be completed to achieve a designed output Got the idea from project management and stage gates; scientists might not have got this ten years ago, but now it's more important (in fact critical in areas like healthcare)
B6	0.3	Job families that sit within functional areas (B2s)
		Probably coloured by how things work around the interviewee e.g. R&D is a function, programme management or product design or sensory science is a job family that sits within the function – have to interact to deliver the output of the function *Spent a lot of time focusing on job families – may have influenced answers here*
B7	1	A goal or objective of the B3 – mission or goal of B3 – or its reason for existing
B8	1	Two are inputs; one is an output. They are actions that feed into a process or are the output of a process. By action, interviewee means events that happen that lead to a business process needing to be set into motion. Actions, happenings…
B9	1	Services – things that are delivered to customers (ultimately)
		Interviewee's function is delivering things to people that need them (internal customers in this case)

B10	0.7	Formal documentation or record of an event that happens that has consequences – a tangibly documenting thing – recorded in history
		Note – maybe lacks the vocabulary or experience related to Information Architecture? Checked – interviewee has never used any BI systems
B11	0.7	It's the goods or service that the customer physically sees; and it's also got legally binding consequences
		Note – could have scored a 1? Didn't say contract, and thought it could have been either (or both) of the above two meanings
B12	0.3	Documentation (sometimes internal) – for record keeping -
B13	1	Product offering (service offering)

H.16 Interviewee: S16

H.16.1 Metadata

Interviewer	Cameron
Date/Time	9^{th} May 2019
Used ArchiMate diagrams?	no

H.16.2 About the Interviewee

Where do you work?	(manufacturing company)
What is your job title?	Credit Controller
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (none)
What did you study at A-level?	German, Applied Business, Sociology
If you went to university, what did you study there?	German Studies
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Management Accounting

H.16.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation

M1	1	Two main stakeholders
M2	0.7	Goals or desires for the two stakeholder groups / primary objectives
		A bit confused with drivers
M3	1	CS is of different relevance/importance to board and customer – there are two ideas of customer satisfaction and the value placed on that
M4	0.3	Direct motivation – means to achieving the objective – targets – means of assessing success
		Some idea…
M5	0.7	Failures to achieve the goal (linked to pink item)
		Close enough to an assessment
M6	0.3	Direct means to achieve the primary objectives / foci Greater specificity than the M4 (driver)
		Got more specific – but didn't recognise as targets in themselves
M7	0.7	Quantitative method of assessing M6 – success within those goals
M8	0.3	Ways to achieve more specific goals
		Yes they are – but reused multiple times
M9	0.3	Means to achieve the linked goals
M10	1	A challenge / obstacle regarding the objective

H.16.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	Thought these might be stakeholders
B2	0	Locations?
		The titles suggest this
B3	0	A means of assessing performance
		I said a customer focus group might be another example of one of these
B4	1	Means of communication

B5	1	Processes and stages of processes
		interviewee has seen these before
B6	0.7	Services – functions – the operations and tasks they undertake
B7	0	Constructive feedback based upon performance review
		Note: Difficultly here in creating good examples
B8	0.3	Start and end points of processes
B9	0.3	Still functions provided by the box within which they are contained
B10	0.3	One of the expected liabilities – started with a NOUN
B11	0	The product they actually sell
B12	0.7	Documentation required
B13	1	What they're selling – the PRODUCT

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	0	You'd expect to have these things there – you can assume they're there
B2	+1	As a location / holder for the others
B3	+1	
B4	0	You can assume these are the methods of communication
B5	+1	
B6	+1	
B7	0	Superfluous – implied by contents of B3
B8	+1	
B9	+1	
B10	0	Implied by what's around it
B11	+1	

B12	0	We assume that there will be documentation
B13	+1	

H.17 Interviewee: S17

H.17.1 Metadata

Interviewer	Cameron
Date/Time	30/10/19
Used ArchiMate diagrams?	No

H.17.2 About the Interviewee

Where do you work?	Non-Governmental Organisation based in India
What is your job title?	Co-founder, CEO
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Chemistry, French, German, English
If you went to university, what did you study there?	French and German
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Psychopathology Statistics

H.17.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	The people you have to keep happy
M2	0.7	The points that ensure the people are happy
M3	1	The priorities – the interpretation by the M1s of how to reach the goal
M4	0.3	Overarching goals – the boxes you have to tick – action points – the components that hold the organisation together - goals isn't the right word

		– all of them have to be maintained for the business to carry on and be successful
M5	1	Current status of the M4s
M6	0.7	Action points to stay in line with the overarching M4s - ways of achieving / maintaining the M4 – what needs to be done
M7	0	Action points specific to this business -
M8	0	
M9	0.7	Data related action points that require specific software or IT specialties
M10	0.3	External problem that's out of the control of the business
		Didn't have a reference to the requirement that it inhibits – but some idea anyway
Does the motivation model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	1	
M2	1	
M3	-1	They are superfluous – where are there two separate ones? They should have similar priorities / interpretations of customer satisfaction
M4	1	
M5	1	No solutions to these on the diagram keep them in because interviewee likes detail!
M6	1	Again detail-oriented
M7	1	They give specific targets
M8	-1	As didn't know what these meant
M9	-1	Don't need specifics to this level in this kind of diagram
M10	1	Need to know about it

H.18 Interviewee: S18

H.18.1 Metadata

Interviewer	Cameron
Date/Time	28th March 2019
Used ArchiMate diagrams?	No

H.18.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Enterprise Technology Architect
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (B)
What did you study at A-level?	Physics, Music, Maths
If you went to university, what did you study there?	Music (BA) Music and Film (MA)
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	

H.18.3 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
A1	1	The applications in use
		They say 'system'
A2	0.3	Pigeon-holing stuff – no particular physical or software concept – potentially could be an API layer that grants access to the external siloed applications – could be API management? Inner box – technical focus
A3	1	Application interface
		We'd had a number of discussions on this specific topic over the past couple of weeks
A4	0.3	Business concepts of functionality?
		Can't see very much that ties these things together
A5	0	High level business process – outer box – business focus

		(a) The text, (b) it encompasses a number of lower-level bits of functionality that comprise the process
A6	0.7	These are processes
		The three on the left are technical (IT) processes; the two on the right are a bit different; obtain travel insurance is a desire, or a process?
A7	1	Inputs to a particular process, or triggers for a process – a combination of both, perhaps?
		Request for a Quotation – is likely to be coming from a person, with accompanying information Data Acquired – data appearing from somewhere, again has information (input) and the data is itself the trigger *Note: I should have had an event that didn't trigger a process?*
A8	1	Modules of technical functionality executed from the user interface
A9	0.3	Data stores – somewhere that you put data – focusing on the place it's stored, to think about synchronisation, backups, etc.
		Not notated correctly – should be drawn with a cylinder (for data) or as something that represents files

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	+1	
A2	+1	Because A2 gives the technical focus … *
A3	+1	The API is of interest; the screens, less so
A4	+1	I told the interviewee what these were (internal functionality) [after part 1 of the process]
A5	+1	… and A5 gives the business focus
A6	+1	Different instances of these are perceived as being of different value – but my method forces a + or - 1
A7	+1	They identify the start boxes of workflows
A8	+1	Helps explain what the user is interacting with (what functionality)
A9	+1	Absolutely mandatory

H.19 Interviewee: S19

H.19.1 Metadata

Interviewer	Cameron
Date/Time	28th March 2019
Used ArchiMate diagrams?	Yes for Business, no for Application

H.19.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Service Architect
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (B)
What did you study at A-level?	English, History, Maths
If you went to university, what did you study there?	Art then switched to Computing Masters in Computer Science
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	PMI, Prince COBIT, ITIL CISSP, CISSM MCSE, MC&E HP-UX

H.19.3 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	Told the interviewee that this wasn't data (because of misleading symbol) – so try again! Could it be an action that the investment manager has to carry out? Action relates to role, right?
		Means data – a barrel represents data – same issue as experienced by Steve Wilson
B2	0.3	A top level organisation? So get that it can involve people, but not the abstraction above that
		Deduced was an OU from the names of the elements of this type
B3	0	

B4	0	A system
		Email is a technical service; also a closed system Phone is a service, also a system in its own right
B5	1	All processes that link into other processes
		An arrow within a box indicates a process – possibly encountered from mainframe days? Also arrows between boxes give it away
B6	0.7	Subunits within the organisation (still not clear on functions)
		These are grouped within the B2s, so they are subunits
B7	0	
B8	0.3	Start of a process? (yes it does trigger a process)
		Doesn't match BPMN notation, so a bit confusing (BPMN uses a circle)
B9	0	
		Doesn't understand differentiation between B6 and B9
B10	0	Something that impacts the organisation – a grouping of processes?
		The notation suggests a process (maybe a grouping of those underneath it) – following BPMN notation
B11	0	An intermediate process
		Again just means processes (because this shape is used within BPMN)
B12	1	This is a document
		This is the symbol for a document in both Visio and BPMN!
B13	0	Looks like a process that hasn't been fulfilled
		Don't see the point of this

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	0	
B2	0	
B3	-1	
B4	0	
B5	+1	
B6	+1	
B7	-1	

B8	+1	Even though it's not obvious what these mean
B9	+1	Although could be achieved by annotating the relations instead
B10	+1	As this is a header for processes (!)
B11	+1	An intermediate process
B12	+1	
B13	0	Don't get it, don't care

H.19.4 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1	Some kind of system
		The words in the boxes
A2	0	Process module
		Something that has to be done
A3	1	Data entry points – data transfer points = application interfaces
		Used them before
A4	0.3	Activities – going to get something, changing something, to prepare something, to complete something – like a process but part of a process Note: *But not the grouping concept that makes up a 'function'*
A5	0.3	Again, a top-level process – something that has to be done (which is true!)
A6		A process – system processes
	1	Series of activities
A7	0.3	Input that requires a result *Note: consistent with business event score*
A8	0	Some form of process
A9	0.3	Electronic document storage
		They're fixed – e.g. customer details don't change regularly – long term storage of data in a system of record
Does the application model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		

Code	Rating	Explanation
A1	0	
A2	+1	
A3	+1	
A4	+1	Doesn't make sense without these
A5	+1	
A6	+1	
A7	0	Not of value in their own right – just a starting point
A8	+1	These are intermediates that you need
A9	0	These are things that you need, but you probably wouldn't miss it

H.20 Interviewee: S20

H.20.1 Metadata

Interviewer	Cameron
Date/Time	10/4/19
Used ArchiMate diagrams?	M yes

H.20.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	3rd Party Assurance Consultant
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) Has been all three (currently a business architect)
What did you study at A-level?	Zoology, Chemistry, Physics
If you went to university, what did you study there?	Electrical & Electronic Engineering
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	ISO27001 PCI Privacy (GDPR)

H.20.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	0.7	People with responsibility who take inputs and drive outputs

		Prime movers within the model – everything else serves their purposes – the model exists for those two groups (people, not functions) – they drive functions
		Everything else is either a function or a value Would have given 1 for a stakeholder Customers are consumers – they can't drive change Whereas the board can influence these things – one's a motivator, one's a consumer – a second symbol would help to differentiate them
M2	0.3	They are both inputs driving outputs (the value of them motivates the board)
		M2s in two separate groups depending on the M1 that they serve – primary and secondary M2 as well (used the value word once?)
M3	0	A metric, has a value attached to it
		Would condense the two clouds into one? Passive and active values
M4	0.3	Outcome Objectives (on the left hand side): something the board sets a value in order to achieve (something we want to achieve) Note: are these bad examples of drivers – they *could* be read as objectives? Variables to achieve the outcome
		"we want to drive to achieve" – several of these are arbitrary -
M5	1	Risks? We set a threshold for success – we fall below that threshold… There are only two, they are both risks – hang on, the third isn't a risk… it's an outcome? Negative thresholds for motivation – Key Performance Indicators?
M6	0.7	Improvements – efficiencies – almost but not quite objectives (objective was used for only one of them)
		With one exception, they are all improvement – all customer data in the UK is an objective so a bit different
M7	1	Specified targets, with one exception (no security breaches – as outside our control) – this is just a synonym for objective
M8	0.3	Design objectives? Design requirements? Don't see these as different to one-off requirements
		Kind of reminiscent of principles?
M9	0	Don't make sense
		PCI DSS compliance is mandatory, specific, defined, imposed by law Integration with existing CRM is none of those things Are these bad examples? They belong in different camps…
M10	1	A constraint:

		A constraint is "in order to achieve your objective, you don't have freedom in this area for various reasons"

H.21 Interviewee: S21

H.21.1 Metadata

Interviewer	Cameron
Date/Time	30/10/19
Used ArchiMate diagrams?	No

H.21.2 About the Interviewee

Where do you work?	Independent
What is your job title?	N/A, was Managing Director
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Physics, Maths
If you went to university, what did you study there?	Started BSc Aero course, then stopped, when straight into employment
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Procurement Marketing M & A, Culture Integration Man Management

H.21.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	Stakeholder
M2	0.7	KPIs – metrics? Set of corporate objectives? Not clear which is the best concept – a set of objectives that would be mostly quant, but some qual – five overarching areas – closer to what 'value' is I think
		These are areas where metrics will appear – but they aren't metrics themselves Obviously understands motivation
M3	1	Need the two stakeholders to give you a proper customer satisfaction column – as they have two perspectives

M4	0.3	The things that the stakeholders have control over – drives the methodology underneath it
		Some understanding of the concept of driving ?
M5	0	All negative statements - either the effect of, or the things you guard against
M6	0.7	All positive statements – immediate actions or improvements of an existing thing – positive benefits that help you achieve the overarching objectives
M7	0.7	Things that you should be aiming for, corporately non-negotiable
		Changed from 0.3 – not clear from material that these have been achieved, rather than should be achieved. Correctly identified, in utility section, they are quantitative
M8	0	Efficiency and cost saving
M9	0	
M10	0	
		Seems to be hanging – noted + and – on the relations

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	1	Without a doubt
M2	0	Implicit if you understand the rest of the information
M3	1	It would show a lack of understanding of the fact that it's a two-sided function
M4	1	Stakeholder is not specific enough
M5	1	These are sufficiently specific
M6	1	
M7	1	They are numeric, firm targets, measurements by which you run the business – the only quants on here
M8	1	You'd be losing an obvious benefit
M9	1	
M10	0	

H.21.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.7	Decision makers, people in charge of an outcome, they all collect information in order to make a decision – they are a person
		Getting closer now to a role – now he said the word 'a person'
B2	0	They are locations – physical tangible things with bricks, walls – places – a batching of functions / activities / people within a location
B3	1	A collection of people from different parts of the business with a common purpose in making a decision – a functional group
B4	1	Bits of kit – tools – conduits for communication
B5	1	Transaction flow – workflow – a process, a list of things that have to happen
B6	1	Functional departments (of people) within the locations (B2)
B7	1	Data supplied to them (B3) that they then have to work on – the objective of that functional group
B8	0.7	An input of information from an external source, that allows the company to start the transaction.
B9	0.7	Customer facing points – incoming and outgoing points of contact
		'explicitly defined exposed business behaviour'
B10	0.7	It's a subject – it drives some processes
		A claim is a claim – probably does get this?
B11	0.3	Has to fit within a set of parameters that the company sells – a filing system - a data bank of what's been sold
B12	0.7	Physical documents – things you can touch – templates
B13	0.7	What do we do – it's the business

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	1	They are people that make a decision – need to know who they are
B2	1	These show how the lines of business go into a batched function
B3	1	
B4	1	it is pretty obvious – but important if you're designing the company, to know who needs emails etc.
B5	1	It shows how the workflow works
B6	1	It shows what functions you have within the business
B7	1	
B8	1	
B9	1	That's where I need to know to keep my people motivated to do all the customer satisfaction stuff
B10	1	
B11	1	
B12	1	It shows me where my standard documentation sits
B13	1	It batches the functions for insurance

H.22 Interviewee: S22

H.22.1 Metadata

Interviewer	Cameron
Date/Time	18th June 2019 (M), 20th June (B), 25th June (A)
Used ArchiMate diagrams?	Yes for Motivation, Yes for Business, No for Application

H.22.2 About the Interviewee

Where do you work?	(utility company)
What is your job title?	Platform Architect (telephony)
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (b)

What did you study at A-level?	Communication Studies Media Studies Business Studies
If you went to university, what did you study there?	Cert HE Open University - Humanities Dip HE Politics, Economics, Sociology
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Cisco CCNA networking Diploma in Business Analysis Diploma – Solution and Enterprise Architecture ITIL foundation Prince 2

H.22.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	Stakeholders
		The text gives it away Don't like the cylinder that represents it!
M2	1	The reasons, desires, of the particular stakeholders
		From the text and placement on the diagram
M3	0.3	The outcomes of a job well done – external to the entire process
		What you'd report on or measure The cloud is not helpful here!
M4	1	Goals – these things are driving, steering, people in a particular direction - objective
		Wheel (ArchiMate symbol) suggests steering – actually not a bad metaphor?
M5	1	What they've noticed in relation to their objectives
		Correctly identified a real world state
M6	1	A target – how you will achieve the objective
		Used M7 symbol in conjunction with M6 symbol as they are related?
M7	0.7	The statistic that you're driving towards
		So the arrow on the target symbol is helpful here Note: not 1 because this is something that has been achieved not something that we want to achieve. This is perhaps a failure in the material, rather than the language?
M8	0.7	A tactic – what we will do (a "how")
		Exclamation mark normally means notice me, warning, exclamation
M9	1	Another how – requirements -

M10	0.3	A challenge – external reference? – an issue outside the organisation
		Note: some recognition of difficulty
Does the motivation model fulfil the purpose specified in the reference material (Y/N)?		y
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	Very useful, definitely
M2	+1	Also useful, they represent the purpose, what's driving the stakeholders
M3	+1	There is value, but also duplication with M4 (driver)
M4	+1	Where we want to go
M5	+1	Adds context (not as valuable as M4)
M6	+1	important
M7	+1	Shows flow of logic by linkage on diagram to M6s
M8	+1	Need to know how we'll achieve a target
M9	0	Can't differentiate between M5 and M9 and M10 - merge with M5s?
M10	0	Can't differentiate again

H.22.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	1	Stakeholders?
		Customer doesn't sound like a role – thinks roles have to be jobs? Previous BA diploma track had very specific definitions of roles and actors (e.g. UML use case diagrams) which didn't match ArchiMate so this made it harder
B2	0.3	Departmental areas and the business itself – so relates to business hierarchy and the functions within that
		Note: scored 0.3 – as the examples given are all business units, but that's not the point…
B3	0.3	Reviews inputs and produces outputs
		Venn diagram – two interconnecting areas (good clue!) -
B4	1	Channels – how people with interact
B5	1	Business process

		They're all lined up sequentially, linked with arrows – looks like steps
B6	0.7	Looks like teams within the departments
		Pretty close to what they are
B7	1	A function / output – something they have to do
B8	0.7	Input or output – triggers something
		Associated with a process – note: events can be triggers
B9	0.3	Filter – a governance or policy or checking function
		Note: an idea of something being done
B10	0.7	Now in document / case / ticket territory – the place that you're going to store all that information – a record – this isn't static – may have multiple
		Note: has the idea that this is a kind of information against which we can track stuff
B11	0.3	A standard document – fixed document that we're going to be referencing against - more permanent than a claim
		Note: elements of something fixed
B12	1	Outputs of that particular object – a document that will summarise the claim
		Relationship to B10s gave it away
B13	0	A function that isn't necessarily a team or department

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	Wouldn't work without customer
B2	+1	
B3	-1	Looks to be redundant
B4	+1	Very important (what interviewee deals with every day!)
B5	+1	Very strange to have processes in the same page as the higher-level concepts – but processes are always valuable
B6	+1	Definitely valuable
B7	-1	Can't see the difference between B6 and B7
B8	0	This could be part of the process?
B9	0	Can't see difference between B9 and B6

B10	+1	useful
B11	+1	Also useful -
B12	+1	How we're presenting information – useful to know
B13	-1	No idea what this is

H.22.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1	Core systems - applications
		From the names
A2	0.3	Almost like a process, or a function – a process outside of an application – (which it is, sort of)
		Note: had already been through business interview and read the explanation of business collaboration and interactions – this is why I didn't use ArchiMate for this layer – the same symbols
A3	0	Read and write type data elements – where it comes from, where it's transmitted, where it's consumed
A4	0.3	Documentation within the systems
		Note: later on, in part 2, called these functions in passing, so changed score to 0.3
A5	0.3	The function contained within the team
		Bears some relation to the definition
A6	0	Stuff they're doing with data – aggregation of something?
A7	1	Data inputs – I want data **trigger**
A8	0.7	Processes or use cases that we require systems to be doing
		Note: what they are doing
A9	0.7	What you'd see if you were given access

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation

A1	+1	Definitely needed
A2	0	Depends upon what it means?
A3	-1	Difficult because meaning isn't clear in the first place
A4	+1	useful
A5	0	Depends upon what it means?
A6	+1	Very important
A7	0	Less useful – almost implied that these things happen
A8	+1	Very important – need to know what the application is being used for
A9	+1	Wanted to score 0 because not done consistently across all apps – when I suggested that interviewee assumes they had been done across all apps, he said they were really important.

H.23 Interviewee: S23

H.23.1 Metadata

Interviewer	Cameron
Date/Time	5th October 2019
Used ArchiMate diagrams?	No

H.23.2 About the Interviewee

Where do you work?	(charity)
What is your job title?	Welfare Benefits Advisor
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (A)
What did you study at A-level?	English, Afrikaans, Maths, Biology, Art, Geography
If you went to university, what did you study there?	BA in Arts
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Market Research Benefits Related courses Massage Training Market Research helped with clarity of communication – has done both quantitative and qualitative research

H.23.3 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	Key decision makers?
		Relates to people – so some understanding
B2	0.3	The main things that are involved – main branches / offices / departments - this is how ArchiSurance is structured
		It's what the example implies – but the 'actor' concept is much more abstract
B3	0.3	The overseeing part of the organisation – the bird's eye view – quality control - governance
B4	0.3	Internal communications, not for the customer
B5	1	The steps involved in processing a claim or getting a new policy When answering B9, said B5 is 'the journey' of my claim etc.
B6	0.7	The workers / departments
B7	0.7	… their primary function (relates to B4)
		It's what they (B3) do
B8	0.3	Beginning and end point?
		It's not clear
B9	0.3	A descriptor of what B5 would be
		In my words – a grouping of processes?
B10	0	Something that may or may not happen
B11	1	It all hangs from this (the policy) …
		Rose's knowledge kicking in …
B12	0.3	Policy Summary – idiot's guide to the policy – is linked to the most important document – related to the contract Submission Form – Claim File Summary -
		Can see that B12s relate to something else
B13	0	A summary of what goes on – internally focused summary The communication with the staff as to the headlines as to what we're all about – doesn't make sense

IQ 11. Does the business model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	+1	
B2	+1	
B3	+1	
B4	-1	Too much detail – lose this – you know they're communicating
B5	+1	
B6	+1	
B7	-1	Duplicate of B3
B8	0	Still not really sure what these are
B9	+1	
B10	+1	Headlines a whole lot of stuff
B11	+1	
B12	+1	
B13	+1	

H.23.4 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	1	The main bits of application that are used
A2	0	Customer-focused – the summary of all the hard work that goes on in the background
		I did point out the similarity with the business collaboration/interaction at the start of the interview…
A3	0	Three primary stages in the claim process – one customer-generated, then administration done, then finally payment issued – document API is part of digesting all this information process.
A4	0.7	Part of the actuarial process – doing all the sums / calculations – computing all the data to get to an outcome
		This is pretty close to the right answer

A5	0	The end product that we get to – the end result – a summary of the rest of the diagram
		Well yes it is, but that's not the point
A6	0	Actually this is actuarial stuff – taking all the information that you've got – from all over the place – to work out risk
A7	0	(unable to suggest what these are)
A8	0	Money?
		Hasn't seen the relation to the applications…
A9	0.7	Taking the three key things ABOUT the customer, claim and policy
		Got the idea of information about…

Does the application model fulfil the purpose specified in the reference material (Y/N)?	Y

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	+1	Central to everything
A2	0	
A3	0	Good for lazy thinking!
A4	+1	Assuming you want to know what the apps are actually doing
A5	0	
A6	+1	
A7	+1	Essential because that's what the whole thing is about
A8	+1	Leave it in to make the diagram idiot-proof!
A9	0	Should have been captured everywhere else

H.24 Interviewee: S24

H.24.1 Metadata

Interviewer	Cameron
Date/Time	31st August 2019
Used ArchiMate diagrams?	Yes

H.24.2 About the Interviewee

Where do you work?	NHS

What is your job title?	Senior Sister
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (a)
What did you study at A-level?	Human Biology, Maths
If you went to university, what did you study there?	Diploma in Nursing
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	City & Guilds Part 1 Adult Education

H.24.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1	1	What needs to be satisfied – the two main stakeholders
M2	0.3	Drivers – the reason people do things (more theoretical)
M3	0	Tangible numbers by which you are judged to be achieving or not achieving
M4	0.7	Drivers – different sort of driver – much more practical – gives you direction
M5	0.3	Things that need to be looked at closely (magnifying glass)
		Hence magnifying glass icon
M6	1	Targets – that we need to aim for
M7	0.7	A very specific target as opposed ot just a general one – has a number attached to it
		Feasible but not quite what the spec says
M8	0.3	Something that company is wanting but is not required (a preferred option) – a barrier that the company has put up themselves
		There is something in this…
M9	0	Computer database emphasis

M10	0.3	An issue out of their control
Does the motivation model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	
M2	+1	
M3	+1	
M4	+1	
M5	+1	
M6	+1	
M7	+1	
M8	+1	
M9	+1	
M10	+1	

H.24.4 Business Layer

IQ 12. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	They are all managers
		Gets the fact that these relate to roles
B2	0	All to do with interacting with the customer, providing service to the customer
B3	1	Collaborative working – joining together
		Venn diagram – overlapping – signalling working together
B4	1	Relates to contact with customers – a means of communication
		Lollipop icon here interpreted as speech bubble
B5	1	Examples of a process – a chain of events, where one follows the other in a very methodical way
B6	0.3	Bread and butter type stuff – gets done in the back office – day to day jobs that are part of the bigger picture but not necessarily very specialised – hands on

		Got an aspect of behind the scenes – not exposed explicitly
B7	0	More practical aspect – data crunching person or type of job
B8	0.3	The start and end points of processes
		True but not really what they represent
B9	0	Admin type work
B10	0	One of the primary reasons for the company
B11	0	Something generated by the admin team which has been tailored to the customer's requirements.
B12	0.7	A summary (user-friendly version) of B11
B13	0.3	Primary driver
		What it's all about

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	
B2	+1	
B3	+1	
B4	+1	
B5	+1	
B6	+1	
B7	0	
B8	+1	
B9	+1	
B10	+1	
B11	+1	
B12	+1	
B13	+1	

H.25 Interviewee: S25

H.25.1 Metadata

Interviewer	Cameron
Date/Time	16th Feb 17:10
Used ArchiMate diagrams?	Yes

H.25.2 About the Interviewee

Where do you work?	HMRC
What is your job title?	Delivery Manager
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Economics, Biology
If you went to university, what did you study there?	No
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Prince Project Office Management Printing Systems (technical)

H.25.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1	0.3	People involved – two people that measure motivation – board and customers
		Text led to this conclusion
M2	0.7	High level objectives of these different groups – importance to those groups – all referred to as customer satisfaction
M3	0	Like a focus group – no real power – subordinate to the board – acknowledged but suffered
M4	0.7	Objective – things that the board are trying to achieve in high level terms – how it will be measured
		Had KPIs, objectives, strategy narratives in his own job

M5	0	Areas of failure, that will affect the objectives – if we get it wrong, this is what would happen
M6	0.7	Aspirational objective – performance indicators – push points – if you can do these things, you'll achieve
M7	0.3	Measurable targets – has a target number
		The symbol led to the idea of target here – e.g. reduce my personnel costs by a specific amount
M8	0.3	Levers for efficiency – if I can do these, I can bring in more efficiency
		i.e. what levers can we pull – what leverage have we got
M9	1	Restriction – things I have to comply with
		Example – had to comply with accessibility
M10	0.3	Again restrictive – a limited choice

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	Allows you to group the elements
M2	0	You'd have expected them to be there anyway – don't really add value
M3	+1	Shows an interaction with the board
M4	+1	Why are we doing this – in every company – what's driving us
M5	+1	Measure – an indicator that you'd have to do something
M6	+1	Business plan - actions
M7	+1	How we're going to measure the business - KPIs
M8	0	Could just be rolled up into a M6
M9	+1	They give you an indication of the constraints you're working in So you know you have to comply – a PM would have to know about these
M10	+1	Some interpretation about how you deal with suppliers – how you handle them

H.25.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation

B1	0.3	Symbol led you to believe that this is data
B2	0.3	Areas where people are based – human interaction
B3	0.7	Where they make some kind of decision – amalgamate information
B4	1	Input / contact channels – different forms of media
B5	1	Flow of information through a process
		Did work in Process Improvement
B6	0.3	Areas of work carried out by the company – denomination for work
B7	0.3	Decision making process
		Reviewing box in context of the text plus those around it
B8	0.3	Start and end of a journey – starting and ending a process
B9	0	Doesn't have any distinct significance or identity (compared to any other shape)
B10	0.3	Information coming **from** a customer (a kind of information)
		Knows this from the words
B11	0.3	Information being supplied **to** a customer (a kind of information)
		Horizontal lines plus text led to this conclusion
B12	0.3	Holding information
		Arrows (realisation) confuses, as a Claim File Summary points to a Claim (why isn't the arrow the other way round?) Clue – information because of the top line
B13		Overall – what the company does - purpose (note – interpreted as 0.3)0.

Does the business model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation

B1	0	Making an assumption that each bit of the business has a manager who would do this stuff
B2	+1	Based upon a correct understanding of the symbol – where there are actually people
B3	0	Not sure what it is – so wouldn't affect things
B4	+1	It shows stuff coming in – gives a flow – what we base a lot of our work on
B5	+1	Need it to show how we work – for example, if I was staffing this company, I'd want to know what we're wanting the staff to do
B6	+1	Tells me what the offices are made up of – processes / sections
B7	0	Doesn't really add anything – because the inside element is obvious given the title of the outside element
B8	+1	A launch point for the processes – what made me start that process
B9	+1	Sets out high level being of the company – what it does for a living
B10	0	Claim is like a duplicate of Claim received – arrows should just go straight to this
B11	0	Again a duplication – meaning of B11 could just be carried by B12 (so have B12 without B11)
B12	+1	Would have to make more of an assumption about what a claim is – what it means
B13	+1	Tells me instantly what the company is about

H.26 Interviewee: G1

H.26.1 Metadata

Interviewer	Cameron
Date/Time	31/10/2019
Used ArchiMate diagrams?	No

H.26.2 About the Interviewee

Where do you work?	Student
What is your job title?	
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c)
What did you study at A-level?	Maths, Physics, Chemistry
If you went to university, what did you study there?	Bachelors in Computer Engineering
What training have you had, since you started work, in any formal or	

structured methods (e.g. technical, project management)	

H.26.3 Motivation Layer

Note that in common with many of the group interviewees, this interviewee did not actually rate their comprehension themselves (contrary to my instructions); they left it blank. I therefore had to rate their answers without knowing what they felt about their own understanding. The scores in the summary spreadsheet have these scores in; they are not in the original completed questionnaires.

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1		It's stakeholders- They are main part of organisation to take care of.
M2		Services- Stakeholders expect from the organisation.
M3		Board- take care of stakeholders' needs
M4		First preference of stakeholders.
M5		Failing in the completing needs of stakeholders.
M6		Improvement of each factor in organisation.
M7		More security in work.
M8		Work done by system itself.
M9		More success in work.
M10		Problems during completing tasks.
Does the motivation model fulfil the purpose specified in the reference material (Y/N)?		

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation

M1	+1	This are the main components of diagram.
M2	+1	Motivation of stakeholders are important.
M3	0	Board for feedback from main stakeholders.
M4	+1	Perspectives of stakeholders can't be neglected.
M5	-1	Negative tasks, does not fulfil purpose.
M6	0	Does not change purpose.
M7	0	
M8	-1	
M9	-1	
M10	-1	

H.26.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1		Main bodies of organisation.
B2		Head departments of organisation.
B3		Boards working in organisation.
B4		Communication methods used in organisation.
B5		Main processes happening in organisation.
B6		Teams or sub departments in main offices.
B7		Finance review task.
B8		Input or output of the processes.
B9		Administrative departments of organisation.
B10		
B11		Legal body of Organisation.

B12		Database about claims and management.
B13		Security given to the customers.

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	
B2	0	
B3	0	
B4	+1	
B5	+1	
B6	+1	
B7	0	
B8	0	
B9	+1	
B10	-1	
B11	+1	
B12	-1	
B13	+1	

H.26.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
A1		Main systems of Organisation.
A2		Finance transaction process.
A3		Webpage or user interface.
A4		Data management and process.

A5		Details about stored data or offers given by organisation.
A6		Specific data processing depending on customer.
A7		Start of a process or inputs of process.
A8		Request for process.
A9		Database of organisation.

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	0	
A2	-1	
A3	+1	
A4	+1	
A5	0	
A6	0	
A7	+1	
A8	0	
A9	+1	

H.27 Interviewee: G2

H.27.1 Metadata

Interviewer	Cameron
Date/Time	31/10/19
Used ArchiMate diagrams?	NO

H.27.2 About the Interviewee

Where do you work?	Student / Petroleum company
What is your job title?	Data Manager

Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) (the highlighted (a) above is taken as the answer)
What did you study at A-level?	Informatics,Marketing, Management,
If you went to university, what did you study there?	Information Systems Management
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Project Management

H.27.3 Motivation Layer

IQ 13. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1		Stakeholders- The persons interested/involved in the process
		Stakeholders analysis in informatics class
M2		Objectives-the specific goals of each group
		Strategy class at school
M3		Specific indicator to measure – department responsible of measuring the indicator
		Not sure
M4		Indicators- what is going to be measured
		Marketing class
M5		Failure reasons – Possible reasons why indicators fails
		Marketing class
M6		Benefits – What are the advantages of measuring indicators
		Informatics class

M7		Results – Consequences of managing indicators
		Undergraduate class
M8		Monitor activities – how is going to be monitored the indicators
		Business class
M9		Relation with other systems
		Informatics
M10		Reasons why an specific objective can me met
		Not sure

IQ 14. Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	Actors should be placed on the model , learned on IT courses
M2	+1	Important to set target objectives, marketing classes
M3	+1	Good to know what will be measured
M4	+1	Good to know how its measured
M5	-1	Its better so separate failure reasons
M6	+1	Marketing classes
M7	+1	
M8	+1	
M9	+1	
M10	-1	Justification reasons are not good in a model- To be diagram or asis do not included this

H.27.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1		Actors – who is performing the action
		Informatics

		Note: actor isn't the right term, but the underlying meaning is kind of correct?
B2		Entities/Sectors- Groups of departments
		Well yes but this is down to the examples
B3		External board – group which verify external decisions
B4		Channel- way of sending the information
B5		Procedure- steps to follow to solve a specific task
B6		Departments – how business are organises
B7		Main activity – purpose of the external entity
B8		Output – what is the final result of the processes or at least expected
B9		External Departments – groups that also have influence or power in the process
B10		Object – the resource by which they will control something
B11		Document – document that affects the procedure
B12		Document delivered – final document where clauses are stated
B13		External department – external group that have control or power in the process

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	+1	
B2	+1	
B3	-1	

B4	+1	
B5	+1	
B6	+1	
B7	-1	
B8	0	
B9	+1	
B10	+1	
B11	0	
B12	0	
B13	+1	

H.27.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1		System – which group of will run the process
A2		Module- part of a system
A3		Front layer – final interface for the user
A4		Module – part of a system
A5		Department – group of people responsible to do a specific task
A6		Processes - list of possible actions or steps
A7		Input – required information to perform the process
A8		Actions- possible options the final user could perform
A9		Output- final results from the process

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	+1	
A2	-1	
A3	+1	
A4	0	
A5	-1	
A6	+1	
A7	+1	
A8	+1	
A9	+1	

H.28 Interviewee: G3

H.28.1 Metadata

Interviewer	Cameron
Date/Time	31.10.2019
Used ArchiMate diagrams?	NO

H.28.2 About the Interviewee

Where do you work?	
What is your job title?	Supply chain analyst
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) A
What did you study at A-level?	Czech Language, English, German, Geography.
If you went to university, what did you study there?	BA (Hons) Business Management and Leadership
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	SAP, IBM Mainframe, ILM L3 coaching in Workplace.

H.28.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation

M1	1	Stakeholders – parties with a stake in the business/process
		Previous experience/business education
M2	1	Objectives – desires/needs of the stakeholders
M3	0.3	Second tier Objectives – this is what we are aiming to achieve by reaching M2 – the primary objectives.
M4	0.3	KPI – key performance indicators – used in judging whether objectives are met
		Previous experience/business education
M5	0.7	Results of failed KPIs – this is what is likely to happen when M4 ends up differently that intended/anticipated
M6	0.3	Instruments that help working towards KPIs
M7	0	Result of failed in M7
M8	0.7	Area for development – this needs to be implemented in order to satisfy the objective
M9	0.3	Do not see a clear meaning behind this, could be ways to fulfil the objectives, not sure
M10	0.3	External factor – cannot be influenced from within the company
		Previous experience/business education
Does the motivation model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	Important to identify the stakeholders
M2	0	Losing some meeting but a business person would identify without it being specifically said
M3	-1	Not so necessary
M4	+1	Needed, KPIs that need to be monitored
M5	0	Not so necessary, you can think that

M6	+1	Important so we know some kind of an action plan
M7	0	Useful in identifying the action plan
M8	-1	Don't understand
M9	0	
M10	+1	Sort of needed, importance to understand the external environment

H.28.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1		Stakeholders – self-explanatory
B2		Business functions/locations – places from which operations are carried out
B3		Infrequent review process
B4		Means of communication
B5		Individual tasks – that are to be carried out by the relevant department when an event occurs
B6		Departments/ teams – within a location/business function
B7		X infrequent task
B8		Outcome – what happens after an event occurs and goes through all the parts in the chain
B9		Front-facing functions or processes/teams
B10		A formalised start of the process
B11		A notification- could be like a final agreement/T&C document etc.
B12		Process initiation – a trigger that starts the whole process

B13		Front-facing department
Does the business model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	-1	Obvious that a management team will have a manager, box no needed
B2	+1	Important to see in which function individual bits sit
B3	0	Not sure, not really related to the primary operations
B4	0	Could be omitted, again obvious that thing will be communicated
B5	0	Relevant to an extent, in high level not needed, if we are delving deeper it's important to understand the underlining processes within the operations
B6	+1	Important to see the relevant teams
B7	-1	Not sure
B8	+1	Important to know the outcome
B9	+1	Seems important high level
B10	+1	
B11	+1	Outcome of the process – importance to see the end process
B12	+1	Important – process initiation
B13	+1	Somehow seems as important

H.28.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1		Applications/data warehouses
A2		Financial/computing systems
A3		Visuals
A4		Data warehouses
A5		Not sure how to call this

A6		The Intend
A7		Process initiation
A8		Step/task/process – that is taken to move forward
A9		Data/information produced/obtained/stored

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	+1	Need to know what systems data go through
A2	-1	Does not influence my understanding
A3	0	Somehow helpful
A4	0	Somehow helpful
A5	+1	
A6	+1	
A7	+1	Process initiation, importance to know how/where the process starts
A8	1+	Importance to know the individual tasks to be taken
A9	+1	We just need data/details/information

H.29 Interviewee: G4

H.29.1 Metadata

Interviewer	Cameron
Date/Time	31/10/2019
Used ArchiMate diagrams?	No

H.29.2 About the Interviewee

Where do you work?	(IT consulting company)
What is your job title?	Junior Management Consultant

Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	(a)
What did you study at A-level?	International Business and Economics, French, International Business Studies
If you went to university, what did you study there?	Principles of Management, Project Management, Business Intelligence and IT-Management
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Agile Professional Scrum Product Owner 1, Agile Professional Scrum Master 1, ITIL Foundation certificate in IT Service Management, Cisco Certified Entry Networking Technician - CCENT

H.29.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1		Stakeholder, group of people, who we need to keep happy.
M2		Things valuable to the different Stakeholders / M1
M3		Customer Satisfaction Report for Stakeholders / M1
M4		Goals, what should the company focus on?
M5		Result when Goals / M4 isn't/wasn't met
M6		Aims of the company? Split up the goals into smaller aims?
M7		
M8		Rules, Rules and internal regulations, how and what to use during a specific process
M9		What needs to be done? Implement a new connection to a IT system etc. so requirements to be able to fulfil the goals and aims
M10		Restriction, Problem What can increase the complicity to achieve the goals/aims?

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	Stakeholder groups are one of the main factors
M2	+1	Graphic should reflect the motivation of the different stakeholders
M3	-1	Not required to display the motivation of the stakeholder groups
M4	+1	Should be there, because it shows the high-level points how the business response or cluster the motivation of the stakeholders
M5	0	Not required to show the response of the business
M6	0	Can be there but the high-level goals/points can be enough too → only nice to have
M7	0	Also nice to have, but too detailed already
M8	-1	Not required, it is an implementation/internal stuff where the business need to care about but nothing to do with the motivation and the response
M9	-1	Not required, it is an implementation stuff but not required to show how the business should respond to the motivation of the stakeholders
M10	-1	Not required, it is a problem/show stopper but not required to show how the business should respond to the motivation of the stakeholders

H.29.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1		Role, Actors. People/roles who are contributing or involved in the business organisation
B2		Business functions, like departments of an organization, but not support functions. Functions who are responsible for the core business → earn money at the end of the day
B3		Joint body, like a board of directors
B4		Ways of contacting the department/organisations, e.g. via phone
B5		Tasks triggered by an event, what is the company actually doing?
B6		Types of tasks, what the department needs to do
B7		Task of the B3

B8		Event/Trigger of process
B9		Support tasks/
B10		1 Main Task
B11		Second main task
B12		
B13		Location of the departments

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	The roles/actors can't be removed, because they are connected to the tasks and departments
B2	+1	Departments are needed to clarify the business structure and operations
B3	-1	Different steering teams/meetings are too detailed
B4	-1	Not relevant for the business structure
B5	+1	Defines the service and the connected processes
B6	+1	Required, because the department are necessary to show the connection between tasks and the business organisation
B7	-1	Same as B3
B8	-1	Trigger who is starting the process is not required to display the process
B9	0	
B10	0	
B11	+1	
B12	+1	
B13	0	Nice to have

H.29.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
A1		Information System
A2		Modules
A3		Interface
A4		Grouped task
A5		
A6		Tasks
A7		Trigger event
A8		Process steps
A9		Details

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	+1	Different applications should be there for an application layer infographic
A2	0	
A3	+1	Application output, necessary to have
A4	0	
A5	0	
A6	+1	Process steps under or in between applications should be there
A7	+1	Trigger should also be included otherwise no knows when the application come to action
A8	0	
A9	+1	Displays details and that is required

H.30 Interviewee: G5

H.30.1 Metadata

Interviewer	Cameron
Date/Time	Oct 31st
Used ArchiMate diagrams?	Yes

H.30.2 About the Interviewee

Where do you work?	Student
What is your job title?	
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c)
What did you study at A-level?	Finance
If you went to university, what did you study there?	Finance
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	

H.30.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1		It's a board of directors -which does this committee having supervisory powers
M2		It's a board of directors -which does this committee having supervisory powers
M3		
M4		
M5		
M6		

M7		t's a target-which does this they need get some
M8		
M9		
M10		

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	-1	
M2	1	I'm really consumed with this one.
M3	0	
M4	1	
M5	1	
M6	-1	
M7	0	
M8	-1	
M9	0	
M10	-1	

H.30.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1		It's a function of bank-which does this bank have different function or professional work
		They are types of people and they have different role
B2		
B3		

B4		This symbol is represents a form of communication and by this way they can communicate more effectively
B5		This symbol is represents a form of communication and by this way they can communicate more effectively
B6		
B7		
B8		
B9		
B10		It's an assertion of a right As money or property.
B11		
B12		
B13		

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	-1	
B2	0	
B3	1	
B4	-1	
B5	0	
B6	0	
B7	-1	

B8	-1	
B9	1	
B10	0	
B11	-1	
B12	1	
B13	-1	

H.30.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1		It’s a different system, application and solution
A2		
A3		It’s a different system, application and solution
A4		
A5		
A6		
A7		
A8		
A9		

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1		
A2		

A3		
A4		
A5		
A6		
A7		
A8		
A9		

H.31 Interviewee: G6

H.31.1 Metadata

Interviewer	Cameron
Date/Time	31/10/2019 14:23
Used ArchiMate diagrams?	Yes

H.31.2 About the Interviewee

Where do you work?	Banking industry
What is your job title?	Credit policy analyst
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	(a)
What did you study at A-level?	Math-English
If you went to university, what did you study there?	Finance
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	Credit analysis, banking products, IB

H.31.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1	1	People who concern about the objectives of company can call stakeholders. Because they are only people in this diagram and they are the important people in the company.

M2	0.7	The main topics of company are be interested to connect with the stakeholders. These are overview of objectives. I learned accounting and finance before so it is clear that there are the words of topic.
M3	0.7	The two willing of customers show of the performance of company. They will look at the company that is good enough to invest so they are influenced to the company.
M4	0.3	The components of sub-main points are focused specifically.
M5	0.3	The method of targets.
M6	0.7	The detail of targets are found in many topics which are expanded from the top view points. These are about the performance of company.
M7	0.3	The specific detail of targets are belong to the customers and some of them have specific number.
M8	0	Maybe, that is the barriers of the systems' company.
M9	0.7	The solution of issue is connected with the other systems.
M10	0.3	The largest problem of company is still not find the solution.

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	N

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	They are the main characters so it has to have identify.
M2	+1	All of this for the business perspective, it is necessary to look overview in one diagram so every boxes must have in the diagram. If one of boxes is deleted, problems about the understanding of stakeholders will appear. Because they don't know the exactly Aimee's [?] of motivation.
M3	+1	
M4	+1	
M5	+1	

M6	+1	
M7	+1	
M8	+1	
M9	+1	
M10	+1	

H.31.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.7	The related people are focused in the business part.
B2	0.7	The overview sectors are connected which cover the relative departments. It separates the parts who have connections with customer, without customers and other companies.
B3	0	It might be the intermediate people who view the information before send to external organizations.
B4	0.3	The technology devices are used to support the company.
B5	0.7	The step of processes in internal company is explained in top view.
B6	0.7	The departments in company are set in the group which have to connect together.
B7	0.3	The intermediate person works with internal and external organization.
B8	0	After and before a claim case.
B9	0.7	The regulations or standards are released from government which the company must concern.
B10	0	The whole topic of claim insurance .
B11	0	The overall topic of policy.

B12	0	Sum up of the each part.
B13	0	The main part of insurance company have to look.

Does the business model fulfil the purpose specified in the reference material (Y/N)?	N

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
B1	+1	There are many connections of the departments and inter-organizations. It's hard to reduce some of it because they will have to know and process something.
B2	+1	
B3	+1	
B4	+1	
B5	+1	
B6	+1	
B7	+1	
B8	+1	
B9	+1	
B10	+1	
B11	+1	
B12	+1	
B13	+1	

H.31.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	0.7	The main systems of company are connected, used and filled.
A2	0	The platform of information exchanges between internal and external company.
A3	0	The interface of the systems are showed in type technology devices.
A4	0	The topics are feeded into the systems to record the data.

A5	0	The intermediate agency responds the transaction.
A6	0.3	The steps of processes in the short words.
A7	0	The data from somewhere is accessed to the system for getting the processes.
A8	0	The action of the systems.
A9	0	The results of the systems

Does the application model fulfil the purpose specified in the reference material (Y/N)?	N

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	+1	
A2	+1	
A3	+1	
A4	+1	
A5	+1	
A6	+1	
A7	+1	
A8	+1	
A9	+1	

H.32 Interviewee: G7

H.32.1 Metadata

Interviewer	Cameron
Date/Time	31/10/2019
Used ArchiMate diagrams?	Y

H.32.2 About the Interviewee

Where do you work?	China
What is your job title?	Product manager
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c) a
What did you study at A-level?	Chinese, Math, English, Physics
If you went to university, what did you study there?	Accounting & Finance
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	None

H.32.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
M1	0.3	I think the diagram looks like a core bar, since these stakeholders is the core of the whole motivation.
		Note: think interviewee should have scored higher, but went with their self-assessed score
M2	0	I don't really know what its meaning,
M3	0	It might mean these things should be research if that would happen.
M4	0.7	Purpose- it
		Note: unable to challenge their 0.7 (self-scoring) and insufficient text given to justify 0.7, so gave 0.3
M5		
M6		
M7		
M8		
M9		

M10		
Does the motivation model fulfil the purpose specified in the reference material (Y/N)?		
For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	These two is the centre of whole process, it must be shows
M2	0	It shows things that the stakeholder concerned about, but it doesn't matter if it would be removed.
M3	-1	
M4	+1	It is important since it shows the main purpose that the stakeholder want to achieve.
M5	-1	
M6	+1	
M7	+1	
M8	+1	
M9	+1	
M10	0	

H.32.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	Same, the stakeholder is the core of the whole system.
B2	0.3	The symbol looks like a people, but I think it actually meaning a team or group.
B3	0.7	
		Note: unable to challenge 0.7 as no explanation given, so scoring 0.3
B4	0.3	
B5	0.3	
B6	0.3	

B7	0.3	
B8	0	
B9	0	
B10	0	
B11	0	
B12	0	
B13	0	

Does the business model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	+1	
B2	+1	
B3	+1	
B4	+1	
B5	+1	
B6	+1	
B7	+1	
B8	+1	
B9	+1	
B10	+1	
B11	+1	
B12	0	
B13	0	

H.32.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	0.7 (changed to 0.3)	*All of these have been scored down to 0.3 as no explanation was given*
A2	0.7	
A3	0.7	
A4	0.7	
A5	0.7	
A6	0.7	
A7	0.7	
A8	0.7	
A9	0.7	

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	+1	
A2	+1	
A3	+1	
A4	+1	
A5	+1	
A6	+1	
A7	+1	
A8	+1	

A9	+1	

H.33 Interviewee: G8

H.33.1 Metadata

Interviewer	Cameron
Date/Time	31.10.2019
Used ArchiMate diagrams?	Yes

H.33.2 About the Interviewee

Where do you work?	Telecommunications manufacturer
What is your job title?	Associate
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	Please answer (a), (b) or (c). C
What did you study at A-level?	Math, English, Chinese
If you went to university, what did you study there?	Information system and Information management
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	HUAWEI big data Certification

H.33.3 Motivation Layer

IQ 15. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1		Customer : Somebody who have the desire to buy something
		Board :
M2		Shareholders Return: share benefits
		Revenues: Total profit for everyone
M3		Customer Satisfaction:Customer
		Customer Satisfaction:Board
M4		Customer Satisfaction:
		Price:
M5		Customer satisfaction survey poor
		Share price falling
M6		All customer data hosted in the UK

		Reduction of costs
M7		No security breaches
		Reduction in complaints to 0.1%
M8		Common use of application:
		Single system of record for each data element:
M9		Compliance with PCI DSS
		Integration with existing CRM
M10		Lack of Suppliers in the UK:

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
M1	+1	
M2	0	
M3	-1	
M4	+1	
M5	-1	
M6	+1	
M7	0	
M8	+1	
M9	+1	
M10	-1	

H.33.4 Business Layer

IQ 16. For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
B1	0.3	
B2	0.3	
B3	0.7	

B4	0.3	
B5	0.7	
B6	0.3	
B7	0.7	
B8	0.7	
B9	0.3	
B10	0.3	
B11	0.3	
B12	0.3	
B13	0.7	

Does the business model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
B1	-1	
B2	1	
B3	1	
B4	1	
B5	-1	
B6	-1	
B7	1	
B8	1	
B9	-1	

B10	1	
B11	-1	
B12	-1	
B13	1	

H.33.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1	0.3	
A2	0.3	
A3	0.7	
A4	0.7	
A5	0.3	
A6	0.7	
A7	0.3	
A8	0.7	
A9	0.3	

Does the application model fulfil the purpose specified in the reference material (Y/N)?	Yes

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.		
Code	Rating	Explanation
A1	-1	
A2	1	
A3	1	

A4	-1	
A5	1	
A6	-1	
A7	0	
A8	0	
A9	-1	

H.34 Interviewee: G9

H.34.1 Metadata

Interviewer	Cameron
Date/Time	31/10/2019
Used ArchiMate diagrams?	No

H.34.2 About the Interviewee

Where do you work?	Student
What is your job title?	
Would you say that you are someone who (a) just wants results from such architects, (b) understands in detail the work of such architects, or (c) uses the work of such architects to guide your own?	(a)
What did you study at A-level?	Finance and Management, Maths, Computer Science.
If you went to university, what did you study there?	BSc. Finance and Management Studies MSc. Finance
What training have you had, since you started work, in any formal or structured methods (e.g. technical, project management)	N/A

H.34.3 Motivation Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
M1		Board-A group of people who manage a company
		Customer-who paid money for the products and services
M2		The things that board cares-their interests
		The things that customer cares-their interests

M3		Customer satisfaction-the degree about how customers satisfy about the products or services
M4		Specific things that the board want to see
		Customers' focus
M5		The details of M4
M6		Things that need to be done to meet the requirements for M4
M7		
M8		
M9		Final results
M10		The reason that caused M4.

Does the motivation model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
M1	+1	
M2	+1	
M3	0	
M4	+1	
M5	-1	
M6	0	
M7	0	
M8	0	
M9	0	
M10	-1	

H.34.4 Business Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.

Code	Rating	Explanation
B1		Managers
		Customer
B2		organisations
B3		Illustration
B4		Different methods
B5		The chain about how to handle claim
		The chain about how to issue new policy
B6		Different apartments for the insurance company
B7		Actions done by investment review board
B8		The result of claim rejected or received
B9		Claim administration
		Policy administration
B10		Claim-the amount paid to policy holder
B11		Policy-the agreement between insurance company and the policyholder who want the insurance to protect sth.
B12		Insurance- the way to protect sth
B13		Submission form-to submit the claim

H.34.5 Application Layer

For each concept listed below, rate your understanding of the concept, and explain it briefly. Explain if you can how your background or role helped or hindered your understanding.		
Code	Rating	Explanation
A1		Different systems
A2		Quotation and payment modules

A3		Different screens
A4		Administration management
A5		Travel insurance-to protect the loss if accidents happen during travel
A6		The process to calculate and process the data of insurance
A7		Request for the inputs
A8		Actions for A3
A9		Details for A4

Does the application model fulfil the purpose specified in the reference material (Y/N)?	

For each concept listed below, if it was removed, would the model better fulfil the stated purpose (score -1), would it not change (score 0) or would it be worse for the loss (score +1). Explain if you can how your background or role led to your rating.

Code	Rating	Explanation
A1	0	
A2	0	
A3	0	
A4	0	
A5	0	
A6	0	
A7	0	
A8	0	
A9	0	

Appendix I Constructing the Test Models

This appendix describes how the test models used for the modelling language quality research were constructed, by starting from existing ArchiSurance material and adding extra elements where necessary to cover the whole of the language in each specific layer within the scope of the research. All of the diagrams in this particular appendix have been taken from the ArchiSurance [62] material.

I.1 Starting Point

The suggested contents of the standard ArchiMate viewpoints are shown in Table 60 above. In sections I.2.1, I.3.1 and I.3.1 below we see which contents (in terms of selection of elements) are used in the ArchiSurance case study [62]. The aim here was to create test models that between them cover the entire range of elements in the ArchiMate language, basing the models as closely as possible upon the ArchiSurance examples. Many of the ArchiSurance models align to specific layers in the ArchiMate metamodel, which provide a convenient way of starting to determine the scope of our test models. For each layer in scope, we used the existing material and then add example(s) of any language elements that were not present in the existing models. We then combined, for each layer, all of the content into a single view.

I.2 Motivation Layer

I.2.1 Current ArchiSurance Motivation Content

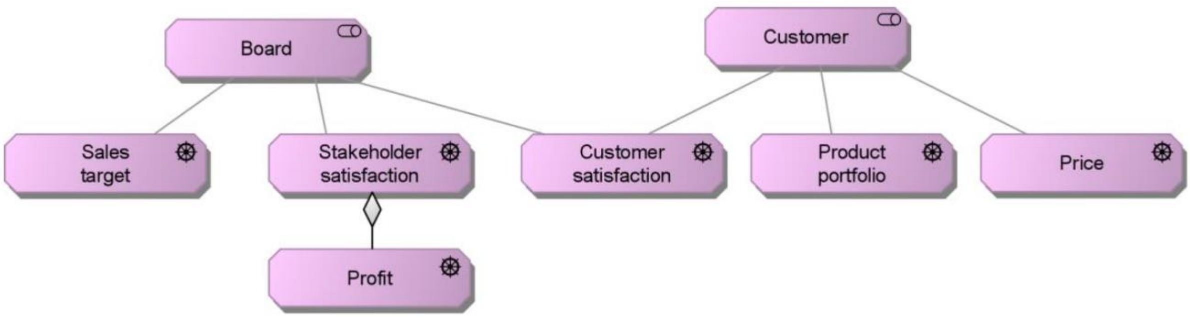
The current ArchiSurance material in this area is as follows:

Table 62 ArchiSurance Motivation Material

Layer	Symbol	Stakeholder	Stakeholder View	Drivers and Goals	Principles	Business Goals and Principles	Goal Realization	Goal Refinement	Introductory View	Requirements Realization
Motivation	Stakeholder	✓	✓							
	Value									
	Meaning									
	Driver	✓	✓	✓						
	Assessment	✓								
	Goal	✓		✓	✓	✓	✓	✓		
	Outcome	✓					✓			
	Principle				✓	✓	✓	✓		
	Requirement		Phase A			1/A	✓	✓		✓
	Constraint						✓	1/A		

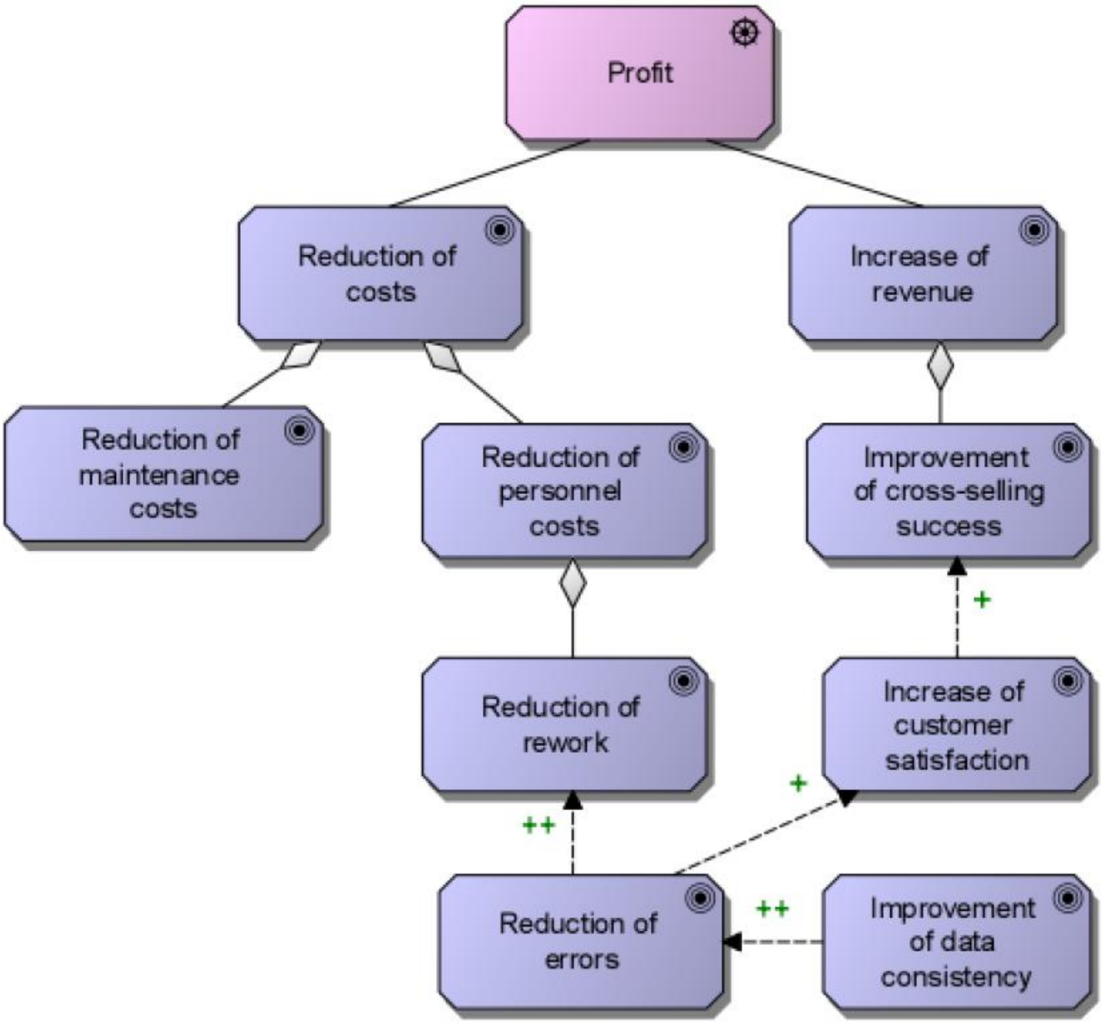
The original ArchiSurance diagrams corresponding to the ‘ticks’ in the above grid are:

Figure 153 Fragment of a Stakeholder View



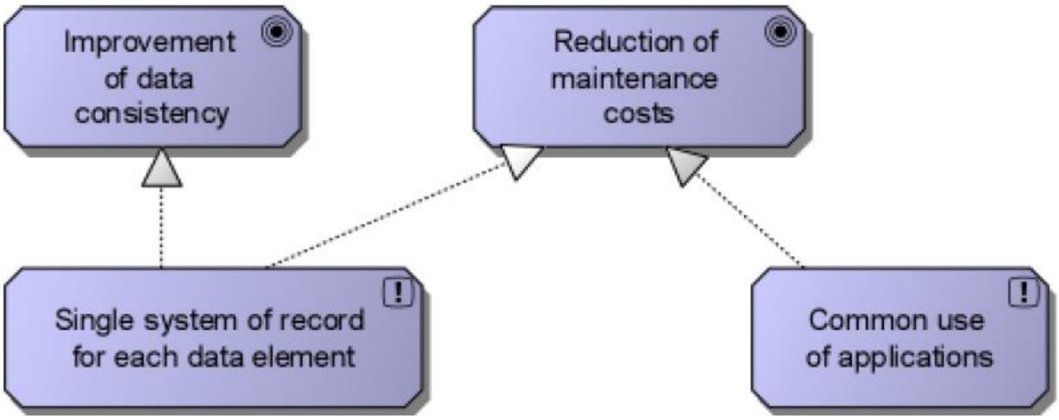
This contains **stakeholders** and **drivers**.

Figure 154 Business Goals Associated with the Driver Profit



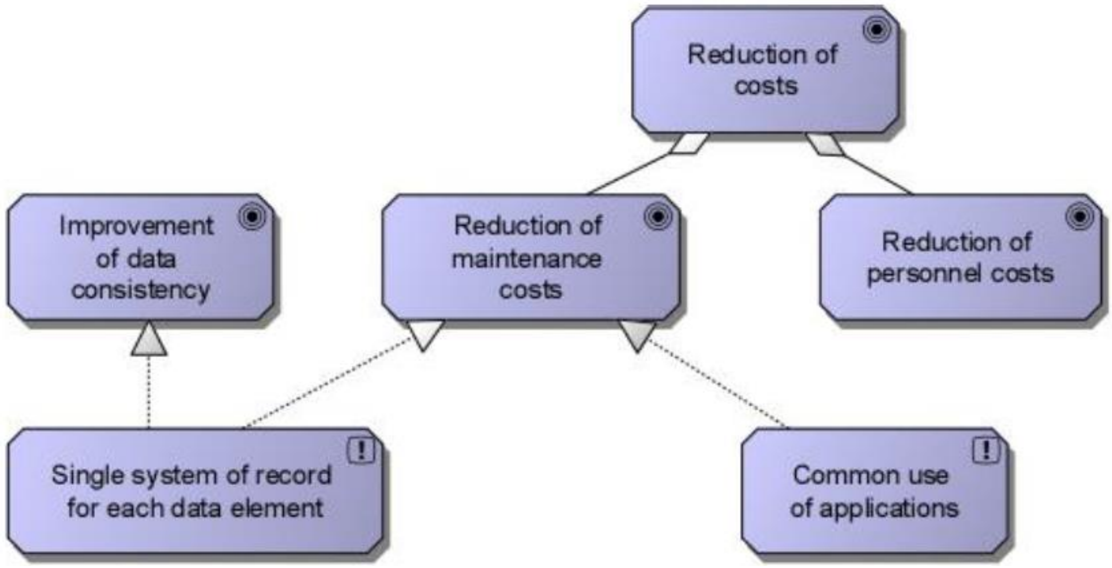
This contains **drivers** and **goals**.

Figure 155 Principles View



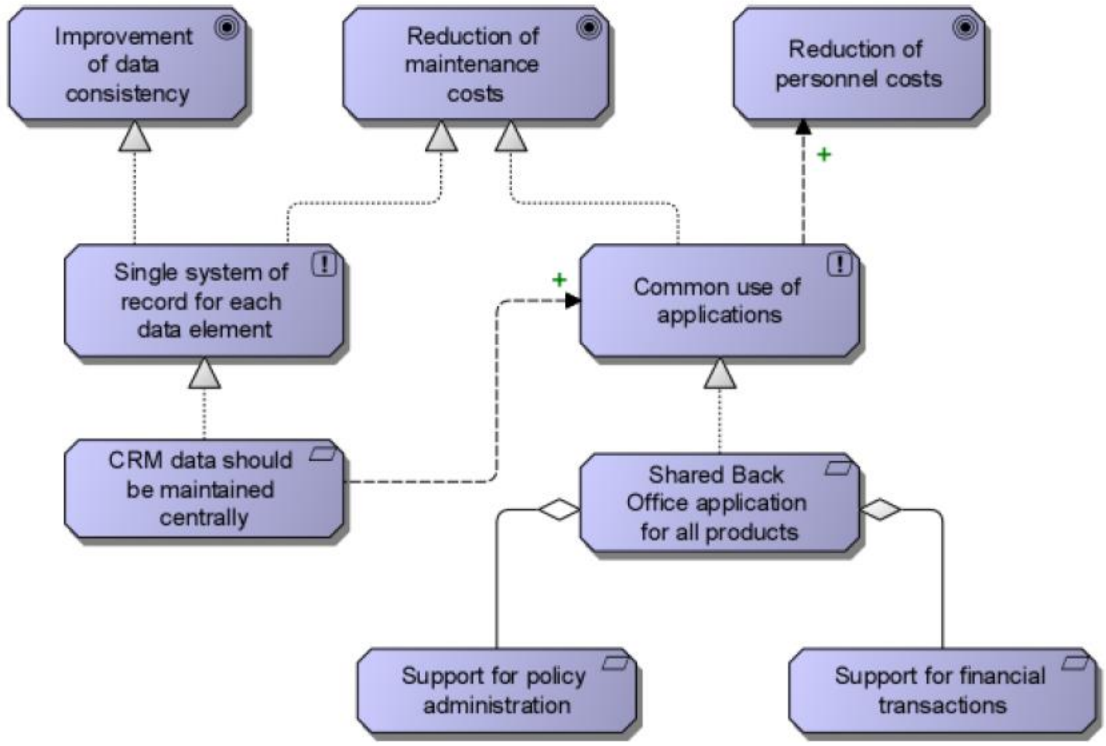
This contains **goals** and **principles**.

Figure 156 Business Goals and Principles



This contains **goals** and **principles**.

Figure 157 Goal Refinement View



This contains goals, principles, and requirements.

The above examples do not contain instances of the following elements from the Motivation layer:

Table 63 Motivation Elements Unused in ArchiSurance

Element	Definition
Value	Value represents the relative worth, utility, or importance of a core element or an outcome
Meaning	The knowledge or expertisepresent in, or the interpretationgiven to, a core element in aparticular context.
Assessment	The result of an analysis of thestate of affairs of the enterprisewith respect to some driver.
Outcome	An end result that has beenachieved.
Constraint	A factor that prevents or obstructsthe realization of goals.

I.2.2 Completing the Motivation Layer

The Motivation layer requires the addition of examples of the elements listed in Table 63 above.

I.3 Business Layer

I.3.1 Current ArchiSurance Business Content

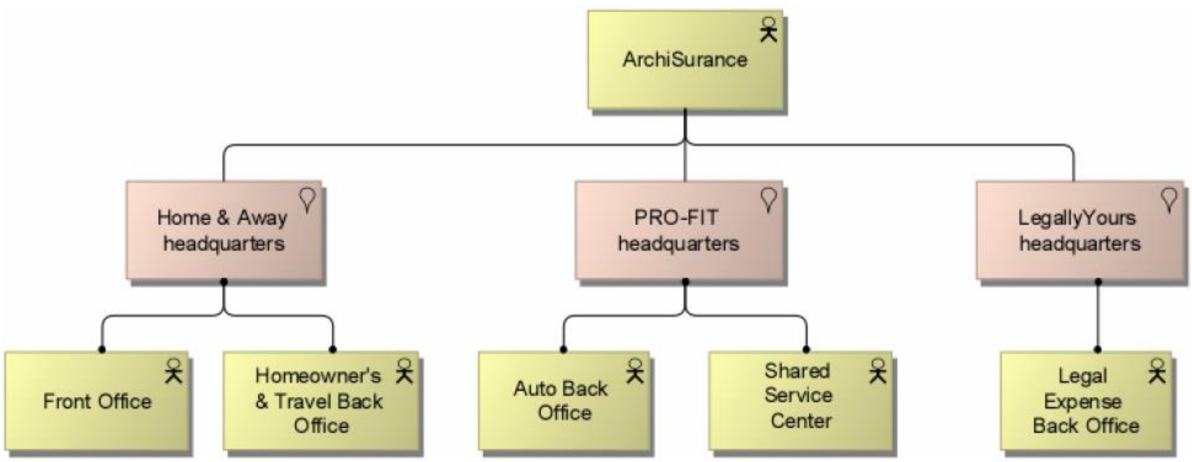
The current ArchiSurance material in this area is as follows:

Table 64 ArchiSurance Business Material

Layer	Symbol	Organisation	Business Function	Process
Business	Business role		✓	
	Business actor	✓	✓	
	Business collaboration			
	Business interface			
	Business process			✓
	Business function		✓	
	Business interaction			
	Business event			✓
	Business service		Phase B	
	Business object			
	Contract			
	Representation			
	Product			

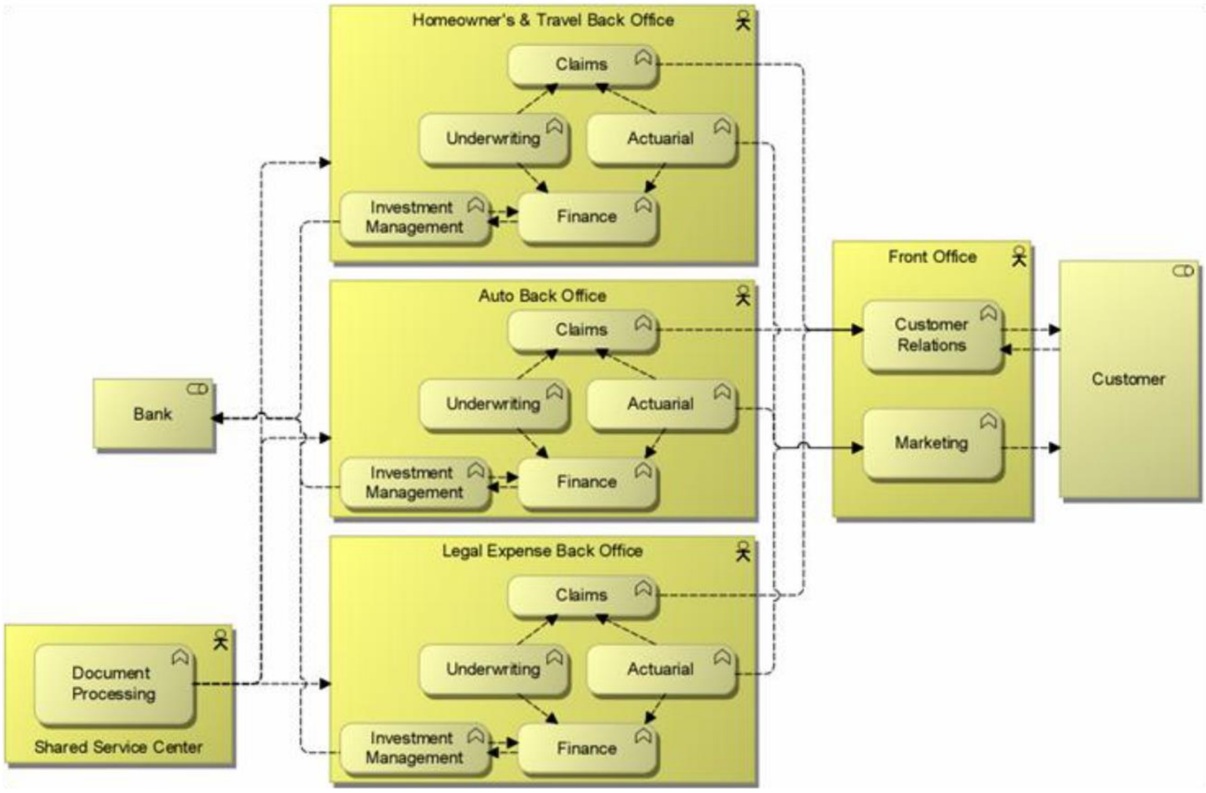
The original ArchiSurance diagrams corresponding to the 'ticks' in the above grid are:

Figure 158 Organisation View



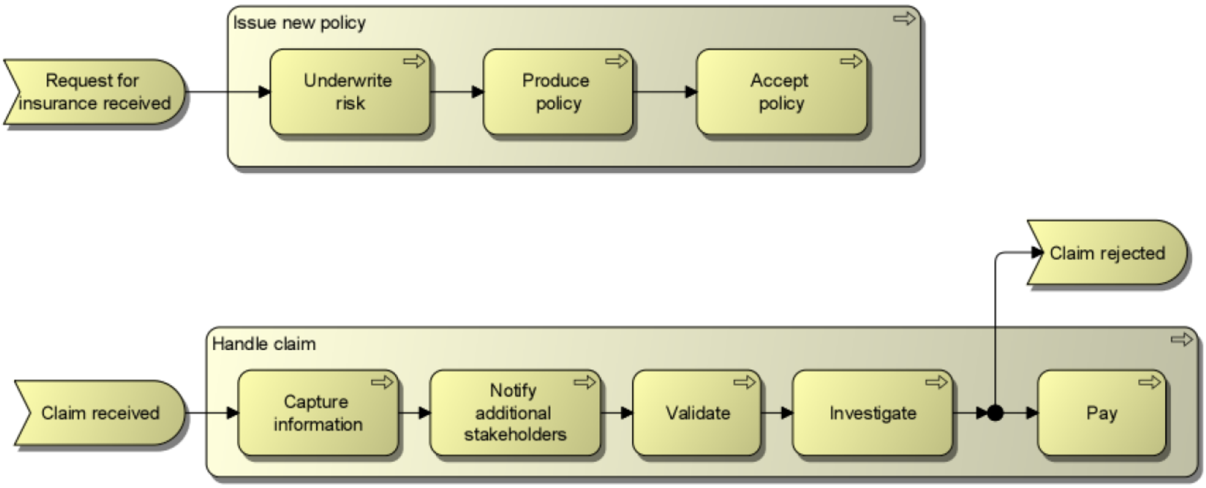
This contains **actors** and **locations** (the latter coming from the Composite layer rather than the Business layer).

Figure 159 Business Function View



This contains actors, business functions and roles.

Figure 160 Business Process View



This contains business events and processes.

The above examples do not contain instances of the following elements from the Business layer:

Table 65 Business Elements Unused in ArchiSurance

Element	Definition
Business Collaboration	A business collaboration is an aggregate of two or more business internal active structure elements[29] that work together to perform collective behavior.
Business interface	A business interface is a point of access where a business service is made available to the environment.
Business interaction	A business interaction is a unit of collective business behavior performed by (a collaboration of) two or more business roles.
Business service	A business service represents an explicitly defined exposed business behavior.
Business object	A business object represents a concept used within a particular business domain.
Contract	A contract represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.
Representation	A representation represents a perceptible form of the information carried by a business object.
Product	A product represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.

[29] This is therefore restricted to business actors and business roles (and other collaborations).

I.3.1 Completing the Business Layer

The Business layer requires the addition of examples of the elements listed in Table 65 above.

I.4 Application Layer

I.4.1 Current ArchiSurance Application Content

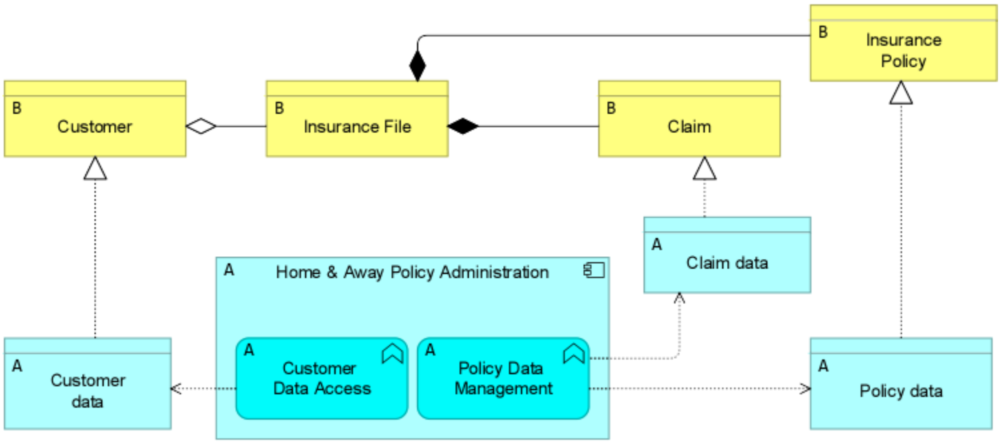
The current ArchiSurance material in this area is as follows:

Table 66 ArchiSurance Application Material

Layer	Symbol	Data Dissemination	Application Usage	Application Co-operation	Application Cooperation	Application Architecture: Gap Analysis	Introductory View	Requirements Realization
Application	Application component	✓	✓	✓	✓	✓	✓	✓
	Application collaboration				1/C			
	Application interface							
	Application function	✓						
	Application interaction							
	Application process							
	Application event							
	Application service		✓					✓
	Data object	✓	Phase C					

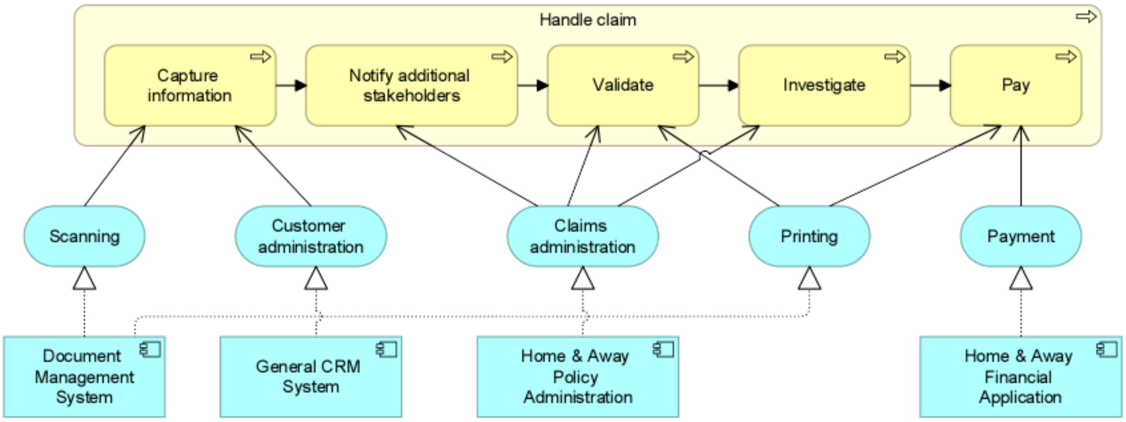
The original ArchiSurance diagrams corresponding to the 'ticks' in the above grid are:

Figure 161 Data Dissemination View



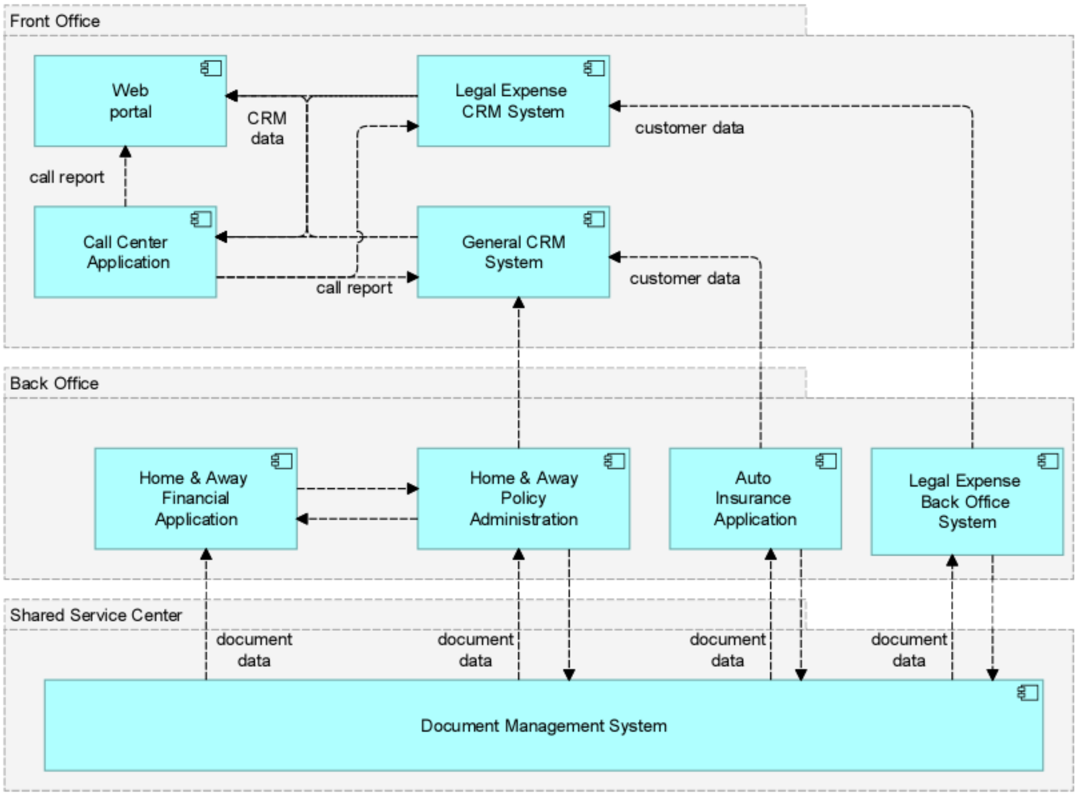
In terms of application layer entities, this contains **application components**, **data objects** and **application functions**.

Figure 162 Application Usage View



This contains **application services** and **application components** from the application layer.

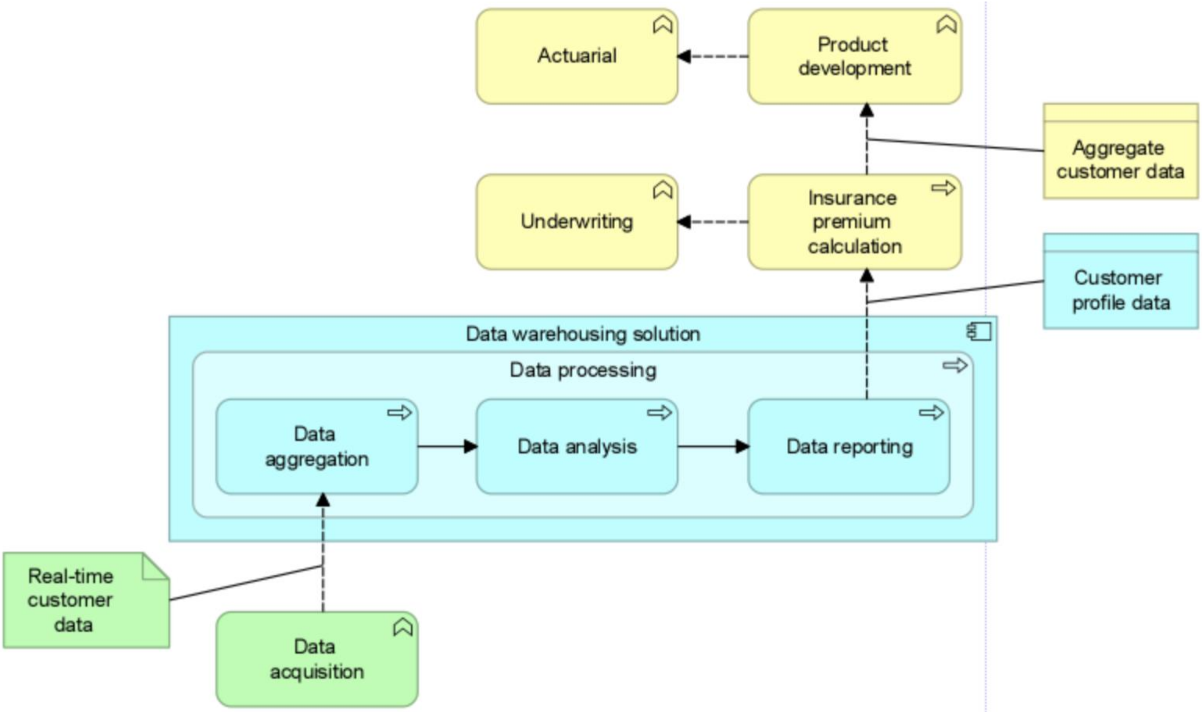
Figure 163 Application Cooperation View



This contains **application components**, linked by data flows.

Additional ArchiSurance content was provided as part of a software distribution kit for BiZZdesign Enterprise Studio [22], which is an Enterprise Architecture modelling tool [223] that provides the ability to model in a number of standard languages including ArchiMate [39].

Figure 164 Application Behaviour (Target)



The above examples do not contain instances of the following elements from the Application layer:

Table 67 Application Elements Unused in ArchiSurance

Element	Definition
Application collaboration	An application collaboration represents an aggregate of two or more application components that work together to perform collective application behavior.
Application interface	An application interface represents a point of access where application services are made available to a user, another application component, or a node.

I.4.2 Completing the Application Layer

The Application layer requires the addition of examples of the elements listed in Table 67 above.

Appendix J Test Models for Modelling Language Effectiveness

Here we present the test models and corresponding narrative descriptions used in the surveys to test the comprehension and utility of the ArchiMate language. These models were constructed using the process described in Appendix H above.

For each layer in scope (Motivation, Business and Application), we present three things: (a) the narrative description, (b) the view (diagram) in ArchiMate format, and (c) the view in an alternative format where element types are distinguished only by the use of colour. Respondents were given (a) and either (b) or (c).

J.1 Motivation View

J.1.1 Narrative Information

J.1.1.1 Purpose

Attempts to capture the motivations of the different stakeholders in the business and sets a detailed context for the business to respond

J.1.1.2 Description

Who's interested in what?

There are two groups of people we need to keep happy. Things valuable to the Board of the company include Shareholder Return, Revenues and Value for Money. Customers are also concerned about Value for Money; they also want to feel that they are getting a good service and are looking for peace of mind.

What should we focus on, and why?

The specific things that the Board wants to see, in order to realise the value they are seeking; they need to ensure Compliance to legislation, they need to hit their Sales target (they are currently failing to meet this), they need to ensure that their own stakeholders are satisfied, partly through continue to focus on Profit (they are failing to satisfy stakeholders as their share price is falling), and they are also targeting Customer satisfaction, because a recent customer survey yielded poor results. Customers also will continue to focus on Price and also Customer satisfaction is, of course, important to them. It should be noted that the Board's definition of Customer Satisfaction is not the same as a Customer's definition!

What are we aiming for? What needs doing?

In order to comply with legislation, we need to host all customer data in the UK. This is made harder for us by the lack of suitable suppliers in the UK.

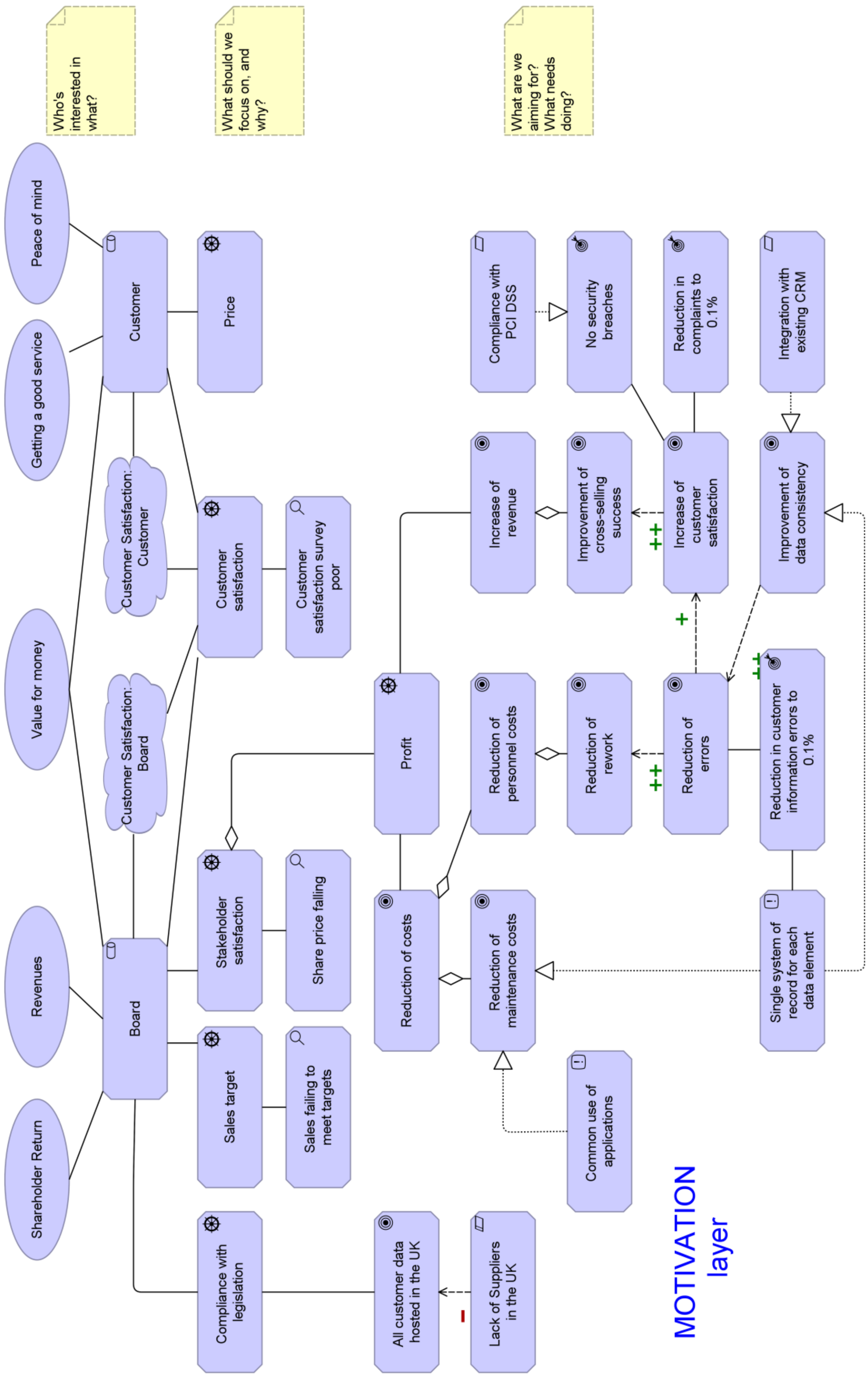
In order to boost profit, we obviously need to reduce our costs and/or increase our revenue.

One way of reducing our costs is to spend less on maintaining our systems and the data within them. To help us get this right over time, we always, in our organisation, apply the following rules when designing solutions: (a) use common applications, and (b) have just a single system of record for each kind of information. Another way of reducing costs is to reduce amount of rework on customer data (for example, personal data, applications, and claims). To do this we need to reduce the errors in this information, and we are aiming to get the amount of errors in customer information down to 0.1%. Having a single system of record is important here as well. We want to improving data consistency as that helps reduce errors and insisting that our new solution integrates with our existing CRM system is one way of doing that.

To increase our revenue, we want to improve cross-selling success, which in turn requires us to increase customer satisfaction levels .Specifically, we want to reduce the number of complaints down to 0.1% of customer contacts, and we also must have no security breaches (which is why we insist on compliance with PCI DSS).

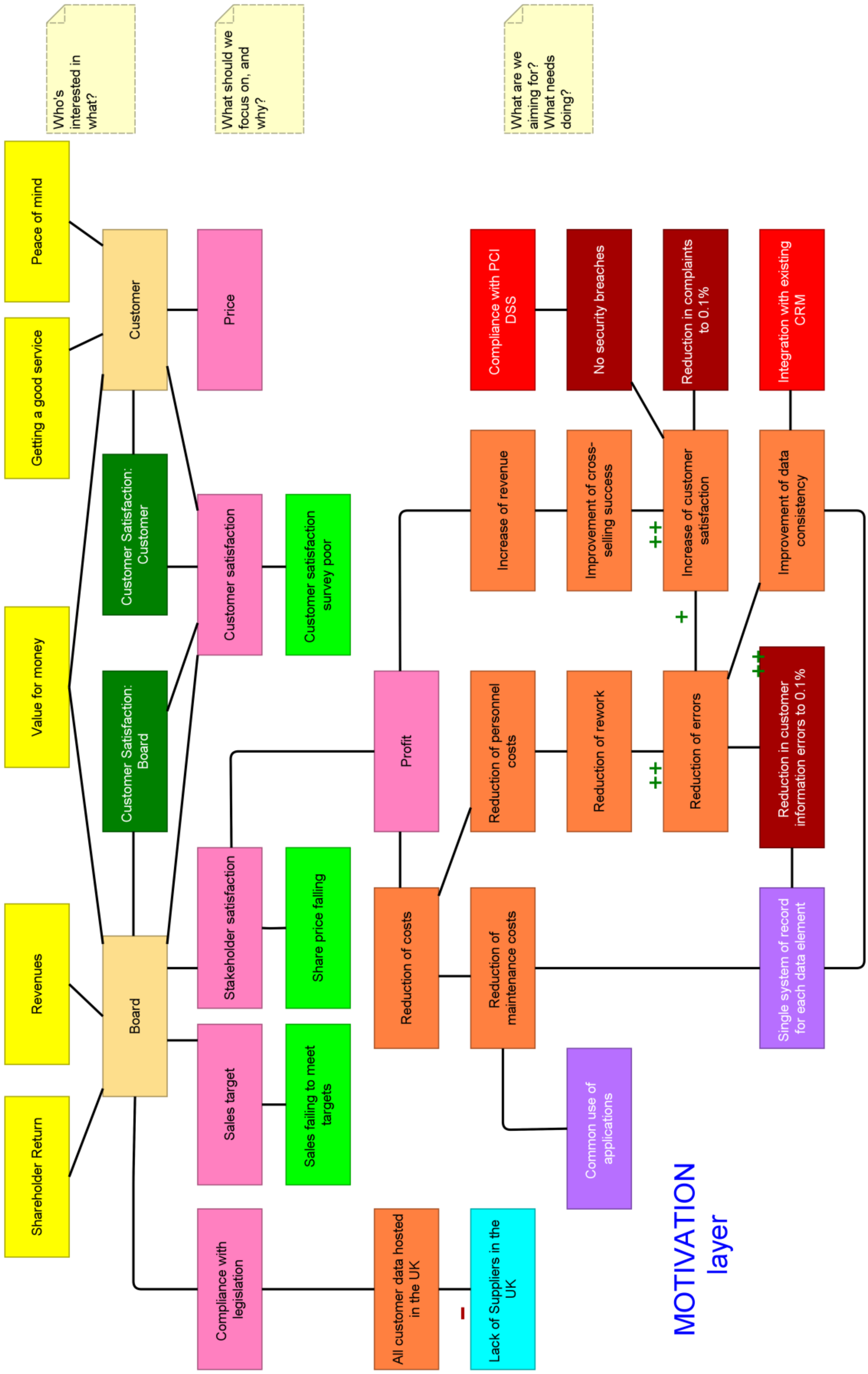
J.1.2 ArchiMate View

Figure 165 ArchiMate Motivation View



J.1.3 Neutral View

Figure 166 Neutral Motivation View



J.2 Business View

J.2.1 Narrative Information

J.2.1.1 Purpose

Attempts to describe the structure of the business and the services that it offers

J.2.1.2 Description

Who are we?

We are ArchiSurance, and we operate out of two locations: our front office functions, plus back office for our homeowners and travel insurance, are located in our Home & Away headquarters. Our shared service centre and our car (auto) insurance is located in our PRO-FIT headquarters.

What does a customer actually buy from us?

Fundamentally, there are two parts to the service that they want from us. They want us to manage their policy, and to handle claims efficiently and fairly. This is the essence of what we are selling.

How is our business organised?

We are split into three main organisations: front-office, which handles customers for all our polices, plus two back-office organisations, one each for our Home & Away and our PRO-FIT (car) policies.

Our front office has one team with marketing people and equipment and another team who are customer-facing, so have the right training and systems to handle that kind of task. On the subject of customers, they are able to contact our front office by either phone or email, unlike our back-office organisations who can only be contacted via email.

In each of our back-office organisations, we are set up to handle five key types of task: claims (obvious), underwriting (where the policy risk is borne), actuarial (calculations of risk that determine policy and risk pricing), finance and related to the finance, one of our teams manages the investments, which is one of the ways we seek to make best use of our available finance. Each of the investment management units has a dedicated investment manager. These investment managers both sit on an Investment Review Board, which is a joint body which also has a representative from our bank (our account manager). Together, this board reviews our investments to ensure that they continue to perform satisfactorily.

One final part of our organisation is our shared service centre which is set up to carry out tasks (such as document processing) required by our other organisations.

What are some of the words you use when you describe your business?

Two of the key words in the insurance sector are "policy" and "claim". Both have a very specific meaning; both of them have tangible forms seen by our clients and our organisation. For example, the essence of a policy is contained within a "policy summary", which contains all the important information about the policy. Similarly, claims manifest themselves in both the submission form (used by clients to submit a claim) and our internal version of the ongoing claim, referred to as a "claim file summary". One key difference between a policy and a claim is that a policy represents a binding agreement that has been made between our organisation and a client.

What do we actually do when you contact us?

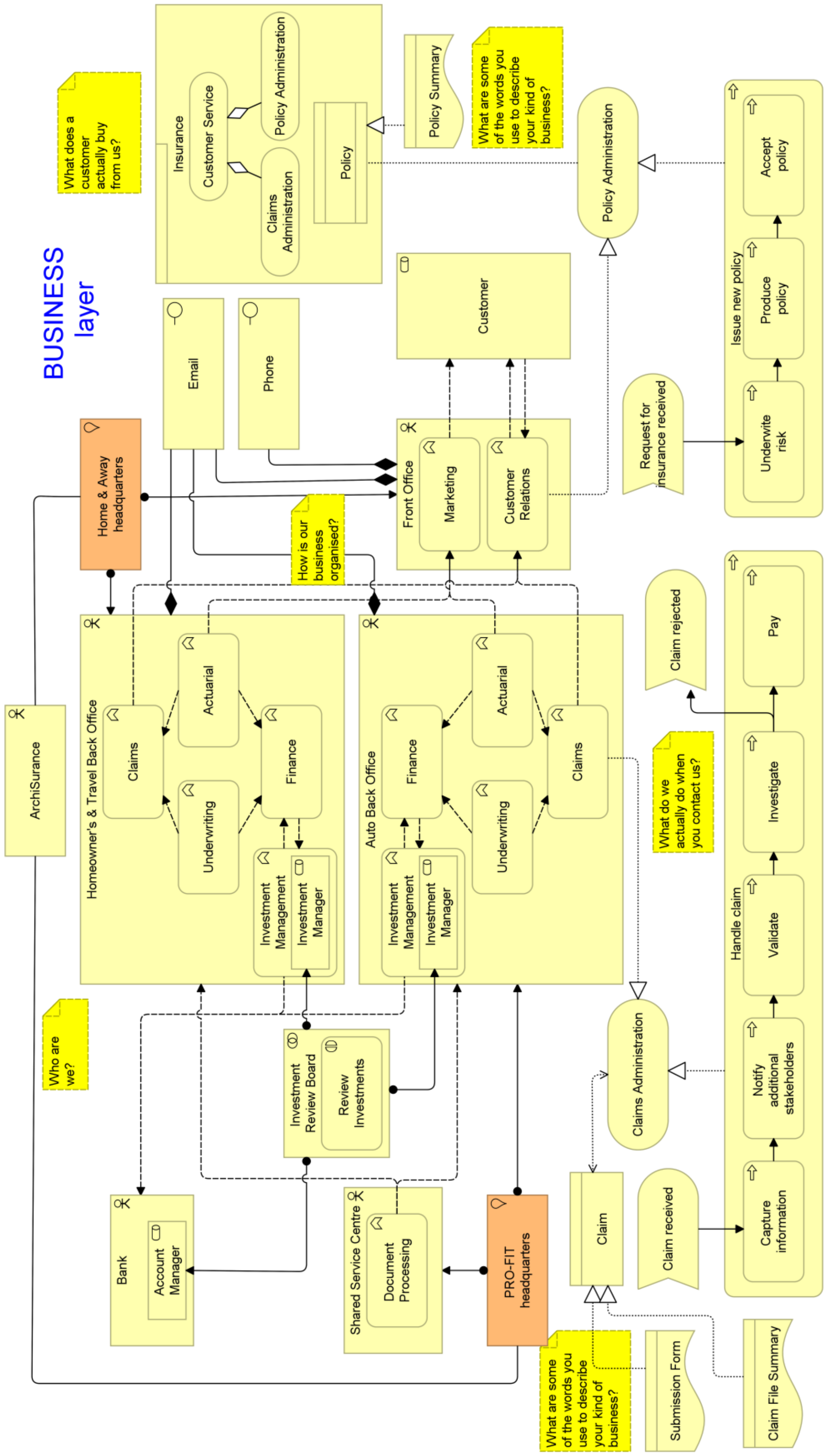
As we said earlier, two of the key things that we do for our customers is administer their policy and then administer any claims arising from that policy. Policy administration is done by our customer relations team, whereas claims administration is done by the claims administration unit within the appropriate back-office division.

For an example of policy administration, obviously a key point here is where a customer requests that we insure them. That request for insurance results in the issuance of a new policy (normally) and that involves a sequence of steps: underwriting the risk, producing the policy and then getting that policy accepted by the customer.

For an example of claims administration, this is generally started when a customer makes a claim that is received by us. We then follow a longer sequence of steps to handle that claim, for example capturing the information, notifying stakeholders and so on, and ultimately this results in one of two outcomes: either we pay the customer, or we reject the claim.

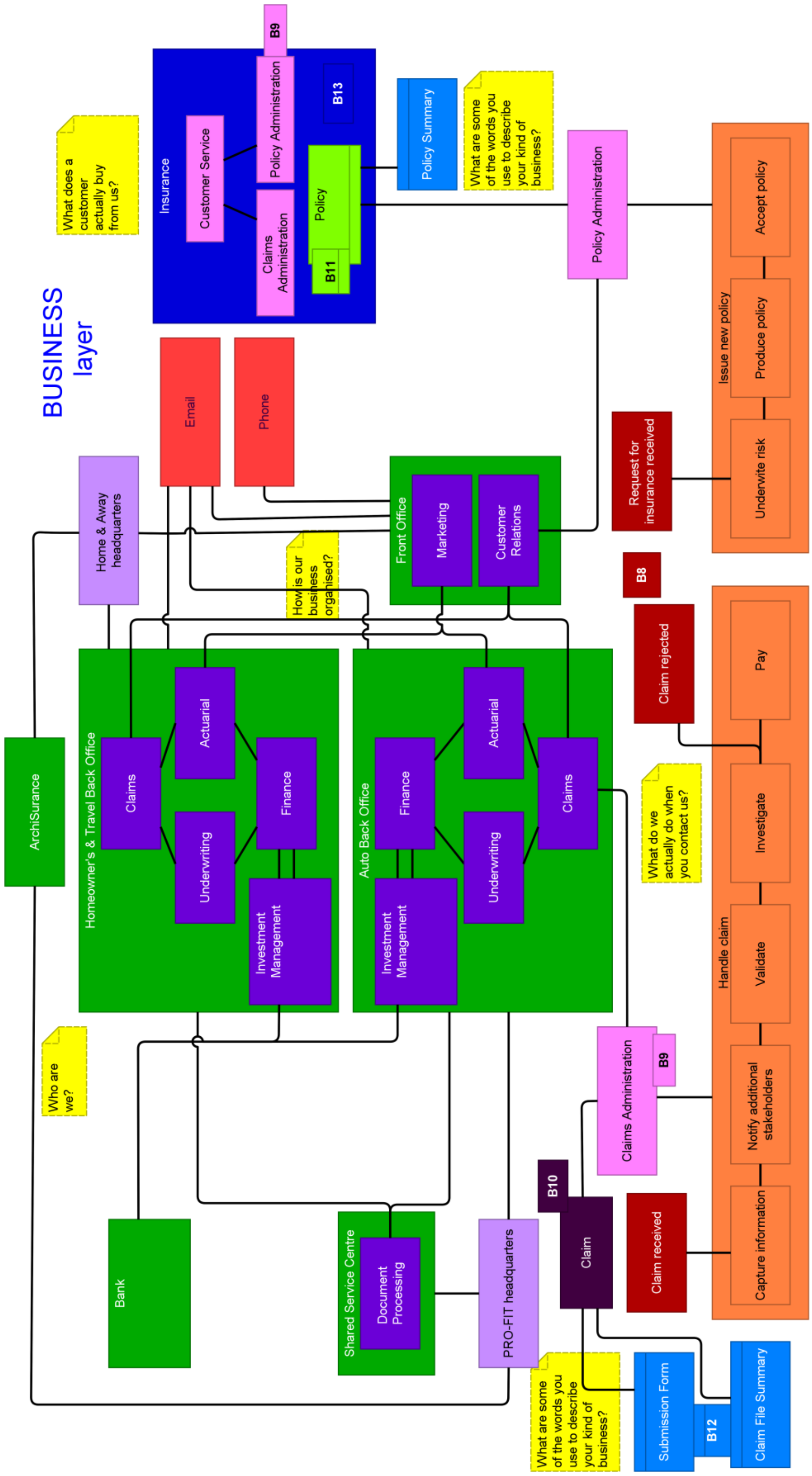
J.2.2 ArchiMate View

Figure 167 ArchiMate Business View



J.2.3 Neutral View

Figure 168 Neutral Business View



J.3 Application View

J.3.1 Narrative Information

J.3.1.1 Purpose

Attempts to describe the structure and operation of the applications supporting the business

J.3.1.2 Description

The main systems that we use are document management, CRM, policy administration and our financial system. There is also a warehouse system that captures data that we receive or generate.

Users have one set of screens used to capture initial information about a claim, and those screens handle both scanning any relevant documentation as well as updating any customer information.

Subsequent processing of the claims is handled via different administration screens, which also trigger any required printing, and finally, there are more screens set up to handle payment of the claim.

The policy administration system must manage both access to detailed customer data as well as information on their claims and policies. This system obviously must exchange data within our main finance system.

It also must talk to our document management system; there's some kind of hook built into that system to allow this to happen; and it also talks to the CRM.

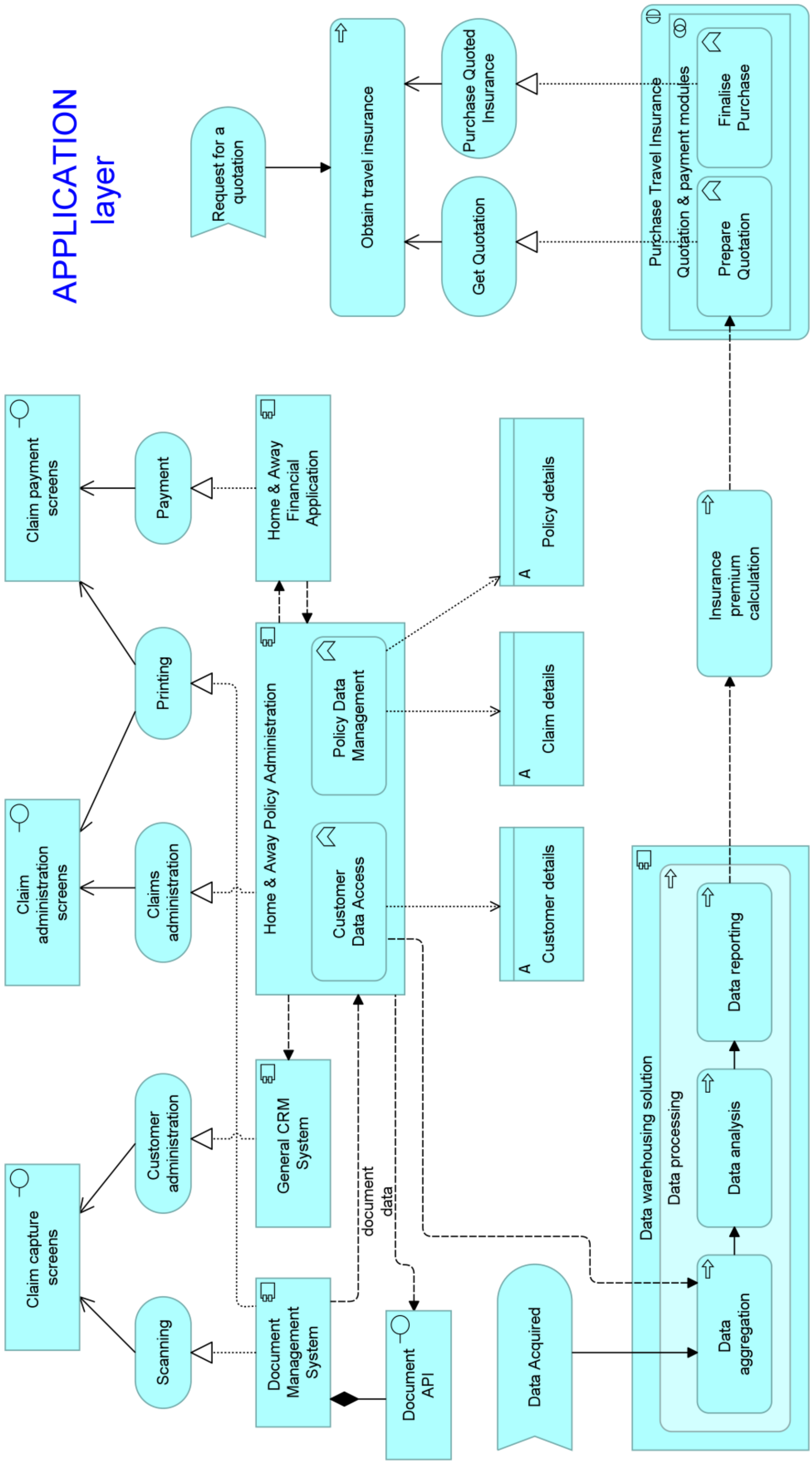
When a request for a quotation is received, our systems need to go through a series of steps in sequence, starting with getting a quotation to the customer and ending (hopefully) with the customer buying the insurance.

This requires the cooperation of two separate parts of our system; the quotation preparation and the finalisation of the purchase.

When data is acquired from any part of our system, it is fed into our warehousing solution, where it goes through a sequence of activities: aggregation, analysis, and reporting. This has a bearing on the ongoing activity of calculating premiums, used in the preparation of quotations.

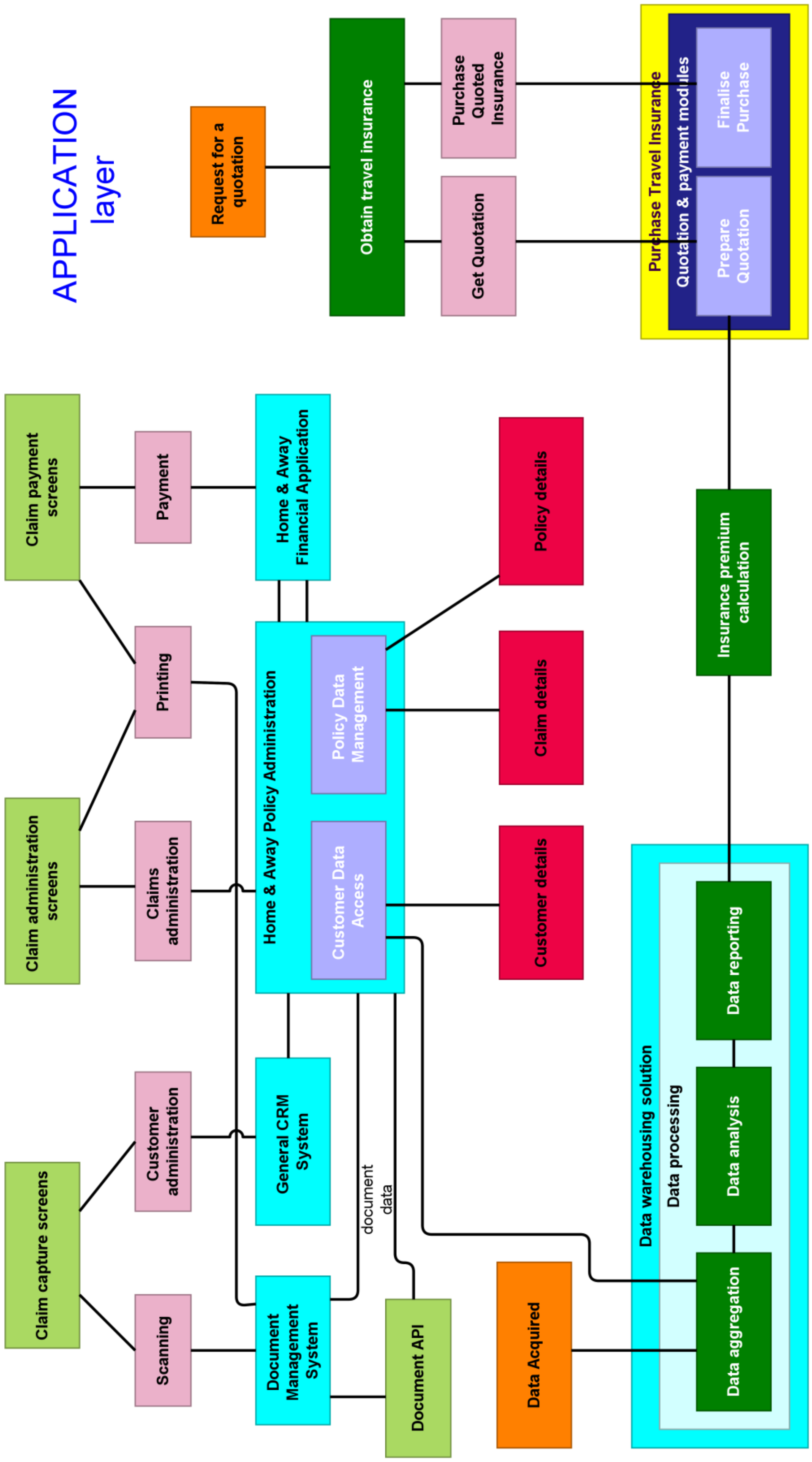
J.3.2 ArchiMate View

Figure 169 ArchiMate Application View



J.3.3 Neutral View

Figure 170 Neutral Application View



J.4 Interviewee Reference Sheets for ArchiMate Concepts

Figure 171 Interviewee Reference (ArchiMate Notation)

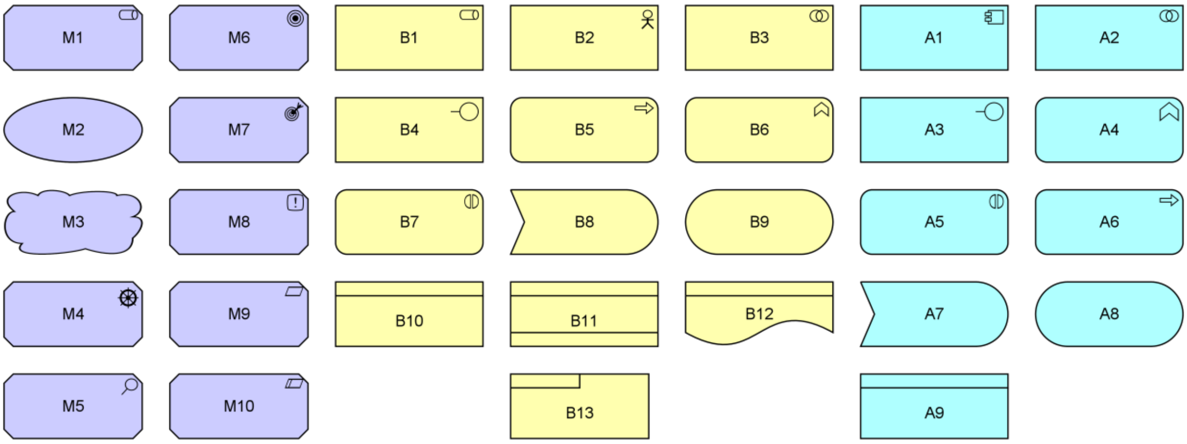


Figure 172 Interviewee Reference (Neutral Notation)

Appendix K Detailed Quantitative Results from Interviews

This appendix contains a summary of all the numeric results from all the interviews, in tabular form.

Table 68 Motivation Layer Raw Data

					Comprehension										Utility									
Date	Job Title	Group	Category	ArchiMate?	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
08/02/2019	Business Architect	No	B	No	1	1	0.7	0.3	1	1	0.7	0.7	0.3	1	1	1	-1	1	-1	1	0	1	-1	-1
16/02/2019	Delivery Manager	No	A	Yes	0.3	0.7	0	0.7	0	0.7	0.3	0.3	1	0.3	1	0	1	1	1	1	1	0	1	1
10/04/2019	3rd Party Assurance Mgr	No	B	Yes	0.7	0.3	0	0.3	1	0.7	1	0.3	0	1	1	1	-1	1	1	1	1	1	0	0
06/05/2019	Programme Lead - Governance	No	A	No	1	1	1	0.7	0.3	0.3	1	0.7	0.3	1	1	1	0	1	1	1	1	1	1	1
18/06/2019	Enterprise Architect (Customer	No	B	No	1	0.7	0	0.3	0.3	1	1	1	0.7	1	1	1	1	1	-1	1	1	0	0	0
16/07/2019	Travel Buyer	No	A	Yes	1	0.3	0.3	0.7	0.7	0.3	0.3	0.3	0	0.7	1	0	-1	1	1	1	0	1	0	0
05/10/2019	Applications Engineer	No	C	Yes	0.3	0	0	0.3	0.7	1	1	0.7	0	0.3	1	1	1	1	-1	1	1	1	1	1
23/07/2019	Freelance IT Consultant / Pasto	No	C	Yes	1	0.7	0	0.7	1	1	0.7	1	1	1	1	1	0	1	1	1	1	1	1	1
09/05/2019	Credit Controller	No		No	1	0.7	1	0.3	0.7	0.3	0.7	0.3	0.3	1	1	1	0	1	-1	1	1	1	1	0
18/06/2019	Platform Architect (telephony)	No	B	Yes	1	1	0.3	1	1	1	0.7	0.7	1	0.3	1	1	1	1	1	1	1	1	0	0
31/08/2019	Senior Sister (nursing)	No		Yes	1	0.3	0	0.7	0.3	1	0.7	0.3	0	0.3	1	1	1	1	1	1	1	1	1	1
16/10/2019	Account Manager (retired)	No	A	Yes	1	1	0	0.3	1	0.7	0.3	0.3	0.3	0.3	1	0	-1	1	1	1	1	1	1	1
30/10/2019	Managing Director (retired)	No	A	No	1	0.7	1	0.3	0	0.7	0.7	0	0	0	1	0	1	1	1	1	1	1	1	0
30/10/2019	CEO, co-founder, charity	No	A	No	1	0.7	1	0.3	1	0.7	0	0	0.7	0.3	1	1	-1	1	1	1	1	-1	-1	1
31/10/2019	Product Manager	Yes	A	Yes	0.3	0	0	0.3	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1	0
31/10/2019	Data Manager	Yes		No	1	0.7	0	0.3	0.3	0.7	0.7	0.3	0.3	1	1	1	1	1	-1	1	1	1	1	-1
31/10/2019	Student	Yes		No	1	0.7	0.3	0.3	0.7	1	0.3	0	0	0.3	1	1	0	1	-1	0	0	-1	-1	-1
31/10/2019	Credit policy analyst	Yes	A	Yes	1	0.7	0.7	0.3	0.3	0.7	0.3	0	0.3	0.3	1	1	1	1	1	1	1	1	1	1
31/10/2019	Junior management consultant	Yes	A	No	1	1	0.3	0.7	1	1	0	0.7	1	1	1	1	-1	1	0	0	0	-1	-1	-1
31/10/2019	Student	Yes		Yes	0.3	0	0	0	0	0	1	0	0	0	-1	1	0	1	1	-1	0	-1	0	-1
31/10/2019	Student / Associate	Yes	C	Yes											1	0	-1	1	-1	1	0	1	1	-1
31/10/2019	Student	Yes	A	No	0	0	0	0	0.3	0.3	0	0	0	0	1	1	0	1	-1	0	0	0	0	-1
31/10/2019	Supply chain analyst	Yes	A	No	1	1	0.3	0.3	0.7	0.3	0	0.7	0.3	0.3	1	0	-1	1	0	1	0	-1	0	1

Table 69 Business Layer Raw Data

					Comprehension													Utility												
Date	Job Title	Group	Category	ArchiMate?	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
16/02/2019	Delivery Manager	No	A	Yes	0.3	0.3	0.7	1	1	0.3	0.3	0.3	0	0.3	0.3	0.3	0.3	0	1	0	1	1	1	0	1	1	0	0	1	1
02/03/2019	COO	No	A	No	1	0.3	0.3	1	1	1	1	1	0.3	0.3	1	0.7	0.7	-1	1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1
20/03/2019	Principal Engineering Manager	No	A	Yes	0.7	0.3	0.3	1	1	0.7	0	0.7	0	0	1	0	0	1	1	0.5	1	1	1	0.5	1	0	-1	1	-1	-1
28/03/2019	Service Architect	No	B	Yes	0.3	0.3	0	0	1	0.7	0	0.3	0	0	0	1	0	0	0	-1	0	1	1	-1	1	1	1	1	1	0
18/06/2019	Freelance IT Consultant (now Pastor)	No	C	Yes	0.7	0.3	0	1	1	1	0	1	0.3	0.7	0.7	0.3	0.3	1	1	-1	1	1	0	-1	1	0	1	1	1	0
16/07/2019	Travel Buyer	No	A	Yes	0.3	0.7	0.7	1	1	0.7	1	0.3	0.3	0	1	0	0	1	1	1	1	1	1	1	1	1	-1	1	1	0
18/06/2019	Enterprise Architect (Customer)	No	B	No	0.7	0.3	1	1	1	1	1	1	0.7	1	0.7	0.7	0.3	1	1	1	1	1	1	-1	1	1	1	1	1	0
18/06/2019	Data Governance Architect	No	B	Yes	1	0.7	0.7	1	1	0.7	0.3	0.7	0.7	1	0.7	0.7	0.3	1	1	1	1	-1	1	1	-1	1	-1	-1	-1	1
06/05/2019	Programme Lead - Governance	No	A	No	0.3	0.3	0.7	1	1	0.3	1	1	1	0.7	0.7	0.3	1	0	1	0	1	1	1	0	1	1	1	1	0	1
09/05/2019	Credit Controller	No		No	0.3	0	0	1	1	0.7	0	0.3	0.3	0.3	0	0.7	1	0	1	1	0	1	1	0	1	1	0	1	0	1
20/06/2019	Platform Architect (telephony)	No	B	Yes	1	0.3	0.3	1	1	0.7	1	0.7	0.3	0.7	0.3	1	0	1	1	-1	1	1	1	-1	0	0	1	1	1	-1
05/10/2019	Welfare Benefits Advisor	No	A	No	0.3	0.3	0.3	0.3	1	0.7	0.7	0.3	0.3	0	1	0.3	0	1	1	1	-1	1	1	-1	0	1	1	1	1	1
31/08/2019	Senior Sister (nursing)	No	A	Yes	0.3	0	1	1	1	0.3	0	0.3	0	0	0	0.7	0.3	1	1	1	1	1	1	0	1	1	1	1	1	1
30/10/2019	Managing Director (retired)	No	A	No	0.7	0	1	1	1	1	1	0.7	0.7	0.7	0.3	0.7	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1
30/10/2019	Initiative Assistant	No	A	No	0.3	0.7	1	1	0	1	1	0.7	1	0	0	0.3	0	1	1	1	1	1	1	-1	1	-1	1	-1	-1	-1
30/10/2019	Teacher, Consultant	No	A	No	0	0	0	1	0.7	0.7	0.3	0.3	0.3	0.3	0	0.7	0.7	0	1	0	0	1	1	0	-1	1	1	1	-1	1
31/10/2019	Product Manager	Yes	A	Yes	0.3	0.7	0.3	0.3	0.3	0.3	0.3	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
31/10/2019	Data Manager	Yes		No	0.7	0.7	0.3	1	1	1	1	0	0	1	0	0.3	0	1	1	-1	1	1	1	-1	0	1	1	0	0	1
31/10/2019	Student	Yes		No	0	0.3	0.3	1	1	0.7	0.3	0.7	0	0	0.3	0	0.3	1	0	0	1	1	1	0	0	1	-1	1	-1	1
31/10/2019	Credit policy analyst	Yes	A	Yes	0.7	0.3	0	0.3	0.7	0.7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
31/10/2019	Junior management consultant	Yes	A	No	1	0.3	1	1	0.7	0.3	1	0.7	0.3	0	0	0	0	1	1	-1	-1	1	1	-1	-1	0	0	1	1	0
31/10/2019	Student	Yes		Yes	1	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	1	-1	0	0	-1	-1	1	0	-1	1	-1
31/10/2019	Student / Associate	Yes	C	Yes	0.3	0.3	0.7	0.3	0.7	0.3	0.7	0.7	0.3	0.3	0.3	0.3	0.7	-1	1	1	1	-1	-1	1	1	-1	1	-1	-1	1
31/10/2019	Student	Yes	A	No	0	0.3	0	0.3	0.3	0.7	1	0.3	0	0.3	0.7	0	0	0	1	0	1	1	1	0	-1	1	-1	0	0	0
31/10/2019	Supply chain analyst	Yes	A	No	0.3	0.3	0	1	1	1	0	0	0.7	0	0.3	0	0	-1	1	0	0	0	1	-1	1	1	1	1	1	1

Table 70 Application Layer Raw Data

					Comprehension									Utility								
Date	Job Title	Group	Category	ArchiMate?	A1	A2	A3	A4	A5	A6	A7	A8	A9	A1	A2	A3	A4	A5	A6	A7	A8	A9
04/03/2019	IT Manager	No	-	Yes	1	0.3	0	0	0	1	0.3	0.3	0.3	1	-0.5	1	1	-0.5	0	1	0	1
28/03/2019	EA (Technology)	No	B	No	1	0.3	1	0.3	0	0.7	1	1	0.3	1	1	1	1	1	1	1	1	1
28/03/2019	Service Architect	No	B	No	1	0	1	0.3	0.3	1	0.3	0	0.3	0	1	1	1	1	1	0	1	0
17/06/2019	Head of IT Strateg	No	A	Yes	1	0	0.3	0.3	0.3	0.7	0.3	0	0.3									
16/10/2019	Account Manager	No	A	Yes	1	0.3	0.3	0.3	0	1	0	1	1	1	0	0	1	0	1	1	1	0
05/10/2019	Welfare Benefits /	No	A	No	1	0	0	0.7	0	0	0	0	0.7	1	0	0	1	0	1	1	1	0
05/10/2019	Applications Engi	No	C	No	1	0.7	1	1	0	0.7	0.3	0	0.7	1	-1	1	1	-1	1	1	1	0
25/06/2019	Platform Archite	No	B	No	1	0.3	0	0.3	0.3	0	1	0.7	0.7	1	0	-1	1	0	1	0	1	1
30/10/2019	Cofounder, Dev H	No	C		0.3	0	0.7	0	0.7	0.7	1	0.3	1	1	-1	1	-1	1	1	1	0	1
30/10/2019	Initiative Assistar	No	A	Yes	1	0.7	0.3	0	0	1	0.7	1	1	1	-1	1	1	-1	1	1	1	1
30/10/2019	Teacher, Consulta	No	A	Yes	0.3	0	0.3	0.3	0	0.7	0.7	0	0.7	1	1	0	0	1	1	1	1	1
31/10/2019	Product Manager	Yes	A	Yes	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	1	1	1	1	1	1	1	1	1
31/10/2019	Data Manager	Yes		No	1	0	0.7	0.3	0	1	0.7	0	0	1	-1	1	0	-1	1	1	1	1
31/10/2019	Student	Yes		No	1	0.3	0.7	0.3	0	0.7	0.7	0	0.3	0	-1	1	1	0	0	1	0	1
31/10/2019	Credit policy anal	Yes	A	Yes	0.7	0	0.7	0.3	0.3	0.7	0.3	0.7	0	1	1	1	1	1	1	1	1	1
31/10/2019	Junior manageme	Yes	A	No	1	0	1	0.3	0	0.3	1	0	0.7	1	0	1	0	0	1	1	0	1
31/10/2019	Student	Yes		Yes	1	0	0	0	0	0	0	0	0									
31/10/2019	Student / Associa	Yes	C	Yes	0.3	0.3	0.7	0.7	0.3	0.7	0.3	0.7	0.3	-1	1	1	-1	1	-1	0	0	-1
31/10/2019	Student	Yes	A	No	1	0.3	0.3	0	0	1	0.3	0	0.7	0	0	0	0	0	0	0	0	0
31/10/2019	Supply chain anal	Yes	A	No	1	0	0.3	0.3	0	0	0.3	0.3	1	1	-1	0	0	1	1	1	1	1

Table 71 Motivation Layer Detailed Analysis

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1				Comprehension										Utility									
2	Group	Category	ArchiMate?	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
3	No	B	No	1	1	0.7	0.3	1	1	0.7	0.7	0.3	1	1	1	-1	1	-1	1	0	1	-1	-1
4	No	A	Yes	0.3	0.7	0	0.7	0	0.7	0.3	0.3	1	0.3	1	0	1	1	1	1	1	0	1	1
5	No	B	Yes	0.7	0.3	0	0.3	1	0.7	1	0.3	0	1	1	1	-1	1	1	1	1	1	0	0
6	No	A	No	1	1	1	0.7	0.3	0.3	1	0.7	0.3	1	1	1	0	1	1	1	1	1	1	1
7	No	B	No	1	0.7	0	0.3	0.3	1	1	1	0.7	1	1	1	1	1	-1	1	1	0	0	0
8	No	A	Yes	1	0.3	0.3	0.7	0.7	0.3	0.3	0.3	0	0.7	1	0	-1	1	1	1	0	1	0	0
9	No	C	Yes	0.3	0	0	0.3	0.7	1	1	0.7	0	0.3	1	1	1	1	-1	1	1	1	1	1
10	No	C	Yes	1	0.7	0	0.7	1	1	0.7	1	1	1	1	1	0	1	1	1	1	1	1	1
11	No		No	1	0.7	1	0.3	0.7	0.3	0.7	0.3	0.3	1	1	1	0	1	-1	1	1	1	1	0
12	No	B	Yes	1	1	0.3	1	1	1	0.7	0.7	1	0.3	1	1	1	1	1	1	1	1	0	0
13	No		Yes	1	0.3	0	0.7	0.3	1	0.7	0.3	0	0.3	1	1	1	1	1	1	1	1	1	1
14	No	A	Yes	1	1	0	0.3	1	0.7	0.3	0.3	0.3	0.3	1	0	-1	1	1	1	1	1	1	1
15	No	A	No	1	0.7	1	0.3	0	0.7	0.7	0	0	0	1	0	1	1	1	1	1	1	1	0
16	No	A	No	1	0.7	1	0.3	1	0.7	0	0	0.7	0.3	1	1	-1	1	1	1	1	-1	-1	1
17	Yes	A	Yes	0.3	0	0	0.3	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1	0
18	Yes		No	1	0.7	0	0.3	0.3	0.7	0.7	0.3	0.3	1	1	1	1	1	-1	1	1	1	1	-1
19	Yes		No	1	0.7	0.3	0.3	0.7	1	0.3	0	0	0.3	1	1	0	1	-1	0	0	-1	-1	-1
20	Yes	A	Yes	1	0.7	0.7	0.3	0.3	0.7	0.3	0	0.3	0.3	1	1	1	1	1	1	1	1	1	1
21	Yes	A	No	1	1	0.3	0.7	1	1	0	0.7	1	1	1	1	-1	1	0	0	0	-1	-1	-1
22	Yes		Yes	0.3	0	0	0	0	0	1	0	0	0	-1	1	0	1	1	-1	0	-1	0	-1
23	Yes	C	Yes											1	0	-1	1	-1	1	0	1	1	-1
24	Yes	A	No	0	0	0	0	0.3	0.3	0	0	0	0	1	1	0	1	-1	0	0	0	0	-1
25	Yes	A	No	1	1	0.3	0.3	0.7	0.3	0	0.7	0.3	0.3	1	0	-1	1	0	1	0	-1	0	1
26																							
27		ArchiMate	12																				
28		Neutral	11																				
29		Count Rows Populated		22	22	22	22	22	22	22	22	22	22	23	23	23	23	23	23	23	23	23	23
30		Total Score		18	13	6.9	9.1	12	14	11	8.3	7.5	11	21	16	-1	23	3	18	15	10	8	2
31		Count Rows ArchiMate		12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
32		Score Rows ArchiMate		7.9	5	1.3	5.3	6	7.1	6.3	3.9	3.6	4.5	10	7	0	12	6	10	9	9	8	4
33		Count Rows Neutral		11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
34		Score Rows Neutral		10	8.2	5.6	3.8	6.3	7.3	5.1	4.4	3.9	6.9	11	9	-1	11	-3	8	6	1	0	-2
35		Count Rows Group		9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
36		Score Rows Group		5.6	4.1	1.6	2.2	3.3	4	2.3	1.7	1.9	2.9	7	6	-2	9	-3	4	3	0	2	-4
37		Count Rows Non-Group		14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
38		Score Rows Non-Group		12	9.1	5.3	6.9	9	10	9.1	6.6	5.6	8.5	14	10	1	14	6	14	12	10	6	6
39		Count Rows Type A		11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
40		Score Rows Type A		8.6	7.1	4.6	4.6	5.3	5.7	2.9	3	3.9	4.2	11	5	-3	11	5	9	7	3	4	4
41		Count Rows Type B		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
42		Score Rows Type B		3.7	3	1	1.9	3.3	3.7	3.4	2.7	2	3.3	4	4	0	4	0	4	3	3	-1	-1
43			Element	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10										
44		Overall	Comprehen	0.81	0.60	0.31	0.41	0.56	0.65	0.52	0.38	0.34	0.52	0.91	0.70	####	1.00	0.13	0.78	0.65	0.43	0.35	0.09
45			Utility	0.91	0.70	####	1.00	0.13	0.78	0.65	0.43	0.35	0.09										
46		ArchiMate	Comprehen	0.66	0.42	0.11	0.44	0.50	0.59	0.53	0.33	0.30	0.38	0.83	0.58	0.00	1.00	0.50	0.83	0.75	0.75	0.67	0.33
47			Utility	0.83	0.58	0.00	1.00	0.50	0.83	0.75	0.75	0.67	0.33										
48		Neutral	Comprehen	0.91	0.75	0.51	0.35	0.57	0.66	0.46	0.40	0.35	0.63	1.00	0.82	####	1.00	####	0.73	0.55	0.09	0.00	####
49			Utility	1.00	0.82	####	1.00	####	0.73	0.55	0.09	0.00	####										
50		Group	Comprehen	0.62	0.46	0.18	0.24	0.37	0.44	0.26	0.19	0.21	0.32	0.78	0.67	####	1.00	####	0.44	0.33	0.00	0.22	####
51			Utility	0.78	0.67	####	1.00	####	0.44	0.33	0.00	0.22	####										
52		Interview	Comprehen	0.88	0.65	0.38	0.49	0.64	0.74	0.65	0.47	0.40	0.61	1.00	0.71	0.07	1.00	0.43	1.00	0.86	0.71	0.43	0.43
53			Utility	1.00	0.71	0.07	1.00	0.43	1.00	0.86	0.71	0.43	0.43										
54		ArchiMate	Comprehen	####	####	####	0.10	####	####	0.06	####	####	####										
55			Utility	####	####	0.09	0.00	0.77	0.11	0.20	0.66	0.67	0.52										
56		Int vs. Grc	Comprehen	0.26	0.19	0.20	0.25	0.28	0.30	0.39	0.28	0.19	0.28										
57			Utility	0.22	0.05	0.29	0.00	0.76	0.56	0.52	0.71	0.21	0.87										
58		Type A	Comprehen	0.78	0.65	0.42	0.42	0.48	0.52	0.26	0.27	0.35	0.38	1.00	0.45	####	1.00	0.45	0.82	0.64	0.27	0.36	0.36
59			Utility	1.00	0.45	####	1.00	0.45	0.82	0.64	0.27	0.36	0.36										
60		Type B	Comprehen	0.9	0.8	0.3	0.5	0.8	0.9	0.9	0.7	0.5	0.8	1	1	0	1	0	1	0.8	0.8	-0.3	-0.3
61			Utility	1	1	0	1	0	1	0.8	0.8	-0.3	-0.3										
62				Stakeholder	Value	Meaning	Driver	Assessment	Goal	Outcome	Principle	Requirement	Constraint										

Table 72 Business Layer Detailed Analysis

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1				Comprehension													Utility												
2	Group	Category	ArchiMate?	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
3	No	A	Yes	0.3	0.3	0.7	1	1	0.3	0.3	0.3	0	0.3	0.3	0.3	0.3	0	1	0	1	1	1	0	1	1	0	0	1	1
4	No	A	No	1	0.3	0.3	1	1	1	1	1	0.3	0.3	1	0.7	0.7	-1	1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1
5	No	A	Yes	0.7	0.3	0.3	1	1	0.7	0	0.7	0	0	1	0	0	1	1	0.5	1	1	1	0.5	1	0	-1	1	-1	-1
6	No	B	Yes	0.3	0.3	0	0	1	0.7	0	0.3	0	0	0	1	0	0	0	-1	0	1	1	-1	1	1	1	1	1	0
7	No	C	Yes	0.7	0.3	0	1	1	1	0	1	0.3	0.7	0.7	0.3	0.3	1	1	-1	1	1	0	-1	1	0	1	1	1	0
8	No	A	Yes	0.3	0.7	0.7	1	1	0.7	1	0.3	0.3	0	1	0	0	1	1	1	1	1	1	1	1	1	-1	1	1	0
9	No	B	No	0.7	0.3	1	1	1	1	1	1	0.7	1	0.7	0.7	0.3	1	1	1	1	1	1	-1	1	1	1	1	1	0
10	No	B	Yes	1	0.7	0.7	1	1	0.7	0.3	0.7	0.7	1	0.7	0.7	0.3	1	1	1	1	-1	1	1	-1	1	-1	-1	-1	1
11	No	A	No	0.3	0.3	0.7	1	1	0.3	1	1	1	0.7	0.7	0.3	1	0	1	0	1	1	1	0	1	1	1	1	0	1
12	No		No	0.3	0	0	1	1	0.7	0	0.3	0.3	0.3	0	0.7	1	0	1	1	0	1	1	0	1	1	0	1	0	1
13	No	B	Yes	1	0.3	0.3	1	1	0.7	1	0.7	0.3	0.7	0.3	1	0	1	1	-1	1	1	1	-1	0	0	1	1	1	-1
14	No	A	No	0.3	0.3	0.3	0.3	1	0.7	0.7	0.3	0.3	0	1	0.3	0	1	1	1	-1	1	1	-1	0	1	1	1	1	1
15	No	A	Yes	0.3	0	1	1	1	0.3	0	0.3	0	0	0	0.7	0.3	1	1	1	1	1	1	0	1	1	1	1	1	1
16	No	A	No	0.7	0	1	1	1	1	1	0.7	0.7	0.7	0.3	0.7	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1
17	No	A	No	0.3	0.7	1	1	0	1	1	0.7	1	0	0	0.3	0	1	1	1	1	1	1	-1	1	-1	1	-1	-1	-1
18	No	A	No	0	0	0	1	0.7	0.7	0.3	0.3	0.3	0.3	0	0.7	0.7	0	1	0	0	1	1	0	-1	1	1	1	-1	1
19	Yes	A	Yes	0.3	0.7	0.3	0.3	0.3	0.3	0.3	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
20	Yes		No	0.7	0.7	0.3	1	1	1	1	0	0	1	0	0.3	0	1	1	-1	1	1	1	-1	0	1	1	0	0	1
21	Yes		No	0	0.3	0.3	1	1	0.7	0.3	0.7	0	0	0.3	0	0.3	1	0	0	1	1	1	0	0	1	-1	1	-1	1
22	Yes	A	Yes	0.7	0.3	0	0.3	0.7	0.7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
23	Yes	A	No	1	0.3	1	1	0.7	0.3	1	0.7	0.3	0	0	0	0	1	1	-1	-1	1	1	-1	-1	0	0	1	1	0
24	Yes		Yes	1	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	1	-1	0	0	-1	-1	1	0	-1	1	-1
25	Yes	C	Yes	0.3	0.3	0.7	0.3	0.7	0.3	0.7	0.7	0.3	0.3	0.3	0.3	0.7	-1	1	1	1	-1	-1	1	1	-1	1	-1	-1	1
26	Yes	A	No	0	0.3	0	0.3	0.3	0.7	1	0.3	0	0.3	0.7	0	0	0	1	0	1	1	1	0	-1	1	-1	0	0	0
27	Yes	A	No	0.3	0.3	0	1	1	1	0	0	0.7	0	0.3	0	0	-1	1	0	0	0	1	-1	1	1	1	1	1	1
28																													
29		ArchiMat	12																										
30		Neutral	13																										
31		Count Rows Populated		25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
32		Total Score		12.5	8	10.6	20.5	20.4	16.5	12.9	12	7.5	7.6	9.3	9	6.6	11	22	8.5	15	17	21	-4.5	9	15	9	12	6	8
33		Count Rows ArchiMate		12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
34		Score Rows ArchiMate		6.9	4.2	4.7	8.9	9.7	6.4	3.6	5	1.9	3	4.3	4.3	1.9	6	10	4.5	9	7	8	1.5	7	7	4	5	5	2
35		Count Rows Neutral		13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
36		Score Rows Neutral		5.6	3.8	5.9	11.6	10.7	10.1	9.3	7	5.6	4.6	5	4.7	4.7	5	12	4	6	10	13	-6	2	8	5	7	1	6
37		Count Rows Group		9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
38		Score Rows Group		4.3	3.2	2.6	6.2	5.7	5	4.3	2.4	1.3	1.6	1.6	0.6	1	2	7	2	4	5	6	-1	1	6	3	3	2	4
39		Count Rows Non-Group		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
40		Score Rows Non-Group		8.2	4.8	8	14.3	14.7	11.5	8.6	9.6	6.2	6	7.7	8.4	5.6	9	15	6.5	11	12	15	-3.5	8	9	6	9	4	4
41		Count Rows Type A		15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
42		Score Rows Type A		6.5	4.8	7.3	12.2	11.7	9.7	8.6	6.6	4.9	2.6	6.3	4	3.7	7	15	7.5	9	12	15	-0.5	6	9	5	9	4	5
43		Count Rows Type B		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
44		Score Rows Type B		3	1.6	2	3	4	3.1	2.3	2.7	1.7	2.7	1.7	3.4	0.6	3	3	0	3	2	4	-2	1	3	2	2	2	0
45			Element	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
46		Overall	Comprehensio	0.50	0.32	0.42	0.82	0.82	0.66	0.52	0.48	0.30	0.30	0.37	0.36	0.26	0.44	0.88	0.34	0.60	0.68	0.84	-0.18	0.36	0.60	0.36	0.48	0.24	0.32
47			Utility	0.44	0.88	0.34	0.60	0.68	0.84	-0.18	0.36	0.60	0.36	0.48	0.24	0.32													
48		ArchiMat	Comprehensio	0.58	0.35	0.39	0.74	0.81	0.53	0.30	0.42	0.16	0.25	0.36	0.36	0.16	0.50	0.83	0.38	0.75	0.58	0.67	0.13	0.58	0.58	0.33	0.42	0.42	0.17
49			Utility	0.50	0.83	0.38	0.75	0.58	0.67	0.13	0.58	0.58	0.33	0.42	0.42	0.17													
50		Neutral	Comprehensio	0.43	0.29	0.45	0.89	0.82	0.78	0.72	0.54	0.43	0.35	0.38	0.36	0.36	0.38	0.92	0.31	0.46	0.77	1.00	-0.46	0.15	0.62	0.38	0.54	0.08	0.46
51			Utility	0.38	0.92	0.31	0.46	0.77	1.00	-0.46	0.15	0.62	0.38	0.54	0.08	0.46													
52		Group	Comprehensio	0.48	0.36	0.29	0.69	0.63	0.56	0.48	0.27	0.14	0.18	0.18	0.07	0.11	0.22	0.78	0.22	0.44	0.56	0.67	-0.22	0.50	0.56	0.38	0.56	0.25	0.25
53			Utility	0.22	0.78	0.22	0.44	0.56	0.67	-0.22	0.50	0.56	0.38	0.56	0.25	0.25													
54		Interview	Comprehensio	0.51	0.30	0.50	0.89	0.92	0.72	0.54	0.60	0.39	0.38	0.48	0.53	0.35	0.56	0.94	0.41	0.69	0.75	0.94	-0.22	0.50	0.56	0.38	0.56	0.25	0.25
55			Utility	0.56	0.94	0.41	0.69	0.75	0.94	-0.22	0.50	0.56	0.38	0.56	0.25	0.25													
56		ArchiMat	Comprehensio	0.14	0.06	-0.06	-0.15	-0.01	-0.24	-0.42	-0.12	-0.27	-0.10	-0.03	0.00	-0.20													
57			Utility	0.12	-0.09	0.07	0.29	-0.19	-0.33	0.59	0.43	-0.03	-0.05	-0.12	0.34	-0.29													
58		Int vs. Grc	Comprehensio	0.03	-0.06	0.21	0.20	0.29	0.16	0.06	0.33	0.24	0.20	0.30	0.46	0.24													
59			Utility	0.34	0.16	0.18	0.24	0.19	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00													
60		Type A	Comprehensio	0.43	0.32	0.49	0.81	0.78	0.65	0.57	0.44	0.33	0.17	0.42	0.27	0.25	0.47	1.00	0.50	0.60	0.80	1.00	-0.03	0.40	0.60	0.33	0.60	0.27	0.33
61			Utility	0.47	1.00	0.50	0.60	0.80	1.00	-0.03	0.40	0.60	0.33	0.60	0.27	0.33													
62		Type B	Comprehensio	0.75	0.40	0.50	0.75	1.00	0.78	0.58	0.68	0.43	0.68	0.43	0.85	0.15	0.75	0.75	0.00	0.75	0.50	1.00	-0.50	0.25	0.75	0.50	0.50	0.50	0.00
63			Utility	0.75	0.75	0.00	0.75	0.50	1.00	-0.50	0.25	0.75	0.50	0.50	0.50	0.00													
64				Business role	Business actor	Business collaboration	Business interface	Business process	Business function	Business interaction	Business event	Business service	Business object	Contract	Representation	Product													

Table 73 Application Layer Detailed Analysis

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1				Comprehension									Utility								
2	Group	Category	ArchiMate?	A1	A2	A3	A4	A5	A6	A7	A8	A9	A1	A2	A3	A4	A5	A6	A7	A8	A9
3	No	-	Yes	1	0.3	0	0	0	1	0.3	0.3	0.3	1	-0.5	1	1	-0.5	0	1	0	1
4	No	B	No	1	0.3	1	0.3	0	0.7	1	1	0.3	1	1	1	1	1	1	1	1	1
5	No	B	No	1	0	1	0.3	0.3	1	0.3	0	0.3	0	1	1	1	1	1	0	1	0
6	No	A	Yes	1	0	0.3	0.3	0.3	0.7	0.3	0	0.3									
7	No	A	Yes	1	0.3	0.3	0.3	0	1	0	1	1	1	0	0	1	0	1	1	1	0
8	No	A	No	1	0	0	0.7	0	0	0	0	0.7	1	0	0	1	0	1	1	1	0
9	No	C	No	1	0.7	1	1	0	0.7	0.3	0	0.7	1	-1	1	1	-1	1	1	1	0
10	No	B	No	1	0.3	0	0.3	0.3	0	1	0.7	0.7	1	0	-1	1	0	1	0	1	1
11	No	C		0.3	0	0.7	0	0.7	0.7	1	0.3	1	1	-1	1	-1	1	1	1	0	1
12	No	A	Yes	1	0.7	0.3	0	0	1	0.7	1	1	1	-1	1	1	-1	1	1	1	1
13	No	A	Yes	0.3	0	0.3	0.3	0	0.7	0.7	0	0.7	1	1	0	0	1	1	1	1	1
14	Yes	A	Yes	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	1	1	1	1	1	1	1	1	1
15	Yes		No	1	0	0.7	0.3	0	1	0.7	0	0	1	-1	1	0	-1	1	1	1	1
16	Yes		No	1	0.3	0.7	0.3	0	0.7	0.7	0	0.3	0	-1	1	1	0	0	1	0	1
17	Yes	A	Yes	0.7	0	0.7	0.3	0.3	0.7	0.3	0.7	0	1	1	1	1	1	1	1	1	1
18	Yes	A	No	1	0	1	0.3	0	0.3	1	0	0.7	1	0	1	0	0	1	1	0	1
19	Yes		Yes	1	0	0	0	0	0	0	0	0									
20	Yes	C	Yes	0.3	0.3	0.7	0.7	0.3	0.7	0.3	0.7	0.3	-1	1	1	-1	1	-1	0	0	-1
21	Yes	A	No	1	0.3	0.3	0	0	1	0.3	0	0.7	0	0	0	0	0	0	0	0	0
22	Yes	A	No	1	0	0.3	0.3	0	0	0.3	0.3	1	1	-1	0	0	1	1	1	1	1
23																					
24		ArchiMat	9																		
25		Neutral	10																		
26		Count Rows Populate		20	20	20	20	20	20	20	20	20	18	18	18	18	18	18	18	18	18
27		Total Score		17	3.8	9.6	6	2.5	12	9.5	6.3	10	13	-0.5	11	9	4.5	13	14	12	11
28		Count Rows ArchiMat		9	9	9	9	9	9	9	9	9	7	7	7	7	7	7	7	7	7
29		Score Rows ArchiMat		6.6	1.9	2.9	2.2	1.2	6.1	2.9	4	3.9	5	2.5	5	4	2.5	4	6	5	4
30		Count Rows Neutral		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
31		Score Rows Neutral		10	1.9	6	3.8	0.6	5.4	5.6	2	5.4	7	-2	5	6	1	8	7	7	6
32		Count Rows Group		9	9	9	9	9	9	9	9	9	8	8	8	8	8	8	8	8	8
33		Score Rows Group		7.3	1.2	4.7	2.5	0.9	4.7	3.9	2	3.3	4	0	6	2	3	4	6	4	5
34		Count Rows Non-Grou		11	11	11	11	11	11	11	11	11	10	10	10	10	10	10	10	10	10
35		Score Rows Non-Grou		9.6	2.6	4.9	3.5	1.6	7.5	5.6	4.3	7	9	-0.5	5	7	1.5	9	8	8	6
36		Count Rows Type A		10	10	10	10	10	10	10	10	10	9	9	9	9	9	9	9	9	9
37		Score Rows Type A		8.3	1.6	3.8	2.8	0.9	5.7	3.9	3.3	6.4	8	1	4	5	3	8	8	7	6
38		Count Rows Type B		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
39		Score Rows Type B		3	0.6	2	0.9	0.6	1.7	2.3	1.7	1.3	2	2	1	3	2	3	1	3	2
40			Element	A1	A2	A3	A4	A5	A6	A7	A8	A9									
41		Overall	Compreher	0.85	0.19	0.48	0.30	0.13	0.61	0.48	0.32	0.52	0.72	####	0.61	0.50	0.25	0.72	0.78	0.67	0.61
42			Utility	0.72	####	0.61	0.50	0.25	0.72	0.78	0.67	0.61									
43		ArchiMat	Compreher	0.73	0.21	0.32	0.24	0.13	0.68	0.32	0.44	0.43	0.71	0.36	0.71	0.57	0.36	0.57	0.86	0.71	0.57
44			Utility	0.71	0.36	0.71	0.57	0.36	0.57	0.86	0.71	0.57									
45		Neutral	Compreher	1.00	0.19	0.60	0.38	0.06	0.54	0.56	0.20	0.54	0.70	####	0.50	0.60	0.10	0.80	0.70	0.70	0.60
46			Utility	0.70	####	0.50	0.60	0.10	0.80	0.70	0.70	0.60									
47		Group	Compreher	0.81	0.13	0.52	0.28	0.10	0.52	0.43	0.22	0.37	0.50	0.00	0.75	0.25	0.38	0.50	0.75	0.50	0.63
48			Utility	0.50	0.00	0.75	0.25	0.38	0.50	0.75	0.50	0.63									
49		Interview	Compreher	0.87	0.24	0.45	0.32	0.15	0.68	0.51	0.39	0.64	0.90	####	0.50	0.70	0.15	0.90	0.80	0.80	0.60
50			Utility	0.90	####	0.50	0.70	0.15	0.90	0.80	0.80	0.60									
51		ArchiMat	Compreher	####	0.02	####	####	0.07	0.14	####	0.24	####									
52			Utility	0.01	0.56	0.21	####	0.26	####	0.16	0.01	####									
53		Int vs. Gro	Compreher	0.06	0.10	####	0.04	0.05	0.16	0.08	0.17	0.27									
54			Utility	0.40	####	####	0.45	####	0.40	0.05	0.30	####									
55		Type A	Compreher	0.83	0.16	0.38	0.28	0.09	0.57	0.39	0.33	0.64	0.89	0.11	0.44	0.56	0.33	0.89	0.89	0.78	0.67
56			Utility	0.89	0.11	0.44	0.56	0.33	0.89	0.89	0.78	0.67									
57		Type B	Compreher	1	0.2	0.7	0.3	0.2	0.6	0.8	0.6	0.4	0.7	0.7	0.3	1	0.7	1	0.3	1	0.7
58			Utility	0.7	0.7	0.3	1	0.7	1	0.3	1	0.7									
59				Application component	Application collaboration	Application interface	Application function	Application interaction	Application process	Application event	Application service	Data object									

Appendix L Underlying Formulae for Analysis of EA Modelling Language Effectiveness

The previous appendix shows the content of the Excel® spreadsheets used to analyse the data from the surveys. It does not however show all of the formulae used to calculate these results.

The detailed formulae used in these sheets are shown below:

L.1 Formulae for Motivation Layer

Table 74 Formulae for Motivation Comprehension Ratings

	E	F	G	H	I	J
27	ArchiMate	=COUNTIF(F$3:F$26,"=Yes")				
28	Neutral	=COUNTIF(F$3:F$26,"=No")				
29	Count Rows Populat		=COUNT(G3:G25)	=COUNT(H3:H25)	=COUNT(I3:I25)	=COUNT(J3:J25)
30	Total Score		=SUM(G3:G25)	=SUM(H3:H25)	=SUM(I3:I25)	=SUM(J3:J25)
31	Count Rows ArchiM		=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")
32	Score Rows ArchiMa		=SUMIF($F$3:$F$25,"Yes",G$3:G$25)	=SUMIF($F$3:$F$25,"Yes",H$3:H$25)	=SUMIF($F$3:$F$25,"Yes",I$3:I$25)	=SUMIF($F$3:$F$25,"Yes",J$3:J$25)
33	Count Rows Neutral		=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")
34	Score Rows Neutral		=SUMIF($F$3:$F$25,"No",G$3:G$25)	=SUMIF($F$3:$F$25,"No",H$3:H$25)	=SUMIF($F$3:$F$25,"No",I$3:I$25)	=SUMIF($F$3:$F$25,"No",J$3:J$25)
35	Count Rows Group		=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")
36	Score Rows Group		=SUMIF($D$3:$D$25,"Yes",G$3:G$25)	=SUMIF($D$3:$D$25,"Yes",H$3:H$25)	=SUMIF($D$3:$D$25,"Yes",I$3:I$25)	=SUMIF($D$3:$D$25,"Yes",J$3:J$25)
37	Count Rows Non-Gr		=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")
38	Score Rows Non-Grc		=SUMIF($D$3:$D$25,"No",G$3:G$25)	=SUMIF($D$3:$D$25,"No",H$3:H$25)	=SUMIF($D$3:$D$25,"No",I$3:I$25)	=SUMIF($D$3:$D$25,"No",J$3:J$25)
39	Count Rows Type A		=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")
40	Score Rows Type A		=SUMIF($E$3:$E$25,"A",G$3:G$25)	=SUMIF($E$3:$E$25,"A",H$3:H$25)	=SUMIF($E$3:$E$25,"A",I$3:I$25)	=SUMIF($E$3:$E$25,"A",J$3:J$25)
41	Count Rows Type B		=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
42	Score Rows Type B		=SUMIF($E$3:$E$25,"B",G$3:G$25)	=SUMIF($E$3:$E$25,"B",H$3:H$25)	=SUMIF($E$3:$E$25,"B",I$3:I$25)	=SUMIF($E$3:$E$25,"B",J$3:J$25)

	E	K	L	M	N	O	P
27	ArchiMate						
28	Neutral						
29	Count Rows Populat	=COUNT(K3:K25)	=COUNT(L3:L25)	=COUNT(M3:M25)	=COUNT(N3:N25)	=COUNT(O3:O25)	=COUNT(P3:P25)
30	Total Score	=SUM(K3:K25)	=SUM(L3:L25)	=SUM(M3:M25)	=SUM(N3:N25)	=SUM(O3:O25)	=SUM(P3:P25)
31	Count Rows ArchiM	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")
32	Score Rows ArchiMa	=SUMIF($F$3:$F$25,"Yes",K$3:K$25)	=SUMIF($F$3:$F$25,"Yes",L$3:L$25)	=SUMIF($F$3:$F$25,"Yes",M$3:M$25)	=SUMIF($F$3:$F$25,"Yes",N$3:N$25)	=SUMIF($F$3:$F$25,"Yes",O$3:O$25)	=SUMIF($F$3:$F$25,"Yes",P$3:P$25)
33	Count Rows Neutral	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")
34	Score Rows Neutral	=SUMIF($F$3:$F$25,"No",K$3:K$25)	=SUMIF($F$3:$F$25,"No",L$3:L$25)	=SUMIF($F$3:$F$25,"No",M$3:M$25)	=SUMIF($F$3:$F$25,"No",N$3:N$25)	=SUMIF($F$3:$F$25,"No",O$3:O$25)	=SUMIF($F$3:$F$25,"No",P$3:P$25)
35	Count Rows Group	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")
36	Score Rows Group	=SUMIF($D$3:$D$25,"Yes",K$3:K$25)	=SUMIF($D$3:$D$25,"Yes",L$3:L$25)	=SUMIF($D$3:$D$25,"Yes",M$3:M$25)	=SUMIF($D$3:$D$25,"Yes",N$3:N$25)	=SUMIF($D$3:$D$25,"Yes",O$3:O$25)	=SUMIF($D$3:$D$25,"Yes",P$3:P$25)
37	Count Rows Non-Gr	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")
38	Score Rows Non-Grc	=SUMIF($D$3:$D$25,"No",K$3:K$25)	=SUMIF($D$3:$D$25,"No",L$3:L$25)	=SUMIF($D$3:$D$25,"No",M$3:M$25)	=SUMIF($D$3:$D$25,"No",N$3:N$25)	=SUMIF($D$3:$D$25,"No",O$3:O$25)	=SUMIF($D$3:$D$25,"No",P$3:P$25)
39	Count Rows Type A	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")
40	Score Rows Type A	=SUMIF($E$3:$E$25,"A",K$3:K$25)	=SUMIF($E$3:$E$25,"A",L$3:L$25)	=SUMIF($E$3:$E$25,"A",M$3:M$25)	=SUMIF($E$3:$E$25,"A",N$3:N$25)	=SUMIF($E$3:$E$25,"A",O$3:O$25)	=SUMIF($E$3:$E$25,"A",P$3:P$25)
41	Count Rows Type B	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
42	Score Rows Type B	=SUMIF($E$3:$E$25,"B",K$3:K$25)	=SUMIF($E$3:$E$25,"B",L$3:L$25)	=SUMIF($E$3:$E$25,"B",M$3:M$25)	=SUMIF($E$3:$E$25,"B",N$3:N$25)	=SUMIF($E$3:$E$25,"B",O$3:O$25)	=SUMIF($E$3:$E$25,"B",P$3:P$25)

Table 75 Formulae for Motivation Utility Ratings

	E	F	Q	R	S	T
27	ArchiMate	=COUNTIF(F$3:F$26,"=Yes")				
28	Neutral	=COUNTIF(F$3:F$26,"=No")				
29	Count Rows Populat		=COUNT(Q3:Q25)	=COUNT(R3:R25)	=COUNT(S3:S25)	=COUNT(T3:T25)
30	Total Score		=SUM(Q3:Q25)	=SUM(R3:R25)	=SUM(S3:S25)	=SUM(T3:T25)
31	Count Rows ArchiM		=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")
32	Score Rows ArchiMa		=SUMIF($F$3:$F$25,"Yes",Q$3:Q$25)	=SUMIF($F$3:$F$25,"Yes",R$3:R$25)	=SUMIF($F$3:$F$25,"Yes",S$3:S$25)	=SUMIF($F$3:$F$25,"Yes",T$3:T$25)
33	Count Rows Neutral		=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")
34	Score Rows Neutral		=SUMIF($F$3:$F$25,"No",Q$3:Q$25)	=SUMIF($F$3:$F$25,"No",R$3:R$25)	=SUMIF($F$3:$F$25,"No",S$3:S$25)	=SUMIF($F$3:$F$25,"No",T$3:T$25)
35	Count Rows Group		=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")
36	Score Rows Group		=SUMIF($D$3:$D$25,"Yes",Q$3:Q$25)	=SUMIF($D$3:$D$25,"Yes",R$3:R$25)	=SUMIF($D$3:$D$25,"Yes",S$3:S$25)	=SUMIF($D$3:$D$25,"Yes",T$3:T$25)
37	Count Rows Non-Gr		=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")
38	Score Rows Non-Grc		=SUMIF($D$3:$D$25,"No",Q$3:Q$25)	=SUMIF($D$3:$D$25,"No",R$3:R$25)	=SUMIF($D$3:$D$25,"No",S$3:S$25)	=SUMIF($D$3:$D$25,"No",T$3:T$25)
39	Count Rows Type A		=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")
40	Score Rows Type A		=SUMIF($E$3:$E$25,"A",Q$3:Q$25)	=SUMIF($E$3:$E$25,"A",R$3:R$25)	=SUMIF($E$3:$E$25,"A",S$3:S$25)	=SUMIF($E$3:$E$25,"A",T$3:T$25)
41	Count Rows Type B		=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
42	Score Rows Type B		=SUMIF($E$3:$E$25,"B",Q$3:Q$25)	=SUMIF($E$3:$E$25,"B",R$3:R$25)	=SUMIF($E$3:$E$25,"B",S$3:S$25)	=SUMIF($E$3:$E$25,"B",T$3:T$25)

	E	U	V	W	X	Y	Z
27	ArchiMate						
28	Neutral						
29	Count Rows Populat	=COUNT(U3:U25)	=COUNT(V3:V25)	=COUNT(W3:W25)	=COUNT(X3:X25)	=COUNT(Y3:Y25)	=COUNT(Z3:Z25)
30	Total Score	=SUM(U3:U25)	=SUM(V3:V25)	=SUM(W3:W25)	=SUM(X3:X25)	=SUM(Y3:Y25)	=SUM(Z3:Z25)
31	Count Rows ArchiM	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")	=COUNTIF($F$3:$F$25,"Yes")
32	Score Rows ArchiMa	=SUMIF($F$3:$F$25,"Yes",U$3:U$25)	=SUMIF($F$3:$F$25,"Yes",V$3:V$25)	=SUMIF($F$3:$F$25,"Yes",W$3:W$25)	=SUMIF($F$3:$F$25,"Yes",X$3:X$25)	=SUMIF($F$3:$F$25,"Yes",Y$3:Y$25)	=SUMIF($F$3:$F$25,"Yes",Z$3:Z$25)
33	Count Rows Neutral	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")	=COUNTIF($F$3:$F$25,"No")
34	Score Rows Neutral	=SUMIF($F$3:$F$25,"No",U$3:U$25)	=SUMIF($F$3:$F$25,"No",V$3:V$25)	=SUMIF($F$3:$F$25,"No",W$3:W$25)	=SUMIF($F$3:$F$25,"No",X$3:X$25)	=SUMIF($F$3:$F$25,"No",Y$3:Y$25)	=SUMIF($F$3:$F$25,"No",Z$3:Z$25)
35	Count Rows Group	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")	=COUNTIF($D$3:$D$25,"Yes")
36	Score Rows Group	=SUMIF($D$3:$D$25,"Yes",U$3:U$25)	=SUMIF($D$3:$D$25,"Yes",V$3:V$25)	=SUMIF($D$3:$D$25,"Yes",W$3:W$25)	=SUMIF($D$3:$D$25,"Yes",X$3:X$25)	=SUMIF($D$3:$D$25,"Yes",Y$3:Y$25)	=SUMIF($D$3:$D$25,"Yes",Z$3:Z$25)
37	Count Rows Non-Gr	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")	=COUNTIF($D$3:$D$25,"No")
38	Score Rows Non-Grc	=SUMIF($D$3:$D$25,"No",U$3:U$25)	=SUMIF($D$3:$D$25,"No",V$3:V$25)	=SUMIF($D$3:$D$25,"No",W$3:W$25)	=SUMIF($D$3:$D$25,"No",X$3:X$25)	=SUMIF($D$3:$D$25,"No",Y$3:Y$25)	=SUMIF($D$3:$D$25,"No",Z$3:Z$25)
39	Count Rows Type A	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")	=COUNTIF($E$3:$E$25,"A")
40	Score Rows Type A	=SUMIF($E$3:$E$25,"A",U$3:U$25)	=SUMIF($E$3:$E$25,"A",V$3:V$25)	=SUMIF($E$3:$E$25,"A",W$3:W$25)	=SUMIF($E$3:$E$25,"A",X$3:X$25)	=SUMIF($E$3:$E$25,"A",Y$3:Y$25)	=SUMIF($E$3:$E$25,"A",Z$3:Z$25)
41	Count Rows Type B	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
42	Score Rows Type B	=SUMIF($E$3:$E$25,"B",U$3:U$25)	=SUMIF($E$3:$E$25,"B",V$3:V$25)	=SUMIF($E$3:$E$25,"B",W$3:W$25)	=SUMIF($E$3:$E$25,"B",X$3:X$25)	=SUMIF($E$3:$E$25,"B",Y$3:Y$25)	=SUMIF($E$3:$E$25,"B",Z$3:Z$25)

Table 76 Formulae for Motivation Comprehension vs. Utility Charts (left half)

	E	F	G	H	I	J	K	L	M	N	O	P
43		Element	=G2	=H2	=I2	=J2	=K2	=L2	=M2	=N2	=O2	=P2
44	Overall	Comprehension	=G30/G29	=H30/H29	=I30/I29	=J30/J29	=K30/K29	=L30/L29	=M30/M29	=N30/N29	=O30/O29	=P30/P29
45		Utility	=Q44	=R44	=S44	=T44	=U44	=V44	=W44	=X44	=Y44	=Z44
46	ArchiMate	Comprehension	=G32/G31	=H32/H31	=I32/I31	=J32/J31	=K32/K31	=L32/L31	=M32/M31	=N32/N31	=O32/O31	=P32/P31
47		Utility	=Q46	=R46	=S46	=T46	=U46	=V46	=W46	=X46	=Y46	=Z46
48	Neutral	Comprehension	=G34/G33	=H34/H33	=I34/I33	=J34/J33	=K34/K33	=L34/L33	=M34/M33	=N34/N33	=O34/O33	=P34/P33
49		Utility	=Q48	=R48	=S48	=T48	=U48	=V48	=W48	=X48	=Y48	=Z48
50	Group	Comprehension	=G36/G35	=H36/H35	=I36/I35	=J36/J35	=K36/K35	=L36/L35	=M36/M35	=N36/N35	=O36/O35	=P36/P35
51		Utility	=Q50	=R50	=S50	=T50	=U50	=V50	=W50	=X50	=Y50	=Z50
52	Interview	Comprehension	=G38/G37	=H38/H37	=I38/I37	=J38/J37	=K38/K37	=L38/L37	=M38/M37	=N38/N37	=O38/O37	=P38/P37
53		Utility	=Q52	=R52	=S52	=T52	=U52	=V52	=W52	=X52	=Y52	=Z52
54	ArchiMate vs. Neutr	Comprehension	=G46-G48	=H46-H48	=I46-I48	=J46-J48	=K46-K48	=L46-L48	=M46-M48	=N46-N48	=O46-O48	=P46-P48
55		Utility	=G47-G49	=H47-H49	=I47-I49	=J47-J49	=K47-K49	=L47-L49	=M47-M49	=N47-N49	=O47-O49	=P47-P49
56	Int vs. Group	Comprehension	=G52-G50	=H52-H50	=I52-I50	=J52-J50	=K52-K50	=L52-L50	=M52-M50	=N52-N50	=O52-O50	=P52-P50
57		Utility	=G53-G51	=H53-H51	=I53-I51	=J53-J51	=K53-K51	=L53-L51	=M53-M51	=N53-N51	=O53-O51	=P53-P51
58	Type A	Comprehension	=G40/G39	=H40/H39	=I40/I39	=J40/J39	=K40/K39	=L40/L39	=M40/M39	=N40/N39	=O40/O39	=P40/P39
59		Utility	=Q58	=R58	=S58	=T58	=U58	=V58	=W58	=X58	=Y58	=Z58
60	Type B	Comprehension	=G42/G41	=H42/H41	=I42/I41	=J42/J41	=K42/K41	=L42/L41	=M42/M41	=N42/N41	=O42/O41	=P42/P41
61		Utility	=Q60	=R60	=S60	=T60	=U60	=V60	=W60	=X60	=Y60	=Z60
62			Stakeholder	Value	Meaning	Driver	Assessment	Goal	Outcome	Principle	Requirement	Constraint

Table 77 Formulae for Motivation Comprehension vs. Utility Charts (right half)

	E	F	Q	R	S	T	U	V	W	X	Y	Z
43		Element										
44	Overall	Comprehension	=Q30/Q29	=R30/R29	=S30/S29	=T30/T29	=U30/U29	=V30/V29	=W30/W29	=X30/X29	=Y30/Y29	=Z30/Z29
45		Utility										
46	ArchiMate	Comprehension	=Q32/Q31	=R32/R31	=S32/S31	=T32/T31	=U32/U31	=V32/V31	=W32/W31	=X32/X31	=Y32/Y31	=Z32/Z31
47		Utility										
48	Neutral	Comprehension	=Q34/Q33	=R34/R33	=S34/S33	=T34/T33	=U34/U33	=V34/V33	=W34/W33	=X34/X33	=Y34/Y33	=Z34/Z33
49		Utility										
50	Group	Comprehension	=Q36/Q35	=R36/R35	=S36/S35	=T36/T35	=U36/U35	=V36/V35	=W36/W35	=X36/X35	=Y36/Y35	=Z36/Z35
51		Utility										
52	Interview	Comprehension	=Q38/Q37	=R38/R37	=S38/S37	=T38/T37	=U38/U37	=V38/V37	=W38/W37	=X38/X37	=Y38/Y37	=Z38/Z37
53		Utility										
54	ArchiMate vs. Neutr	Comprehension										
55		Utility										
56	Int vs. Group	Comprehension										
57		Utility										
58	Type A	Comprehension	=Q40/Q39	=R40/R39	=S40/S39	=T40/T39	=U40/U39	=V40/V39	=W40/W39	=X40/X39	=Y40/Y39	=Z40/Z39
59		Utility										
60	Type B	Comprehension	=Q42/Q41	=R42/R41	=S42/S41	=T42/T41	=U42/U41	=V42/V41	=W42/W41	=X42/X41	=Y42/Y41	=Z42/Z41
61		Utility										

L.2 Formulae for Business Layer

Table 78 Formulae for Business Comprehension Ratings

	E	F	G	H	I	J	K	L	M
29	ArchiMate	=COUNTIF(F$3:F$26,"=Yes")							
30	Neutral	=COUNTIF(F$3:F$27,"=No")							
31	Count Rows Populat		=COUNT(G3:G27)	=COUNT(H3:H27)	=COUNT(I3:I27)	=COUNT(J3:J27)	=COUNT(K3:K27)	=COUNT(L3:L27)	=COUNT(M3:M27)
32	Total Score		=SUM(G3:G27)	=SUM(H3:H27)	=SUM(I3:I27)	=SUM(J3:J27)	=SUM(K3:K27)	=SUM(L3:L27)	=SUM(M3:M27)
33	Count Rows ArchiM		=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes"	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")
34	Score Rows ArchiMa		=SUMIF($F$3:$F$27,"Yes",G$3:G$2	=SUMIF($F$3:$F$27,"Yes",H	=SUMIF($F$3:$F$27,"Yes",I$3:I$25)	=SUMIF($F$3:$F$27,"Yes",J$3:J$2	=SUMIF($F$3:$F$27,"Yes",K$3:K	=SUMIF($F$3:$F$27,"Yes",L$3:L$	=SUMIF($F$3:$F$27,"Yes",M$3:M$25)
35	Count Rows Neutral		=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")
36	Score Rows Neutral		=SUMIF($F$3:$F$27,"No",G$3:G$2	=SUMIF($F$3:$F$27,"No",H$	=SUMIF($F$3:$F$27,"No",I$3:I$27)	=SUMIF($F$3:$F$27,"No",J$3:J$27	=SUMIF($F$3:$F$27,"No",K$3:K$	=SUMIF($F$3:$F$27,"No",L$3:L$	=SUMIF($F$3:$F$27,"No",M$3:M$27)
37	Count Rows Group		=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")
38	Score Rows Group		=SUMIF($D$3:$D$27,"Yes",G$3:G$	=SUMIF($D$3:$D$27,"Yes",H	=SUMIF($D$3:$D$27,"Yes",I$3:I$27)	=SUMIF($D$3:$D$27,"Yes",J$3:J$	=SUMIF($D$3:$D$27,"Yes",K$3:	=SUMIF($D$3:$D$27,"Yes",L$3:L	=SUMIF($D$3:$D$27,"Yes",M$3:M$27)
39	Count Rows Non-Gr		=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No"	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")
40	Score Rows Non-Gr		=SUMIF($D$3:$D$27,"No",G$3:G$2	=SUMIF($D$3:$D$27,"No",H	=SUMIF($D$3:$D$27,"No",I$3:I$27)	=SUMIF($D$3:$D$27,"No",J$3:J$2	=SUMIF($D$3:$D$27,"No",K$3:K	=SUMIF($D$3:$D$27,"No",L$3:L$	=SUMIF($D$3:$D$27,"No",M$3:M$27)
41	Count Rows Type A		=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")
42	Score Rows Type A		=SUMIF($E$3:$E$27,"A",G$3:G$27)	=SUMIF($E$3:$E$27,"A",H$3	=SUMIF($E$3:$E$27,"A",I$3:I$27)	=SUMIF($E$3:$E$27,"A",J$3:J$27)	=SUMIF($E$3:$E$27,"A",K$3:K$	=SUMIF($E$3:$E$27,"A",L$3:L$2	=SUMIF($E$3:$E$27,"A",M$3:M$27)
43	Count Rows Type B		=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
44	Score Rows Type B		=SUMIF($E$3:$E$25,"B",G$3:G$25)	=SUMIF($E$3:$E$25,"B",H$3	=SUMIF($E$3:$E$25,"B",I$3:I$25)	=SUMIF($E$3:$E$25,"B",J$3:J$25)	=SUMIF($E$3:$E$25,"B",K$3:K$	=SUMIF($E$3:$E$25,"B",L$3:L$2	=SUMIF($E$3:$E$25,"B",M$3:M$25)

	E	N	O	P	Q	R	S
29	ArchiMate						
30	Neutral						
31	Count Rows Populat	=COUNT(N3:N27)	=COUNT(O3:O27)	=COUNT(P3:P27)	=COUNT(Q3:Q27)	=COUNT(R3:R27)	=COUNT(S3:S27)
32	Total Score	=SUM(N3:N27)	=SUM(O3:O27)	=SUM(P3:P27)	=SUM(Q3:Q27)	=SUM(R3:R27)	=SUM(S3:S27)
33	Count Rows ArchiM	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")
34	Score Rows ArchiMa	=SUMIF($F$3:$F$27,"Yes",N$3:N$25)	=SUMIF($F$3:$F$27,"Yes",O$3:O$25)	=SUMIF($F$3:$F$27,"Yes",P$3:P$25)	=SUMIF($F$3:$F$27,"Yes",Q$3:Q$25)	=SUMIF($F$3:$F$27,"Yes",R$3:R$25)	=SUMIF($F$3:$F$27,"Yes",S$3:S$25)
35	Count Rows Neutral	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")
36	Score Rows Neutral	=SUMIF($F$3:$F$27,"No",N$3:N$27)	=SUMIF($F$3:$F$27,"No",O$3:O$27)	=SUMIF($F$3:$F$27,"No",P$3:P$27)	=SUMIF($F$3:$F$27,"No",Q$3:Q$27)	=SUMIF($F$3:$F$27,"No",R$3:R$27)	=SUMIF($F$3:$F$27,"No",S$3:S$27)
37	Count Rows Group	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")
38	Score Rows Group	=SUMIF($D$3:$D$27,"Yes",N$3:N$27)	=SUMIF($D$3:$D$27,"Yes",O$3:O$27)	=SUMIF($D$3:$D$27,"Yes",P$3:P$27)	=SUMIF($D$3:$D$27,"Yes",Q$3:Q$27)	=SUMIF($D$3:$D$27,"Yes",R$3:R$27)	=SUMIF($D$3:$D$27,"Yes",S$3:S$27)
39	Count Rows Non-Gr	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")
40	Score Rows Non-Gr	=SUMIF($D$3:$D$27,"No",N$3:N$27)	=SUMIF($D$3:$D$27,"No",O$3:O$27)	=SUMIF($D$3:$D$27,"No",P$3:P$27)	=SUMIF($D$3:$D$27,"No",Q$3:Q$27)	=SUMIF($D$3:$D$27,"No",R$3:R$27)	=SUMIF($D$3:$D$27,"No",S$3:S$27)
41	Count Rows Type A	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")
42	Score Rows Type A	=SUMIF($E$3:$E$27,"A",N$3:N$27)	=SUMIF($E$3:$E$27,"A",O$3:O$27)	=SUMIF($E$3:$E$27,"A",P$3:P$27)	=SUMIF($E$3:$E$27,"A",Q$3:Q$27)	=SUMIF($E$3:$E$27,"A",R$3:R$27)	=SUMIF($E$3:$E$27,"A",S$3:S$27)
43	Count Rows Type B	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
44	Score Rows Type B	=SUMIF($E$3:$E$25,"B",N$3:N$25)	=SUMIF($E$3:$E$25,"B",O$3:O$25)	=SUMIF($E$3:$E$25,"B",P$3:P$25)	=SUMIF($E$3:$E$25,"B",Q$3:Q$25)	=SUMIF($E$3:$E$25,"B",R$3:R$25)	=SUMIF($E$3:$E$25,"B",S$3:S$25)

Table 79 Formulae for Business Utility Ratings

	E	F	T	U	V	W
29	ArchiMate	=COUNTIF(F$3:F$26,"=Yes")				
30	Neutral	=COUNTIF(F$3:F$27,"=No")				
31	Count Rows Pop		=COUNT(T3:T27)	=COUNT(U3:U27)	=COUNT(V3:V27)	=COUNT(W3:W27)
32	Total Score		=SUM(T3:T27)	=SUM(U3:U27)	=SUM(V3:V27)	=SUM(W3:W27)
33	Count Rows Arch		=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")
34	Score Rows Arch		=SUMIF($F$3:$F$27,"Yes",T$3:T$25)	=SUMIF($F$3:$F$27,"Yes",U$3:U$25)	=SUMIF($F$3:$F$27,"Yes",V$3:V$25)	=SUMIF($F$3:$F$27,"Yes",W$3:W$25)
35	Count Rows Neu		=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")
36	Score Rows Neut		=SUMIF($F$3:$F$27,"No",T$3:T$27)	=SUMIF($F$3:$F$27,"No",U$3:U$27)	=SUMIF($F$3:$F$27,"No",V$3:V$27)	=SUMIF($F$3:$F$27,"No",W$3:W$27)
37	Count Rows Grou		=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")
38	Score Rows Grou		=SUMIF($D$3:$D$27,"Yes",T$3:T$27	=SUMIF($D$3:$D$27,"Yes",U$3:U$27	=SUMIF($D$3:$D$27,"Yes",V$3:V$27	=SUMIF($D$3:$D$27,"Yes",W$3:W$27
39	Count Rows Non		=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")
40	Score Rows Non		=SUMIF($D$3:$D$27,"No",T$3:T$27)	=SUMIF($D$3:$D$27,"No",U$3:U$27)	=SUMIF($D$3:$D$27,"No",V$3:V$27)	=SUMIF($D$3:$D$27,"No",W$3:W$27)
41	Count Rows Type		=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")
42	Score Rows Type		=SUMIF($E$3:$E$27,"A",T$3:T$27)	=SUMIF($E$3:$E$27,"A",U$3:U$27)	=SUMIF($E$3:$E$27,"A",V$3:V$27)	=SUMIF($E$3:$E$27,"A",W$3:W$27)
43	Count Rows Type		=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
44	Score Rows Type		=SUMIF($E$3:$E$25,"B",T$3:T$25)	=SUMIF($E$3:$E$25,"B",U$3:U$25)	=SUMIF($E$3:$E$25,"B",V$3:V$25)	=SUMIF($E$3:$E$25,"B",W$3:W$25)
45		Element	=T2	=U2	=V2	=W2

	E	X	Y	Z	AA	AB
29	ArchiMate					
30	Neutral					
31	Count Rows Pop	=COUNT(X3:X27)	=COUNT(Y3:Y27)	=COUNT(Z3:Z27)	=COUNT(AA3:AA27)	=COUNT(AB3:AB27)
32	Total Score	=SUM(X3:X27)	=SUM(Y3:Y27)	=SUM(Z3:Z27)	=SUM(AA3:AA27)	=SUM(AB3:AB27)
33	Count Rows Arch	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")
34	Score Rows Arch	=SUMIF($F$3:$F$27,"Yes",X$3:X$25)	=SUMIF($F$3:$F$27,"Yes",Y$3:Y$25)	=SUMIF($F$3:$F$27,"Yes",Z$3:Z$25)	=SUMIF($F$3:$F$27,"Yes",AA$3:AA$2	=SUMIF($F$3:$F$27,"Yes",AB$3:AB$25)
35	Count Rows Neu	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")
36	Score Rows Neut	=SUMIF($F$3:$F$27,"No",X$3:X$27)	=SUMIF($F$3:$F$27,"No",Y$3:Y$27)	=SUMIF($F$3:$F$27,"No",Z$3:Z$27)	=SUMIF($F$3:$F$27,"No",AA$3:AA$27	=SUMIF($F$3:$F$27,"No",AB$3:AB$27)
37	Count Rows Grou	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")
38	Score Rows Grou	=SUMIF($D$3:$D$27,"Yes",X$3:X$27	=SUMIF($D$3:$D$27,"Yes",Y$3:Y$27	=SUMIF($D$3:$D$27,"Yes",Z$3:Z$27	=SUMIF($D$3:$D$27,"Yes",AA$3:AA$2	=SUMIF($D$3:$D$27,"Yes",AB$3:AB$27)
39	Count Rows Non	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")
40	Score Rows Non	=SUMIF($D$3:$D$27,"No",X$3:X$27)	=SUMIF($D$3:$D$27,"No",Y$3:Y$27)	=SUMIF($D$3:$D$27,"No",Z$3:Z$27)	=SUMIF($D$3:$D$27,"No",AA$3:AA$27	=SUMIF($D$3:$D$27,"No",AB$3:AB$27)
41	Count Rows Type	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")
42	Score Rows Type	=SUMIF($E$3:$E$27,"A",X$3:X$27)	=SUMIF($E$3:$E$27,"A",Y$3:Y$27)	=SUMIF($E$3:$E$27,"A",Z$3:Z$27)	=SUMIF($E$3:$E$27,"A",AA$3:AA$27)	=SUMIF($E$3:$E$27,"A",AB$3:AB$27)
43	Count Rows Type	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
44	Score Rows Type	=SUMIF($E$3:$E$25,"B",X$3:X$25)	=SUMIF($E$3:$E$25,"B",Y$3:Y$25)	=SUMIF($E$3:$E$25,"B",Z$3:Z$25)	=SUMIF($E$3:$E$25,"B",AA$3:AA$25)	=SUMIF($E$3:$E$25,"B",AB$3:AB$25)
45		=X2	=Y2	=Z2	=AA2	=AB2

	E	AC	AD	AE	AF
29	ArchiMate				
30	Neutral				
31	Count Rows Pop	=COUNT(AC3:AC27)	=COUNT(AD3:AD27)	=COUNT(AE3:AE27)	=COUNT(AF3:AF27)
32	Total Score	=SUM(AC3:AC27)	=SUM(AD3:AD27)	=SUM(AE3:AE27)	=SUM(AF3:AF27)
33	Count Rows Arch	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")	=COUNTIF($F$3:$F$27,"Yes")
34	Score Rows Arch	=SUMIF($F$3:$F$27,"Yes",AC$3:AC$25)	=SUMIF($F$3:$F$27,"Yes",AD$3:AD$25)	=SUMIF($F$3:$F$27,"Yes",AE$3:AE$25)	=SUMIF($F$3:$F$27,"Yes",AF$3:AF$25)
35	Count Rows Neu	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")	=COUNTIF($F$3:$F$27,"No")
36	Score Rows Neut	=SUMIF($F$3:$F$27,"No",AC$3:AC$27)	=SUMIF($F$3:$F$27,"No",AD$3:AD$27)	=SUMIF($F$3:$F$27,"No",AE$3:AE$27)	=SUMIF($F$3:$F$27,"No",AF$3:AF$27)
37	Count Rows Grou	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")	=COUNTIF($D$3:$D$27,"Yes")
38	Score Rows Grou	=SUMIF($D$3:$D$27,"Yes",AC$3:AC$27)	=SUMIF($D$3:$D$27,"Yes",AD$3:AD$27)	=SUMIF($D$3:$D$27,"Yes",AE$3:AE$27	=SUMIF($D$3:$D$27,"Yes",AF$3:AF$27
39	Count Rows Non	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")	=COUNTIF($D$3:$D$27,"No")
40	Score Rows Non	=SUMIF($D$3:$D$27,"No",AC$3:AC$27)	=SUMIF($D$3:$D$27,"No",AD$3:AD$27)	=SUMIF($D$3:$D$27,"No",AE$3:AE$27)	=SUMIF($D$3:$D$27,"No",AF$3:AF$27)
41	Count Rows Type	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")	=COUNTIF($E$3:$E$27,"A")
42	Score Rows Type	=SUMIF($E$3:$E$27,"A",AC$3:AC$27)	=SUMIF($E$3:$E$27,"A",AD$3:AD$27)	=SUMIF($E$3:$E$27,"A",AE$3:AE$27)	=SUMIF($E$3:$E$27,"A",AF$3:AF$27)
43	Count Rows Type	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")	=COUNTIF($E$3:$E$25,"B")
44	Score Rows Type	=SUMIF($E$3:$E$25,"B",AC$3:AC$25)	=SUMIF($E$3:$E$25,"B",AD$3:AD$25)	=SUMIF($E$3:$E$25,"B",AE$3:AE$25)	=SUMIF($E$3:$E$25,"B",AF$3:AF$25)
45		=AC2	=AD2	=AE2	=AF2

Table 80 Formulae for Business Comprehension vs. Utility Charts (left)

	E	F	G	H	I	J	K	L	M
45		Element	=G2	=H2	=I2	=J2	=K2	=L2	=M2
46	Overall	Comprehension	=G32/G31	=H32/H31	=I32/I31	=J32/J31	=K32/K31	=L32/L31	=M32/M31
47		Utility	=T46	=U46	=V46	=W46	=X46	=Y46	=Z46
48	ArchiMate	Comprehension	=G34/G33	=H34/H33	=I34/I33	=J34/J33	=K34/K33	=L34/L33	=M34/M33
49		Utility	=T48	=U48	=V48	=W48	=X48	=Y48	=Z48
50	Neutral	Comprehension	=G36/G35	=H36/H35	=I36/I35	=J36/J35	=K36/K35	=L36/L35	=M36/M35
51		Utility	=T50	=U50	=V50	=W50	=X50	=Y50	=Z50
52	Group	Comprehension	=G38/G37	=H38/H37	=I38/I37	=J38/J37	=K38/K37	=L38/L37	=M38/M37
53		Utility	=T52	=U52	=V52	=W52	=X52	=Y52	=Z52
54	Interview	Comprehension	=G40/G39	=H40/H39	=I40/I39	=J40/J39	=K40/K39	=L40/L39	=M40/M39
55		Utility	=T54	=U54	=V54	=W54	=X54	=Y54	=Z54
56	ArchiMate vs. Neut	Comprehension	=G48-G50	=H48-H50	=I48-I50	=J48-J50	=K48-K50	=L48-L50	=M48-M50
57		Utility	=G49-G51	=H49-H51	=I49-I51	=J49-J51	=K49-K51	=L49-L51	=M49-M51
58	Int vs. Group	Comprehension	=G54-G52	=H54-H52	=I54-I52	=J54-J52	=K54-K52	=L54-L52	=M54-M52
59		Utility	=G55-G53	=H55-H53	=I55-I53	=J55-J53	=K55-K53	=L55-L53	=M55-M53
60	Type A	Comprehension	=G42/G41	=H42/H41	=I42/I41	=J42/J41	=K42/K41	=L42/L41	=M42/M41
61		Utility	=T60	=U60	=V60	=W60	=X60	=Y60	=Z60
62	Type B	Comprehension	=G44/G43	=H44/H43	=I44/I43	=J44/J43	=K44/K43	=L44/L43	=M44/M43
63		Utility	=T62	=U62	=V62	=W62	=X62	=Y62	=Z62
64			Business role	Business actor	Business collaboration	Business interface	Business process	Business function	Business interaction

	E	N	O	P	Q	R	S
45		=N2	=O2	=P2	=Q2	=R2	=S2
46	Overall	=N32/N31	=O32/O31	=P32/P31	=Q32/Q31	=R32/R31	=S32/S31
47		=AA46	=AB46	=AC46	=AD46	=AE46	=AF46
48	ArchiMate	=N34/N33	=O34/O33	=P34/P33	=Q34/Q33	=R34/R33	=S34/S33
49		=AA48	=AB48	=AC48	=AD48	=AE48	=AF48
50	Neutral	=N36/N35	=O36/O35	=P36/P35	=Q36/Q35	=R36/R35	=S36/S35
51		=AA50	=AB50	=AC50	=AD50	=AE50	=AF50
52	Group	=N38/N37	=O38/O37	=P38/P37	=Q38/Q37	=R38/R37	=S38/S37
53		=AA52	=AB52	=AC52	=AD52	=AE52	=AF52
54	Interview	=N40/N39	=O40/O39	=P40/P39	=Q40/Q39	=R40/R39	=S40/S39
55		=AA54	=AB54	=AC54	=AD54	=AE54	=AF54
56	ArchiMate vs. Neut	=N48-N50	=O48-O50	=P48-P50	=Q48-Q50	=R48-R50	=S48-S50
57		=N49-N51	=O49-O51	=P49-P51	=Q49-Q51	=R49-R51	=S49-S51
58	Int vs. Group	=N54-N52	=O54-O52	=P54-P52	=Q54-Q52	=R54-R52	=S54-S52
59		=N55-N53	=O55-O53	=P55-P53	=Q55-Q53	=R55-R53	=S55-S53
60	Type A	=N42/N41	=O42/O41	=P42/P41	=Q42/Q41	=R42/R41	=S42/S41
61		=AA60	=AB60	=AC60	=AD60	=AE60	=AF60
62	Type B	=N44/N43	=O44/O43	=P44/P43	=Q44/Q43	=R44/R43	=S44/S43
63		=AA62	=AB62	=AC62	=AD62	=AE62	=AF62
64		Business event	Business service	Business object	Contract	Representation	Product

Table 81 Formulae for Business Comprehension vs. Utility Charts (right)

	E	F	T	U	V	W
45		Element	=T2	=U2	=V2	=W2
46	Overall	Comprehension	=T32/T31	=U32/U31	=V32/V31	=W32/W31
47		Utility				
48	ArchiMate	Comprehension	=T34/T33	=U34/U33	=V34/V33	=W34/W33
49		Utility				
50	Neutral	Comprehension	=T36/T35	=U36/U35	=V36/V35	=W36/W35
51		Utility				
52	Group	Comprehension	=T38/T37	=U38/U37	=V38/V37	=W38/W37
53		Utility				
54	Interview	Comprehension	=T40/T39	=U40/U39	=V40/V39	=W40/W39
55		Utility				
56	ArchiMate vs. Ne	Comprehension				
57		Utility				
58	Int vs. Group	Comprehension				
59		Utility				
60	Type A	Comprehension	=T42/T41	=U42/U41	=V42/V41	=W42/W41
61		Utility				
62	Type B	Comprehension	=T44/T43	=U44/U43	=V44/V43	=W44/W43
63		Utility				

	E	X	Y	Z	AA	AB
45		=X2	=Y2	=Z2	=AA2	=AB2
46	Overall	=X32/X31	=Y32/Y31	=Z32/Z31	=AA32/AA31	=AB32/AB31
47						
48	ArchiMate	=X34/X33	=Y34/Y33	=Z34/Z33	=AA34/AA33	=AB34/AB33
49						
50	Neutral	=X36/X35	=Y36/Y35	=Z36/Z35	=AA36/AA35	=AB36/AB35
51						
52	Group	=X38/X37	=Y38/Y37	=Z40/Z39	=AA40/AA39	=AB40/AB39
53						
54	Interview	=X40/X39	=Y40/Y39	=Z40/Z39	=AA40/AA39	=AB40/AB39
55						
56	ArchiMate vs. Ne					
57						
58	Int vs. Group					
59						
60	Type A	=X42/X41	=Y42/Y41	=Z42/Z41	=AA42/AA41	=AB42/AB41
61						
62	Type B	=X44/X43	=Y44/Y43	=Z44/Z43	=AA44/AA43	=AB44/AB43
63						

	E	AC	AD	AE	AF
45		=AC2	=AD2	=AE2	=AF2
46	Overall	=AC32/AC31	=AD32/AD31	=AE32/AE31	=AF32/AF31
47					
48	ArchiMate	=AC34/AC33	=AD34/AD33	=AE34/AE33	=AF34/AF33
49					
50	Neutral	=AC36/AC35	=AD36/AD35	=AE36/AE35	=AF36/AF35
51					
52	Group	=AC40/AC39	=AD40/AD39	=AE40/AE39	=AF40/AF39
53					
54	Interview	=AC40/AC39	=AD40/AD39	=AE40/AE39	=AF40/AF39
55					
56	ArchiMate vs. Ne				
57					
58	Int vs. Group				
59					
60	Type A	=AC42/AC41	=AD42/AD41	=AE42/AE41	=AF42/AF41
61					
62	Type B	=AC44/AC43	=AD44/AD43	=AE44/AE43	=AF44/AF43
63					

L.3 Formulae for Application Layer

Table 82 Formulae for Application Comprehension Ratings

	E	F	G	H	I	J
24	ArchiMate	=COUNTIF(F$3:F$22,"=Yes")				
25	Neutral	=COUNTIF(F$3:F$22,"=No")				
26	Count Rows Populate		=COUNT(G3:G22)	=COUNT(H3:H22)	=COUNT(I3:I22)	=COUNT(J3:J22)
27	Total Score		=SUM(G1:G22)	=SUM(H1:H22)	=SUM(I1:I22)	=SUM(J1:J22)
28	Count Rows ArchiMat		=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")
29	Score Rows ArchiMate		=SUMIF($F$3:$F$22,"Yes",G$3:G$22)	=SUMIF($F$3:$F$22,"Yes",H$3:H$22)	=SUMIF($F$3:$F$22,"Yes",I$3:I$22)	=SUMIF($F$3:$F$22,"Yes",J$3:J$22)
30	Count Rows Neutral		=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")
31	Score Rows Neutral		=SUMIF($F$3:$F$22,"No",G$3:G$22)	=SUMIF($F$3:$F$22,"No",H$3:H$22)	=SUMIF($F$3:$F$22,"No",I$3:I$22)	=SUMIF($F$3:$F$22,"No",J$3:J$22)
32	Count Rows Group		=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")
33	Score Rows Group		=SUMIF($D$3:$D$22,"Yes",G$3:G$22)	=SUMIF($D$3:$D$22,"Yes",H$3:H$22)	=SUMIF($D$3:$D$22,"Yes",I$3:I$22)	=SUMIF($D$3:$D$22,"Yes",J$3:J$22)
34	Count Rows Non-Grou		=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")
35	Score Rows Non-Grou		=SUMIF($D$3:$D$22,"No",G$3:G$22)	=SUMIF($D$3:$D$22,"No",H$3:H$22)	=SUMIF($D$3:$D$22,"No",I$3:I$22)	=SUMIF($D$3:$D$22,"No",J$3:J$22)
36	Count Rows Type A		=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")
37	Score Rows Type A		=SUMIF($E$3:$E$22,"A",G$3:G$22)	=SUMIF($E$3:$E$22,"A",H$3:H$22)	=SUMIF($E$3:$E$22,"A",I$3:I$22)	=SUMIF($E$3:$E$22,"A",J$3:J$22)
38	Count Rows Type B		=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")
39	Score Rows Type B		=SUMIF($E$3:$E$22,"B",G$3:G$22)	=SUMIF($E$3:$E$22,"B",H$3:H$22)	=SUMIF($E$3:$E$22,"B",I$3:I$22)	=SUMIF($E$3:$E$22,"B",J$3:J$22)

	E	K	L	M	N	O
24	ArchiMate					
25	Neutral					
26	Count Rows Populate	=COUNT(K3:K22)	=COUNT(L3:L22)	=COUNT(M3:M22)	=COUNT(N3:N22)	=COUNT(O3:O22)
27	Total Score	=SUM(K1:K22)	=SUM(L1:L22)	=SUM(M1:M22)	=SUM(N1:N22)	=SUM(O1:O22)
28	Count Rows ArchiMat	=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")	=COUNTIF($F$3:$F$22,"Yes")
29	Score Rows ArchiMate	=SUMIF($F$3:$F$22,"Yes",K$3:K$22)	=SUMIF($F$3:$F$22,"Yes",L$3:L$22)	=SUMIF($F$3:$F$22,"Yes",M$3:M$22)	=SUMIF($F$3:$F$22,"Yes",N$3:N$22)	=SUMIF($F$3:$F$22,"Yes",O$3:O$22)
30	Count Rows Neutral	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")
31	Score Rows Neutral	=SUMIF($F$3:$F$22,"No",K$3:K$22)	=SUMIF($F$3:$F$22,"No",L$3:L$22)	=SUMIF($F$3:$F$22,"No",M$3:M$22)	=SUMIF($F$3:$F$22,"No",N$3:N$22)	=SUMIF($F$3:$F$22,"No",O$3:O$22)
32	Count Rows Group	=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")	=COUNTIF($D$3:$D$22,"Yes")
33	Score Rows Group	=SUMIF($D$3:$D$22,"Yes",K$3:K$22)	=SUMIF($D$3:$D$22,"Yes",L$3:L$22)	=SUMIF($D$3:$D$22,"Yes",M$3:M$22)	=SUMIF($D$3:$D$22,"Yes",N$3:N$22)	=SUMIF($D$3:$D$22,"Yes",O$3:O$22)
34	Count Rows Non-Grou	=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")	=COUNTIF($D$3:$D$22,"No")
35	Score Rows Non-Grou	=SUMIF($D$3:$D$22,"No",K$3:K$22)	=SUMIF($D$3:$D$22,"No",L$3:L$22)	=SUMIF($D$3:$D$22,"No",M$3:M$22)	=SUMIF($D$3:$D$22,"No",N$3:N$22)	=SUMIF($D$3:$D$22,"No",O$3:O$22)
36	Count Rows Type A	=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")	=COUNTIF($E$3:$E$22,"A")
37	Score Rows Type A	=SUMIF($E$3:$E$22,"A",K$3:K$22)	=SUMIF($E$3:$E$22,"A",L$3:L$22)	=SUMIF($E$3:$E$22,"A",M$3:M$22)	=SUMIF($E$3:$E$22,"A",N$3:N$22)	=SUMIF($E$3:$E$22,"A",O$3:O$22)
38	Count Rows Type B	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")
39	Score Rows Type B	=SUMIF($E$3:$E$22,"B",K$3:K$22)	=SUMIF($E$3:$E$22,"B",L$3:L$22)	=SUMIF($E$3:$E$22,"B",M$3:M$22)	=SUMIF($E$3:$E$22,"B",N$3:N$22)	=SUMIF($E$3:$E$22,"B",O$3:O$22)

Table 83 Formulae for Application Utility Ratings

	E	F	P	Q	R	S
24	ArchiMate	=COUNTIF(F$3:F$22,"=Yes")				
25	Neutral	=COUNTIF(F$3:F$22,"=No")				
26	Count Rows Populate		=COUNT(P3:P22)	=COUNT(Q3:Q22)	=COUNT(R3:R22)	=COUNT(S3:S22)
27	Total Score		=SUM(P3:P22)	=SUM(Q1:Q22)	=SUM(R1:R22)	=SUM(S1:S22)
28	Count Rows ArchiMat		=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2
29	Score Rows ArchiMate		=SUMIF($F$3:$F$22,"Yes",P$3:P$22)	=SUMIF($F$3:$F$22,"Yes",Q$3:Q$22)	=SUMIF($F$3:$F$22,"Yes",R$3:R$22)	=SUMIF($F$3:$F$22,"Yes",S$3:S$22)
30	Count Rows Neutral		=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")
31	Score Rows Neutral		=SUMIF($F$3:$F$22,"No",P$3:P$22)	=SUMIF($F$3:$F$22,"No",Q$3:Q$22)	=SUMIF($F$3:$F$22,"No",R$3:R$22)	=SUMIF($F$3:$F$22,"No",S$3:S$22)
32	Count Rows Group		=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1
33	Score Rows Group		=SUMIF($D$3:$D$22,"Yes",P$3:P$22)	=SUMIF($D$3:$D$22,"Yes",Q$3:Q$22)	=SUMIF($D$3:$D$22,"Yes",R$3:R$22)	=SUMIF($D$3:$D$22,"Yes",S$3:S$22)
34	Count Rows Non-Grou		=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1
35	Score Rows Non-Grou		=SUMIF($D$3:$D$22,"No",P$3:P$22)	=SUMIF($D$3:$D$22,"No",Q$3:Q$22)	=SUMIF($D$3:$D$22,"No",R$3:R$22)	=SUMIF($D$3:$D$22,"No",S$3:S$22)
36	Count Rows Type A		=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1
37	Score Rows Type A		=SUMIF($E$3:$E$22,"A",P$3:P$22)	=SUMIF($E$3:$E$22,"A",Q$3:Q$22)	=SUMIF($E$3:$E$22,"A",R$3:R$22)	=SUMIF($E$3:$E$22,"A",S$3:S$22)
38	Count Rows Type B		=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")
39	Score Rows Type B		=SUMIF($E$3:$E$22,"B",P$3:P$22)	=SUMIF($E$3:$E$22,"B",Q$3:Q$22)	=SUMIF($E$3:$E$22,"B",R$3:R$22)	=SUMIF($E$3:$E$22,"B",S$3:S$22)

	E	T	U	V	W	X
24	ArchiMate					
25	Neutral					
26	Count Rows Populate	=COUNT(T3:T22)	=COUNT(U3:U22)	=COUNT(V3:V22)	=COUNT(W3:W22)	=COUNT(X3:X22)
27	Total Score	=SUM(T1:T22)	=SUM(U1:U22)	=SUM(V1:V22)	=SUM(W1:W22)	=SUM(X1:X22)
28	Count Rows ArchiMat	=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2	=COUNTIF($F$3:$F$22,"Yes")-2
29	Score Rows ArchiMate	=SUMIF($F$3:$F$22,"Yes",T$3:T$22)	=SUMIF($F$3:$F$22,"Yes",U$3:U$22)	=SUMIF($F$3:$F$22,"Yes",V$3:V$22)	=SUMIF($F$3:$F$22,"Yes",W$3:W$22)	=SUMIF($F$3:$F$22,"Yes",X$3:X$22)
30	Count Rows Neutral	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")	=COUNTIF($F$3:$F$22,"No")
31	Score Rows Neutral	=SUMIF($F$3:$F$22,"No",T$3:T$22)	=SUMIF($F$3:$F$22,"No",U$3:U$22)	=SUMIF($F$3:$F$22,"No",V$3:V$22)	=SUMIF($F$3:$F$22,"No",W$3:W$22)	=SUMIF($F$3:$F$22,"No",X$3:X$22)
32	Count Rows Group	=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1	=COUNTIF($D$3:$D$22,"Yes")-1
33	Score Rows Group	=SUMIF($D$3:$D$22,"Yes",T$3:T$22)	=SUMIF($D$3:$D$22,"Yes",U$3:U$22)	=SUMIF($D$3:$D$22,"Yes",V$3:V$22)	=SUMIF($D$3:$D$22,"Yes",W$3:W$22)	=SUMIF($D$3:$D$22,"Yes",X$3:X$22)
34	Count Rows Non-Grou	=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1	=COUNTIF($D$3:$D$22,"No")-1
35	Score Rows Non-Grou	=SUMIF($D$3:$D$22,"No",T$3:T$22)	=SUMIF($D$3:$D$22,"No",U$3:U$22)	=SUMIF($D$3:$D$22,"No",V$3:V$22)	=SUMIF($D$3:$D$22,"No",W$3:W$22)	=SUMIF($D$3:$D$22,"No",X$3:X$22)
36	Count Rows Type A	=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1	=COUNTIF($E$3:$E$22,"A")-1
37	Score Rows Type A	=SUMIF($E$3:$E$22,"A",T$3:T$22)	=SUMIF($E$3:$E$22,"A",U$3:U$22)	=SUMIF($E$3:$E$22,"A",V$3:V$22)	=SUMIF($E$3:$E$22,"A",W$3:W$22)	=SUMIF($E$3:$E$22,"A",X$3:X$22)
38	Count Rows Type B	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")	=COUNTIF($E$3:$E$22,"B")
39	Score Rows Type B	=SUMIF($E$3:$E$22,"B",T$3:T$22)	=SUMIF($E$3:$E$22,"B",U$3:U$22)	=SUMIF($E$3:$E$22,"B",V$3:V$22)	=SUMIF($E$3:$E$22,"B",W$3:W$22)	=SUMIF($E$3:$E$22,"B",X$3:X$22)

Table 84 Formulae for Application Comprehension vs. Utility Charts (left)

	E	F	G	H	I	J	K	L	M	N	O
40		Element	=G2	=H2	=I2	=J2	=K2	=L2	=M2	=N2	=O2
41	Overall	Comprehension	=G27/G26	=H27/H26	=I27/I26	=J27/J26	=K27/K26	=L27/L26	=M27/M26	=N27/N26	=O27/O26
42		Utility	=P41	=Q41	=R41	=S41	=T41	=U41	=V41	=W41	=X41
43	ArchiMate	Comprehension	=G29/G28	=H29/H28	=I29/I28	=J29/J28	=K29/K28	=L29/L28	=M29/M28	=N29/N28	=O29/O28
44		Utility	=P43	=Q43	=R43	=S43	=T43	=U43	=V43	=W43	=X43
45	Neutral	Comprehension	=G31/G30	=H31/H30	=I31/I30	=J31/J30	=K31/K30	=L31/L30	=M31/M30	=N31/N30	=O31/O30
46		Utility	=P45	=Q45	=R45	=S45	=T45	=U45	=V45	=W45	=X45
47	Group	Comprehension	=G33/G32	=H33/H32	=I33/I32	=J33/J32	=K33/K32	=L33/L32	=M33/M32	=N33/N32	=O33/O32
48		Utility	=P47	=Q47	=R47	=S47	=T47	=U47	=V47	=W47	=X47
49	Interview	Comprehension	=G35/G34	=H35/H34	=I35/I34	=J35/J34	=K35/K34	=L35/L34	=M35/M34	=N35/N34	=O35/O34
50		Utility	=P49	=Q49	=R49	=S49	=T49	=U49	=V49	=W49	=X49
51	ArchiMate vs. Neutral	Comprehension	=G43-G45	=H43-H45	=I43-I45	=J43-J45	=K43-K45	=L43-L45	=M43-M45	=N43-N45	=O43-O45
52		Utility	=G44-G46	=H44-H46	=I44-I46	=J44-J46	=K44-K46	=L44-L46	=M44-M46	=N44-N46	=O44-O46
53	Int vs. Group	Comprehension	=G49-G47	=H49-H47	=I49-I47	=J49-J47	=K49-K47	=L49-L47	=M49-M47	=N49-N47	=O49-O47
54		Utility	=G50-G48	=H50-H48	=I50-I48	=J50-J48	=K50-K48	=L50-L48	=M50-M48	=N50-N48	=O50-O48
55	Type A	Comprehension	=G37/G36	=H37/H36	=I37/I36	=J37/J36	=K37/K36	=L37/L36	=M37/M36	=N37/N36	=O37/O36
56		Utility	=P55	=Q55	=R55	=S55	=T55	=U55	=V55	=W55	=X55
57	Type B	Comprehension	=G39/G38	=H39/H38	=I39/I38	=J39/J38	=K39/K38	=L39/L38	=M39/M38	=N39/N38	=O39/O38
58		Utility	=P57	=Q57	=R57	=S57	=T57	=U57	=V57	=W57	=X57
59			Application component	Application collaboration	Application interface	Application function	Application interaction	Application process	Application event	Application service	Data object

Table 85 Formulae for Application Comprehension vs. Utility Charts (right)

	E	F	P	Q	R	S	T	U	V	W	X
40		Element									
41	Overall	Comprehension	=P27/P26	=Q27/Q26	=R27/R26	=S27/S26	=T27/T26	=U27/U26	=V27/V26	=W27/W26	=X27/X26
42		Utility									
43	ArchiMate	Comprehension	=P29/P28	=Q29/Q28	=R29/R28	=S29/S28	=T29/T28	=U29/U28	=V29/V28	=W29/W28	=X29/X28
44		Utility									
45	Neutral	Comprehension	=P31/P30	=Q31/Q30	=R31/R30	=S31/S30	=T31/T30	=U31/U30	=V31/V30	=W31/W30	=X31/X30
46		Utility									
47	Group	Comprehension	=P33/P32	=Q33/Q32	=R33/R32	=S33/S32	=T33/T32	=U33/U32	=V33/V32	=W33/W32	=X33/X32
48		Utility									
49	Interview	Comprehension	=P35/P34	=Q35/Q34	=R35/R34	=S35/S34	=T35/T34	=U35/U34	=V35/V34	=W35/W34	=X35/X34
50		Utility									
51	ArchiMate vs. Neutral	Comprehension									
52		Utility									
53	Int vs. Group	Comprehension									
54		Utility									
55	Type A	Comprehension	=P37/P36	=Q37/Q36	=R37/R36	=S37/S36	=T37/T36	=U37/U36	=V37/V36	=W37/W36	=X37/X36
56		Utility									
57	Type B	Comprehension	=P39/P38	=Q39/Q38	=R39/R38	=S39/S38	=T39/T38	=U39/U38	=V39/V38	=W39/W38	=X39/X38
58		Utility									

Appendix M Statistical Treatment of Differential Results

This appendix contains details of the statistical analysis of the differential results for comprehension and utility that forms part of the analysis contained in 7.7.5.5 above.

We start with the use of 'box and whisker' diagrams, originally created by Tukey and discussed for example in [224]. We are looking to compare the average comprehension and utility scores (as summarised in Table 42 above) across survey respondents, and so to do this we start by taking average figures for each survey respondent across each ArchiMate "layer" on which they responded. This was done by adding two 'average' columns to the tables of raw data shown in Appendix K above, so for example a column was inserted in the motivation results set (Table 68 above) between the "Comprehension M10" column and the "Utility M1" column, averaging the values for each respondent, as shown below:

Figure 173 Averaging Comprehension Figures for Each Respondent (Motivation layer)

=AVERAGE(H4:Q4)

G	H	I	J	K	L	M	N	O	P	Q	R
	Comprehension										
ArchiMa	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	**average**
No	1	1	0.7	0.3	1	1	0.7	0.7	0.3	1	**0.77**
No	1	1	1	0.7	0.3	0.3	1	0.7	0.3	1	**0.73**
No	1	0.7	0	0.3	0.3	1	1	1	0.7	1	**0.7**

The extra column is shown in bold. Exactly the same was done for the business layer and application layer, and for the utility values as well. In order to give an overall figure, an additional table of averages was created that contained the concatenated averages from the motivation, business and application data sets.

Each of these data sets was made the subject of a 'box and whisker' diagram; the motivation, business and application layer data. In addition, combined (overall) plots were created by concatenating the average figures, for each respondent, across all three layers, using the following table (figures are of course taken from the individual tables created for each layer; and the layer column is included here only for traceability purposes, not because it was used to create the Combined plots):

Table 86 Combined Average Comprehension and Utility Scores

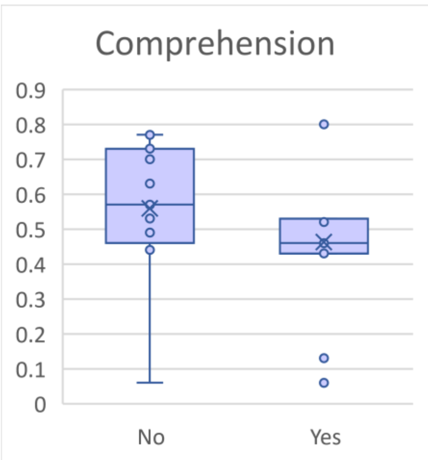
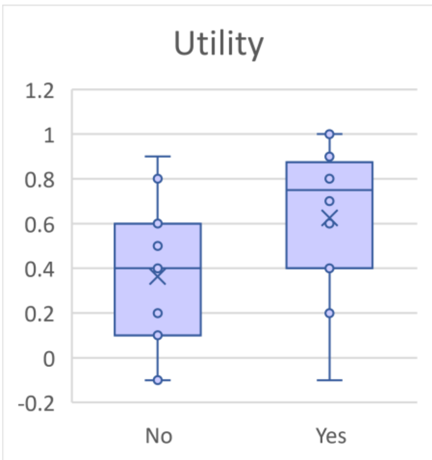
Layer	ArchiMate used	Comprehension	Utility
Motivation	No	0.77	0.10
Motivation	Yes	0.43	0.80

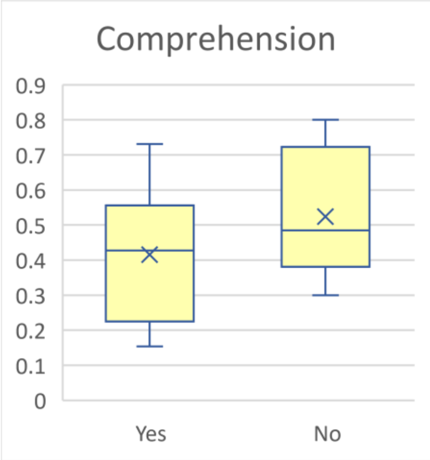
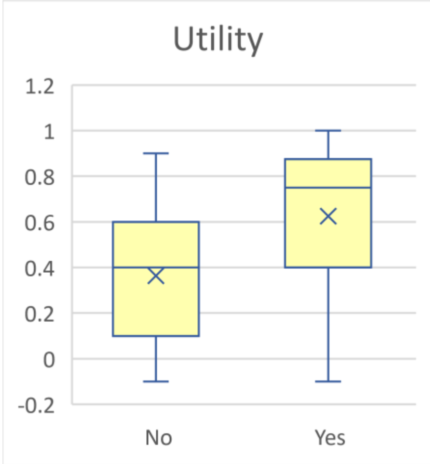
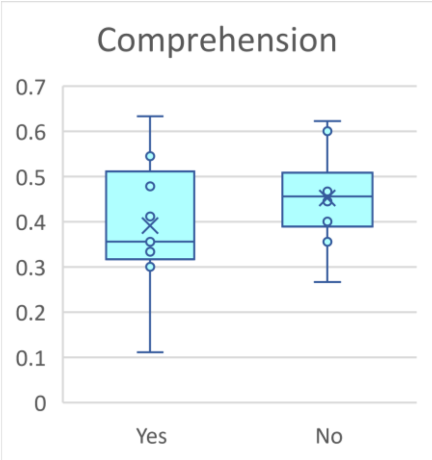
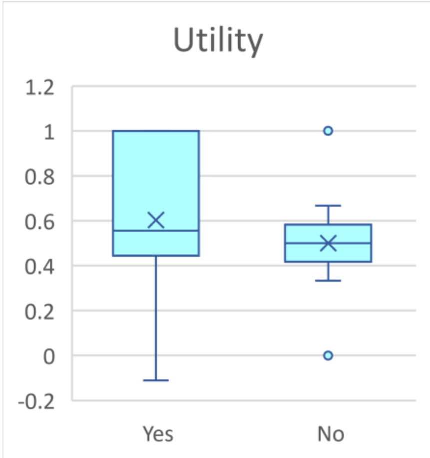
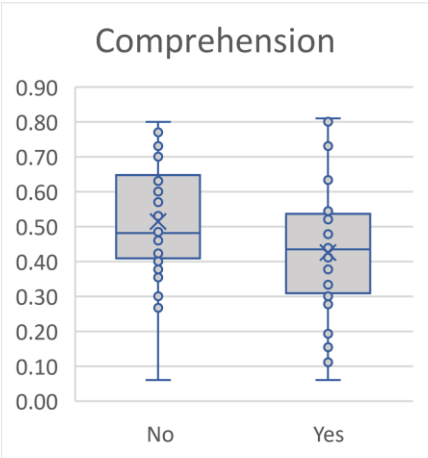
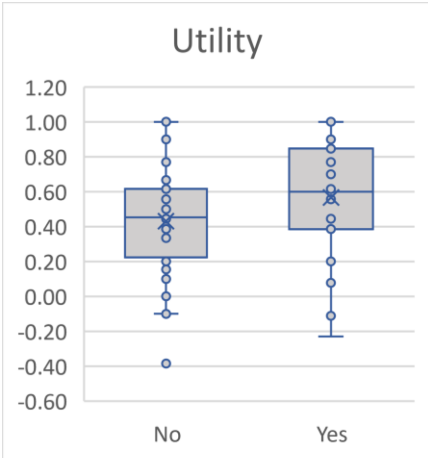
Layer	ArchiMate used	Comprehension	Utility
Motivation	Yes	0.53	0.60
Motivation	No	0.73	0.90
Motivation	No	0.70	0.50
Motivation	Yes	0.46	0.40
Motivation	Yes	0.43	0.80
Motivation	Yes	0.81	0.90
Motivation	No	0.63	0.60
Motivation	Yes	0.80	0.80
Motivation	Yes	0.46	1.00
Motivation	Yes	0.52	0.70
Motivation	No	0.44	0.80
Motivation	No	0.57	0.40
Motivation	Yes	0.06	0.40
Motivation	No	0.53	0.60
Motivation	No	0.46	-0.10
Motivation	Yes	0.46	1.00
Motivation	No	0.77	-0.10
Motivation	Yes	0.13	-0.10
Motivation	Yes		0.20
Motivation	No	0.06	0.10
Motivation	No	0.49	0.20
Business	Yes	0.42	0.62
Business	No	0.74	-0.38
Business	Yes	0.44	0.38
Business	Yes	0.28	0.38
Business	Yes	0.56	0.46
Business	Yes	0.54	0.77
Business	No	0.80	0.77
Business	Yes	0.73	0.23
Business	No	0.72	0.69
Business	No	0.43	0.62
Business	Yes	0.64	0.38
Business	No	0.42	0.62
Business	Yes	0.38	0.92
Business	No	0.73	1.00
Business	No	0.54	0.23
Business	No	0.38	0.38
Business	Yes	0.19	0.85
Business	No	0.54	0.46
Business	No	0.38	0.38
Business	Yes	0.21	1.00
Business	No	0.48	0.15

Layer	ArchiMate used	Comprehension	Utility
Business	Yes	0.15	-0.23
Business	Yes	0.45	0.08
Business	No	0.30	0.23
Business	No	0.35	0.46
Application	Yes	0.36	0.44
Application	No	0.62	1.00
Application	No	0.47	0.67
Application	Yes	0.36	
Application	Yes	0.54	0.56
Application	No	0.27	0.56
Application	No	0.60	0.44
Application	No	0.48	0.44
Application	Yes	0.63	0.56
Application	Yes	0.33	0.78
Application	Yes	0.30	1.00
Application	No	0.41	0.44
Application	No	0.44	0.33
Application	Yes	0.41	1.00
Application	No	0.48	0.56
Application	Yes	0.11	
Application	Yes	0.48	-0.11
Application	No	0.40	0.00
Application	No	0.36	0.56

The box and whisker diagrams are shown below:

Table 87 Box and Whisker Analysis of Effect of ArchiMate Notation

<table>
<tr><td>Layer</td><td colspan="2">Average per-respondent scores for non-ArchiMate vs ArchiMate notation</td></tr>
<tr><td>Motivation</td><td>
</td><td>
</td></tr>
</table>

Business		
Application		
Combined (Motivation + Business + Application)		

Care should be taken when examining these plots, as for some reason we have been unable to ascertain, Excel has in some cases put the ‘No’ (i.e. without ArchiMate notation) plot on the left, and in other cases on the right. These plots would tend to agree with our earlier suggestion that comprehension scores decrease but utility scores increase when using ArchiMate (as opposed to the coloured boxes used in the alternative ‘neutral’ notation).

The literature previous referred to, related to box and whisker diagrams, cautions that these kind of plots cannot be used for inference; instead, some kind of T-test [225] is required, in particular to evaluate the possibility that these results are due to change (the null hypothesis). We therefore carried out T-tests using the Excel Analysis Toolpak software [226], starting by rearranging our overall tables of comprehension and utility scores into adjacent scores showing 'without' and with' ArchiMate:

Table 88 Rearrangement of Tables Without and With ArchiMate

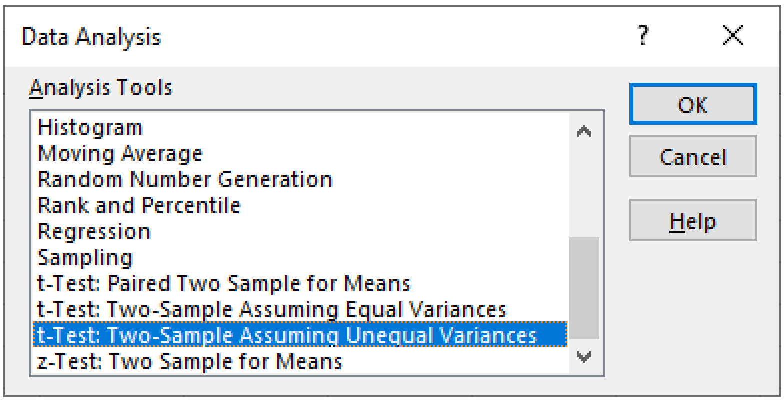
Comprehension		Utility	
No	**Yes**	**No**	**Yes**
0.77	0.43	0.10	0.80
0.73	0.53	0.90	0.60
0.70	0.46	0.50	0.40
0.63	0.43	0.60	0.80
0.44	0.81	0.80	0.90
0.57	0.80	0.40	0.80
0.53	0.46	0.60	1.00
0.46	0.52	-0.10	0.70
0.77	0.06	-0.10	0.40
0.06	0.46	0.10	1.00
0.49	0.13	0.20	-0.10
0.74		-0.38	0.20
0.80	0.42	0.77	0.62
0.72	0.44	0.69	0.38
0.43	0.28	0.62	0.38
0.42	0.56	0.62	0.46
0.73	0.54	1.00	0.77
0.54	0.73	0.23	0.23
0.38	0.64	0.38	0.38
0.54	0.38	0.46	0.92
0.38	0.19	0.38	0.85
0.48	0.21	0.15	1.00
0.30	0.15	0.23	-0.23
0.35	0.45	0.46	0.08
0.62	0.36	1.00	0.44
0.47	0.36	0.67	
0.27	0.54	0.56	0.56
0.60	0.63	0.44	0.56
0.48	0.33	0.44	0.78
0.41	0.30	0.44	1.00
0.44	0.41	0.33	1.00
0.48	0.11	0.56	

0.40	0.48
0.36	

0.00	-0.11
0.56	

We then used the T-test twice, once for each pair of columns, invoked using the following Excel command:

Figure 174 Invoking the T-Test in Excel



The results for the Comprehension scores are as follows:

Table 89 Output from Excel T-Test for Comprehension Scores

	Variable 1	*Variable 2*
Mean	0.514341	0.424901
Variance	0.028087	0.035372
Observations	34	32
Hypothesized Mean Difference	0	
df	62	
t Stat	2.035113	
P(T<=t) one-tail	0.02306	
t Critical one-tail	1.669804	
P(T<=t) two-tail	0.04612	
t Critical two-tail	1.998972	

The key figure we require is the p-value, two-tail, which is calculated here to be 0.04612, under the 0.05 threshold; thus we can reject the null-hypothesis and say that this result, for the Comprehension figures, is statistically significant.

We obtained results for the Utility scores using the same technique, and they are as follows:

Table 90 Output from Excel T-Test for Utility Scores

	Variable 1	*Variable 2*
Mean	0.429864	0.566722

Variance	0.099389	0.124447
Observations	34	31
Hypothesized Mean Difference	0	
df	60	
t Stat	-1.6431	
P(T<=t) one-tail	0.052797	
t Critical one-tail	1.670649	
P(T<=t) two-tail	0.105594	
t Critical two-tail	2.000298	

The p-value above is 0.105594; this is greater than the 0.05 threshold value and therefore we are not entitled, on the basis of this data, to reject the null hypothesis; the data is not strong enough.

Appendix N References

1. RIBA. *Architecture.com by RIBA*. 2015 [cited 2015 31st December]; Available from: https://www.architecture.com.
2. Schekkerman, J., *Trends in enterprise architecture 2005: How are organizations progressing.* Institute for Enterprise Architecture Developments, Amersfoort, 2005.
3. Varnus, J. and N. Panaich, *TOGAF 9 Survey Results Presentation*. 2009, The Open Group.
4. Pettey, C. and R.v.d. Meulen, *Gartner Says Enterprise Architecture Practitioners Significantly Influenced $1.1 Trillion*. 2013, Gartner: Stamford, Connecticut.
5. *Definition of 'enterprise'*, in *dictionary.com*. 2020.
6. *Definition of 'enterprise'*, in *Cambridge Dictionary*. 2020, Cambridge University Press.
7. The Open Group, *The Open Group Architecture Framework*, in *Executive Overview*. 2018.
8. *ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description*. 2011.
9. Op't Land, M., et al., *Enterprise architecture: creating value by informed governance*. 2008: Springer Science & Business Media.
10. Chotalia, P., *Evaluation of financial health of sampled private sector banks with Altman Z-score model.* Circulation in more than 85 countries, 2012. **7**.
11. Schuermann, T., *Stress testing banks.* International Journal of Forecasting, 2014. **30**(3): p. 717-728.
12. *Data Protection Act 2018*, U. Government, Editor. 2018: legislation.gov.uk.
13. Aikman, I., *Coronavirus supermarkets latest: non-essential goods guidelines for England*, in *Which? News*. 2020, Which?
14. The Open Group, *The TOGAF Standard, Version 9.2*. 2018, The Open Group,.
15. The Open Group, *ArchiMate® 3.1 Specification*. 2019.
16. Rood, M.A. *Enterprise architecture: definition, content, and utility*. in *Third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1994.* . 1994.
17. Brand, S., *Magic Quadrant for Enterprise Architecture Tools*. 2015, Gartner.
18. Sparx. *Sparx Systems: EA Pricing & Purchasing*. 2015 [cited 2015 27th March]; Available from: http://www.sparxsystems.eu/enterprisearchitect/ea-pricing-purchasing/.
19. CACI. *CACI Mood Software*. 2020 [cited 2020 14th June]; Available from: https://www.moodsoftware.co.uk/.
20. Troux. *Troux Architect*. 2016 [cited 2015 28th March]; Available from: http://www.troux.com/enterprise-architecture/products#architect.
21. Software AG. *IT Portfolio Management*. 2020 [cited 2020 9th June]; Available from: https://www.softwareag.com/en_corporate/platform/alfabet/it-portfolio-management.html.
22. BiZZdesign. *Enterprise Studio*. 2017 [cited 2017 2nd March]; Available from: http://www.bizzdesign.com/enterprise-studio.
23. dictionary.com, *Synonyms for models*, in *Thesaurus.com*. 2020, dictionary.com.

24. Spence, C. and V. Michell. *Assessing the Business Risk of Technology Obsolescence through Enterprise Modelling*. in *Symposium on Business Modeling and Software Design*. 2011. Sofia: SciTePress.

25. Lagerström, R., et al., *A method for creating entreprise architecture metamodels: applied to systems modifiability.* International Journal of Computer Science and Applications, 2009. **6**(5): p. 89-120.

26. Alshuqayran, N., N. Ali, and R. Evans. *A systematic mapping study in microservice architecture*. in *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*. 2016. IEEE.

27. Chappell, D.A., *Enterprise service bus*. 2004: " O'Reilly Media, Inc.".

28. Oxford Learner's Dictionaries, *Definition of Software*, in *Oxford Learner's Dictionaries*. 2020.

29. The Open Group, *The Open Group Architectural Framework (TOGAF) 9.1*. 2011.

30. The Open Group, *The Open Group Architecture Framework (TOGAF) 8.1.1*. 2006.

31. The Open Group, *The Open Group Architectural Framework (TOGAF) 9*. 2009.

32. The Open Group, *The Open Group Architectural Framework (TOGAF) 9.2*, in *Preliminary Phase*. 2018, The Open Group,. p. 51-64.

33. ISO/IEC/IEEE, *ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description*. 2011.

34. Standish Group, *The CHAOS Report*. 1994, The Standish Group. p. 8.

35. Mellor, S.J., et al., *Model-Driven Architecture*, in *Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings*, J.-M. Bruel and Z. Bellahsene, Editors. 2002, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 290-297.

36. Stachowiak, H., *Allgemeine Modelltheorie*. 1973, Wien: Springer-Verlag.

37. Liu, K., *Semiotics in Information Systems Engineering*. 2009: Cambridge University Press. 232.

38. Barroero, T., G. Motta, and G. Pignatelli. *Business Capabilities Centric Enterprise Architecture*. 2010. Berlin, Heidelberg: Springer Berlin Heidelberg.

39. The Open Group, *Archimate 3.0 Specification*. 2016.

40. Object Management Group, *Unified Modelling Language Version 2.5*. 2015.

41. Reeves, C.A. and D.A. Bednar, *Defining Quality: Alternatives and Implications.* The Academy of Management Review, 1994. **19**(3): p. 419-445.

42. Church of England. *Book of Common Prayer*. 1762; Available from: https://www.churchofengland.org/sites/default/files/2019-10/The%20Book%20of%20Common%20Prayer%201662.pdf.

43. Krogstie, J., *A Semiotic Approach to Quality in Requirements Specifications*, in *Organizational Semiotics*, K. Liu, et al., Editors. 2002, Springer US. p. 231-249.

44. Popescu, G., *Designing modeling notations readers understand*, in *Information and Communications Faculty*. 2016, EPFL.

45. Oberhauser, R. and C. Pogolski. *VR-EA: Virtual Reality Visualization of Enterprise Architecture Models with ArchiMate and BPMN*. 2019. Cham: Springer International Publishing.

46. Kruchten, P., *The Software Architect*, in *Software Architecture: TC2 First Working IFIP Conference on Software Architecture (WICSA1) 22–24 February 1999, San Antonio, Texas, USA*, P. Donohoe, Editor. 1999, Springer US: Boston, MA. p. 565-583.

47. Spence, C. and V. Michell, *Experiences of Tool-based Enterprise Modelling as part of Architectural Change Management*, in *Symposium on Business Modeling and Software Design*, B. Shishkov, Editor. 2015, SciTePress: Milan.

48. Krogstie, J., O.I. Lindland, and G. Sindre. *Defining Quality Aspects for Conceptual Models*. in *IFIP8.1 working conference on Information Systems Concepts (ISCO3)*. 1995. Marburg, Germany.

49. Object Management Group, *Unified Modeling Language*. 2017.

50. The Open Group, *The Open Group Architectural Framework (TOGAF) 9.2*, in *Deliverables, Artifacts, and Building Blocks*. 2018. p. 14.

51. Spence, C. and V. Michell, *Measuring the Quality of Enterprise Architecture Models.* Journal of Enterprise Architecture, 2016. **12**(3).

52. Ogden, C.K. and I.A. Richards, *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Vol. 29. 1923: K. Paul, Trench, Trubner & Company, Limited.

53. Peirce, C.S., *Collected Papers of Charles Sanders Peirce, ed. C. Hartshorne and P. Weiss*. 1931, C. Hartshorne & P. Weiss). Harvard University Press, Cambridge, Mass.

54. Thomas J. Madden, Kelly Hewett, and M.S. Roth, *Managing Images in Different Cultures: A Cross-National Study of Color Meanings and Preferences.* Journal of International Marketing, 2000. **8**(4): p. 90-107.

55. Wout, J.v.t., et al., *The Integrated Architecture Framework Explained*. 2010, Berlin Heidelberg: Springer-Verlag.

56. Lindland, O.I., G. Sindre, and A. Solvberg, *Understanding quality in conceptual modeling.* Software, IEEE, 1994. **11**(2): p. 42-49.

57. Frank, U. *Power-Modelling: Toward a More Versatile Approach to Creating and Using Conceptual Models*. in *Fourth Internatioal Symposium on Business Modeling and Software Design*. 2014. Luxembourg: SCITEPRESS.

58. Shanks, G., E. Tansley, and R. Weber, *Using ontology to validate conceptual models.* Communications of the ACM, 2003. **46**(10): p. 85-89.

59. Geihs, K., *Middleware challenges ahead.* Computer, 2001. **34**(6): p. 24-31.

60. Zachman, J. *Zachman International Enterprise Architecture*. 2015 [cited 2015 5th February]; Available from: http://www.zachman.com/about-the-zachman-framework.

61. Spence, C. and V. Michell, *Measuring the Quality of Enterprise Architecture Models.* Journal of Enterprise Architecture, 2016. **12**(3): p. 64-73.

62. Jonkers, H., I. Band, and D. Quartel, *ArchiSurance Case Study*. 2012, The Open Group.

63. Members of the College of Senior Scholars. *Senior Scholars' Basket of Journals*. 2011 [cited 2017 2nd February]; Available from: https://aisnet.org/?SeniorScholarBasket.

64. *Google Scholar*. 2020 [cited 2020 18th November 2020]; Available from: https://scholar.google.co.uk/.

65. Cambridge University Press, *Definition of “Enterprise”*, in *Cambridge Advanced Learner’s Dictionary and Thesaurus*. 2016, Cambridge University Press.

66. Henderson, J.C. and N. Venkatraman, *Strategic alignment: leveraging information technology for transforming organizations.* IBM Syst. J., 1993. **32**(1): p. 4-16.

67. Merriam-Webster, *Definition of SYSTEM*, in *Merriam-Webster Online Dictionary*. 2017.

68. Rood, M.A. *Enterprise architecture: definition, content, and utility*. in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1994. Proceedings., Third Workshop on*. 1994.

69. Jørgensen, M. and K. Moløkken-Østvold, *How large are software cost overruns? A review of the 1994 CHAOS report.* Information and Software Technology, 2006. **48**(4): p. 297-301.

70. Eveleens, J.L. and C. Verhoef, *The rise and fall of the Chaos report figures.* IEEE Software, 2010. **27**(1): p. 30-36.

71. Lankhorst, M.M., H.A. Proper, and H. Jonkers, *The Architecture of the ArchiMate Language*, in *Enterprise, Business-Process and Information Systems Modeling*, T. Halpin, et al., Editors. 2009, Springer Berlin Heidelberg. p. 367-380.

72. Capgemini, *IAF v4.5 overview V1.0 artifacts & views*. 2006, Capgemini. p. 1.

73. Zachman, J.A., *A framework for information systems architecture.* IBM Systems Journal, 1987. **26**(3): p. 276-292.

74. Zachman, J.A., *The framework for enterprise architecture: background, description and utility.* Zachman International, 1996.

75. Defense, U.S.D.o., *The DoDAF Architecture Framework Version 2.02*. 2010.

76. Defence, M.o., *MOD Architecture Framework*. 2012, UK Government.

77. *NATO Architecture Framework Version 3*. 2007.

78. Williams, T.J., *Development of GERAM, A Generic Enterprise Reference Architecture and Enterprise Integration Methodology*, in *Integrated Manufacturing Systems Engineering*, P. Ladet and F. Vernadat, Editors. 1995, Springer US. p. 279-288.

79. Force, I.-I.T., *GERAM: Generalised enterprise reference architecture and methodology.* IFIP-IFAC Task Force on Architectures for Enterprise Integration March Version, 1999. **1**(3).

80. Kosanke, K., *CIMOSA — Overview and status.* Computers in Industry, 1995. **27**(2): p. 101-109.

81. Vlietstra, J., *A summary of the CIMOSA reference architecture*, in *Architectures for Enterprise Integration*, P. Bernus, L. Nemes, and T. Williams, Editors. 1996, Springer US. p. 70-101.

82. Chen, D. and G. Doumeingts, *The GRAI-GIM reference model, architecture and methodology*, in *Architectures for Enterprise Integration*, P. Bernus, L. Nemes, and T. Williams, Editors. 1996, Springer US. p. 102-126.

83. Williams, T.J., *The Purdue Enterprise Reference Architecture*, in *Proceedings of the JSPE/IFIP TC5/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing*. 1993, North-Holland Publishing Co. p. 43-64.

84. Saha, P., *Analyzing the open group architecture framework from the GERAM perspective.* The Open Group, Tech. Rep, 2004.

85. Noran, O., *A Mapping of Individual Architecture Frameworks (GRAI, PERA, C4ISR, CIMOSA, ZACHMAN, ARIS) onto GERAM*, in *Handbook on Enterprise Architecture*, P. Bernus, L. Nemes, and G. Schmidt, Editors. 2003, Springer Berlin Heidelberg. p. 65-210.

86. Franke, U., et al. *EAF2- A Framework for Categorizing Enterprise Architecture Frameworks*. in *Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD '09. 10th ACIS International Conference on*. 2009.

87. US Department of Defense Architecture Methodology Working Group, *Technical Architecture Framework for Information Management (TAFIM)*. 1993, Versions.

88. Gerber, A., P. Kotze, and A.v.d. Merwe. *Towards the Formalisation of the TOGAF Content Metamodel using Ontologies*. in *Proceedings of the 12th International Conference on Enterprise Information Systems*. 2010. INSTICC.

89. The Open Group, *Archimate 1.0 Specification*. 2009, The Open Group. p. 122.

90. The Open Group, *Archimate 2.0 Specification*. 2012.

91. modelpractice. *General Model Theory by Stachowiak*. 2012 [cited 2015 7th December]; Available from: https://modelpractice.wordpress.com/2012/07/04/model-stachowiak/.

92. Krogstie, J. and H. Jørgensen, *Quality of Interactive Models*, in *Advanced Conceptual Modeling Techniques*, A. Olivé, M. Yoshikawa, and E.K. Yu, Editors. 2003, Springer Berlin Heidelberg. p. 351-363.

93. Scaife, M. and Y. Rogers, *External cognition: how do graphical representations work?* International journal of human-computer studies, 1996. **45**(2): p. 185-213.

94. ISO, *ISO 9000:2015(en) Quality management systems - Fundamentals and vocabulary*, in *quality*. 2015.

95. Wicks, A.M. and C.J. Roethlein, *A Satisfaction-Based Definition of Quality.* The Journal of Business and Economic Studies, 2009. **15**(1): p. 82-97,110-111.

96. Heidegger, M., *Being and Time - A Translation of Sein und Zeit*. SUNY Series in Contemporary Continental Philosophy, ed. D.J. Schmidt. 1996, New York: State University of New York Press. 487.

97. Harman, G., *Technology, objects and things in Heidegger.* Cambridge Journal of Economics, 2009. **1**.

98. Lange, M., J. Mendling, and J. Recker, *An empirical analysis of the factors and measures of Enterprise Architecture Management success.* European Journal of Information Systems, 2016. **25**(5): p. 411-431.

99. Kotusev, S., *Enterprise architecture and enterprise architecture artifacts: Questioning the old concept in light of new findings.* Journal of Information Technology, 2019. **34**(2): p. 102-128.

100. Stamper, R., *Signs, norms, and information systems.* Signs of Work, eds. Holmqvist B. et al, Walter de Gruyter, Berlin, 1996.

101. Yong Ahn, J., S. Ki Kim, and K. Soo Han, *On the design concepts for CRM system.* Industrial Management & Data Systems, 2003. **103**(5): p. 324-331.

102. Davis, A., et al. *Identifying and measuring quality in a software requirements specification*. in *[1993] Proceedings First International Software Metrics Symposium*. 1993.

103. Cambridge Dictionary, *Meaning of “pants”*, in *Cambridge Dictionary*. 2017.

104. Niemi, E. and S. Pekkola. *Enterprise architecture quality attributes: A case study*. in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. 2013. IEEE.

105. Bernus, P., *Enterprise models for enterprise architecture and ISO9000:2000.* Annual Reviews in Control, 2003. **27**(2): p. 211-220.

106. Krogstie, J., O.I. Lindland, and G. Sindre, *Defining quality aspects for conceptual models.* ISCO, 1995. **1995**: p. 216-231.

107. Green, T.R.G. and M. Petre, *Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework.* Journal of Visual Languages & Computing, 1996. **7**(2): p. 131-174.

108. Steinman, S.B. and K.G. Carver, *Visual programming with Prograph CPX*. 1995: Manning Publications Co.

109. Kalkman, C.J., *LabVIEW: A software system for data acquisition, data analysis, and instrument control.* Journal of Clinical Monitoring, 1995. **11**(1): p. 51-58.

110. Goodman, N., *Languages of art: An approach to a theory of symbols*. 1968: Hackett publishing.

111. Moody, D., *The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering.* IEEE Transactions on Software Engineering, 2009. **35**(6): p. 756-779.

112. Lankhorst, M., *Enterprise Architecture at Work: Modelling, Communication and Analysis (The Enterprise Engineering Series)*. 2013, Springer.

113. Moody, D. and J. van Hillegersberg. *Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML family of diagrams*. in *International Conference on Software Language Engineering*. 2008. Springer.

114. Moody, D.L., P. Heymans, and R. Matulevičius, *Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation.* Requirements Engineering, 2010. **15**(2): p. 141-175.

115. Checkland, P. and S. Holwell, *Information, systems and information systems: making sense of the field.* 1997.

116. Schulenklopper, J. and E. Rommes, *Why They Just Don't Get It: Communicating about Architecture with Business Stakeholders.* IEEE Software, 2016. **33**(3): p. 13-19.

117. Wertheimer, M., *Laws of organization in perceptual forms.* 1938.

118. Goldstein, E.B. and J. Brockmole, *Sensation and perception*. 2016: Cengage Learning.

119. Boisot, M.H., *Knowledge Assets: Securing Competitive Advantage in the Information Economy*. 1998: Oxford University Press.

120. Bean, S., *Producing Enterprise Architecture Content that Counts.* Journal of Enterprise Architecture, 2013. **2013**(3): p. 15-20.

121. Kvanvig, J.L., *The value of knowledge and the pursuit of understanding*. 2003: Cambridge University Press.

122. Gordon, E.C., *Understanding in Epistemology*, in *Internet Encyclopedia of Philosophy*. 2020.

123. Guizzardi, G., *On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models*, in *Proceedings of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006*. 2007, IOS Press: Vilnius, Lithuania. p. 18–39.

124. Copeland, B.J., *The Genesis of Possible Worlds Semantics.* Journal of Philosophical Logic, 2002. **31**(2): p. 99-137.

125. Recker, J., *Opportunities and constraints: the current struggle with BPMN.* Business Process Management Journal, 2010.

126. Object Management Group, *Business Process Model and Notation™ (BPMN™) Version 2.0.2*. 2011.

127. Wand, Y. and R. Weber, *On the ontological expressiveness of information systems analysis and design grammars.* Information systems journal, 1993. **3**(4): p. 217-237.

128. Recker, J.C., *Understanding process modelling grammar continuance: a study of the consequences of representational capabilities*. 2008, Queensland University of Technology.

129. Muehlen, M.z. and J. Recker, *How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation*, in *Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE*, J. Bubenko, et al., Editors. 2013, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 429-443.

130. Zarwin, Z., et al., *Natural modelling.* Journal of Object Technology, 2014. **13**(3).

131. Williams, R.V., *Punched Cards: A Brief Tutorial.«.* IEEE annals of the history of computing-Web extra, 2002.

132. Davies, I., et al., *How do practitioners use conceptual modeling in practice?* Data & Knowledge Engineering, 2006. **58**(3): p. 358-380.

133. Bjekovic, M., *Pragmatics of Enterprise Modelling Languages: A Framework for Understanding and Explaining*, in *Institute for Computing and Information Sciences*. 2018, Radboud University: Nijmegen. p. 269.

134. Egyedi, T.M., *Standard-compliant, but incompatible?!* Computer Standards & Interfaces, 2007. **29**(6): p. 605-613.

135. Thalheim, B. *Syntax, semantics and pragmatics of conceptual modelling*. in *International Conference on Application of Natural Language to Information Systems*. 2012. Springer.

136. Clark, H.H., *Arenas of language use*. 1992: University of Chicago Press.

137. Frank, U., *Evaluating modelling languages: relevant issues, epistemological challenges and a preliminary research framework.* 1998.

138. Wyssusek, B., M. Schwartz, and B. Kremberg. *The Philosophical Foundation of Conceptual Knowledge–A Sociopragmatic Approach*. in *Supplementary Proceedings of the 9th Int. Conf. on Conceptual Structures*. 2001. Citeseer.

139. Kotusev, S., *Yet Another Taxonomy for Enterprise Architecture Artifacts.* Journal of Enterprise Architecture, 2019.

140. Kotusev, S., M. Singh, and I. Storey, Journal of Enterprise Architecture, 2017. **13**(1).

141. Maissel, J., *Wanted –A Reference Architecture for Enterprise Architecture Repositories.* Journal of Enterprise Architecture, 2017. **13**(1).

142. Günther, W.A. and W. Heijstek, *Measuring Enterprise Architecture Effectiveness using Key Performance Indicators.* Journal of Enterprise Architecture, 2015. **2015**(1): p. 41-54.

143. Behara, G.K. and P. Palli, *Enterprise Architecture Metrics.* Journal of Enterprise Architecture, 2015. **2015**(3): p. 23-29.

144. Sharp, A. and P. McDermott, *Workflow Modeling: Tools for Process Improvement and Application Development*. 2001: Artech House, Inc.

145. Cameron, B.H. and E. McMillan, *Analyzing the Current Trends in Enterprise Architecture Framework.* Journal of Enterprise Architecture, 2013. **2013**(1).

146. Darling, R., *Reducing Communication Overhead in Enterprise Architecture.* Journal of Enterprise Architecture, 2010. **6**(3): p. 41-53.

147. Suurmond, C. *IT Systems in Business: Model or Reality?* 2018. Cham: Springer International Publishing.

148. Jonkers, H., et al. *Towards a language for coherent enterprise architecture descriptions*. in *Seventh IEEE International Enterprise Distributed Object Computing Conference, 2003. Proceedings.* 2003.

149. Lankhorst, M., *Enterprise Architecture at Work: Modelling, Communication and Analysis*. 2013: Springer Berlin Heidelberg.

150. Bernaert, M., et al., *CHOOSE: Towards a metamodel for enterprise architecture in small and medium-sized enterprises.* Information Systems Frontiers, 2016. **18**(4): p. 781-818.

151. Dam, H.K., L.-S. Lê, and A. Ghose, *Managing changes in the enterprise architecture modelling context.* Enterprise Information Systems, 2016. **10**(6): p. 666-696.

152. Perez-Castillo, R., et al., *A systematic mapping study on enterprise architecture mining.* Enterprise Information Systems, 2019. **13**(5): p. 675-718.

153. Bryman, A. and E. Bell, *Business Research Methods*. 3rd ed. 2011: Oxford University Press. 765.

154. Einstein, A., *Erklarung der Perihelionbewegung der Merkur aus der allgemeinen Relativitatstheorie.* Sitzungsber. preuss.Akad. Wiss, 1915. **47**: p. 831.

155. Heit, E., *Properties of inductive reasoning.* Psychonomic Bulletin & Review, 2000. **7**(4): p. 569-592.

156. *Definition of Positivism*, in *Merriam-Webster*. 2017, Merriam-Webster.

157. Spohrer, J., et al., *Steps Toward a Science of Service Systems.* Computer, 2007. **40**(1): p. 71-77.

158. Vygotsky, L.S., *Thought and language*, ed. M. Cole, et al. 1986, Cambridge, MA: MIT Press.

159. Burge, T., *Philosophy of Language and Mind: 1950-1990.* The Philosophical Review, 1992. **101**(1): p. 3-51.

160. Howard, D., *Who Invented the “Copenhagen Interpretation”? A Study in Mythology.* Philosophy of Science, 2004. **71**(5): p. 669-682.

161. Mawson, T.J., *The rational inescapability of value objectivism.* Think, 2008. **6**(17-18): p. 15-21.

162. Diesing, P., *Objectivism vs. Subjectivism in the Social Sciences.* Philosophy of Science, 1966. **33**(1/2): p. 124-133.

163. Swanson, R.A. and E.F.H. III, eds. *Research in Organisations: Foundations and Methods in Inquiry*. 2005, Berrett-Koehler Publishers, Inc.: San Francisco.

164. Holton, E.F. and M.F. Burnett, *The Basics of Quantitative Research*, in *Research in Organisations: Foundations and Methods in Inquiry*, R.A. Swanson and E.F.H. III, Editors. 2005, Berrett-Koehler Publishers, Inc.: San Francisco.

165. Aspers, P., et al., *What is Qualitative in Qualitative Research.* Qualitative sociology, 2019. **42**(2): p. 139-160.

166. Maxwell, J.A., *Designing a Qualitative Study*, in *The SAGE handbook of Applied Social Research Methods*, L. Bickman and D.J. Rog, Editors. 2009, Sage.
167. Kuckartz, U., *Qualitative Text Analysis - A Guide to Methods, Practice & Using Software*. 2 ed, ed. K. Metzler. 2014: Sage.
168. Howe, K.R., *Two Dogmas of Educational Research.* Educational Researcher, 1985. **14**(8): p. 10-18.
169. Doyle, L., A.-M. Brady, and G. Byrne, *An overview of mixed methods research.* Journal of Research in Nursing, 2009. **14**(2): p. 175-185.
170. Tashakkori, A. and C. Teddlie, *The past and future of mixed methods research: From data triangulation to mixed model designs.* Handbook of mixed methods in social and behavioral research, 2003: p. 671-701.
171. Cresswell, J.W. and V.L.P. Clark, *Designing and Conducting Mixed Methods Research*. 2007, Thousand Oaks: Sage Publications. 274.
172. Caracelli, V.J. and J.C. Greene, *Crafting mixed-method evaluation designs.* New Directions for Evaluation, 1997. **1997**(74): p. 19-32.
173. Leech, N.L. and A.J. Onwuegbuzie, *A typology of mixed methods research designs.* Quality & Quantity, 2009. **43**(2): p. 265-275.
174. Gomm, R., M. Hammersley, and P. Foster, eds. *Case Study Method - Key Issues, Key Texts*. 2000, Sage.
175. Fry, H., S. Ketteridge, and S. Marshall, *A Handbook for Teaching and Learning in Higher Education*. 2 ed. 2003: Kogan Page.
176. Eisenhardt, K.M. and M.E. Graebner, *Theory Building from Cases: Opportunities and Challenges.* The Academy of Management Journal, 2007. **50**(1): p. 25-32.
177. Seawright, J. and J. Gerring, *Case Selection Techniques in Case Study Research:A Menu of Qualitative and Quantitative Options.* Political Research Quarterly, 2008. **61**(2): p. 294-308.
178. Moreland, J.P. and W.L. Craig, *Argumentation and Logic*, in *Philosophical Foundations for a Christian Worldview*. 2003, InterVarsity Press. p. 28-67.
179. Duff, P.A., *Beyond generalizability: Contextualization, complexity, and credibility in applied linguistics research*, in *Inference and Generalizability in Applied Linguistics*, M. Chalhoub-Deville, C.A. Chapelle, and P.A. Duff, Editors. 2006, John Benjamins Publishing. p. 248.
180. Groves, R.M., et al., *Survey methodology*. Vol. 561. 2011: John Wiley & Sons.
181. Wolf, C., et al., *The SAGE Handbook of Survey Methodology*. 2016, London, UNITED KINGDOM: SAGE Publications.
182. Jobe, J.B. and D.J. Mingay, *Cognition and survey measurement: History and overview.* Applied Cognitive Psychology, 1991. **5**(3).
183. Jick, T.D., *Mixing Qualitative and Quantitative Methods: Triangulation in Action.* Administrative Science Quarterly, 1979. **24**(4): p. 602-611.
184. Denzin, N.K., *The research act: A theoretical introduction to sociological methods*. 2017: Transaction publishers.
185. Noble, H. and R. Heale, *Triangulation in research, with examples.* Evidence Based Nursing, 2019. **22**(3): p. 67-68.
186. Cohen, L., L. Manion, and K. Morrison, *Research Methods in Education*. Eighth;8; ed. Vol. 1. 2017, London: Routledge Ltd.

187. Angus-Leppan, P.V., *THE HEIGHT OF MOUNT EVEREST.* Survey Review, 1982. **26**(206): p. 367-385.
188. Leuf, B. and W. Cunningham, *The Wiki way: quick collaboration on the Web*. 2001: Addison-Wesley Longman Publishing Co., Inc.
189. Wikimedia Foundation. *Wikipedia*. 2019 [cited 2020 23rd February]; The Free Encyclopedia]. Available from: https://www.wikipedia.org/.
190. Oppenheim, A.N., *Questionnaire design, interviewing and attitude measurement*. 2000: Bloomsbury Publishing.
191. Rattray, J. and M.C. Jones, *Essential elements of questionnaire design and development.* Journal of clinical nursing, 2005. **16**(2): p. 234-243.
192. Bryman, A. and E. Bell, *Self-completion questionnaires*, in *Business Research Methods*. 2011, Oxford University Press: Oxford. p. 230-246.
193. Tourangeau, R. and T.W. Smith, *Asking Sensitive Questions: The Impact of Data Collection Mode, Question Format, and Question Context.* The Public Opinion Quarterly, 1996. **60**(2): p. 275-304.
194. *Definition of Metric*, in *Collins English Dictionary*. 2020, Collins.
195. Klubeck, M., *Planning a Good Metric*, in *Metrics: How to Improve Key Business Results*. 2011, Apress: Berkeley, CA. p. 57-81.
196. English, T. *Monty Python and the Holy Grail: Airspeed Velocity of an Unladen Swallow*. 2020 [cited 2020 7th November]; Available from: https://interestingengineering.com/monty-python-and-the-holy-grail-airspeed-velocity-of-an-unladen-swallow.
197. Quine, W.v.O., *Set theory and its logic*. Vol. 9. 1969: Harvard University Press.
198. Simon, H.A., *The sciences of the artificial*. Vol. 136. 1969: MIT press.
199. Davis, G.B. and M.H. Olson, *Management information systems: conceptual foundations, structure, and development (2nd ed.)*. 1985: McGraw-Hill, Inc. 693.
200. March, S.T. and G.F. Smith, *Design and natural science research on information technology.* Decision Support Systems, 1995. **15**(4): p. 251-266.
201. Hevner, A.R., et al., *Design science in information systems research.* MIS Q., 2004. **28**(1): p. 75-105.
202. Kalicki, J., *Note on truth-tables.* Journal of Symbolic Logic, 1950. **15**(3): p. 174-181.
203. Bennett, D. *Origins of the Venn Diagram*. 2015. Cham: Springer International Publishing.
204. Steenbergen, M.v., *Maturity and effectiveness of enterprise architecture*. 2011, Utrecht University.
205. Object Management Group, *Business Process Model and Notation™ (BPMN™) Version 2.0*. 2011.
206. Mendling, J., *Event-Driven Process Chains (EPC)*, in *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*. 2008, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 17-57.
207. Chen, P.P.-S., *The entity-relationship model - toward a unified view of data.* ACM Transactions on Database Systems, 1976. **1**(1): p. 9-36.
208. Merriam-Webster Online Dictionary. *Definition of Truth*. 2019 [cited 2020 1st March]; Available from: https://www.merriam-webster.com/dictionary/truth.

209. The Open Group, *The Open Group Architectural Framework (TOGAF) 9.2*, in *Architectural Artifacts*. 2018. p. 319-349.

210. Au-Yong-Oliveira, M., et al. *The Rise of the Unicorn: Shedding Light on the Creation of Technological Enterprises with Exponential Valuations*. 2018. Cham: Springer International Publishing.

211. The Open Group, *ArchiMate 3.1 Specification*, in *Language Customization Mechanisms*. 2019, The Open Group,. p. 116-123.

212. LInkedIn. *LinkedIn ArchiMate Group*. 2020 [cited 2020 6th June]; Available from: https://www.linkedin.com/groups/50758/.

213. Dillman, D.A., *Mail and other self-administered questionnaires.* Handbook for survey research, 1983: p. 359-377.

214. The Open Group, *ArchiMate® 3.0 Specification*, in *Derivation Rule for Structural and Dependency Relationships*. 2016. p. 36-38.

215. The Open Group, *ArchiMate® 3.1 Specification*, in *Relationships (Normative)*. 2019. p. 128-140.

216. Lankhorst, M., *derivations (personal email)*, C. Spence, Editor. 2020.

217. Eades, P. and X. Lin. *Spring algorithms and symmetry*. 1997. Berlin, Heidelberg: Springer Berlin Heidelberg.

218. Lankhorst, M., *Enterprise architecture modelling—the issue of integration.* Advanced Engineering Informatics, 2004. **18**(4): p. 205-216.

219. BiZZdesign. *ArchiMate 3.1 Overview Poster*. 2019 [cited 2020 15th March]; Available from: http://tiny.cc/ArchiMate31Poster.

220. Rumbaugh, J., I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual, The*. 2004: Pearson Higher Education.

221. Lankhorst, M.M., H.A. Proper, and H. Jonkers, *The architecture of the archimate language*, in *Enterprise, business-process and information systems modeling*. 2009, Springer. p. 367-380.

222. Schobbens, P.Y., P. Heymans, and J.C. Trigaux. *Feature Diagrams: A Survey and a Formal Semantics*. in *14th IEEE International Requirements Engineering Conference (RE'06)*. 2006.

223. McGovern, J. and M. Resnick, *Critical Capabilities for Enterprise Architecture Tools*. 2019, Gartner.

224. Nuzzo, R.L., *The box plots alternative for visualizing quantitative data.* PM&R, 2016. **8**(3): p. 268-272.

225. Kim, T.K., *T test as a parametric statistic.* Korean journal of anesthesiology, 2015. **68**(6): p. 540.

226. *Analysis Toolpak*. 2020 [cited 2020 14th November 2020]; Available from: https://www.excel-easy.com/data-analysis/analysis-toolpak.html.