University of Reading

Considering the Feasibility of Semantic Model Design in the Built-Environment

Article

Published Version

Creative Commons: Attribution 3.0 (CC-BY)

Open Access

Grzybek, H., Xu, S., Gulliver, S. ORCID: https://orcid.org/0000-0002-4503-5448 and Fillingham, V. (2014) Considering the Feasibility of Semantic Model Design in the Built-Environment. Buildings (2014), 4 (4). pp. 849-879. ISSN 2075-5309 doi: 10.3390/buildings4040849 (Grzybek, H.; Xu, S.; Gulliver, S.; Fillingham, V. Considering the Feasibility of Semantic Model Design in the Built-Environment. Buildings 2014, 4, 849-879.) Available at https://centaur.reading.ac.uk/38123/

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

Published version at: http://www.mdpi.com/2075-5309/4/4/849

To link to this article DOI: http://dx.doi.org/10.3390/buildings4040849

Publisher: MDPI

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the End User Agreement.

www.reading.ac.uk/centaur

University of Reading

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

Buildings **2014**, *4*, 849-879; doi:10.3390/buildings4040849

OPEN ACCESS

buildings

ISSN 2075-5309

www.mdpi.com/journal/buildings/

Article

Considering the Feasibility of Semantic Model Design in the Built-Environment

Hubert Grzybek [1], Shen Xu [1], Stephen Gulliver [1,*] and Victoria Fillingham [2]

[1] Henley Business School, University of Reading, Reading RG6 6UD, UK; E-Mails: hubert.grzybek2@gmail.com (H.G.); s.xu@pgr.reading.ac.uk (S.X.)

[2] School of Construction Management and Engineering, University of Reading, Reading RG6 6UD, UK; E-Mail: v.l.fillingham@pgr.reading.ac.uk

***** Author to whom correspondence should be addressed; E-Mail: s.r.gulliver@henley.reading.ac.uk; Tel.: +44-118-378-4422; Fax: +44-118-378-4421.

External Editor: Arto Kiviniemi

Received: 3 April 2014; in revised form: 14 October 2014 / Accepted: 26 October 2014 / Published: 4 November 2014

Abstract: Building Information Modeling (BIM) is the process of structuring, capturing, creating, and managing a digital representation of physical and/or functional characteristics of a built space [1]. Current BIM has limited ability to represent dynamic semantics, social information, often failing to consider building activity, behavior and context; thus limiting integration with intelligent, built-environment management systems. Research, such as the development of Semantic Exchange Modules, and/or the linking of IFC with semantic web structures, demonstrates the need for building models to better support complex semantic functionality. To implement model semantics effectively, however, it is critical that model designers consider semantic information constructs. This paper discusses semantic models with relation to determining the most suitable information structure. We demonstrate how semantic rigidity can lead to significant long-term problems that can contribute to model failure. A sufficiently detailed feasibility study is advised to maximize the value from the semantic model. In addition we propose a set of questions, to be used during a model's feasibility study, and guidelines to help assess the most suitable method for managing semantics in a built environment.

Keywords: Building Information Modeling (BIM); semantic; modeling; feasibility

---

1. Introduction

The term semantics is defined in the context of linguistics as the study of meaning. The field of semantics focuses on the relationship between a sign, a word, a sentence or symbol, and what it commonly represents. Whilst semantics is most commonly associated with linguistics, it is also an important area in other domains, e.g., psychology [2], computer science [3], philosophy and metaphysics [4].

The field of metaphysics has been of interest since the earliest philosophers defined taxonomies, which classified types of entities into a number of pre-set categories. Early philosophers made a distinction between types of concepts, *i.e.*, categories of things, and particular instances of those types-a distinction that remains a core part of how semantics is embedded within models. All models and ontologies are therefore conceptual representations "things" that are said to exist, and are of interest within a particular application domain.

Multiple alternative means of classifying concepts in the world exist. Aristotle distinguished ten basic types of entity including: Substance, quantity, quality, relation, place, time, situation/position, condition/state, action, and passion [5]. Kant defined four categories: Quantity, quality, relation and modality [6]; and Grossmann identified eight categories, which consist of: Individuals, properties, relations, classes, structures, quantifiers, facts, negation; many of which still remain in common use of modeling [7]. The variety of classification demonstrates how difficult it is to reach a consensus on conceptual structures. Nonetheless, models, tools, and information systems, rely on conceptual domain representations to define the structure of data that is utilized by systems. Within the construction domain, IFC, for example, defines entities that relate to the physical properties of a built space: e.g., Geometric Constraint, Geometric Model, Geometry, Material Property, Material, Presentation Appearance, Presentation Definition, Presentation Dimensioning, Presentation Organization, *etc.*; as well as entities that support construction planning and project management: Actors, Cost, Date, Time, Time Series, Control, Process, Product, *etc.*, however, does not easily allow building fabric to be linked with building usage, which is of key importance to the ultimate building owner.

A significant disparity appears between models that are formed for use within a specific domain, such as within the building domain (IFC [3], CityGML, gbXML [8], the Art & Architecture Thesaurus [9], UniClass, CI/SfB, *etc.*), and those that are highly abstract and non-domain specific. The value of the model languages rests on a conceptual representation of the domain in terms of important concepts, their attributes, the relationships between them (*i.e.*, the domain semantics), and the rules that govern the structures application to specific modeling scenarios (*i.e.*, language semantics). Alignment between domain semantics and modeling language semantics is therefore key to maximizing model value.

2. Semantics in the Construction Domain

All models should support human activity and support the intended interpretation. As a result, semantics are critical to model value; ensuring they are both easy to use and their results can easily be understood. Without explicit semantics, users are expected to interpret, infer or guess the meaning of model elements. A lack of semantics, however, is common. Computer Aided Design (CAD) systems, for example, provide sophisticated functionality for developing a wide range of drawings, yet most

traditional CAD drawings rely on user interpretation of constructed, primitive geometric shapes; and are not semantically marked-up with relationships and labels.

Within modern CAD there are certain functions that go beyond basic geometric shapes, in order to add meaning to models, e.g., blocks, inserts and layers. Blocks (a set of shapes that have been linked together) simplify design, yet can also support the addition of building model semantics; since blocks can be linked to specific physical objects. Layers are groupings of graphical entities with labels. A building plan may include a specific layer, for example, that relates to CCTV camera objects. By grouping these objects within a specific layer, a user is able to change set characteristics of specific object types, allowing the user to distinguish between objects within the drawing; or even to switch the layer off, causing them to disappear from the visualization. Layers can therefore be used to add semantic depth to CAD drawings. Moreover, since layers have labels, the existence of an object within a layer implies that objects are related to the layer's label.

Designers, however, do not use such features in a coherent or consistent way. CAD software imposes no rules on how blocks, inserts and layers are to be used by designers, which often leads to inconsistency of use. As a result, the drawing interpreter requires intuition and experience to define what designers intended to represent. Existing geometric CAD systems also fail to support explicitly defined relationships between drawing elements and/or non-graphical information. For instance, geometric CAD tools do not support the creation of a label to signify the concept "room". The software maintains information concerning the size and position of the label object; yet no explicit relationship can be formed between the label and the objects that it semantically represents. As a result of this lack of relationship, the model is unable to provide useful functionality to represent these concept types.

Building Information Modeling (BIM), based on object-oriented modeling language principles, was developed in order to capture the semantics of the physical space, and to facilitate a repository of product information shared by construction participants. BIM was advocated as a means of solving many long-standing problems in the construction industry, however the problem of interoperability between BIM tools has frequently been seen as a barrier to the uptake of BIM. To deal with interoperability challenges, a number of information standards have emerged.

The Industry Foundation Classes (IFC)—and its derivative Construction Operations Building Information Exchange (COBie)—is the standard mandated by the UK Government, as a method for converting BIM models into, and exported BIM models from, various applications. The IFC object model was developed to represent concept types in the building domain, and combines 3D geometric building plan visualizations with rich, relational semantics, thus making the format both human and machine-readable (in Extensible Markup Language (XML)). IFC, however, has limited capability for capturing object-based semantics in context of human activity, which limits the potential for automated analysis of building use. Moreover, IFC does not support complex or multiple stakeholder views/interactions, and does not support consideration of building activity context; an issue that limits the use of IFC with intelligence tools as part of a whole life solution. Moreover Cheng, Trivedi, and Law (2002) [10] report the unsuitableness of XML in the scheduling ontology, identifying problems converting Process Specification Language (PSL) to XML; because of the time concept in scheduling.

There appears limited consideration for the alignment between language semantics and domain semantics, with inadequate research considering the suitableness of the language. Accordingly, we

conducted a literature review concerning the ontologies used in the building sector. We based our search on several keywords, including: Semantics model, construction, building, built-environment, ontology, resource description framework (RDF), Web Ontology Language (OWL), IFC, *etc.* From 2002 to 2014, we identified more than 130 papers that meet the criteria, however in order to maintain the scope of this paper, we reduced the literature base to 105 papers (see Table 1); as 25 papers did not focus on domain ontology, but instead discussed system architecture and web-services. Since ontology engineers in the built-environment do not appear to employ ontology languages consistently, each paper was tagged for its domain activity, grouped into the same domain problems and classified into the built-environment. Traditionally construction software used data models based on entity-relationship specifications [11]. In our review we identified several key ontology languages, including: XML, which is extended from Generalized Markup Language (GML); SL, which is based on Knowledge Interchange Format (KIF); OWL, which is based on RDF [12]; ifcXML, which is an XML format defined by ISO 10303-28 (STEP-XML) [13]; Conceptual models, which are models made of the composition of concepts abstracting from the domain concepts–though typically papers do not specify the language; CityGML, which is an XML-based language; Suggested Upper Merged Ontology (SUMO), which is written in the SUO-KIF language [14]; Taxonomy, which was developed as part of the e-COGNOS project (Consistent knowledge management across projects and between enterprises in the construction domain—IST–2000-28671); Reference model, which is an abstract framework or domain-specific ontology consisting of an interlinked set of clearly defined concepts; Description Logic (DL), which is a logic-based model derived from first-order logic (FOL); and IFC-based ontologies, which are based on IFC data structures, but where the ontology language have not been clearly specified.

Table 2 presents ontology languages in different built-environments. The numbers within Table 2 represents the number of identified occurrences of different classification structures described in recent literature. OWL was identified as the most accepted ontology language, across a range of industries, and is also dominant when undertaking built-environment classification; however detailed built-environment classification is beyond the scope of this research paper. Definition of ifcXML and CityGML, within Table 2, is certainly not a clear-cut decision, and it is important that the reader understands the issues involved. XML tagging is used in both, subject to the xsd schema, to define both object and non-object information. Although a XML Schema is employed for ontology modeling, and ultimately helps the user specify the resultant file structure (ISO 10303-28, 2007), both ifcXML and CityGML primarily describes object data. Although many might disagree with our logic, the authors have decided to place both ifcXML and CityGML under the "OO" definition. In Table 3, we provide supporting evidence that even when undertaking the same activity there are different structures being employed to model the domain semantics in the construction industry. Thus the question arises: What is the typical process of knowledge engineering, and how could the process be aligned with domain semantics, especially corresponding to its domain activities?

Table 1. Sematic models of professional domains in the built-environment.

Sources	Tagged Activity	Domain Problem	Built Environment
[15]	Designing	Architectural Design	Building Design
[16]	Engineering		
[17]	Designing	Conceptual Design	
[18]	Appraising	Energy Efficiency	
[19,20]	Collaborating	Interoperability	
[21]	Checking	Regulation	
[22]	Designing	Structural Design	
[23]	Monitoring	Building	Building Performance
[24]	Managing	Defect Management	
[25–38]	Appraising	Energy Efficiency	
[39]	Checking	Regulation	
[40]	e-Learning	Knowledge Representation	Construction Project Management
[41]	Learning		
[42]	Sharing		
[43,44]	Managing	Construction	
[45–51]	Estimating	Cost Estimation	
[52]	Integrating	Interoperability	
[53]	Managing	Plant Construction	
[54]	Collaborating	Process Management	
[55]	Exchanging	Project Management	
[56–59]	Managing		
[60]	Sharing		
[10,61]	Scheduling	Project Schedule	
[62–67]	Checking	Regulation	
[68,69]	Appraising	Risk Management	
[70,71]	Managing	Site Management	
[72]	Controlling	Spatial Relation	
[73]	Modelling	Architectural Reconstruction	Existing Building
[74]	Optimizing	Energy Consumption	Facility Management
[75]	Simulating	Hydrodynamic Processes	
[76–78]	Collaborating	Interoperability	
[79,80]	Collaborating	Knowledge Representation	
[81]	Sharing		
[82–84]	Collaborating	Interoperability	General Construction
[85–87]	Implementing		
[88]	Integrating		
[89]	Conceptualizing		
[90–93]	Modelling	Context	Smart homes
[94]	Controlling	Home Automation	
[95]	Collaborating	Interoperability	
[96]	Implementing	Building Environment	Sustainability
[97,98]	Appraising		
[99]	Collaborating	Interoperability	
[100]	Sharing	Knowledge Representation	
[101]	Managing	Carbon management	

Table 1. *Cont.*

Sources	Tagged Activity	Domain Problem	Built Environment
[102]	Modelling	Architecture Design	Urban Design
[103,104]	Modelling	Civil Engineering	
[105]	Planning	Design	
[106]	Researching	Energy Efficiency	
[107]	Planning	Structure	
[108–110]	Collaborating	Interoperability	
[111]	Appraising	Landscaping	
[112]	Designing		
[113]	Planning	Panels Management	
[114]	Planning	Project Plan	
[115]	Identifying	Slum	
[116]	Inferring	Spatial Relation	
[117]	Planning		
[118]	Identifying	Structural Design	
[119]	Modelling	Urban Modelling	

Table 2. Ontology languages in different built environments.

Language & Structure \ Domain	Building Design	Building Performance	Construction Project Management	Existing Building	Facility Management
Frame, OO	**6**	**11**	**19**	**1**	**5**
Not specified		1			
ifcXML			1		1
CityGML					
OWL	5	10	18	1	4
OWL 2	1				
UML + OWL					
Hierarchy			**3**		
Not specified			2		
Taxonomy			1		
Network			**2**		**1**
Conceptual modelling					1
PSL			2		
Tagging		**1**	**1**		
Not specified					
XML		1	1		
Generic	**2**	**2**	**5**		**1**
Not specified	1		1		
Reference model		1			
Suggested Upper Merged Ontology (SUMO)		1			
IFC-based	1		4		1
Not specified		**2**	**4**		
Not specified		2	4		
Relational		**1**			
DL					
Not specified		1			
Grand Total	**8**	**17**	**34**	**1**	**7**

Table 2. *Cont.*

Language & Structure \ Domain	General Construction	Smart Homes	Sustainability	Urban Design	Total
Frame, OO	**7**	**4**	**2**	**12**	**68**
Not specified				1	2
ifcXML					2
CityGML				1	1
OWL	7	4	2	9	60
OWL 2				1	2
UML+OWL				1	1
Hierarchy					**3**
Not specified					2
Taxonomy					1
Network					**3**
Conceptual modelling					1
PSL					2
Tagging			**1**	**2**	**4**
Not specified			1		1
XML				1	3
Generic			**2**	**1**	**13**
Not specified			1	1	4
Reference model					1
Suggested Upper Merged Ontology (SUMO)					1
IFC-based			1		7
Not specified		**1**		**3**	**10**
Not specified		1		3	10
Relational	**1**	**1**	**1**		**4**
DL	1				1
Not specified		1	1		3
Grand Total	**8**	**6**	**6**	**18**	**105**

Note: Frame, OO, frame-based object-oriented model; all the structures will be explained in later section.

Table 3. Classification Structures to Domain Activities.

Domain Activities \ Language & Structure	Frame, OO	Generic	Hierarchy	Relational	Network	Tagging	Not specified	Grand Total
Appraising	14	2		2			2	20
Checking	6						2	8
Collaborating	10	1			1	2	1	15
Conceptualizing	1							1
Controlling	2							2

Table 3. *Cont.*

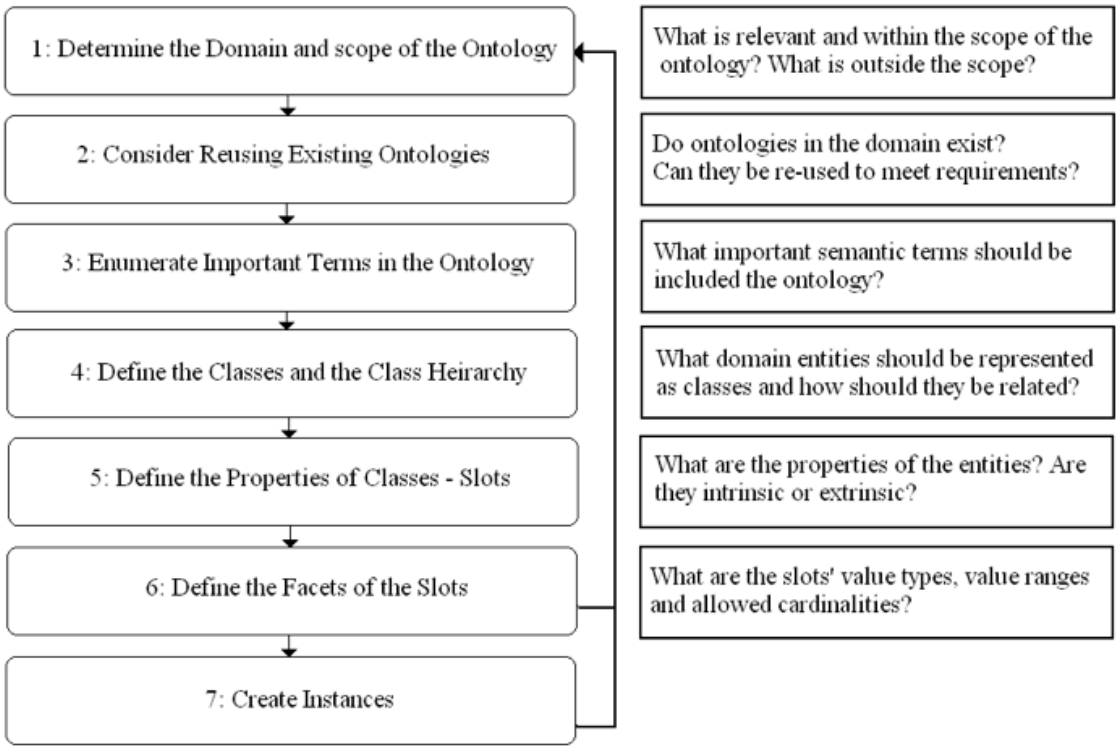
Domain Activities \ Language & Structure	Frame, OO	Generic	Hierarchy	Relational	Network	Tagging	Not specified	Grand Total
Designing	3	1					1	5
e-Learning	1							1
Estimating	1	4	1				1	7
Identifying	1					1		2
Inferring	1							1
Integrating	4		1	1				6
Learning	1							1
Managing	7	1	1		1	2		12
Modelling	6	1		1		0	1	9
Monitoring	1							1
Optimizing		1						1
Planning	4						1	5
Researching							1	1
Scheduling					1	1		2
Sharing	2	2						4
Simulating	1							1
Grand Total	**65**	**13**	**3**	**4**	**3**	**7**	**10**	**105**

3. An Ontology Development Guide

The ontology development guide, developed by Noy and McGuinness [120], divides ontology selection into seven steps that should be completed during the initial phase of model development (see Figure 1). Whilst Noy and McGuinness define their process as iterative, the precise nature of iteration was left unclear. It is expected that all steps are completed in the correct order during the initial design phase. It is possible, however, in successive iterations for designers to change the order of the steps, or choose to avoid steps if no change has occurred. The first step of the process aims to identify the domain and scope of the ontology. This step should involve the designers answering a number of questions: What will the ontology represent? What will the ontology be used for? What types of questions should the information in the ontology aim to answer? Who will use and maintain the ontology? The first two questions are straightforward, since the intention for the model ontology should be based on a pre-specified or understood requirement. The third question, however, is more complex, as it requires designers to list the types of questions that the semantic ontology should be capable of answering. This list should be comprehensive in order to avoid later-stage re-designs if the ontology is unable to provide answers to important questions. Listing such questions will not only assist in specifying the requirements for the ontology, but also provide a means of testing whether the completed design has a good fit with the modeled space and its intended purpose. The second step of the process involves the identification and consideration of existing ontologies that represent the same domain. In most cases it is more beneficial to re-use, modify or extend existing ontologies than developing a new one. This is particularly important if the intended system is expected to interact with

other systems within the same domain, as semantic discrepancies can be avoided by establishing a single domain ontology.

Figure 1. Steps in Ontology Development Process based on (Noy and McGuinness, [120]).



If no suitable ontology exists, designers should move to step three, which requires the listing of important semantic terms that should be included in the ontology. The terms in this list should be based on concepts that they wish to make statements about, or wish to explain to a user, including any relevant properties. The fourth and fifth steps aim to arrange these concepts into a coherent conceptual model, with step four focusing on the definition of concept types (also commonly referred to as "classes") and a class hierarchy and step five focusing on class properties (or "slots"). The development of the class hierarchy involves identifying concept types at different levels of abstraction and relating them together to form a hierarchical structure. This process can be approached using either a top-down, bottom-up or middle-out approach. A top-down approach involves identifying the most abstract concepts first and then relating their more specific "sub-classes", whilst a bottom-up reverses this by focusing instead on the most specific types and then identifying their more abstract "super-types". A third alternative, which is favored by [121,122], is to use a "middle-out" that focuses primarily on the most fundamental concepts used within the domain, *i.e.*, those that are most commonly used and those that will constitute the middle levels of the class hierarchy, before moving on to more abstract and more specific concept types.

The fifth step involves the definition of class properties. The properties for each class should describe the internal structure of the concept type. For this step it is advised that designers look back at the list developed in step three, since the concepts that have not been added to the class hierarchy are likely to represent class properties. It is important to relate properties with the most general classes that may have the property, since all sub-classes will automatically inherit the property from their parent

class. Having defined the class hierarchy and class properties, the sixth step focuses on the facets of properties. For each property, it is important to identify the data type to be used for holding the value for each instance, e.g., string, integer or floating-point number, the number of allowed values (cardinality), domain range, and minimum and maximum allowed values. Finally the seventh phase creates an instance of the concept type in the ontology. Instantiating classes can be performed once the ontology is complete or at the end of an iteration to test whether the ontology meets its requirements, before moving on to the next round of model development.

4. Classification Structures Used in Conceptual Modeling

Shirky proposed criteria for assessing the suitability of ontological classification [123]. In terms of the domain to be modeled, the key issues affecting the suitability of ontological classification include:

- Domain corpus (the number of important concept types in the semantic domain);
- Presence of formal categories (the extent to which categories are simple and derivable);
- Stability of entities (possibility or likelihood of entity types change);
- Restricted or unrestricted entity set (whether the set of important entities is inherently limited or grows continuously);
- Presence of clear boundaries between entities.

Shirky argues that, for building space with a small set of constituent entities, the presence of formalized categories and a highly stable and restricted set of entities with clear edges is well-suited to ontological classification. As adoption of the wrong classification scheme can lead to project failure, or to the retrospective need to re-engineer the building model due to a lack of flexibility in the resulting design, it is important that the correct classification scheme is selected. In this section we consider restricted subsets of classification schemes, to allow the reader to see how each could impact use of semantic information in building models.

4.1. Hierarchical Classification Schemes

A hierarchical classification scheme maps data into a tree structure. Each node has one parent node and may have multiple children nodes. Each child node represents the sub-categorization of the parent node, by facilitating increased categorization complexity. Although Noy and McGuinness' traditional design approach is appropriate in many domains, the adoption of hierarchical structures leads to significant problems when faced with ongoing modifications, extensions or adaptations; such as in the case of building models.

Shirky [123] argues, however, against the use of hierarchical schemes of classification, arguing that such classification schemes are imperfect due to related "context errors". Shirky states that it is difficult to design ontologies for expected capabilities in advance, as it burdens the designers with the task of understanding what the end user is thinking, and making appropriate predictions about future needs. It is worth mentioning that whilst ontologies are likely to need modification during their use, Noy *et al.* [120] conceded that maintaining ontologies is a significant challenge, and must also be considered by model designers; particularly for domains where fundamental structures are expected to

change, such as construction. Shirky does, however, accept that in certain scenarios, ontologies provide an excellent solution and these cases can be assessed using a number of criteria.

Hierarchical ontological classification schemes seem appropriate where expert users can be guaranteed, *i.e.*, those who act within a coordinated environment, and who have an authoritative source of judgment, however in modeling scenarios where users are uncoordinated, limited value can be gained from a hierarchical classification schemes.

Overall, Shirky's arguments are heavily critical of ontologies, showing that use of hierarchical structures is inherently limited. Shirky highlights that hierarchical ontologies are difficult to design and that their representation of the target domain is likely to become increasingly inaccurate over time as the domain evolves.

4.2. Network Models

Holub [124] argued against the use of hierarchical models, stating that the lost of inheritance leads to a loss of flexibility, tight coupling between objects and components, and the fragile base class problem. The fragile base class problem is an issue that results from tight coupling of components, and it is introduced through hierarchical object models. As a result of this tight coupling, change made to a base class may not affect that class itself, yet semantic errors may occur when related sub-classes inherit behavior. The added risk of introducing bugs into a group of classes, through change made to their parent class, reduces the semantic modelers' ability to make change.

Inheritance supports specialization, however does not allow natural relationships to be modeled. Network models, although largely hierarchical, allows each record to have multiple parent and child records, permitting the formation of pointer-based graph structures; a collection of nodes (vertices), with connections (pointers) between them. Although multiple inheritance and graph structures allow a more natural modeling of relationships between entities, tight coupling can occur. If, for example, data in object A is dependent on B, B data is dependent on C, and C data is dependent on A, any data being updated could have a pervasive impact.

The concept of coupling revolves around the interdependence of components [125]. Tight coupling implies that model components are highly dependent upon each other. This results in an increased level of complexity when attempting model change; as change to a single component potentially has a resultant effect on all components that depend upon it. Loose coupling avoids this situation by ensuring that dependencies between components are minimized, and that components are able to operate with no or little knowledge of the structure of other components.

Although widely implemented in the 1970s, network models are often also coupled with application functionality. Any changes in data structure results in a need to change the application. Subsequently, a shift towards relational-based models occurred in the 1980s, as data content and application function can be abstracted.

4.3. Relational Models

Relational models, proposed by Edgar Codd [126], represent data that is grouped into relations via use of association. Using a relational model allows the user to specify, query, and project tables of information from predefined data structures without having an understanding of how data is physically

stored. Although relational data classification schemes are ubiquitous, there are a number of key limitations when using relational models to model built space; *i.e.*, relational models are non-flexible and non-temporal in nature.

Since the data structure has to be defined before use, the model designer should, to maximize the value of the model, know explicitly the information that is to be contained in the model in advance of development, and limit subsequent changes in scope; not practically possible if the physical space changes significantly. In addition to limitations in handling change, relational models do not naturally capture information about time; and are therefore not ideal when understanding interaction. Relational model data normally reflects the current state (the AS IS), however, a change in state (the AS WAS) is not captured in the dataset unless the model is specifically designed to monitor such change. If you wanted to identify why people were opening the window, for example, then you would need to identify what attributes need be captured, as the building model reflects only the current state of the model, and the relational structure would have to be redesigned and extended to capture this temporal data.

With the construction industry ubiquitously embracing BIM, and with an increased focus of IFC on management of information through-life (*i.e.*, in both "construction" and "operation" phrases), it seems clear that capture of user activity data is critical to support facility management (*i.e.*, space and cost optimization) via use of big-data and data mining technologies.

4.4. Object-Oriented Model Design

Holub [124] argues for the use of interfaces, which allows access to object attributes/functions without specifying any actual behavior. Interfaces simplify change in the data structure, and do not suffer from replicated error; as is the case with inheritance-based object models. The key to capturing semantics effectively is to ensure that the structure is sufficiently flexible to support future change. Holub [124] argues that the use of interfaces lies at the heart of this flexibility, and that the common use of concrete classes forces modelers to use predefined semantic structures, thus restricting the potential for future change, and increasing the risk of long-term failure.

Whilst the construction domain may seem explicit, it is difficult to reach agreement on the exact nature of data structures. As a result, the problem of explicitly defining a domain's structure in terms of concepts and their relationships can be identified as a "messy", "ill-defined" or "ill-structured" problem; since there is typically a wide difference of opinion as to how the problem should be resolved and there are often multiple viable solution alternatives [127]. In practical terms, this means that different individuals developing conceptual models defining the structure of built space develop a number of distinct solutions, each of which will be more or less acceptable to others in the group and, crucially, it will be difficult to reach a consensus regarding which solution fits the domain best; hence the development of multiple construction object-orientated ontologies (e.g., CityGML, gbXML, *etc.*), which all have pro's and con's for specific user groups.

As a result of these issues, guidelines for developing conceptual models, such as those proposed by Noy and McGuinness [120], encourage an iterative development approach where each iteration is assessed in order to ensure that a consensus between key stakeholders is reached. The authors also advise the adoption of conceptual structures based on a consideration of the model's purpose and any

envisaged future semantic extensions. These supplementary considerations act as a method of reducing conflict and uncertainty when faced with a number of viable alternatives. This advice implies that conceptual building models are inherently biased in terms of representing the particular perspective of the model designers, the original purpose of the building, and the predictions made by the model designer as to the evolution of the conceptual structures and their anticipated future uses.

Conceptual models are therefore limited in that they support a particular view of a building's semantic structure, and are designed to favor certain predicted extensions over others. Shirky [123] provides detail concerning the types of scenarios that benefit from using ontologies, and those that should use alternative methods. Shirky argues that when modeling scenarios where ontologies are unlikely to succeed, such as the classification of web content, alternative approaches to semantic description, such as the use of "tagging" or a generic data model, should be used.

4.5. Tagging

Tagging is the process of creating links to content, and attributing a descriptive semantic label to that link. The label associated with the tag is then used as a semantic description of the referenced resource. The end result of tagging by numerous users is a system of classification that has been developed collaboratively by a group of people [128]. In practice, the process of semantically marking up content using tagging is, in many ways, the reverse of adopting an ontology-based approach. Specifically, unlike ontologies, tags can be created *after* the content has been published, and anyone is free to tag the content using terms that they feel suitably describes that content. As a result, semantic tagging is much more organic, and can rely on communities of users to label content. Tagging, therefore, does not suffer from the problems commonly associated with ontologies, *i.e.*, the ontology designer needing to predict future needs. Also, rather than relying on a small body of experts to develop a standardized view of the conceptual structure of a domain, a large group of non-expert users can be used to accomplish the same task, with the most commonly used labels becoming the strongest descriptors of domain concepts.

Shirky [123] sees the distinction between ontologies and tagging as a fundamental philosophical question. Specifically, whether the world makes sense or whether we make sense of the world. Ontology designers adopt the view that the world makes sense. However, any alternative perspective of the domain's structure requires formal reconciliation, which is not a trivial issue and often results in semantic interoperability problems between related, and possibly overlapping, ontologies. Tagging begins with the perspective that people make sense of the world. As a result, its foundation does not rely on a single perspective of the domain's structure, and every perspective is treated as equally valuable as its alternatives. The value of tagging comes from aggregating these varied perspectives and feeding the results back to the users. Whilst Shirky's support for the use of tagging is limited to managing the semantics of web-based content, it has already been used in a wide range of more specific contexts. Spiteri's results [129], for example, echo those of Shirky, suggesting that tagging can provide a powerful and flexible means for increasing the user-friendliness and interactivity.

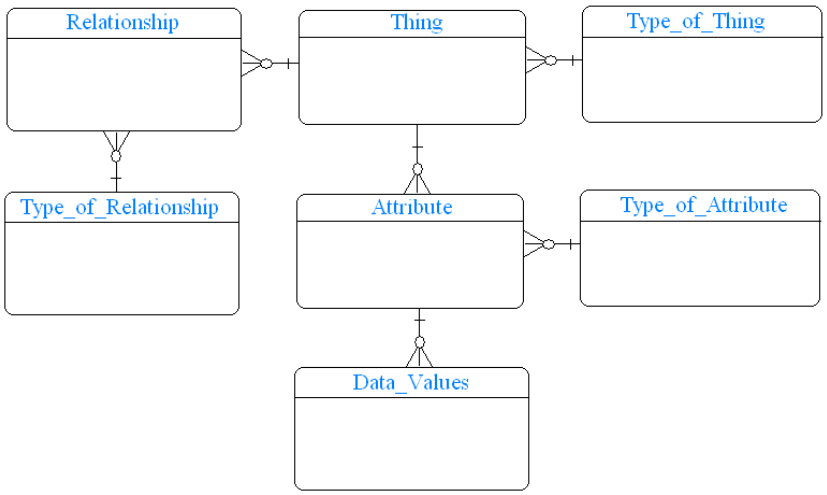
Traditional data modeling techniques are often very similar to the ontology guide presented by Noy *et al.* [120]. The result of the design phase is a conceptual data schema that represents concept types, as well as their attributes and relationships. The implementation of the conceptual data schema

then relies on the designers creating an appropriate table structure to support future use; *i.e.*, inclusion of correct tables, related attributes, and the links between the tables within the chosen database system. The implementation of conceptual data models, by mapping concepts to database tables, concept attributes to table columns, and relationships with primary and foreign keys, results in a reduced level of flexibility within the resultant design; due to tight coupling between the data schema components themselves and the semantic use [130]. Minor change to table structure requires subsequent complex modification to any related tables as well as any triggers, stored procedures and associated code that accesses the table within the application logic. An alternative to traditional data modeling is the adoption of a generic data model, such as entity-attribute-value (EAV) [131].

4.6. Generic Data Models

Generic data models provide a generic structure where both concept type structures and their instances are managed as records within generic tables. Typical EAV schema implementations, for example, feature around six tables, three of which represent conceptual structures of entities, attributes and relationships, with the other three being used to store instances. For example, we can create the concept type "person" as a record in the concept type table, and instances of this type in the instances table. Figure 2 shows an example EAV data schema where the "Type_of" prefix denotes tables used for storing concept types, whilst the remaining tables are used to store instances of those concepts.

Figure 2. Example EAV conceptual data model.



Generic data model structures contrast significantly with the traditional database design and implementation method; where the structure of concept types is represented by table structure, and concept instances as records within those tables. This distinction may initially seem unimportant, however, it has profound consequences for the flexibility of the overall design. For instance, any minor modifications to existing entities, attributes or relationships on a schema using standard development techniques will result in the need to modify all coupled components within both the database and semantic logic. Conversely, change to generic data model schema requires significantly fewer modifications since the conceptual definition of concepts, their attributes, and their relationships are stored explicitly as records within generic table structures, rather than implicitly through non-generic

table structures. As a result, by using the generic data model schema, any changes to the conceptual semantic model of the domain are represented by the database, and no change is required to the underlying database tables; thus reducing coupling between system components, and facilitating future change and maintenance.

The adoption of a generic data model means that change to the conceptual domain model does not need to be reflected in other system components. For instance, adding attributes to a concept type will require semantic logic to be modified in order to take the new attribute into account, however by recording the conceptual structures explicitly as records within generic tables, far fewer system changes must be made to accommodate modifications to the conceptual representation of the domain. As a result of these benefits, various versions of generic data models have been used in domains that feature sparse (*i.e.*, including missing values) and/or heterogeneous data sets, or where the conceptual structure of the domain is expected to change on a frequent basis.

Despite the benefits of adopting generic data models, and the fact that such approaches are already fully supported by most database vendors, generic data models are rarely adopted due to a number of associated disadvantages. Databases using the generic data models risk poorer performance and some scalability issues, which prevent the use of valuable database features such as foreign keys [132]. Generic data models also rely on additional code to manage the generic data model and require the development of more complex queries [133]. Despite these issues, however, generic data models have been successfully applied; for example, the company "4 projects" integrates IFC entities within a Cassandra database (a generic data model) to support flexible integration of information from disparate stakeholders. Cassandra allows tables to be created, dropped, and altered at runtime without blocking updates and queries, facilitating a level of flexibility that is not achievable using XML or relational data models.

5. Defining Solution Feasibility

Sterman [134] produced a set of questions that model designers should seek to answer:

- What is the problem?
- What is the boundary of the model? What factors are endogenous? Exogenous? Excluded? Are soft variables included? Are feedback effects properly taken into account? Does the model capture possible side effects, both harmful and beneficial?
- What is the time horizon relevant to the problem? Does the model include components that may change significantly over the time horizon?
- Are people assumed to act rationally in order to optimize their performance? Does the model take non-economic behavior (organizational realities, non-economic motives, political factors, cognitive limitations) into account?
- Does the model assume people have perfect information about the future and about the way the system works, or does it take into account the limitations, delays and errors in acquiring information that plague decision makers in the real world?
- Are appropriate time delays, constraints and possible bottlenecks taken into account?
- Is the model robust in the face of extreme variations in input assumptions?

- Are the policy recommendations derived from the model sensitive to plausible variations in its assumptions?
- Are the results of the model reproducible? Or are model results adjusted (add factored) by the model builder?
- Is the team that built it currently operating the model? How long does it take for the model team to evaluate a new situation, modify the model, and incorporate new data?
- Is the model documented? Is the documentation publicly available? Can third parties use the model and run their own analyses with it?

Prior to the creation of a building model, a sufficiently detailed feasibility study is advised to ensure use of correct data structures, in order to maximize the value of semantic information. In the following sections we have also developed a set of questions that should be considered as part of the project's feasibility study. The aim of these questions is to draw the building model designers' attention to issues that may be initially overlooked, but which can quickly reduce the validity of a model. It is likely that if such feasibility studies were more commonplace, many building projects would be cancelled before designs were developed. Such analysis aims to reduce wasted effort and resources in the developing of models that, ultimately, can never achieve their target objectives.

5.1. Assessment of the Model's Purpose

Models must have a clear purpose, which should relate to a particular stakeholder need. Since models are abstractions of real world space and systems, the objective of a building model can never be the representation of an complex social or interactive spaces. Total modeling is likely to be futile, since the resulting model will be as complex as the real-world space, and therefore will be as difficult to understand and validate. Consequently the building model designers at the beginning of a project should ask questions such as:

- Is the purpose of the building model clear and well defined?
- Does the purpose entail a large model boundary? If so, is assess whether resources and time available to develop a sufficiently detailed model in order to meet its requirements.
- Given the assessed complexity of the problem, what is the likelihood that the building model will be semantically appropriate, given the state of the art in modeling techniques, technologies and the project's context (team, available resources, support *etc.*).
- Is a building model necessary or adequate in solving the construction problem?

5.2. Domain Assessment

One of the key issues that should be considered when assessing the building semantic modeling feasibility is the domain being represented. Specifically, the designers should assess the complexity of the domain, including the domain's size, number of conceptual entities, their attributes and possible relationships, number of feedback effects between variables, number of interrelated sectors that affect one another and complexity of system behavior. Questions in this phase should include:

- Is the semantic domain to be addressed by the model clear?

- What is the size of the domain in terms of the number of variables, entities, attributes and relationships to be represented?
- To what extent are the building model and semantic domain stakeholders in agreement as to its structure, as expressed via, for instance, ontology?
- Is the semantic domain highly interrelated with other sectors? How does this complicate the building model? What would be the effect of treating related sectors as exogenous and will this affect the model's validity?
- What is the perceived difficulty of modeling the structure or behavior of the domain?

5.3. Difficulty of Accurately Representing the Domain: Physical Versus Social Systems

Many of Sterman's arguments [134] against optimization and simulation models are limited to those addressing domain semantics, which involve human actors and decision-makers, such as government policy development, ecological and economic models. These domains are likely to involve more complexity since human decision makers don't always behave rationally and are, to some extent, unpredictable. Models used to solve physical domain problems, e.g. considering the physical and chemical properties of the building, are more definable, separable and verifiable. Solutions to physical problems can often be found more easily than their social counterparts due to the relative predictability of the physical world's behavior. Consequently, semantic model designers should be aware of the added complexity of attempting to develop models to represent social systems in physical space, and should ask feasibility questions such as:

- Is the team intending to develop the model aware of the differences in complexity and methods between developing social and physical models?
- Is the team sufficiently skilled in the relevant domains, *i.e.*, physical/social, to develop an accurate model?

5.4. Understanding and Assessment of Alternative Modeling Methods

As Sterman identified [134], many modeling projects are initiated using methods that are not suitable to solve the objective. It is possible that in many such cases, semantic model designers simply opt for their preferred method without considering other, possibly more suitable, alternatives. As a result, model designers should aim to have an understanding of a range of different modeling methods, such as simulation and optimization, including their relative capabilities, advantages and disadvantages before selecting the method to be used for a particular semantic issue. Feasibility questions related to modeling methods should include:

- Which semantic modeling methods could be used to solve the problem?
- What are the capabilities, advantages and disadvantages of using a particular method?
- Will the results that may be gained from adopting a particular method meet the model's purpose? For example, optimization models cannot be used to predict behavior.
- Which methods are applicable to the semantic domain?

5.5. Model Boundary

The problem of defining a suitable model boundary is difficult, and is applicable to all types of models. This process requires the presence of a clear problem definition, but within complex domains, such as the construction industry, it can be extremely difficult to identify the effects that one model component may have on other components. The focus of this phase should be on identifying key variables and relationships within the domain, identifying which should be modeled as endogenous and exogenous, and the effects of these choices. Questions related to the model's boundary should include:

- What are the key variables and relationships in the problem domain?
- Which factors should be included as endogenous, which as exogenous?
- Are there key factors or a significant number of factors, which can only be represented, as exogenous? If so, to what extent will this undermine the model's semantic validity?
- What other domains/sectors are related to the problem domain via feedback relationships? Should these also be included in the model?
- Sensitivity analysis: To what extent does alternative plausible estimates of exogenous variables affect the model's output?
- How frequently are the exogenous variables likely to change, what impact will this have on the model's results and semantic validity?

5.6. Analysis of Required Input Data

Based on the previous questions, designers should focus on the ability to find or create input data for the model. If problems arise in this section, it is likely that the validity of the model semantics will be significantly reduced. Feasibility questions for this section include:

- Is data to be used as input for the model available or does it have to be collected?
- How easy is it to collect input data, if required?
- If the data is already available, is it accurate, complete, up-to-date, unbiased and from a reliable source?
- If the data must be collected, does it consist primarily of hard or soft variables? Does the modeling team have sufficient expertise to collect data?

5.7. Reliance on Assumptions

Model designers should list all assumptions concerning intended building model semantics; i.e. input data, modeling method and underlying theories. Tasks for this section should include: identify and record as many assumptions as possible relating to the modeling project, the quality and quantity of input data, underlying theory, model boundary and the nature and behavior of the real-world system.

5.8. Assessment of Modeling Team

The assessment of the modeling team is a more practical issue, but nonetheless an important one. It is important to identify whether the modeling team has sufficient expertise and range of skills to successfully complete the modeling project. Questions for this phase should include:

- Which skills are required to complete the project successfully?
- Expertise in which domains is required/preferred?
- Does the team have sufficient expertise in all required areas?
- If the model's problem domain includes multiple related sectors, seek out experts in related fields and gain a better understanding of the number and importance of possible feedback effects. Also consider modifying team and process to develop the model using a multi-disciplinary approach as part of an effort to broaden the model's boundary and increase its validity.
- If the proposed model design ignored related domains, what effect is this likely to have on the model's semantic validity?

5.9. Inclusion of Delays and Human Error for Optimization and Forecasting Models

As identified by Sterman [134], models seeking to provide advice as to what should be done in the future, given a particular scenario, should include both delays and the likelihood of human error to affect the choice's implementation. For instance, models advising on optimal combinations of energy sources should include the available power sources, as well as the costs associated with building new facilities in order to take advantage of change in resource combinations. Questions in this phase should include:

- What types of delay are significant when implementing the guidelines, which will be produced by the model?
- What values should be assigned to the delay lengths and are these delays fixed, or, are they variable and dependent on other factors?
- How can we represent the effects of human error in implementing the model's advise? How will errors affect delays?

5.10. Technical Considerations

Model designers should also assess whether the proposed modeling project includes any issues that require novel or unusual technical requirements. In such cases, designers should assess whether suitable technology exists, is sufficiently mature, supported and can be acquired at an acceptable cost. Technical issues relates to the ability to supply sufficient storage space for managing model semantic data, processing capacity, security features, and clarity concerning the collection, storage and manipulation of unusual data sets. As a result of such failure, model designers should not forget to identify and assess the risks that can lead to project failure, including those that can be generally applied to modeling software.

5.11. Fostering Trust in Models' Structure and Results

Sterman [134] advises that, in order for a model's stakeholders to have trust in the model semantics, the model should be open and available to all interested parties. Its structure, behavior, input data and all other relevant components should be well documented, and this documentation should also be made publicly available. This will allow stakeholders to use the model semantics, test assumptions, and

assess the effect of alternative assumptions on the model's output. Conversely, results produced by "black box" models, which do not grant the user access to the model's structure or behavior, are less likely to be trusted and must be accepted by their users as a matter of faith. Consequently, model semantic designers should ask:

- Are the purpose, structure and behavior of the model, or any of its components, classified? If not, consider making them and their documentation available publicly, or to interested parties.
- Can and should the model be made publicly, or widely, available?

If the model is developed as a "black box" solution, to what extent will this undermine user's trust in the model? This set of questions is intended to draw model designers' attention to issues that present an increased risk of project failure, or affect the extent to which a developed model will be trusted by its users.

6. Managing Project Feasibility and Domain Semantics

The lack of critical analysis concerning different modeling techniques is not limited to conceptual semantic modeling methods. In many cases, construction projects fail due to a lack of consideration for the project's feasibility, and a lack of understanding of how physical space links to building semantics and activity. We present a process that allows model designers to assess the likelihood of project success, and subsequently select the most appropriate approach to represent domain semantics.

6.1. Feasibility Study and Risk Assessment

The first phase of the process requires the completion of the feasibility study. The questions, expanded in Section 5, are intended to provide a general guideline in order to accommodate a wide range of building modeling projects. As a result, designers should pay special attention to issues that make the project or any of its aspects in some way unusual or novel, and assess whether this novelty is associated with increased risk of project failure. The purpose of this phase is to gain an understanding of the risks that affect the project so that they can be controlled or mitigated during project development. This feasibility phase should help model designers identify any major risks that indicate likely project risk/failure and, as a result, should lead designers to reconsider project outcomes.

6.2. Selection of Method for Managing Domain Semantics

The second step of the process is to select a suitable method for modeling the semantics of the domain. The choices in this section include traditional, ontology-based approaches (hierarchy, network, relationship, object models *etc.*), and newer alternatives (such as tagging, generic data models). It is worth noting at this stage that this paper has considered only a sub-set of all possible alternatives. In this step, designers need to consider the nature of the semantic and the requirements of the model in the context of the criteria raised by Shirky [123]. The purpose of this step is to assess the extent to which the modeling problem relies on, or would benefit from, a more flexible approach to managing domain semantics than the standard ontology design techniques. Reaching a suitable decision in this step is critical in order to avoid future problems.

6.3. Selection of Implementation Method

The third step in the process encourages designers, who intend to base their model on a domain ontology, to consider and select between managing the domain's semantics using traditional relational database design methods and the adoption of generic data models. In this step it is important to understand how the relative advantages, disadvantages and trade-offs between the two approaches will impact the model's development, particularly in terms of support for future change. If frequent change to the building's conceptual model is expected, designers should consider adopting a generic data model. Alternatively, if designers are confident that they are able to develop a stable conceptual domain model that will not require frequent or major change during the expected lifetime of the building model, the adoption of traditional database implementation methods is reasonable.

In scenarios where requirements include the advantages of adopting both traditional and generic implementations, designers can also consider a hybrid approach where the domain concepts that are expected to change are stored using a generic data model, whilst stable entities that require high speed access and scalability can be implemented using a standard approach. This step encourages designers to consider alternative options that may prove to provide more powerful and flexible capabilities. In some cases this assessment is likely to lead to either more flexible and extensible models, or the avoidance of project failure where success is reliant on the ability to maintain and make numerous change to conceptual domain semantic structures throughout the life of the model.

If the tagging approach is adopted, designers may not need to develop a conceptual domain semantic model, as the meaning of the domain entities will be collaboratively decided upon by the model's users. Designers will, however, certainly need to specify rules or manage tags in order to distinguish between those that are unusual from those that are commonly applied, and therefore most strongly associated with by the user community. Designers should be careful, however, as tagging might restrict the potential for the future inclusion of behavior since existing examples of tagging in action have focused on providing feedback to users based on the relationships between tags, rather than through any behavior associated with this network. Consequently, folksonomies are a poor choice for problems that require the model to associated multiple object states, state transitions and behaviors with a conceptual domain model developed using social tagging.

7. Conclusions

Although BIM research is moving towards inclusion of more complex semantics in building modeling, it is important that model designers appropriately consider semantic information structures. In this paper we have discussed a number of key issues related to the development of building-based models with a focus on conceptual semantics models. We have demonstrated how model designers often make important decisions based on the methods that they prefer, or are most comfortable with, and that, in many cases, these decisions are not optimal for capture building semantics. Based on evidence compiled by Sterman [134], we have also shown that there are certain common sets of problems that frequently undermine the validity of model results; many of which, at least in part, can be addressed by asking relevant questions, making appropriate design decisions, and/or identifying that the proposed model is simply feasible. In a hope to support this process we have developed a process,

and a set of questions, to guide the designers in assessing and selecting appropriate methods for managing domain semantics.

Acknowledgments

We would like to thank Faithful and Gould for their ongoing research support.

Author Contributions

The original research, by Hubert Grzybek, was written for inclusion in his PhD thesis. This work was edited, adapted, and extended for publication by Stephen Gulliver; who also managed the paper revision and publication process. Shen Xu strengthened the contextualization of this work, by supplying much of the literature, and by providing feedback concerning paper contributions. Victoria Fillingham provided valuable feedback for each of the publication drafts, and contributed significantly to the final formatting and editing of the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2011.
2. Smith, E.; Shoben, E.; Rips, L. Structure and process in semantic memory: A featural model for semantic decisions. *Psychol. Rev.* **1974**, *81*, 214–241.
3. Guarino, N.; Borgo, S.; Masolo, C. Logical Modelling of Product Knowledge: Towards a Well-Founded Semantics for STEP. In Proceedings of the European Conference on Product Data Technology, Sophia Antipolis, France, 16–17 April 1997; pp. 183–190.
4. Ludlow, P. Semantics, Tense, and Time: An Essay in the Metaphysics of Natural Language; MIT Press: Cambridge, MA, USA, 1999.
5. Aristotle, I. Metaphysics. In *Complete Works of Aristotle*; Barnes, J., Princeton, N., Eds.; Princeton University Press: Princeton, NJ, USA, 1995; pp. 73–90.
6. Kant, I.; Caygill, H.; Banham, G. *Critique of Pure Reason*, 2nd ed.; Smith, N.K., Ed.; Palgrave Macmillan: Basingstoke, UK, 2007.
7. Guarino, N. Formal Ontology in Information Systems. In Proceedings of the 1st FOIS Conference (FOIS'98), Trento, Italy, 6–8 June 1998; IOS Press: Amsterdam, The Netherlands, 1998; pp. 3–15.
8. Codd, E.F. Extending the Database Relational Model to Capture More Meaning. *ACM Trans. Database Syst.* **1979**, *4*, 397–434.
9. Petersen, T. Developing a New Thesaurus for Art and Architecture. *Libr. Trends* **1990**, *38*, 644–658.

10. Cheng, J.; Trivedi, P.; Law, K.H. Ontology Mapping between PSL and XML-Based Standards for Project Scheduling. In Proceedings of the 3rd International Conference on Concurrent Engineering in Construction, Berkeley, CA, USA, 1–2 July 2002.
11. ISO 16739 Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries; International Organization for Standardization: Geneva, Switzerland, 2013.
12. ISO/TS 15926–8 Industrial automation systems and integration—Integration of life-cycle data for process plants including oil and gas production facilities—Part 8: Implementation methods for the integration of distributed systems: Web Ontology Language (OWL) implementation; International Organization for Standardization: Geneva, Switzerland, 2011.
13. ISO 10303–28 Industrial automation systems and integration—Product data representation and exchange—Part 28: Implementation methods: XML representations of EXPRESS schema and data; International Organization for Standardization: Geneva, Switzerland, 2007.
14. Niles, I.; Pease, A. Towards a Standard Upper Ontology. In Proceedings of the International Conference on Formal Ontology in Information Systems, Ogunquit, ME, USA, 17–19 October 2001; pp. 2–9.
15. Aksamija, A.; Grobler, F. Architectural ontology: Development of machine-readable representations for building design drivers. *Comput. Civil Eng.* **2007**, *2007*, 168–175, doi:10.1061/40937(261)21.
16. Si, J.; Wang, Y. IFC-based construction engineering domain ontology development. *World Acad. Sci. Eng. Technol.* **2012**, *6*, 431–434.
17. Kim, H.; Grobler, F. Building ontology to support reasoning in early design. *Comput. Civil Eng.* **2007**, *19*, 151–159.
18. McGibbney, L.J.; Kumar, B. The WOMBRA Project: A Web-Based Ontology-Enhanced Multi-Purpose Building-Regulation Retrieval Application for Scottish Technical Standards. In Proceedings of the 28th International Conference of CIB W78, Sophia Antipolis, France, 26–28 October 2011; p. 64.
19. Yang, Q.Z.; Zhang, Y. Semantic interoperability in building design: Methods and tools. *Comput.-Aided Des.* **2006**, *38*, pp. 1099–1112.
20. Garcia, L.E.R. Ontological CAD Data Interoperability Framework. In Proceedings of the 4th International Conference on Advances in Semantic Processing, Florence, Italy, 25–30 October 2010; pp. 79–82.
21. Zhang, L.; Issa, R.R.A. Development of IFC-based Construction Industry Ontology for Information Retrieval from IFC Models. In Proceedings of the 2011 Eg-Ice Workshop, University of Twente, The Netherlands, 6–8 July 2011.
22. Zhang, X.; Di, R.; Liang, Y. Ontology Based Knowledge Modeling for Structural Engineering Experiment Information Management. In Proceedings of the 2010 9th International Conference on Grid and Cooperative Computing, Nanjing, China, 1–5 November 2010; pp.40–45.
23. Dibley, M.; Li, H.; Rezgui, Y.; Miles, J. An ontology framework for intelligent sensor-based building monitoring; *Autom. Constr.* **2012**, *28*, 1–14.

24. Katranuschkov, P.; Rybenko, K.; Scherer, R.J. Ontology-Based Dynamic Process Support on the Example of Defect Management. In Proceedings of the 26th CIB W078 Conference Managing IT in Construction, Istanbul, Turkey, 1–3 October 2009; pp. 339–350.
25. Xu, J.; Lee, Y.-H.; Tsai, W.-T.; Li, W.; Son, Y.-S.; Park, J.-H.; Moon, K.-D. Ontology-Based Smart Home Solution and Service Composition. In Proceedings of the International Conference on Embedded Software and Systems (ICESS2009), Hangzhou, China, 25–27 May 2009; pp. 297–304.
26. Kumar, V.; Tomic, S.; Pellegrini, T.; Fensel, A.V.; Mayrhofer, R. User Created Machine-readable Policies for Energy Efficiency in Smart Homes. In Proceedings of Ubicomp 2010 Workshop: Ubiquitous Computing for Sustainable Energy (UCSE 2010), Copenhagen, Denmark, 26–29 September 2010.
27. Tomic, S.; Fensel, A.; Pellegrini, T. SESAME Demonstrator: Ontologies, Services and Policies for Energy Efficiency. In Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS 2010), Graz, Austria, 1–3 September 2010.
28. Wicaksono, H.; Rogalski, S.; Kusnady, E. Knowledge-Based Intelligent Energy Management Using Building Automation System. In Proceedings of the 9th International Power and Energy Conference, Singapore, 27–29 October 2010.
29. Daouadji, A.; Nguyen, K.-K.; Lemay, M.; Cheriet, M. Ontology-Based Resource Description and Discovery Framework for Low Carbon Grid Networks. In Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm), Gaithersburg, MD, USA, 4–6 October 2010.
30. Rossello Busquet, A.; Brewka, L.; Soler, J.; Dittmann, L. OWL Ontologies and SWRL Rules Applied to Energy Management. In Proceedgs of the 13th International Conference on Computer Modelling and Simulation, IEEE, Cambridge, UK, 30 March–1 April 2011; pp. 446–450.
31. Han, J.; Jeong, Y.-K.; Lee, I. Efficient Building Energy Management System Based on Ontology, Inference Rules, and Simulation. In Proceedings of the International Conference on Intelligent Building and Management, Singapore, 2–4 May 2011; pp. 295–299.
32. Kofler, M.; Reinisch, C.; Kastner, W. A semantic representation of energy- related information in future smart homes. *Energy Build.* **2011**, *47*, 169–179.
33. Shah, N.; Chao, K.; Zlamaniec, T.; Matei, A. Ontology for home energy management domain. *Commun. Comput. Inform. Sci.* **2011**, *167*, 337–347.
34. Kofler, M.; Reinisch, C.; Kastner, W. An Ontological Weather Representation for Improving Energy-Efficiency in Interconnected Smart Home Systems. In Proceedings of the Applied Simulation and Modelling/Artificial Intelligence and Soft Computing (ASC2012), Napoli, Italy, 25–27 June 2012.
35. Curry, E.; Jentzsch, A.; Cyganiak, R. Cross-domain Building Optimisation Using Scenario Modelling and Linked Data. In Proceedings of the 1st Workshop Linked Data in Architecture and Construction (LDAC2012), Ghent, Belgium, 28–29 March 2012.
36. Nemirovski, G.; Sicilia, A.; GalÃ þn, F.; Massetti, L.M. Ontological Representation of Knowledge Related to Building Energy Efficiency. In Proceedings of the SEMAPRO 2012, The 6th International Conference on Advances in Semantic Processing, Barcelona, Spain, 23–28 September 2012.

37. Cotterell, M.; Zheng, J.; Sun, Q.; Wu, Z.; Champlin, C.; Beach, A. Facilitating Knowledge Sharing and Analysis in Energy Informatics with the Ontology for Energy Investigations (OEI). In Proceedings of the 7th International Conference on Formal Ontology in Information Systems (FOIS'12), Graz, Austria, 24–27 July 2012; Volume 12.
38. Gursel, I.; Sariyildiz, S.; Akin, O.; Stouffs, R. Modeling and visualization of lifecycle building performance assessment. *Adv. Eng. Inform.* **2009**, *23*, 396–417.
39. Pauwels, P.; Deursen, D. V; Verstraeten, D.R.; de Meyer, R.; de Walle, R. V; Campenhout, J.V. A semantic rule checking environment for building performance checking. *Autom. Constr.* **2011**, *20*, 506–518.
40. Ahmed, V.; Shaik, A.; Aouad, G. An ontology of construction education for e-learning via the semantic web. *Archit. Eng. Des. Manag.* **2006**, *2*, 87–99.
41. Pathmeswaran, R.; Ahmed, V. SWmLOR: Technologies for developing semantic web-based mobile learning object repository. *Built Hum. Environ. Rev.* **2011**, *2*, 1–10.
42. Argüello, M.; El-Hasia, A.; Lees, M. Using Semantic Web Technologies to Bridge the Language Gap between Academia and Industry in the Construction Sector. In *Applications and Innovations in Intelligent Systems XIV*; Springer: Berlin, Germany, 2007, pp. 135–148.
43. El-Diraby, T.; Lima, C.; Feis, B. Domain taxonomy for construction concepts: Toward a formal ontology for construction knowledge. *J. Comput. Civil Eng.* **2005**, *19*, 394–406.
44. Lima, C.; El-Diraby, T.; Stephens, J. Ontology-based optimisation of knowledge management in e-construction. *ITcon* **2005**, *10*, 305–327.
45. Staub-French, S.; Fischer, M.; Kunz, J.; Ishii, Kos.; Paulson, B. A feature ontology to support construction cost estimating. *Artif. Intell. Eng. Des. Anal. Manuf.* **2003**, *17*, 133–154.
46. Lee, C.-S.; Wang, M.-H.; Chen, J.J. Ontology-based intelligent decision support agent for CMMI project monitoring and control. *Int. J. Approx. Reason.* **2008**, *48*, 62–76.
47. Abanda, F.H.; Tah, J.H. M.; Pettang, C.; Manjia, M. An ontology-driven house-building labour cost estimation in Cameroon. *ITcon* **2011**, *16*, 617–634.
48. Fidan, G.; Dikmen, I.; Tanyer, A.; Birgonul, M. Ontology for relating risk and vulnerability to cost overrun in international projects. *J. Comput. Civil Eng.* **2011**, *25*, 302–315.
49. Ma, Z.; Wei, Z. Framework for Automatic Construction Cost Estimation Based on BIM and Ontology Technology. In Proceedings of the 29th CIB W78 International Conference, Beirut, Lebanon, 17–19 October 2012.
50. Kim, H.; Anderson, K.; Lee, S.H.; Hildreth, J. Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology. *Autom. Constr.* **2013**, *35*, 285–295.
51. Lee, S.-K.; Kim, K.-R.; Yu, J.-H. BIM and ontology-based approach for building cost estimation. *Autom. Constr.* **2014**, *41*, 96–105.
52. Scherer, R.J.; Schapke, S.-E. A distributed multi-model-based management information system for simulation and decision-making on construction projects. *Adv. Eng. Inform.* **2011**, *25*, 582–599.
53. El-Diraby, T.E.; Briceno, F. Taxonomy for outside plant construction in telecommunication infrastructure: Supporting knowledge-based virtual teaming. *J. Infrastruct. Syst.* **2005**, *11*, 110–121.

54. El-Gohary, N.; El-Diraby, T. Domain ontology for processes in infrastructure and construction. *J. Constr. Eng. Manag.* **2010**, *136*, 730–744.
55. Ruikar, D.; Anumba, C.J.; Duke, A.; Carrillo, P.M.; Bouchlaghem, N.M. Using the semantic web for project information management. *Facilities* **2007**, *25*, 507–524.
56. Cheng, J.; Kumar, B.; Law, K.H. A question answering system for project management applications. *J. Adv. Eng. Inform.* **2002**, *16*, 277–289.
57. Cheng, J.; Law, K.H.; Kumar, B. Integrating Project Management Applications as Web Services. In Proceedings of the 2nd International Conference on Innovation in Architecture, Engineering and Construction, Loughborough, UK, 25–27 June 2003.
58. El-Diraby, T.; Kashif, K.F. Distributed ontology architecture for knowledge management in highway construction. *J. Constr. Eng. Manag.* **2005**, *131*, 591–603.
59. Ramaprasad, A.; Prakash, A.N.; Rammurthy, N. Construction Project Management System (CPMS): An Ontological Framework. In Proceedings of the Research and Education Conference (PMIREC), Pune, India, 9 December 2011.
60. Venugopal, M.; Eastman, C.M.; Sacks, R.; Teizer, J. Semantics of model views for information exchanges using the industry foundation class schema. *J. Adv. Eng. Inform.* **2012**, *26*, 411–428.
61. Cheng, J.; Law, K.H. Using Process Specification Language for Project Information Exchange. In Proceedings of the 3rd International Conference on Concurrent Engineering in Construction, Berkeley, CA, USA, 1–2 July 2002.
62. Yurchyshyna, A.; Faron-Zucker, C.; Le Thanh, N.; Lima, C.; Zarli, A. Towards an Ontology-Based Approach for Conformance Checking Modeling in Construction. In Proceedings of the 24th W78 Conference, Maribor, Slovenia, 27–29 June 2007.
63. Yurchyshyna, A.; Faron-Zucker, C.; Le Thanh, N.; Lima, C.; Zarli, A. Ontological Approach for the Conformity-Checking Modelling in Construction. In Proceedings of the 10th International Conference on Enterprise Formation Systems (ICEIS2008), Barcelona, Spain, 12–16 June 2008.
64. Yurchyshyna, A.; Zarli, A. An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction. *Autom. Constr.* **2009**, *18*, 1084–1098.
65. Kim, H.; Grobler, F. Design coordination in Building Information Modeling (BIM) using ontological consistency checking. *Comput. Civil Eng.* **2009**, 410–420, doi:10.1061/41052(346)41.
66. Yurchyshyna, A.; Faron-Zucker, C.; Le Thanh, N.; Zarli, A. Knowledge capitalisation and organisation for conformance checking model in construction. *Int. J. Knowl. Eng. Soft Data Paradig.* **2010**, *2*, 15–32.
67. Zhong, B.T.; Ding, L.Y.; Luo, H.B.; Zhou, Y.; Hu, H.M. Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Autom. Constr.* **2012**, *28*, 58–70.
68. El-Diraby, T.A.; Gill, S M. A taxonomy for construction terms in privatized-infrastructure finance: Supporting semantic exchange of project risk information. *Constr. Manag. Econ.* **2006**, *24*, 271–285.
69. Tserng, H.P.; Yin, S.Y.L.; Dzeng, R.J.; Wou, B.; Tsai, M.D.; Chen, W.Y. A study of ontology-based risk management framework of construction projects through project life cycle. *Autom. Constr.* **2009**, *18*, 994–1008.

70. Elghamrawy, T.; Boukamp, F. A Vision for a Framework to Support Management of and Learning from Construction Problems. In Proceedings of the 25th International Conference on Formation Technology in Construction: Improving the Management of Construction Projects Through IT Adoption, Santiago, Chile, 15–17 July 2008.
71. Elghamrawy, T.; Boukamp, F.; Kim, H.-S. Ontology-Based Semi-Automatic Framework for Storing and Retrieving On-Site Construction Problem Information—An RFID-Based Case Study. In Proceedings of the 2009 Construction Research Congress, Seattle, WA, USA, 5–7 April 2009; pp. 457–466.
72. Lee, J.S.; Lee, Y.S.; Min, K.M.; Kim J.H.; Kim, J.J. Building Ontology to Implement the BIM (Building Information Modeling) Focused on Pre-Design Stage. In Proceedings of the 25th International Symposium on Automation and Robotics Construction, Vilnius, Lithuania, 27–29 June 2008.
73. Cruz, C.; Marzani, F.; Boochs, F. Ontology-driven 3D reconstruction of architectural objects. *Comput. Vision Imag. Comput. Graph. Theory Appl.* **2007**, *2007*, 47–54.
74. Dibley, M.J.; Li, H.; Miles, J.C.; Rezgui, Y. Towards intelligent agent based software for building related decision support. *Adv. Eng. Inform.* **2011**, *25*, 311–329.
75. Islam, A.S.; Piasecki, M. Ontology based web simulation system for hydrodynamic modelling. *Simul. Model. Pract. Theory* **2008**, *16*, 754–767.
76. Schevers, H.A.; Mitchell, J.; Akhurst, P.; Marchant, D.M.; Bull, S.; McDonald, K.; Drogemuller, R.M.; Linning, C. Towards digital facility modelling for Sydney Opera house using IFC and Semantic Web technology. *Inform. Technol. Constr.* **2007**, *12*, 347–362.
77. Curry, E.; O'Donnell, J.; Corry, E. Building Optimisation Using Scenario Modeling and Linked Data. In Proceedings of the 1st Workshop Linked Data in Architecture and Construction (LDAC2012), Ghent, Belgium, 28–29 March 2012.
78. Curry, E.; O'Donnell, J.; Corry, E.; Hasan, S.; Keane, M.; O'Riain, S. Linking building data in the cloud: Integrating cross-domain building data using linked data. *J. Adv. Eng. Inform.* **2012**, *27*, 206–219.
79. Vanlande, R.; Cruz, C.; Nicolle, C. IFC and buildings lifecycle management. *Autom. Constr.* **2008**, *18*, 70–78.
80. Mauher, M.; Smokvina, V. Municipal asset and property management system for the web collaborative environment. Available online: http://www.majorcities.eu/generaldocuments/pdf/municipal_asset_and_property_management_system_for_the_web_collaborative_environment.pdf (accessed on 31 October 2014).
81. Vanlande, R.; Nicolle, C. Semantic Web technologies for facilities management. In Proceedings of the 2nd International Conference on Digital Formation Management, Lyon, France, 28–31 August 2007.
82. Pauwels, P.; de Meyer, R.; van Campenhout, J. Interoperability for the design and construction industry through semantic web technology. *Semant. Multimed. Lect. Notes Comput. Sci.* **2010**, *6725*, 143–158.
83. Pauwels, P.; Deursen, D.V. IFC/RDF: Adaptation, Aggregation and Enrichment. In Proceedings of the 1st Workshop Liked Data Architecture and Construction (LDAC2012), Ghent , Belgium, 28 March 2012.

84. Schevers, H.; Drogemuller, R. Converting Industry Foundation Classes to the Web Ontology Language. In Proceedings of the 1st International Conference on Semantics, Knowledge and Grid, Washington, DC, USA, 27–29 November 2005; p. 73.
85. Shen, L.; Chua, D.K.H. Application of Building Information Modeling (BIM) and Information Technology (IT) for Project Collaboration. In Proceedings of the EPPM, Singapore, 20–21 September 2011.
86. Madrazo, L.; Costa, G. Open Product Modelling and Interoperability in the AEC Sector. In Proceedings of the 1st Workshop Linked Data Architecture and Construction (LDAC2012), Ghent, Belgium, 28–29 March 2012.
87. Törmä, S.; Oraskari, J.; Hoang, N. Distributed Transactional Building Information Management. In Proceedings of the 1st Workshop Linked Data Architecture and Construction (LDAC2012), Ghent, Belgium, 28–29 March 2012.
88. Scherer, R.J.; Katranuschkov, P.; Kadolsky, M.; Laine, T. Ontology-Based Building Information Model for Integrated Lifecycle Energy Management. In *eWork Ebus. Architecture, Engineering and Construction*; CRC Press: Boca Raton, FL, USA, 2012; pp. 951–956.
89. Pauwels, P.; Deursen, D.V.; de Roo, J.; Ackere, T.V.; de Meyer, R.; de Walle, R.V. Three-dimensional information exchange over the Semantic Web for the domain of architecture, engineering, and construction. *Artif. Intell. Eng. Des. Anal. Manuf.* **2011**, *25*, 317–332.
90. Gu, T.; Wang, X.H.; Zhang, D.Q. An Ontology-Based Context Model in Intelligent Environments. In Proceedings of the Communication Networks and Distributed Systems Modelling and Simulation Conference, San Diego, CA, USA, 17–24 January 2004.
91. Ricquebourg, V.; Durand, D.; Menga, D.; Marhic, B.; Delahoche, L.; Loge, C.; Jolly-Desodt, A.-M. Context Inferring in the Smart Home: An SWRL Approach. In Proceedings of the AAW 2007 Proceedings of the 21st International Conference on Advanced Formation Networking and Applications Workshops, Niagara Falls, Canada, 21–23 May 2007.
92. Meshkova, E.; Riihijarvi, J.; Mahonen, P.; Kavadias, C. Modelling the Home Environment Using Ontology with Applications in Software Configuration Management. In Proceedings of the 15th International Conference on Telecommunications (ICT2008), 16–19 June 2008.
93. Reinisch, C.; Kofler, M.; Iglesias, F.; Kastner, W. ThinkHome energy efficiency in future smart homes. *J. Embed. Syst.* **2011**, *2011*, 1–18.
94. Valiente-Rocha, P.A.; Lozano-Tello, A. Ontology and SWRL-based learning model for home automation controlling. *Adv. Intell. Soft Comput.* **2010**, *72*, 79–86.
95. Abdulrazak, B.; Chikhaoui, B.; Gouin-Vallerand, C.; Fraikin, B. A standard ontology for smart spaces. *Int. J. Web Grid Serv.* **2010**, *6*, 244–268.
96. Tah, J.H. M.; Abanda, F.H. Sustainable building technology knowledge representation: Using semantic web techniques. *J. Adv. Eng. Inform.* **2011**, *25*, 547–558.
97. Garrido, J.; Requena, I. Proposal of ontology for environmental impact assessment: An application with knowledge mobilization. *Expert Syst. Appl.* **2011**, *38*, 2462–2472.
98. Edum-Fotwe, F.T.; Price, A.D.F. A social ontology for appraising sustainability of construction projects and developments. *Int. J. Proj. Manag.* **2009**, *27*, 313–322.

99. Rizzolia, A.E.; Schimak, G.; Donatellic, M.; Hrebicek, J.; Avellinoe, G.; Monf J.L.; Athanasiadis, I. TaToo: Tagging Environmental Resources on the Web by Semantic Annotations. In Proceedings of the International Environmental Modelling and Software Society (iEMSs) 2010 International Congress on Environmental Modelling and Software Modelling for Environment's Sake, 5th Biennial Meeting, Ottawa, Canada, 5–8 July 2010.
100. Kumazawa, T.; Saito, O.; Kozaki, K.; Matsui, T.; Mizoguchi, R. Toward knowledge structuring of sustainability science based on ontology engineering. *Sustain. Sci. Manag.* **2009**, *4*, 99–116.
101. Li, H.J.; Rezgui, Y.; Miles, J.C.; Wilson, I. Low Carbon Ontology Development Using Information Retrieval Techniques. In eWork and eBusess Architecture, Engineering and Construction Proceedings of the European Conference on Product and Process Modelling, Cork, Ireland, 14–16 September 2010.
102. Mignard, C.; Gesquiere, G.; Nicolle, C. SIGA3D: A Semantic BIM Extension to Represent Urban Environment. In Proceedings of the 5th International Conference on Advances Semantic Processing, Lisbon, Portugal, 20–25 November 2011.
103. Trausan-Matu, S.; Neacsu, A. An Ontology-Based Intelligent Information System for Urbanism and Civil Engineering Data. In Proceedings of the 2nd Workshop of the COST Action C21-Towntology, Turin, Italy, 17–18 October 2007.
104. El-Diraby, T.E.; Osman, H. A domain ontology for construction concepts in urban infrastructure products. *Autom. Constr.* **2011**, *20*, 1120–1132.
105. Lacasta, J.; Nogueras-Iso, J.; Zarazaga-Soria, F.J.; Muro-Medrano, P.R. Generating an Urban Domain Ontology through the Merging of Cross-Domain Lexical Ontologies. In Proceedings of the 2nd Workshop of the COST Action C21-Towntology, Turin, Italy, 17–18 October 2007.
106. Lannon, S.; Linovski, O. Ontologies for the Classification of Urban Characteristics: Opportunities for Urban Designers. In Proceedings of the Fall Conference of the COST Action C21-Towntology: Urban Ontologies for an Improved Communication Urban Development Projects, Liège, France, 9–10 March 2009; pp. 59–67.
107. Kaza, N.; Hopkins, L.D. Ontology for land development decisions and plans. *Ontol. Urban Dev.* **2007**, *61*, 47–59.
108. MÃctral, C.; Falquet, G.; Cutting-Decelle, A.F. Towards Semantically Enriched 3D City Models: An Ontology-Based Approach. In Proceedings of the GeoWeb 2009 Academic Track-Cityscapes, Vancouver, Canada, 27–31 July 2009.
109. Metral, C.; Billen, R.; Cutting-Decelle, A.F.; van Ruymbeke, M. Ontology-based approaches for improving the interoperability between 3D urban models. *J. Inform. Technol. Constr.* **2010**, *15*, 169–184.
110. El-mekawy, M.; Östman, A. Semantic Mapping: An Ontology Engineering Method for Integrating Building Models in IFC and CityGML. In Proceedings of the 3rd ISDE Digital Earth Summit, Nessebar, Bulgaria, 12–14 June 2010.
111. Abanda, H.; Ng'ombe, A.; Tah, J.H.M.M.; Keivani, R. An ontology-driven decision support system for land delivery in Zambia. *Expert Syst. Appl.* **2011**, *38*, 10896–10905.
112. Lepczyk, C.A.; Lortie, C.J.; Anderson, L.J. An ontology for landscapes. *Ecol. Complex.* **2008**, *5*, 272–279.

113. Belhadef, H.; Kholladi, M.K. Urban ontology-based geographical information system. *J. Theor. Appl. Inform. Technol.* **2009**, *9*, 139–154.
114. Schevers, H.A.J.; Trinidad, G.; Drogemuller, R.M. Towards integrated assessments for urban development. *ITcon* **2006**, *11*, 225–236.
115. Kohli, D.; Sliuzas, R.; Kerle, N.; Stein, A. An ontology of slums for image- based classification. *Comput. Environ. Urban Syst.* **2012**, *36*, 154–163.
116. Beetz, J.; van Leeuwen, J.; de Vries, B. Towards a Topological Reasoning Service for IFC-Based Building Information Models in a Semantic Web Context. In Proceedings of the Joint International Conference on Computing and decision Making in Civil and Building Engineering, Montréal, Canada, 14–16 June 2006.
117. Montenegro, N.; Gomes, G.C.; Urbano, P.; Duarte, J.P. A land use planning ontology: LBCS. *Future Internet* **2012**, *4*, 65–82.
118. Finat, J.; Delgado, J.F.; Martinez, R.; Hurtado, A.; Fernandez, J.J.; San Jose, J.I.; Martinez, J. Constructors of geometric primitives in domain ontologies for urban environments. *ITcon* **2010**, *15*, 149–158.
119. Metral, C.; Cutting-Decelle, A.-F. Ontologies for interconnecting urban models. *Adv. Inform. Knowl. Process* **2011**, *1*, 105–122.
120. Noy, N.; McGuinness, D. Ontology development 101: A guide to creating your first ontology. *Stanford Knowl. Syst. Lab. Tech. Rep.* **2001**, *15*, 1–25.
121. Uschold, M.; Gruninger, M. Ontologies: Principles, methods and applications. *Knowl. Eng. Rev.* **1996**, *11*, 93–136.
122. Rosch, E. Principles of Categorization. In *Concepts Core Readings*; MIT Press: Cambridge, MA, USA, 1999.
123. Shirky, C. Ontology is Overrated—Categories, Links, and Tags. Available online: http://www.shirky.com/writings/ontology_overrated.html?goback=.gde_1838701_member_179729766 (accessed 31 October 2014).
124. Holub, A. Why extends is evil. Available online: http://www.javaworld.com/article/2073649/core-java/why-extends-is-evil.html (accessed on 11 September 2014).
125. Alghamdi, J.S. Measuring Software Coupling. In Proceedings of the 6th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems, Stevens Point, WI, USA, 16–19 February 2007; pp. 6–12.
126. Dong, B.; Lam, K.P.; Huang, Y.C. A Comparative Study of the IFC and gbXML Informational Infrastructures for Data Exchange in Computational Design Support Environments Geometry Information. In Proceedings of the 10th International IBPSA Conference, 3–6 September 2007; pp. 1530–1537.
127. Mitroff, I.I.; Mason, R.O. Structuring III-structured policy issues: Further explorations in a methodology for messy problems. *Strateg. Manag. J.* **1980**, *1*, 331–342.
128. Zeldman, J.; Marcotte, E. *Designing with Web Standards*, Voices That Matter; Pearson Education: Upper Saddle River, NJ, USA, 2009.
129. Spiteri, L. The structure and form of folksonomy tags: The road to the public library catalog. *Inform. Technol. Libr.* **2013**, *26*, 13–25.

130. Ambler, S.W. Refactoring for Fitness. Available online: http://www.drdobbs.com/refactoring-for-fitness/184414821 (accessed on 31 October 2014).
131. Williams, B. Father of all Data Models Data Model. Available online: http://www.databaseanswers.org/data_models/father_of_all_models/ (accessed on 11 September 2014).
132. Andrews, T. Tony Andrews on Oracle and Databases: OTLT and EAV: The Two Big Design Mistakes All Beginners Make. Available online: http://tonyandrews.blogspot.co.uk/2004/10/otlt-and-eav-two-big-design-mistakes.html (accessed on 11 September 2014).
133. Broersma, R. Richard's Stuff: EAV—The Good and Bad. Available online: http://richardbroersma.blogspot.co.uk/2009/02/eav-good-and-bad.html (accessed on 11 September 2014).
134. Sterman, J. A Skeptic's Guide to Computer Models. In *Managing a Nation: The Microcomputer Software Catalogue*; Barney, G.O., Kreutzer, W.B., Garrett, M.J., Eds; Westview Press: Boulder, CO, USA; 1991.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).