Domain Ontology for Construction Knowledge

Tamer E. El-Diraby[1]

Abstract: An ontology is a claim on/for knowledge that attempts to model what is known about a domain of discourse. A domain ontology does not aim to exhaustively list all concepts in a domain, but rather to build an abstract (yet extendable) philosophical (yet practical) conceptualization of the essence of knowledge in a domain. At the core of any ontology is an ontological model—an architecture of how the world (in a domain) behaves (or becomes). The ontology categorizes construction knowledge across three main dimensions: concept, modality, and context. Concept encompasses five key terms: entity (further subdivided into generic and secondary), environmental element, abstract concept, attribute, and system (combinations of the previous four types). Modality is a means for generating a variety of types for each of the described concepts. Context allows for linking concepts in a variety of ways—creating different worlds. **DOI: 10.1061/(ASCE)CO.1943-7862.0000646.**

CE Database subject headings: Information management; Construction management; Knowledge-based systems.

Author keywords: Ontology; Semantic web; Knowledge representation; Interoperability.

Introduction

2-3 2 notes:

An informatics ontology is a formal description of what is known about a domain of discourse. It includes hierarchies (taxonomies) of concepts, their relationships, and the axioms (limitations) that describe their behavior. Informatics ontologies are not meant to be a data-exchange standard. Nonetheless, they can interact fairly well with data standards to support interoperable computer programs. Although ontologies are related more to knowledge representation than to reasoning, they normally can be complemented by artificial intelligence (AI) tools to enhance their decision-support capabilities. An upper ontology is the most abstract type of ontology, typically aiming at capturing universal cross-domain conceptualization at a philosophical level. A domain ontology is also philosophical and is interested with categorization of universals, but within a certain domain of discourse. An application ontology is the least philosophical of all types and is meant to instantiate a domain ontology for use by a computer software system—typically, to enrich the semantic representation of terms.

4-5 2 notes: 6-7 2 notes:

Although listing concepts, linking them, and describing their behavior are important, ontology is, fundamentally, a claim on (or a debate about) knowledge. At the core of any informatics ontology is an ontological model—a philosophical view of how concepts are structured. The main questions of such models relates to (conceptual) universals: Do they exist (does the author/participant believe in them)? What are they? How do they relate to each other? Do they change/morph? If so, under what conditions? It can be argued that the advantage of building an ontology is twofold: its use to formally represent and share human knowledge in a machine-interpretable way (the informatics contribution); and it use to spur dialogue among users about its ontological model—that is, what is knowledge (in a domain) and how to represent it (the philosophical contribution). The rich history of building ontologies in the domain of philosophy has shown that the second advantage could be more beneficial and long lasting.

8-9 2 notes:

This paper presents an iteration for a domain ontology for construction knowledge (DOCK 1.0). The objective is not to exhaustively list concepts, but rather to build a conceptual architecture of key terms in construction, their relationships, and behavior. This ontology was developed on the basis of a mix of constructivist and pragmatic epistemologies. Constructivism denies the existence of an absolute ontology (reality is socially constructed) and advocates a bottom-up modeling approach. Pragmaticism focuses on modeling knowledge based on its practicality and the benefits gained from such knowledge (for more, see El-Diraby 2012). Based on this, and recognizing the impossibility of a perfect model, the aim of DOCK is to find a good enough skeleton to describe the main concepts relating to construction knowledge. Can the proposed categories and their structure help represent, fairly adequately, most of the possible conceptual understanding of/about knowledge in the construction domain based on suitable perspectives of the perception of such knowledge?

As with many claim to knowledge, there must be for sure a set of much-welcomed arguments about it—especially the consistency of the proposed model (and especially given the constructivist-pragmatic foundations of DOCK). Although consistency is important, one should, however, remember that "...an ontology is a conceptual hierarchy. Although it can be, it is not necessarily an axiomatic system. It is fashioned to formulate ontologies axiomatically to test their consistency... But consistency is too weak a test for acceptability, for there is unlimited supply of consistent ontologies. Quine suggested long ago that with enough ingenuity and flexibility, any set of ideas can be formalized. Formalization is a trivial property of any and all sets of belief, coherent or not..." (Swindler 1991).

A review of some related efforts in modeling construction knowledge is presented first. This is followed by a description of the scope and limitations of DOCK, and then its development methodology. Before presenting the concepts of DOCK, a summary of the ontological model is presented to put their definitions in context. Following this summary, DOCK main concepts, relations, and axioms are presented.

[1]Associate Professor and Director, Centre for Civil Informatics (i2c), Dept. of Civil Engineering, Univ. of Toronto, 35 St. George St., Toronto, ON, Canada M5S 1A4. E-mail: tamer@ecf.utoronto.ca

Note. This manuscript was submitted on March 26, 2012; approved on October 8, 2012; published online on October 10, 2012. Discussion period open until December 1, 2013; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 139, No. 7, July 1, 2013. © ASCE, ISSN 0733-9364/2013/7-768-784/$25.00.

Modeling Construction Knowledge

Models of construction knowledge span three broad categories: classification systems and thesauri, product and process models, and ontologies. The first category is the most prominent and oldest. By far, these classification systems (such as the Swedish classification of construction terms: *sfb*, *Uniclass*, and *MasterFormat*) focused on product categorization with limited attention to ontological modeling. Product models such as the Industry Foundation Classes (IFC) also had limited ontological features because they were geared towards assuring interoperable exchange of product data (in contrast to semantic knowledge). Some models, such as the general construction object model (GenCOM), however, aimed at modeling the major terms in construction products and processes with some semantic tilt. The International Framework for Dictionaries (IFD) is closely related to IFC and building information modeling (BIM). It is closer to a thesaurus of construction terms with aims to create multilingual dictionaries or ontologies. It is meant as a reference library intended to support improved interoperability in the building and construction industry (IFD 2012).

A set of ontology-related work has emerged lately in the architectural/engineering/construction domain. For example, the development of taxonomies (Woestenenk et al. 2000), or subdomain-level ontologies (Katanuschik et al. 2002; Staub-French et al. 2003; Gehre et al. 2006). With few exceptions—most notably, the work of Ekholm (see Ekholm 1996) and Eir (see Eir 2004), the majority of construction ontologies focused on formalization rather than conceptualization (in true linguistic or philosophical fashion) and were concentrated at the subdomain (or the application) level. Interestingly, some of the most relevant philosophical contributions could be found in lean construction and organizational management (see Koskela 2000; Abdelhamid 2004).

Scope and Limitations

Because of its name and its aim of modeling (i.e., being), there is a rather false expectation that ontology should be exhaustive in scope and inclusive in its terminology. Further, the introduction of ontology to machine learning added a similar, false expectation: that there could exist a set of universal axioms to provide precise venues for reasoning. Nothing can be further from the truth. Ontology is a claim on/for knowledge. It starts new questions as much as it answers questions. Consequently, the domain ontology presented in this paper is not meant to be the universal representation of construction knowledge, but rather an attempt to generate one of many possible general representations. To this end, the value (and the fundamental contribution of an ontology) is in the coherence of its ontological model, its suitability/adaptability to support general (not universal) situations, and its extendibility. To this end, minimizing ontological commitments (especially in terms of axioms) is a fundamental lesson learned (Gruber 1993; Guarino 1995; Noy and McGuinness 2001). This, of course, refers to adopting a clear scope and using a weak domain theory that reduces conflict, not minimizing the depth or rigor of an ontology. The issues presented in the following subsections shaped the scope of DOCK.

Epistemology

The DOCK is built on the basis of a balance between contemporary pragmatism and constructivist epistemology. Every concept has possibilities (a variety of shapes or modes of existence) and probabilities (a variety of shapes or modes of existence in a certain context). Concepts and their models have to be built primarily (or, at least, initially) on the basis of bottom-up modeling of the social collective understandings of domain experts. Conceptualization is not a mechanical process (top down and deterministic). It has to be built through consensus between users. Such consensus is relative, given that it is developed by a limited group of users. Ultimately, conceptualization cannot reach the level of good enough from the first attempt. It has to go through iterative development to allow for adequate testing of its representativeness and rebalancing of the interests of the consenting group. This is not to say that a top-down approach has no place in ontology development. As always, a balance has to be reached (for more, see El-Diraby 2012).

Universality and Completeness

The proposed ontology is not a universal description of construction terms. The reality is that good-enough solutions to the classification dilemma have to be accepted. This has been recognized since Aristotle acknowledged that his original taxonomy (which focused on the genus) was not complete, later suggesting that accidental properties (called differentia) have to be added. Being among the first in construction (with a focus on the continuous issue of conceptualization), this ontology aims to present an initial good-enough ontological model for construction concepts. The DOCK is an open-world ontology. In a closed-world model, everything that is not included in the model is assumed to be false (i.e., the model includes all the true possible assertions). In contrast, in an open-world model, everything that is not included in the model is simply unknown or undefined—no assumptions are made about items not listed in the model (Mazzocchi 2010).

Methodology

Ontologies have been developed in many domains of knowledge—for example, philosophy, cognitive science, language, and information retrieval. The style of modeling in each of these domains varies markedly. Most of the informatics ontologies seem to follow a rational–intuitive approach that is guided by (or rooted in) artificial intelligence. For example, *WordNet* (a linguistic and information-retrieval schema) is built on the basis of the principal of naive semantics with a cognitive-science bent (Fellbaum 1998). Additional major ontologies that follow the same lines include CYC (extracted from the word *encyclopedia*) ontology (Lenat 1995) and Toronto Virtual Enterprise Ontology (TOVE), which was developed by Fox and Gruninger (1997).

The TOVE approach follows a pure AI mentality: An ontology is a formal model. Researchers have to establish a set of competency questions (model specifications), develop the model (extensively using predicate calculus and first-order or descriptive logic), and then test how the ontology satisfies competency questions. This formal, axiomatic approach was also used by others (see Staab and Maedche 2002) because the main function of ontology in the 1990s was to reason in the context of (the then-new field of) automated processes or virtual (online) transactions (Guarino 1995). Some researchers added features that relate to testing the usability of an ontology as part of an application. Gomez-Perez et al. (1996) advocated using input from end-users and focus groups as part of life-cycle management of ontology. For a nice comparison of these and other approaches, see Almeida (2009). Lately, with the advancement of the social Web, there is an added emphasis on the linguistic and communication aspects of ontology (in contrast to formal reasoning). This can be seen in the adoption of ontologies in search engines (such as Google) and the attempt to create a full domain-specific language in the biomedical field (Bodenreider 2004). This has led to the development of linguistically rich philosophical/abstract ontologies. Examples of such movement can be seen in

J. Constr. Eng. Manage. 2013.139:768-784.

three major upper ontologies: standard upper ontology (SUO), descriptive *ontology* for linguistic and cognitive engineering (DOLCE), and CYC (Jureta et al. 2009).

The DOCK attempted to analyze and benchmark these three upper ontologies. No one upper ontology was targeted specifically, but postdevelopment analysis reveals a closer link to DOLCE structure and some influence of CYC terms and axioms. The IFC and existing classification systems had more influence on DOCK (especially with regard to terms used); TOVE, too, was a major influence and inspiration (especially in concept structure and axioms). There are clear resemblances between construction and manufacturing in many aspects that made many of TOVE's structures and assertions intuitively relevant to construction. Being concerned with conceptualization over reasoning, however, DOCK is far less formal than TOVE.

Developed over a span of 9 years through benchmarking best practices from Fox-Gruninger and Gomez-Perez approaches, DOCK evolved over three phases, as shown in Fig. 1. In each phase, DOCK was used to develop a set of subdomain ontologies and a set of semantic Web-based applications. Input from industry experts (through formal interviews) and end-users (in the form of focus groups) was solicited throughout the development of DOCK, as listed in Fig. 1. A total of 145 experts were involved, including 120 in one-on-one interviews, 21 in focus-group format, and four in the form of peer reviews (the roles of each group are clarified in subsequent sections). First, a domain taxonomy for construction terms was developed and tested as part of the e-COGNOS project (El-Diraby et al. 2005). Sets of application taxonomies/ontologies were then developed in six related domains. These included highway construction (El-Diraby and Kashif 2005), environmental aspects of highway construction (El-Diraby and Wang 2005), outside plant construction in telecommunication (El-Diraby and Brecino 2005), stakeholder engagement (El-Gohary et al. 2006), building construction (El-Diraby and Zhang 2006), and privatized finance (El-Diraby and Gill 2006). The taxonomy was used as part of the e-COGNOS platform and in developing a Web-based platform for integrating life-cycle costing (El-Diraby 2006).

Second, a beta version of DOCK was created on the basis of the lessons learned from the e-COGNOS taxonomy and the application ontologies previously discussed. This version was used to develop two subdomain ontologies: an ontology for construction processes (El-Gohary and El-Diraby 2010a) and a product ontology (Osman and El-Diraby 2011b). A portal for integrating project processes (El-Gohary and El-Diraby 2010b) and a portal for coordinating urban utility routing (Osman and El-Diraby 2011a) were developed on the basis of these subdomain ontologies.

Incorporating all lessons learned from the steps described, DOCK 1.0 is the final (third) version of this domain ontology. Like the other ontologies, it was the subject of interviews with experts and was used to develop a new subdomain ontology for construction actors along with a port[illegible] social semantic communications (Zhang and El-Diraby 2012).

Validation of informatics ontologies is an elusive task. In fact, in the view of the author, it may not be suitable to use the word validation with ontology. Instead, evaluation is more appropriate. The rationale for this is based on the following reasons/arguments:

- The philosophical nature of domain ontology: Unlike application ontologies, domain ontologies are high-level claims about the fundamentals of knowledge in a domain. By presenting (and challenging) the very philosophical aspects of the understanding of domain knowledge, ontologies create as many questions (and objections) as answers. In fact, once an ontology is developed, it requires some modifications (to answer the new questions). The development of applied ontology, therefore, circulates between two major phases: theorizing to practicing and practicing to theorizing. Development steps and basic assumptions have to be revisited and calibrated frequently in an iterative manner before a stable and adequate ontology is reached.
- Relativism of abstract models: An ontology is composed of three visible elements: taxonomy of concepts, relationships, and

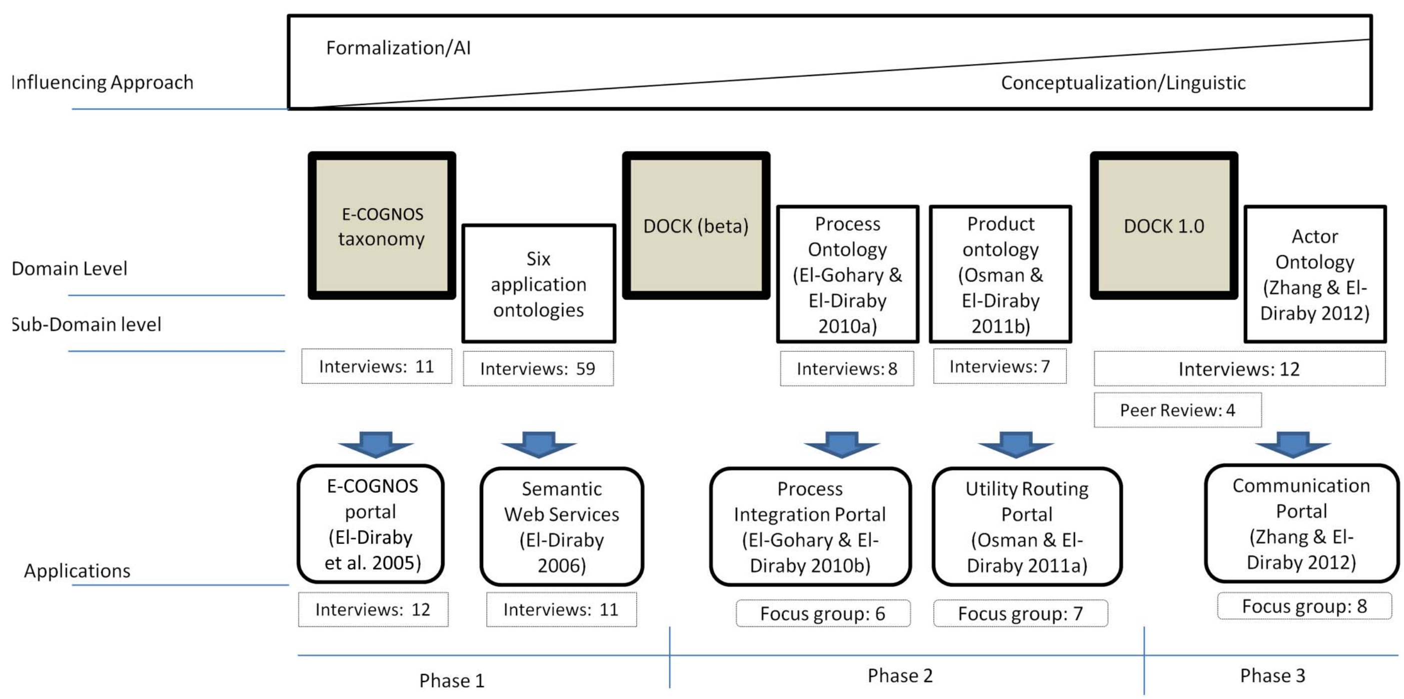

Fig. 1. Development of DOCK

axioms. Behind these are arguably the most important elements of an ontology: epistemological assumptions and the ontological model. Epistemological model/assumptions define knowledge-discovery modes (i.e., how we know what we know). An ontological model is the base layout (wireframe) of the main concept categories. The abstract nature of both models makes them essentially relative—it is hard to find universal consensus on them. Discussions about these two elements have been the subject of never-ending debate (and bitter disagreement) between philosophers over the last two millennia (for more, see El-Diraby 2012).
- The semiformal nature of DOCK: The DOCK was designed as an open-world ontology with reduced rigidity. Although DOCK is programmed in Web ontology language (OWL), it is not fully represented in situation calculus or first-order logic. This was the case given owing to the subjective nature of construction knowledge and the desire to enhance the textual/semantic clarity of DOCK.

One of the most important steps in managing the quality of theory building is to design a research methodology that triangulates tools to test work quality. The DOCK was developed in an iterative fashion in which intermediate evaluations were done after each phase of development, and a variety of tools were used to conduct these evaluations, as described in the following sections.

Exposing and Falsifying Assumptions about Epistemology and Ontological Model

Researchers have to constantly expose and critique their own assumptions to eliminate biases and test adherence to scope. At the end of each phase of DOCK development, an internal analysis was conducted to test the formalization of working hypothesis (primarily the epistemological and ontological models). Hypotheses and elements of these models were formally articulated and questioned after each step.

Scope Management and Competency Questions

Competency questions are developed to check the adherence of the ontology to the hypotheses (described previously). The research team used these questions to test the adequacy (extent and consistency) of modeling for each major concept. Four sets of competency questions were investigated: first, partonomy/inheritance (i.e., testing the part–whole and is–a relationships between major concepts and their parts or child concepts); second, modality (i.e., what varieties of the concept exist, where, when, and under which conditions?); third, descriptive questions (what attributes are essential to describe a concept? Do they change? What are the optional attributes, and when are they needed?); and fourth, relational questions (what other concepts must be modeled to fully cover the semantics of a concept? For example, in modeling a process, what are the actors, products, and mechanisms that must exists to fully describe the process?).

Consistency Checking

The ontology was represented in OWL using the Protégé ontology editor (Stanford Center for Biomedical Informatics Research, Stanford, CA). Reasoning is performed in the Protégé editor by utilizing a description logic reasoner called Racer (2005), which checks the consistency of an OWL ontology and flags any inconstancies or redundancies.

Expert Interviews

During the development of DOCK, 120 experts were interviewed and formally surveyed. A survey was conducted after each upgrade of DOCK, or after the development of a subdomain ontology. In total, 12 surveys were conducted with 7–12 experts in each. All surveys included the following questions, for which experts were asked to answer using a scale of 1–5, with 5 being most favorable:
- Navigation/traversing: The interviewees were asked to locate 25 concepts in the taxonomy and assess how easy it was to locate them. The average response for this question varied, ranging between 3.2 and 4.3.
- Categorization and abstraction consensus: Interviewees were asked to indicate whether they agreed or disagreed with the categorization of the 25 concepts (described previously) as suggested in the taxonomy. The average response for this question ranged between 2.9 and 4.1. Further, interviewees were asked to categorize 25 additional new concepts, which were intentionally removed from the ontology. Experts were asked to rate the ease of this categorization. The average response for this question ranged between 3.8 and 4.7.
- Representation and familiarity of concepts: After the interviewees became aware of the contents of the taxonomy, they were asked whether any major concepts were missing and/or should be added to the taxonomy. They were also asked to state where the concepts should they be added and to assess how easy it was to find a parent concept for them. The average response for this question ranged between 3.2 and 4.3.
- Quality of the ontological model: Experts were asked about the overall validity of the taxonomy, whether the taxonomy adequately covered the domain, and whether the taxonomy was biased to one subdomain. The average response for this question ranged between 3.4 and 4.4.
- Usability: Experts were asked to assess the potential benefits of using the taxonomy. This was a subjective question, and thus no scale was used.

Application Development

Developing a software on the basis of the ontology puts the usability and its coherence to the test. It can also help in assessing the lexical relevance of the terms, especially if the software uses social-Web tools. During its three phases, DOCK was used as the basis of nine different programs:
1. E-COGNOS platform;
   - The e-COGNOS platform is a large European project that aimed at developing a comprehensive set of ontology-centered Web services that support a full cycle of knowledge management (KM) in the construction domain. The KM cycle includes seven main activities: acquisition, transformation, indexing, updating, refreshing, searching/discovering, and sharing/dissemination.
2. A Web-based system for collaborative life-cycle costing;
   - This system represents costs as a hierarchy of cost elements. Each cost element has a dollar value that could be deterministic, probabilistic, or fuzzy. Several indigenous and exogenous factors (also represented in hierarchies) can have a set of impacts on the values of these costs. Through the analysis of different impact possibilities and probabilities, a decision maker can study various alternative scenarios and define the optimum set of costs and their values. A set of Web services are used to capture cost elements, factors, and their impacts (see El-Diraby 2006).

J. Constr. Eng. Manage. 2013.139:768-784.

3. An e-society Web portal;
   - This application was used to showcase highway-design features to local communities. By browsing through the portal, a user can learn about project elements, the impact of each element on sustainability issues, who is sponsoring each element, and what efforts have been made to reduce the impact of such elements on local communities (see El-Diraby and Wang 2005).
4. A system for creating communities of interest;
   - This application allowed users to profile their areas of interest through using tags from the ontology. An agent-based system was used to search registered users and suggest matches on the basis of their semantic profiles (El-Diraby and Brecino 2005).
5. A risk registry;
   - A prototypical semantic Web-based portal for communicating project risks was developed to illustrate the use of the taxonomy. In this portal, project partners are able to post and view risk items and their statuses, to observe who is handling these items, and to be aware of which decisions have been made to manage these items and which lessons learned are available to address these risks (see El-Diraby and Gill 2006).
6. A corporate memory system;
   - An agent-based system for supporting semiautomatic generation of reports, such as lessons learned, work forms, and meeting agendas, this supports easier access to and retrieval of these reports in future projects (see Zhang and El-Diraby 2012).
7. A process integration portal;
   - The Web-based application allowed different users collaborating in a project to semantically describe their work processes. It allowed each user to drag and drop work processes from the ontology along with their attributes (cost, duration, and location, for example) and related concepts (such as resources, constraints, and objectives). The portal then used a matching algorithm to help find synergies between these processes and to support the creation of a unified work flow between all project stakeholders (see El-Gohary and El-Diraby 2010a, b).
8. A system for coordinating utility routing in urban areas;
   - A geographic information system (GIS) based system to help visualize route data, interact with users, and support the needed discussions among stakeholders, the portal also includes a set of reasoners to study and manage conflicting constraints. The system is capable of extracting the attributes of each routing option, testing the interaction/conflicts between route attributes and the constraints of the surrounding area, studying the impacts of a route as stipulated in the ontology, referring users to existing best practices to help enhance routes or address conflicts, and develop objective measures for comparing different routes when needed (see Osman and El-Diraby 2011a, b).
9. A social semantic system for communication in construction projects;
   - Three technologies were utilized to address the challenges in information exchange and knowledge sharing. First, a semantic Web, which provides for more linguistic-friendly representation of knowledge. Second, a social Web, which allows people to share, reconfigure, and generate knowledge. Finally, publish/subscribe (pub/sub) systems, which provide for a push–pull scenario for information exchange. In the proposed system, any knowledge item (KI) (e.g., a document, website, or blog) is represented (tagged) with a semantic vector that describes its contents. The developer of the KI can push (share) this to his or her social network. On the other end of the spectrum, system users can build semantic profiles for their areas of expertise and/or interests. The system can pull (find) the most relevant KIs and forward them to the user. The system can also link peers with similar or complementary interests to one another to establish virtual ad hoc teams. The system was evaluated through input from two focus groups (see Zhang and El-Diraby 2012).

In each of these applications, the ontology was used primarily to store concept hierarchy and reasoning rules (through the axioms). In a few (such as application 7 in the preceding list), the ontology was used to facilitate semantic profiling of project processes and study possible matches. Each process was represented through a semantic vector representing all its related concepts (as expressed in the taxonomy). Semantic similarity measures were used to find related processes (processes that use the same resources or involve the same actor, for example). These processes typically belonged to independent schedules by different project parties. A coordinator can then link these processes to create an integrated flow of data (across jurisdictions). In other applications (such as application 8 in the preceding list), the ontology was used to create interoperability between different data formats. Data about infrastructure products (which are typically represented according to different GIS data standards) were matched to the concepts in the taxonomy to create consistency. In application 9, the ontology was used to semantically annotate documents (understanding the semantics of text). This was used to match documents to one another (during search) and also to user semantic profiles to route documents to suitable users based on their interests.

After each application development (and usage), an internal analysis by research teams and programmers was conducted to evaluate consistency and ease of use of ontology. Typically, this included expanding the ontology scope (areas that should be covered), amending the categorization approach (primarily providing more clarity about concepts and their modalities), and, more importantly, revisiting the structure and format of axioms (many turned out to be domain specific and not as universal as first thought).

Focus Groups and User Forums

Three sets of user forums were conducted in which domain experts were asked to use three different applications developed on top of DOCK (see Osman and El-Diraby 2011b; El-Gohary and El-Diraby 2010a; Zhang and El-Diraby 2012). Each forum included six to eight potential users (for a total of 21 users). Each forum started with a detailed demonstration of the system and its functionalities. Then, users were asked to experiment with the system through a set of demonstration cases. At the end, users were asked the same set of questions used in the expert surveys described previously (to evaluate the ontology). The results of their responses were augmented in the average scores provided previously. They were further asked questions about the software itself. User comments and discussions were recorded. The typical benefits listed by users can be categorized as follows:

- Improved usability: Positive feedback about the human-friendly nature of semantic systems was the first and most consistent comment by end-users.
- Consistency in data representation: The role of ontology in interoperability was typically the second observation by end-users. In the view of most, coordinating work flows and integrating data from various sources are evolving as major problems in today's work environment.

J. Constr. Eng. Manage. 2013.139:768-784.

- Time savings: In general, users noticed that automating parts of their work will save them time (which is typical of most software).
- Knowledge capturing and formalization: Users typically provided this type of comment later. It was also primarily observed by senior users. This is attributable primarily to the fact that many users were not engaged in knowledge-management tasks—most were frontline engineers and decision makers.

Benchmarking and Interdisciplinary Peer Review

Comparing the ontological model with ontologies in other domains provides significant insights. Similarly, input from peers in other domains can be very enlightening. Four informal interviews were conducted with developers and professors who work in the areas of ontology development and management in other domains. The objective was to critique the ontological model of DOCK and to gather insight on how to integrate it with other ontologies.

Main Ontological Model

Construction thing—the base concept—is categorized in a three-dimensional matrix (see Fig. 2). The first dimension encompasses the three pillars of any ontology: concept, relationship, and axiom. The second, orthogonal, dimension is modality (means for creating varieties of the things). For example, applying the modalities electronic, secure, and approved to the concept, document, produces the types: electronic document, secure document, and approved document, respectively. It also assures that approved electronic document will/can be recognized as a possible type of document. Recognizing modality as an orthogonal dimension to concept creates a flatter topology. Not doing so would have mandated defining a significantly lengthy taxonomy with many redundant layers.

Concept is categorized into five dimensions: entity, environmental element, abstract concept, system, and attribute (which is orthogonal to the first four). Entity is categorized (along two main modalities) into generic and secondary entities. Generic entities encompass three genuine (ever reoccurring) concepts in any informatics ontology (especially AI-based ontologies): action, product, and actor. An action is a process or an event. A product or a service is output of (a set of) actions that engage actors. A product is divided into knowledge (abstract constructs in the minds of human actors), knowledge item (a tangible representation of knowledge), decision, and physical product. Physical product is divided into simple products (such as a column), complex products (such as a bridge), and construction aids (such as a scaffold). Actor is divided into individual (such as civil engineer), organization (such as bank), and other actors (such as a driver-vehicle unit in transportation studies).

From the perspective of the process that produced it, a product is the output. For another, it could be used as a resource or can be a constraint [as in Icam DEFinition for Function Modeling (IDEF0)]. Secondary entity casts a concept to a secondary function in a certain situation. Equipment is a product of a manufacturing process (base context). However, in the context of every construction process, it is a resource. Contract (stipulations) is a product of the contracting process. For almost every other process, it plays the role of a constraint.

Secondary entity is divided into two main concepts: input and condition. Input is further divided into resource and mechanism. Condition refers to all concepts that are related to a process-actor-product continuum without being inputs. If it plays a controlling function, a condition becomes a constraint.

Next in the proposed model are environment and abstract concept. These refer to concepts that typically are outside the construction domain but interact with it. Environmental element is divided into natural element (such as air, soil, and water), artificial element (such as regulations, interest rate, monetary policy, and labor supply), and virtual element (such as a computer object). Abstract concept includes (the definition of) things such as time, space, risk, motive, and interest (of humans). The proposed model, in a true pragmatic fashion, intentionally does not make any claim on the structure of concepts in these two categories—whether they are

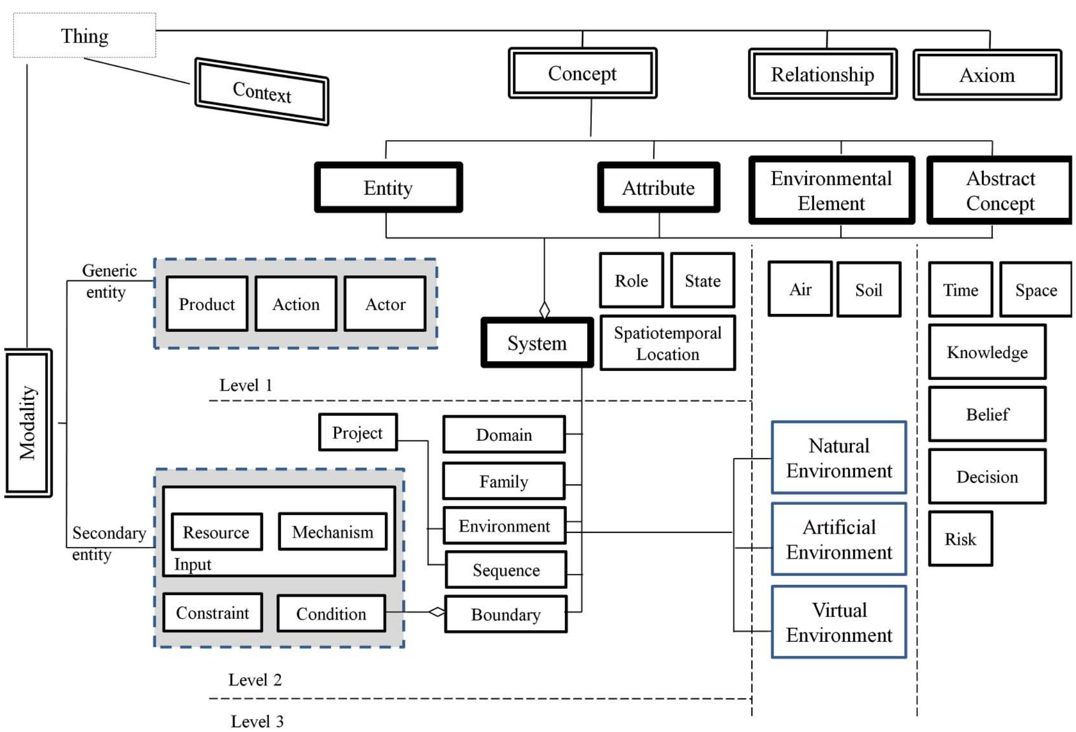

Fig. 2. Ontological model

J. Constr. Eng. Manage. 2013.139:768-784.

individuals or universals, or whether they are real or not. Such choice is immaterial to this ontology and is better left to philosophers. These categories are included nominally in DOCK.

A system is a collection of the elements described. Based on the modality of collating concepts, a system has a set of subcategories (domain, family, sequence, environment, and boundary). These will be explained subsequently.

Finally, attribute is orthogonal to all categories (entity, environmental element, system, and abstract concept). For example, cost is an attribute of any product, duration is an attribute of any process, and skill is an attribute of any human actor. The relationship between attributes and modalities is not that straightforward. Attributes are subject to modalities. One can use the modality, physical, to categorize attributes into physical and nonphysical attributes. Paradoxically, at the same time, modalities can (in some cases) be seen as having attributes. For example, one can talk about the prevalence of a modality. This is a typical causality dilemma—solving it is out of scope for construction researchers. Though DOCK asserts that attributes have modalities, it is silent about the nature of modalities.

These categories represent roots of concept trees (utilizing is–a relationship). Typically, they will also be linked cross-tree by a set of relationships and will obey a set of axioms. The final dimension, context, allows for creating variations on the standard concept behavior (through changing, modifying, or relaxing their linkages to other concepts and their adherence to which axioms), hence creating new worlds (scenarios in which concepts behave differently from the standard perspective).

It is important to point out that concept is organized into three levels that can be seen as a measure of uniqueness (or independence). Concepts in level 1 (entity, environmental element, abstract concept, system, and attribute) represent the core concepts. Level 2 includes the two main modalities of entity and the subtypes of system. The final level is meant for concepts that suit an application ontology. The following sections further describe some of the main concepts in the model.

Abstract Concept

Abstract concepts are meant to encompass mental constructs that could be relevant to construction terms. It is not the aim of this ontology to address the holistic structure or nature of abstract terms (this is a job better left to philosophers). Some of the main concepts under this category include knowledge, risk, decision, motivation, time, and space.

Regarding time, the representation of temporal aspects of construction works is of great importance. Time is modeled in a manner similar to TOVE and the time ontology suggested by Hobbs (2002), both of which are based on Allen's temporal logic (see Allen and George 1997). Time is represented by points and periods (interval) on a continuous time line as follows (see Fig. 2):

- Instant (point in time): This represents an instantaneous (snapshot) occurrence of time: a date, a start time, the finish time, a milestone (in a schedule or a contract); and
- Interval (chronoid): This is a continuous period of time that is bound by two time points (start and finish) and could include additional intermediate time points. Allen shows 13 different types of inter-interval relationships (Allen 1983).

Abstract concepts are normally used to describe an attribute (for example, the motive of an actor). In other cases, it is the (abstract) parent concept of a construction concept. For example, a stage (of a process or a product) and the duration of an activity are types of chronoid. Project risk (an attribute) is defined in terms of risk. Cost-analysis knowledge (a product) is defined in terms of knowledge.

Environmental Element

This is an unordered set of terms (the categorization and organization of which are the domain of knowledge of other experts) that aim to describe the surrounding environment to construction operations. They are classified into three main groups:

1. Natural element (such as air, water, plants, animals);
2. Artificial element (all manufactured/artificial substances, such as a building, a roadway link, or a manufactured metal; also refers to logical environments, such as law, regulation, policy); and
3. Virtual environment (whereas the preceding two groups describe natural and manufactured physical matter, this group is dedicated to conceptual artifacts such as computer objects or a field in a database).

Project

Before presenting the detailed ontological model of entity, it is suitable to define project—the typical container of construction entities. A project is a system (collection) of entities in which a set of actors are engaged in a sequence of processes (and events) to utilize a multitude of resources to produce a product or deliver a service. A project life cycle is composed of a chronologically bound set of phases. Each phase encompasses a set of processes and involves a set of actors with defined roles. The default phases of a construction project were adopted from those of the Construction Industry Institute: business planning, pre–project planning, implementation, operation, and decommissioning.

Action

This entity defines the main perdurant components of a project. In other words, the time line of a project is composed of a sequence of actions that progress from the start of the project until the end. Actions are entities that happen rather than endure. The output/consequence of an action is either a product (normally for processes) or a change in state (normally for events and some processes). For example, the completion of a design process advances product life cycle and the state of a product into designed product. The design-evaluation process advances the design life cycle into one of (at least) two states: approved design or unapproved design. The bid-evaluation process (normally) changes the status (role) of one bidder into the winning bidder. The bid-award process (normally) changes the role of the winning bidder into the contractor.

Process

A process is a time-consuming perdurant entity that engages a set of actors and possibly consumes resources to produce an outcome. The outcome could be a product, a service, or a change in the attributes or states of other entities. Process includes systematic application of human knowledge/skills, techniques, and resources to achieve an outcome that is normally a product of significance to a project.

Processes are the main concept for representing and controlling temporal aspects of construction work. As shown in Fig. 3, a process is part of a project phase. A process could be composed of a set of subprocesses. A subprocess could be composed of a set of activities. An activity is composed of a set of tasks. The life cycle

J. Constr. Eng. Manage. 2013.139:768-784.

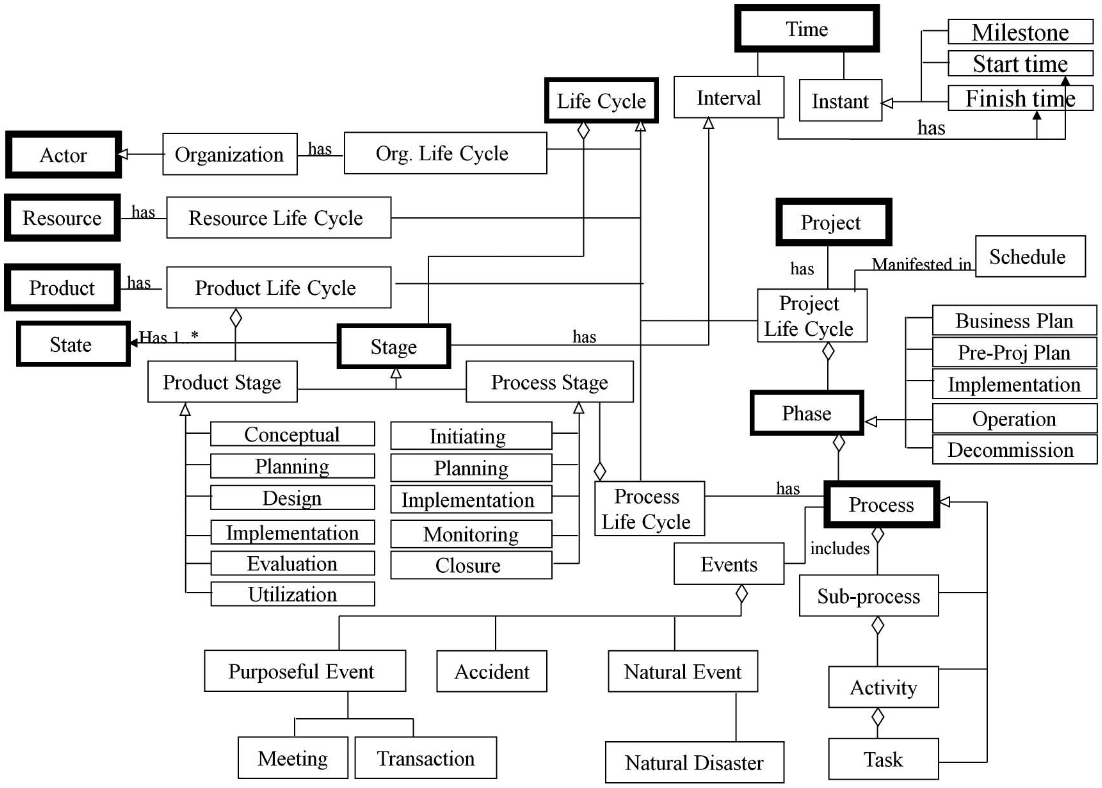

Fig. 3. Time representation in DOCK

of a process is the composition of its subparts. A process stage is a type of time interval. It can encompass one or more activities/subprocesses. They add up to the same life cycle of the process (however, they are not necessarily aligned with the start or end of activities or subprocesses). The default types of process stages are (in chronological order): initiation, planning, implementation, monitoring and control, closure, usage, and decommissioning. Of course, not every life cycle will include all types of stages. Further, during its working each stage could go through one or more operational states (which is an attribute of a process stage). The default operational states of a process stage are dormant, stopped, resuming, executing, and finished.

Event

Just like processes, events are perduring entities that could consume time and space and use resources. In physics, an event is an instantaneous action that takes place at a point in time, where the state of the world changes. Events are typically contrasted with processes, which occurs across intervals of time. Typically, an event marks the start or the end of a process. However, events are not necessarily instantaneous—some can consume time. Such events are divided into two main categories:

1. All events are triggered intentionally and carried out by a human for a dedicated purpose. This includes, for example, a meeting, a phone call, or a party. Some of these concepts (such as meetings) share many common attributes with process. For example, they include application of knowledge, skills, and tools to produce an outcome. However, one does not build a project through just a set of meetings. To help in the demarcation, DOCK used a specific set of events in contrast to an open door for adding processes. Intentional events are categorized as
   - Meeting; and
   - Transaction, subdivided into administrative (such as a submittal) and business (such as a purchase order).
2. Unintentional events are categorized as
   - Natural event: A nonartificial event that takes place as an act of God, a natural event includes reoccurring natural events (such as sunrise, sunset, rain, and heat and cold waves) and catastrophic events (such as earthquakes or hurricanes). These events can also be classified as earth-related natural events (such as earthquake or soil failure) and weather-related natural events (such as rain or hurricane).
   - Accident: This refers to events that happen unexpectedly—more of unforeseen or unlikely events. Unlike natural events, accidents may have a human as an instigator (primarily through negligence).

Product/Service

Product and service are the output of processes and some purposeful events (primarily meetings). Product has four major subcategories:

1. Knowledge;
   - Almost every action that involves humans produces or amends human knowledge. This mental construct remains in the mind of the actor unless it is manifested through a means suitable for communicating it. An engineer possesses design knowledge, and when designing a new bridge, first establishes a mental construct of the bridge. Another example is the lesson learned at the end of a project. Such lessons present wisdom that a person gains through project execution. A person can recall such wisdom (or knowledge) in future projects.

J. Constr. Eng. Manage. 2013.139:768-784.

2. Knowledge item;
   - While knowledge refers to a mental construct (something in the minds of humans), a knowledge item is the physical or symbolic manifestation of such knowledge. To illustrate the difference between knowledge and a knowledge item, consider a bridge design; the mental conceptualization of how the bridge should be designed is knowledge. In contrast, design documents (including specifications and drawings) are knowledge items that manifest the mental construct. Similarly, a database of lessons learned is a manifestation of lessons learned. Theoretically, there is a piece (or pieces) of knowledge that proceeds every knowledge item.
3. Physical product;
   - This refers to tangible products, including basic products such as a beam, a column, or a footing; complex products such as a house, a bridge, a highway, or a water-distribution system; and construction aids (such as a scaffold).
4. Decision;
   - Not all actions lead to a physical product. Some actions (for example, bid-evaluation processes) lead to a decision (selection of winner).

A product is developed/realized through a set of product stages (type of interval). Such stages could encompass/span a set of processes or even a set of projects. The sequence of product stages is called a product life cycle. A product stage normally corresponds to a similar process stage (see Fig. 3). They typically correspond to a modality of a product. The default product stages (and the corresponding product modalities) include

- Conceptual stage (corresponds to conceptual product);
- Planning stage (corresponds to a product under planning or a planned product);
- Design stage (corresponds to a product under design or a designed product);
- Implementation stage (corresponds to product under construction);
- Evaluation stage (can be linked to finished, ongoing, or disputed product); and
- Utilization (typically linked to a finished product).

During its working, each stage could go through one or more operational states (which is an attribute of a product stage). The default operational states of a product stage are dormant, stopped, resuming, executing, and completed.

Actor and Role

Actors can hardly be separated from their roles. In fact, discerning the two concepts has been a long-standing dilemma in knowledge representation. Steimann (2000) summarized 15 features for the actor–role continuum in conceptual modeling and the multitude of modeling approaches that have been presented in this regard. The following list summarizes the proposed (pragmatic) approach in discerning roles and actors:

1. The definition of roles is process driven. They are stereotypical functions that hold irrespective of the actor who is performing them (role holds no matter who is doing it). As suggested by the work of Sowa (2000), a role is meaningful only in the context of a relationship to a process or an actor. The role of a project manager can be performed by an engineer (civil, mechanical, electrical), an architect, or a technician. In other words, a role is an externally assigned attribute of an actor.
2. Actor is a reflection of identity. It relates to innate capabilities/competencies of humans and organizations that, hence, characterize their identities. By virtue of education, an engineer possesses certain (stable) capabilities and attributes that hold no matter what is assigned in a project (actor identity holds irrespective of what is being done). An engineer knows how to use mathematical and logical notations to formulate and solve problems even if working as a CEO of a company, designer, or carpenter.
3. An actor may play different roles simultaneously or in different contexts. Actors can acquire and abandon roles dynamically.

Fundamentally, an actor in DOCK is very much attached to the education/training of an individual or what an organization can claim as a stable track record of business. For example, an engineer, an architect, and a certified accountant are all actors. A contracting firm, a manufacturing company, a design firm, and a bank are all actors, too. In contrast, a designer, a project manager, a quality manager, a field engineer, a contractor, a supplier, an owner, and a client are all roles. To illustrate the rationale, a contracting firm can assume the role of designer, contractor, or project manager. A manufacturing firm can assume the role of quality manager, supplier, and owner.

Given the specificity of education (and stable experience) as demarcation means, DOCK includes a limited number of actors and an extended number of roles. The basic types are individual (such as a civil engineer, laborer, and certified surveyor), organization (such as contracting firm, bank, and court), and other actors [for example, an agent (in software) and a driver-vehicle unit (in transportation simulation)]. Almost anything else in the actor-role continuum is, by default, a role.

Still, there could be some linguistic confusion. Take again, for example, the concept of carpenter or the concept of accountant—these are generally seen as actors but can also be seen a roles, too (in a small project or small municipality, an engineer can do such jobs). To help resolve this, the proposed model utilizes postterm qualifiers to discern such linguistic ambiguity. A carpenter (experienced), carpenter (certified), or carpenter (unionized) all designate an actor—someone who has innate/stable qualifications on the basis of education, expertise, or designation. If carpenter is used without any qualifiers, then it is interpreted as role. Of course, users can ascertain that by using the keyword role as postterm qualifier: carpenter (role). Similarly, project manager refers to a role unless a qualifier is designated—e.g., project manager (certified), which indicates an actor with formal education/experience/designation in project management.

Input (Resource and Mechanism)

The representation of resources in this model started from that of TOVE: "Being a resource is not an innate property of an object, but is a property that is derived from the role an object plays with respect to an activity. Hence, the resource ontology includes the concepts of a resource being divisible, quantifiable, consumable, reusable, a component of, committed to, and having usage and consumption specifications" (Tham et al. 1994). Process can use or consume a resource—a piece of equipment is used, whereas energy is consumed. Similar to TOVE, there are four different status predicates for resources: committed, enabled, disabled, reenabled, completed.

In many models, resources encompass human resources, too. However, DOCK, concerned with semantic clarity, limited the definition of resources to physical and abstract concepts (entities) that are utilized in actions (processes and events). Human resources and participating organizations are included under the umbrella of

J. Constr. Eng. Manage. 2013.139:768-784.

actors (to highlight the active role of human and organizational participants). Further, and highlighting the importance of knowledge, DOCK includes a separate secondary entity (called mechanism, as in IDF0) to refer to knowledge products that can be an input to a process. This includes the following:

- Guide: This is a fundamental tool that helps formalize a metaphorical model for analyzing a problem, a process, or a product. This includes theories, such as the theory of structures and the theory of architecture; and algorithms, such as scheduling algorithms and resource-allocation algorithms. It also includes strategies. If formal theoretical representations are not available, strategies represent the fundamental mechanism to guide the processing of work. As an example, company strategy and project implementation strategy are two major tools in managing most processes. Finally, guides also include best practice. At lower levels of work, best practices are one important tool in forming a model of work entities.
- Method/technique: These are the approaches that help workers perform their work (based on the theoretical metaphor that was built using guides). This includes methods such as management by objective (for managing a task) and techniques such as the lift-slab technique (in erecting a building).
- Measure/test: After theorizing the work (using guides) and using some techniques for executing them, the conformance of the output to the objectives has to be measured. This is done through conducting tests (for physical entities) and using metrics (for other entities).

Condition and Constraint

Condition labels concepts as being boundary of or related to an entity, without making any assessment of the role they play. This includes laws, code, specifications (including owner specifications, manufacturer specifications), user requirements, physical conditions (such as topography and weather), market conditions, and cultural norms. Opportunities (business or otherwise) and threats (such as construction risk) are types of condition.

Constraint is, however, the most important type of condition. Whereas conditions describe related concepts without making any claim on their influence, constraints are meant to cast a condition to a controlling role that restricts the feasible behavior/attributes of an entity. Constraints are categorized across many modalities, such as domain (e.g., engineering and legal constraints) or span of control (controllable and uncontrollable constraints or avoidable and unavoidable constraints). They are also categorized according to their norms into norm of conduct, norm of competence, and norm of conformance.

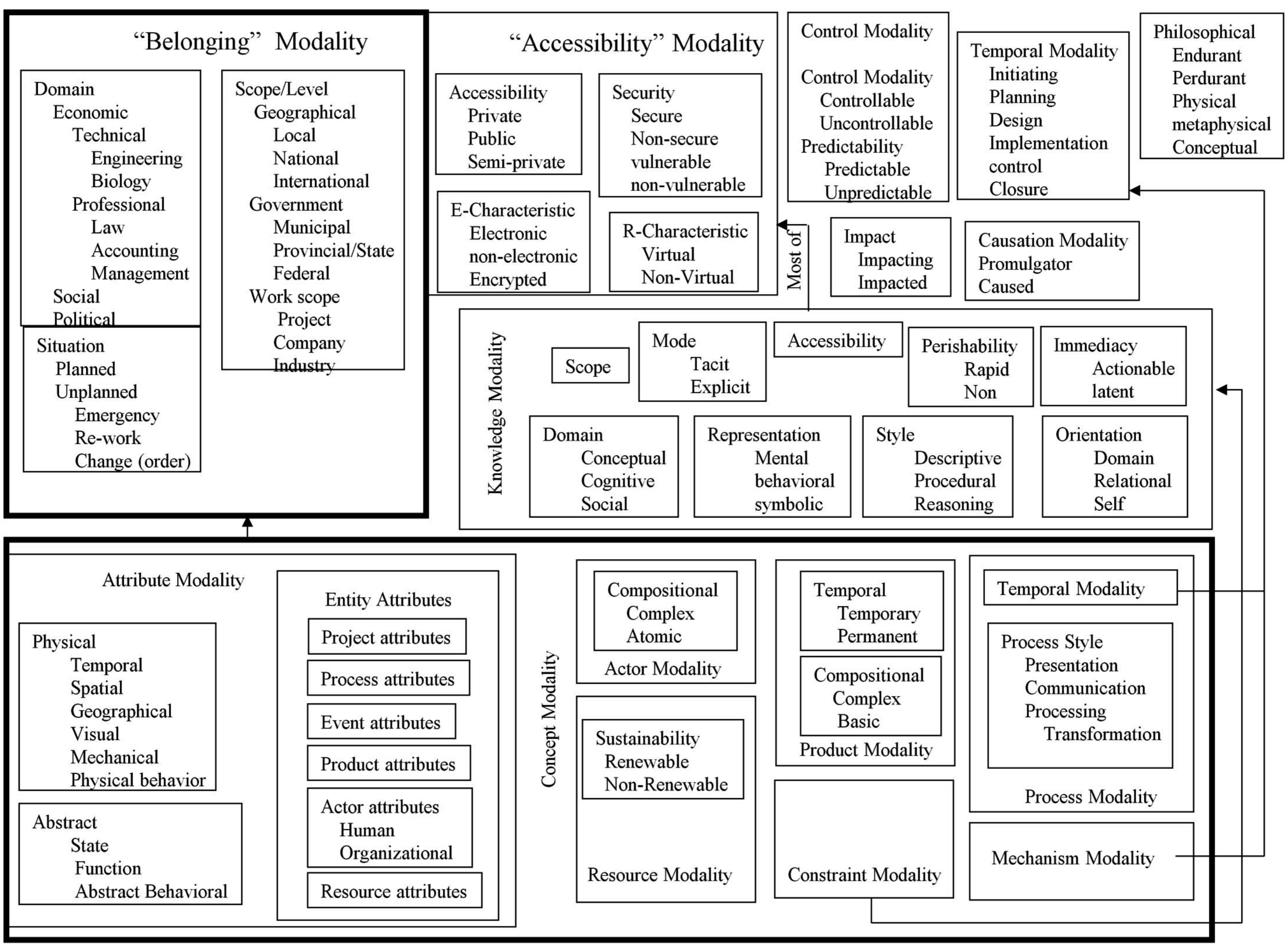

Fig. 4. Modalities in DOCK

Modality (Possibilities)

The modality concept provides means for binding a type to a basic concept, thus creating a new child concept. Some modalities, such as the following, are generic modalities that can be applied to all concepts (see Fig. 4):

- Belonging modality: categorizes concepts on the basis of their belonging to a domain of knowledge (such as engineering process versus administrative process), the level they belong to (such as project-level process and company-level process), or the situation they belong to (planned and unplanned). For example, the design process is an engineering process that normally takes place at the project level. It could belong to a planned or an emergency project.
- Accessibility modality: categorizes access to entities into (as a sample) public accessibility (such as public versus private document), security (such as secure website), e-characteristics (such as electronic versus nonelectronic document), and reality characteristic (such as virtual and nonvirtual organizations).
- Control modality: includes modalities related to the predictability (predictable and unpredictable) and controllability (controllable and uncontrollable) of an entity.
- Temporal modality: relates mostly to events and processes, including the following categories: initiation, planning, design, implementation, control, and closure. For example, in the context of a project, the pre–project planning process is categorized as initiating processes, and others are categorized as implementation processes (such as the erection process).

Other modalities are relevant to a specific concept. They reference basic modalities and include additional features that are only applicable to the type of concept at hand. For example (as shown in Fig. 4), knowledge modalities include scope, mode (explicit or tacit), perishability (rapid or none), immediacy, nature (conceptual, cognitive, social), representation (mental, behavioral, symbolic), style (descriptive, procedural, reasoning), and orientation (domain specific, relational, and self-oriented).

Attribute

An attribute is an abstraction or description of a characteristic or a property of an entity. It normally reflects means to help identify and distinguish it from other entities. It could have a value. Attributes are presented in a multidimensional view (see Fig. 5). The basic dimension divides attributes into physical and abstract concepts. On another dimension, attributes are categorized into intrinsic and extrinsic attributes. Other dimensions include behavioral attributes, usability attributes, state, functional attributes, and temporal and spatial attributes.

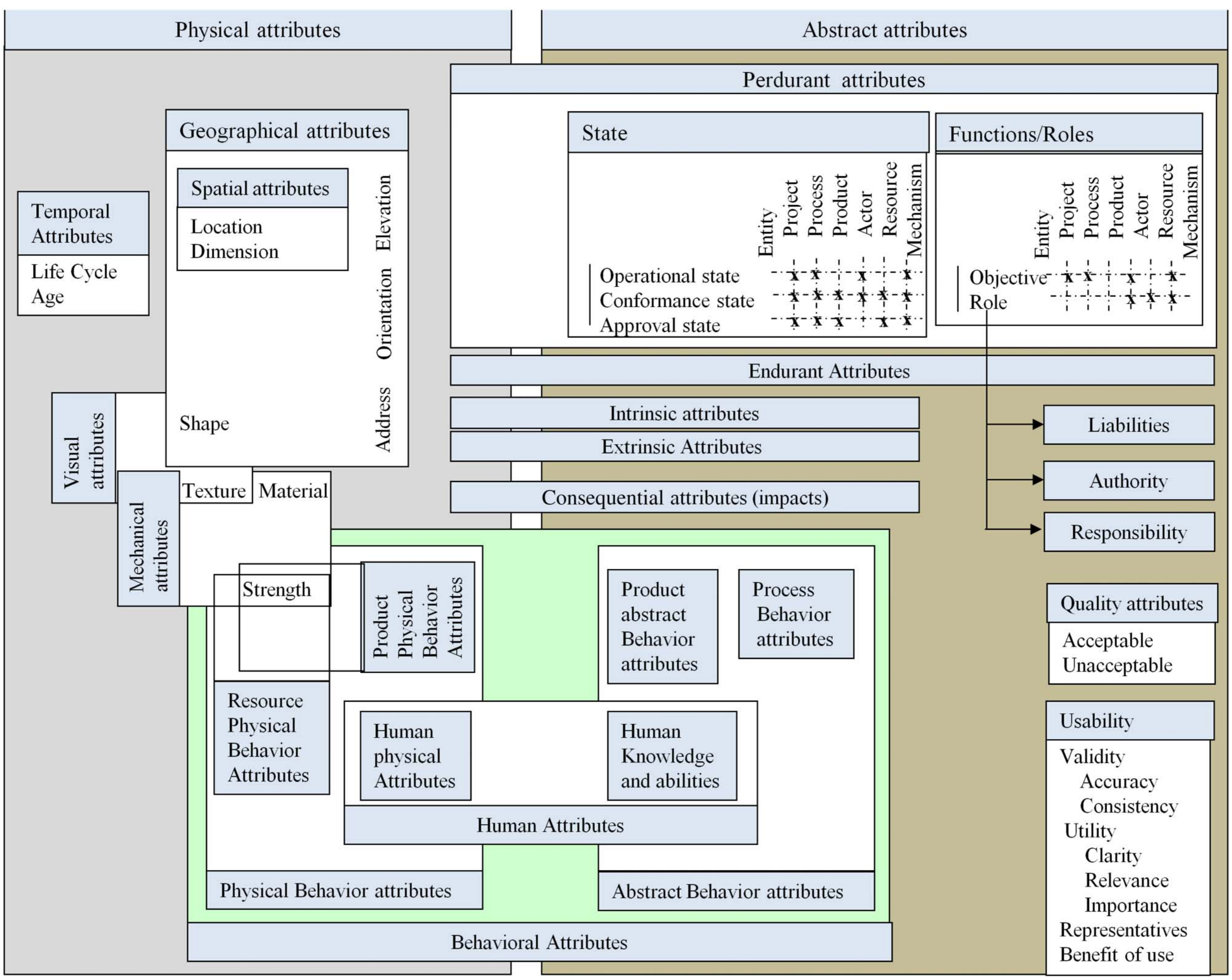

Fig. 5. Attribute modalities in DOCK

J. Constr. Eng. Manage. 2013.139:768-784.

System

A system is a collection of entities along with their attributes and conditions that represent a meaningful set. It includes the following types:

- Domain: a coherent collection of concepts that form a cluster of knowledge. One can talk about the domain of sustainability, which encompasses all the products, processes, resources, and actors related to sustainability. Similarly, one can talk about the domain of productivity management, project delivery system, safety engineering, quality assurance, and policy-making.
- Family: a bag of concepts that share a common modality. They do not have to belong to a coherent domain of knowledge. A family can be seen as a type-sensitive system (set of entities that belong to the same type). For example, the family of e-entities includes e-processes (such as e-bidding) and e-products (such as electronic purchase orders).
- Sequence: an ordered list of concepts—a life cycle (of processes), a schedule, a chain of command.
- Environment: a mix of related concepts that share a common theme or a spatiotemporal location. The built environment refers to the artificial built products in an area. A transportation network is an environment of road links and driver–car units. A software platform is a virtual environment. A project is a child of sequence and environment.
- Boundary: includes/collates all entity conditions. Boundary can be seen as the semantic vector of an entity—its extended signature.

Context

The previously defined terms are the roots of concept taxonomies, which are built using is–a relationships. In any ontology, these concepts are related to cross-taxonomy using a set of relationships. The configuration and nature of these relationships (and related axioms) morphs from one context to another: "It is not the mere problem of identifying an entity in the world that is central to the ontological representation of the world, but the ability to reidentify an entity in all its possible forms, or more formally reidentification in all possible worlds" (Tamma and Bench Capon 2002).

Contexts are possible worlds. Through varying the type (modality) of relationships and reformatting some axioms, a context creates a new world from a generic one (one with default perspective). For example, design-bid-build and design-build are two contexts that are driven from the project delivery system, in which the sequences of processes, their relationships, and axioms are different. In other words, context is a means to create a subdomain ontology by weaving concepts in a different way. To take an example, the domain of construction management includes all relevant concepts in construction management with a specific set of relations

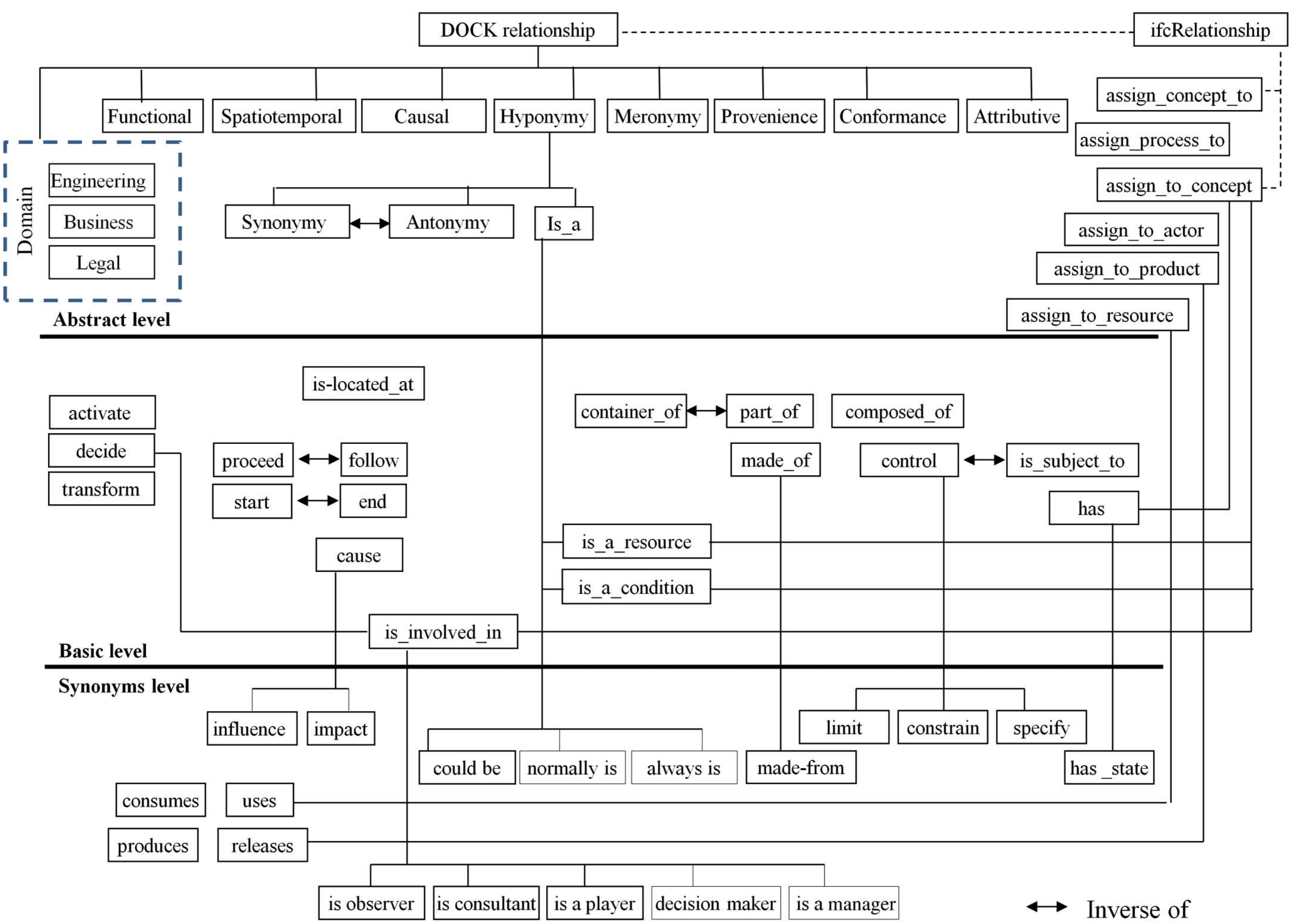

Fig. 6. Relationships in DOCK

J. Constr. Eng. Manage. 2013.139:768-784.

Table 1. Taxonomy of Relationships in DOCK and Their Semantics

Type (objective)	Relationship/subtypes	Example
Hyponymy	Is–a/always is/normally is/to an extent is/could be	—
Synonymy	Is the same as	—
	Is like	—
Antonymy	Opposites	—
	Negation of	—
Meronymy	Part-of	—
	Composed of	—
Provenience	Made of	Product is made of material
	Made from	Green product is made from renewable sources
Spatial	Is located at	Project is located at place
Temporal	Takes place at/happens at	Project takes place on date
	Proceeds (follows)	Process A proceeds Process B
	Start (end)	Kick-off meeting (event) is start of project
Attributive	Has	Product has shape; actor has role; actor has skill; attribute has value
Contingency/causal	Causes/results in	Contract results in liabilities
	Influences	Owner influences design
	Impacts	Weather impacts excavation
Conformance/constraint	Is subject to/adheres to	Design adheres to code
	Applies to	Code applies to design
	Controls/limits/constrains	Contract controls construction schedule
Function		
Activation	Enacts/initiates	Owner initiates project
	Requests	Owner requests bid
Decision	Approves/permits/accepts	Government approves project plans
	Is involved/manages/participates/observes	Users participate in pre–project planning process
Transformation	Manifests	Knowledge item manifests knowledge
	Utilizes	Process utilizes method
	Uses	Process uses resources
	Guides	Strategy guides project or design
General	Enhances	—
	Subject of	—
	Exemplifies	—

and axioms. To generate the context of emergency construction (in contrast to a typical construction context), some relations need to be amended. Similarly, the context of building construction will have specific relationships and axioms that are different from the context of civil or infrastructure construction.

Relationships

In an ontology (or an object model), relationships create combinations of concepts. According to relational theory, one can understand reality or physical systems only through (1) their relative position (link) to other objects, and (2) their intrinsic properties. In a relative universe, the linkage of a concept to others (in the ontology) enriches its definition and clarifies its context. Relations are one of the indirect tools to support the presentation of tacit knowledge about concepts. Through triangulating concepts (through their relationships to other concepts), ontologies can embed (at least partially) some of the tacit knowledge available about concepts. Finally, and more importantly, relations are one of the most fundamental tools to enrich the expressiveness and linguistic aspects of an ontology.

Relations in DOCK are organized in a set of taxonomies, in which a relation can have subtypes according to modalities (see Fig. 6):

1. Hyponymy (is–a): This category aims at defining the type, or kind, of a concept. Such relationships are the backbone of building taxonomy trees. Some variations on is–a could include the following relations:
   a. Synonymy;
      - The following relationships represent additional means for describing types and categories (in a softer manner): is_the_same_as, is_like, and is_similar_to.

   b. Antonymy;
      (1) The following set of relationships can contribute to defining the category or type of a concept by excluding it from a certain domain (or group of concepts):
         (*a*) Opposites;
            - This relationship establishes that Concept A is the opposite of Concept B (for example, a local project is seen as opposite to a national project, and it is also opposite to an international project).

         (*b*) Negation of;
            - This relationship is more restrictive than opposite because it creates the exact negation of a concept (for example, a renewable resource is the negation of a nonrenewable resource). Although opposite illustrates a contrast between two concepts, negation establishes that a concept is the complete and only inverse of another concept.
2. Meronymy (whole–part): There are two ways for partitioning complex concepts: decomposition into subclasses (taxonomies) and decomposition into parts (partonomies). The DOCK recognizes the following fundamental whole–part relationships: part_of, container_of, and composed_of. The first is from the perspective of the part, and the latter two assume the perspective of the whole.

J. Constr. Eng. Manage. 2013.139:768-784.

3. Cross-tree relationships (behavioral/descriptive relationships): This type of relationship is key for enriching semantics in DOCK (see Table 1). This type of relationship enriches the representation of knowledge about a concept in a manner similar to triangulations. In other words, the essence and behavior of a concept becomes more clear and better defined through clarifying its relative position and relationship to other concepts.

Fuzzy Representation of Relationships

Some of these relationships have been further detailed through the use of fuzzy hedges, which allow for the creation of relationship flavors. All fuzzy forms of relations are meant to support a fundamental paradigm in DOCK; every concept has a possibility (potentiality) and a probability (of belonging). For example, is–a has the following fuzzy subtypes:

- Always_is (possibility): for example, "time management is_always a project objective";
- Normally_is (possibility): for example, "a transit authority is_normally a government agency";
- To_an_extent_is (possibility): for example, "the requirement analysis process is–a engineering process, and to_an_extent_is a business process" (the assertion in this example is that requirements analysis is fundamentally an engineering process, but, at the same time, it includes some features of a business process; another example could be "project payroll process is–a project-level process and to_an_extent_is a corporate-level process"); and
- Could_be (probability): for example, "a design firm could_be a contracting company."

In the whole-part aspect, included-in and includes are used as softer types (possibilities) of part-of and composed. Also, could-belong-to is used for representing a probabilistic version of part-of. Similarly, the involved-in relation (as in an actor involved in a process) could take different flavors (possibilities). An actor could manage, participate, or just observe a process.

Axioms

Axioms are rules and formal assertions from which other assertions follow by the application of logical rules. They represent

Table 2. Sample Axioms

Axiom modality	Sample axiom	Purpose/role of axiom
Consistency (general) axioms		
Part–whole	$\forall x, y$ part_of$(x, y) \leftrightarrow$ container_of(y, x)	Container_of is inverse of part_of
	$\forall x, y, z$ part_of$(z, y) \wedge$ part_of$(y, x) \rightarrow$ part_of(z, x)	Part_of is transitive
	$\forall x, y, z$ composed_of$(x, y, z) \rightarrow$ part_of$(y, x) \wedge$ part_of(z, x)	Composed_of entails part_of
	$\forall x, y$ disjoint$(x, y) \rightarrow \neg\exists z$ part_of$(z, x) \wedge$ part_of(z, y)	Definition of disjoint concepts
Time	$\forall T$ Time$(T) \rightarrow$ Instant$(T) \vee$ Interval(T)	Time has two subtypes only: instant and interval
	$\forall t1, t2$ before$(t1, t2) \rightarrow t1 < t2$; $\forall t1, t2$ after$(t1, t2) \rightarrow t1 > t2$	Defining before and after
	$\forall t1, t2$ Instant$(t1) \wedge$ Instant$(t2) \rightarrow$ before$(t1, t2) \vee t1 = t2 \vee$ before$(t2, t1)$	Time is linear total order
	$\forall t1, t2, T$ start$(t1, T) \wedge$ start$(t2, T) \rightarrow t1 = t2$; $\forall t1, t2, T$ end$(t1, T) \wedge$ end$(t2, T) \rightarrow t1 = t2$	Unique start and finish of each interval
	$\forall T1, T2$ Interval$(T1) \wedge$ Interval$(T2) \wedge$ proceed$(T1, T2) \rightarrow$ proceed_FS$(T1, T2) \vee$ proceed_FF$(T1, T2) \vee$ proceed_SF$(T1, T2) \vee$ proceed_SS$(T1, T2)$	Four types of precedence (these are subsets of Allen's 13 options for interval relationships)
State	$\forall p1, v1, v2$(process$(p1) \vee$ product$(p1) \vee$ project$(p1) \vee$ resource$(p1)) \wedge$ has_state$(p1, v1) \wedge$ has_state$(p1, v2) \rightarrow v1 = v2$	Process, product, project, or resource must have one state only
Causality	p_cause$(x, y, p) \wedge p \neq 0 \rightarrow$ cause(x, y)	Causality can be probabilistic
	$\forall x, y$ cause$(x, y) \rightarrow$ proceed(x, y)	Causality necessitates precedence
Construction-specific axioms		
Definitional	$\forall p1, tf1$ process$(p1) \wedge$ total_float$(tf1) \wedge$ has_float$(p1, tf1) \wedge tf1 = 0 \rightarrow$ critical$(p1)$	Process is critical if it has zero total float
	$\forall x$ state$(x) \rightarrow x =$ executing $\vee\ x =$ resuming $\vee\ x =$ dormant $\vee\ x =$ stopped $\vee\ x =$ suspended $\vee\ x =$ completed	Entity can be in state of executing, resuming, dormant, stopped, suspended, or completed
Attributive	$\forall p1, v1, v2$(process$(p1) \vee$ product$(p1) \vee$ project$(p1) \vee$ resource$(p1)) \rightarrow \exists Cd, Cx, Cs, Cr, Ct$ dormant_cost$(Cd, p1) \vee$ executing_cost$(Cx, p1) \vee$ suspended_cost$(Cs, p1) \vee$ resuming_cost$(Cr, p1)$	Every process, product, project, or resource could have attribute: dormant cost, executing cost, suspended cost, or resuming cost
	$\forall p1, v1, v2$(process$(p1) \vee$ product$(p1) \vee$ project$(p1) \vee$ resource$(p1)) \rightarrow$ total_cost$(Ct, p1)$	Each, however, must have attribute: total cost
State control	$\forall p1, s1, v1$ process$(p1) \wedge$ process_stage$(s1) \wedge$ has_state$(s1, v1) \wedge$ has_stage$(p1, s1) \wedge \neg(\exists s2, v2$ has_stage $(p1, s2) \wedge$ has_state$(s2, v2) \wedge (v2 =$ executing $\vee\ v2 =$ resuming $\vee\ v2 =$ dormant $\vee\ v2 =$ stopped $\vee\ v2 =$ suspended$)) \rightarrow v1 =$ completed	Process is complete if all its stages are completed
	$\forall p1, v1$ process$(p1) \wedge$ state$(s1) \wedge$ has_state$(p1, v1) \wedge v1 =$ change $\rightarrow \exists sp1, v2$ sub_process$(sp1, p1) \wedge$ has_state$(sp1, v2) \wedge (v2 =$ rework $\vee\ v2 =$ emergency $\vee\ v2 =$ change$)$	If process is in state of change, then at least one of its subprocesses is in state of rework, emergency, or change (order)
Behavioral	$\forall x$ process$(x) \rightarrow \exists y, z$ actor$(y) \wedge$ role$(z) \wedge$ involved(y, x, z)	Process must have at least one actor involved with at least one role
	$\forall Cd, Cx, Cs, Cr, Ct, s1$ stage$(s1) \wedge$ dormant_cost$(Cd, s1) \wedge$ executing_cost$(Cx, s1) \wedge$ suspended_cost$(Cs, s1) \wedge$ resuming_cost$(Cr, s1) \wedge$ total_cost$(Ct, s1) \rightarrow Ct = Cd + Cx + Cs + Cr$	Cost of stage (and, consequently, costs of processes, products, and projects) is sum of costs of all its possible states

J. Constr. Eng. Manage. 2013.139:768-784.

means to control the behavior of concepts and explain how a multitude of them behave or change their behavior in different situations. Within informatics ontologies, an axiom is a formal logical expression used to describe the behaviour of a one or a set of concept(s). Used collectively, a set of axioms can be used to deduct further results. The representation of axioms is by far the most demanding (and, to an extent, controversial) part of ontology development. Compared with concept taxonomies and relationship modeling, there is definitely less agreement among researchers about what axioms are, how they should be modeled, how to manage them, and even how to represent them. "Often axiom specification in ontology modeling environments is restricted to what subsumption offers in a description logics framework . . . or to what the ontology engineer encodes in some kind of first-order logic language . . . , or axiom modeling is neglected at all...This situation is detrimental to the modeling of large-scale ontologies, because it aggravates engineering and maintenance of large sets of axioms" (Staab and Maedche 2000).

The development of axioms is a challenging endeavor even in relatively more objective domains. For example, in a discussion of spatial reasoning (a well-developed domain), Guesgen (1989) mentions that "commonsense reasoning often is of a qualitative nature rather than a quantitative one" and lists three main challenges:

1. Imprecision;
   - Many relationships or assertions are fuzzy. Those that have exact boundaries and use quantitative measures are less frequent and may not provide valuable addition.
2. Uncertainty;
   - Often, one cannot decide what the relationship between two objects is. This is, of course, more so a problem outside such a well-developed domain as spatial representation.
3. Context;
   - The applicability of an axiom depends on the level of granularity of the ontology, its context, and the perspective it assumes. Certainly, an ontology for construction developed by (or for) a lawyer will be different from an ontology developed by engineers.

It is important to remember that an ontology is not a single universal model of the knowledge—just one of many possible models. It does not have to exhaustively incorporate all possible axioms that are relevant in the domain—just a core set. In fact, "sciences do no try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work" (von Neumann 1955). Consequently, when it comes to axioms, minimalism in ontological commitment (not in clarity and coverage) is a virtue (Gruber 1993).

In DOCK, axioms have three main taxonomies (types), which are directly related to the typical functions of axioms:

1. Consistency axioms;
   - These are meant to protect the integrity of (almost any) ontology and assure consistent reasoning, and are independent of the context of future subdomain ontologies. At the core, these encompass the main axioms of the set theory, such as: axiom of extensionality, axiom of empty set, and axiom of union (for more, see Potter 2004). Formalizing these axioms requires extensive and highly philosophical analysis/theorizing about fundamental concepts/terms such as is–a, in, part-of, and composed-of. What does it mean that Concept A is in Location B or Time C, is part of Organization D or Project E, or is composed of a set of other concepts? A variety of philosophical approaches exist in this regard. Formalization of these is the domain of upper ontologies.
2. Domain axioms;
   - These are construction specific and are the main claims proposed by the ontology on the cross-tree behavior. How do construction processes relate to products and projects? How can the role actors in each be described/controlled? What rules exist for change in state? What rules exist for managing major issues such as schedule and cost?
3. Dynamic axioms;
   - These are context-dependent axioms that are only applicable within a specific context.

Table 2 shows a sample of DOCK axioms, their modalities, and role; DOCK attempted to benchmark existing upper ontologies. Being a pragmatic and constructivist ontology, many of the realist and foundationalism-influenced assertions of basic formal ontology (BFO) were not suitable for DOCK. Instead, it is influenced by DOLCE—owing to its pragmatic, objective categorization structure. It is also interested in CYC approach, given its rich semantics and extensive mereological analysis. Linking DOCK to these two upper ontologies (to benefit from their logical structure) is the first task for future research into extending DOCK. Additional research will conduct a comparative analysis/review of direct and indirect domain axioms of major construction product- and process-modeling approaches to distill an additional set of domain-level axioms and support efforts to better formalize dynamic (context-level) axioms to support the generation of future application ontologies in a consistent manner.

Consistency axioms have priority over the other two types of axioms, and domain axioms have priority over dynamic ones. In DOCK, axioms are designed as objects that could have attributes—primarily priority (or precedence) and context of applicability. Axioms were assigned priority to help with future merger exercises. Experience has shown that in many cases, axioms of different ontologies conflict. If axioms are assigned priority (a kind of importance weight), then the most important ones will prevail upon conflict.

Conclusions and Future Work

The DOCK is a constructivist and pragmatic claim for knowledge about construction concepts. Its objectives are to provide a shared (base) platform to develop additional subdomain and application-level ontologies; to facilitate the linking of construction-informatics systems to already existing ontologies in other domains; and, fundamentally, to represent a conceptual model that can spur discussions about how to represent construction knowledge. Human knowledge is multifaceted, and many views of the same concept can coexist. Knowledge is also dynamic, and the representation of the same concept can change from one context to the other. Consequently, in a certain domain of knowledge it is very much expected (and justified) that there will be varying ontologies to describe the same knowledge. Although ontological universality is unachievable, work on developing and fine-tuning ontologies should not be stopped, and the hardest work should be on the area of ontology integration (not unification). More importantly, the validity, relevance, and adequacy of the ontological models of these varying ontologies have to be analyzed to

J. Constr. Eng. Manage. 2013.139:768-784.

discover and better understand human knowledge and how to represent it.

At the core of the DOCK approach is the recognition that the representation of knowledge is contextual. Concepts, relationships, and axioms have possibilities (varying modes of existence), each of which has a probability. Context, modality, and concept were used as the main dimensions in DOCK. Modality generates types of concepts. Contexts weave these concepts into various worlds by changing/modifying (the modality of) their interrelationships and axioms.

Future work on DOCK will focus on three major thrusts: first, linking DOCK to CYC and DOLCE to enhance the quality of consistency axioms and provide for better chances of linking DOCK to ontologies in other domains; second, analyzing the direct and indirect assumptions (axioms) of related product and process models to enrich the construction-specific axioms; and finally, providing formal means to generate contexts through usage of descriptive logic tools.

References

Abdelhamid, T. S. (2004). "The self-destruction and renewal of Lean Construction theory: A prediction from Boyd's theory." *Proc., 12th Conf. of the Int. Group for Lean Construction*, International group of Lean construction (IGLC), Elsinore, Denmark.

Allen, J. F. (1983). "Maintaining knowledge about temporal intervals." *Commun. ACM*, 26(11), 832–843.

Allen, J. F., and George, F. (1997). "Actions and events in interval temporal logic." *Spatial and temporal reasoning*, O. Stock, ed., Kluwer Academic, Dordrecht, Netherlands.

Almeida, M. B. (2009). "A proposal to evaluate ontology content." *Appl. Ontol.*, 4(3–4), 245–265.

Bodenreider, O. (2004). "The unified medical language system (UMLS): Integrating biomedical terminology." *Nucleic Acids Res.*, 32(1), D267–D270.

Eir, A. (2004). "Construction informatics." Ph.D. thesis, Technical University of Denmark, Copenhagen, Denmark.

Ekholm, A., and Fridqvist, S. (1996). "Modelling of user organisations, buildings and spaces for the design process." *Proc., CIB W78 Workshop, Construction on the Information Highway*, Bled, Slovenia.

El-Diraby, T. E. (2006). "A web-services environment for collaborative management of product life cycle costs." *J. Constr. Eng. Manage.*, 132(3), 300–313.

El-Diraby, T. E. (2012). "On the epistemology of construction informatics." *J. Constr. Eng. Manage.*, 138(1), 53–65.

El-Diraby, T. E., and Brecino, F. (2005). "A taxonomy for outside plant construction in telecommunication infrastructure." *J. Infrastruct. Syst.*, 11(2), 110–121.

El-Diraby, T. E., and Gill, S. (2006). "A taxonomy for construction terms in privatized infrastructure finance: Toward a semantic cross-industry exchange of knowledge." *J. Constr. Manage. Econ.*, 24(2), 271–285.

El-Diraby, T. E., and Kashif, K. (2005). "A distributed ontology architecture for knowledge management in highway construction." *J. Constr. Eng. Manage.*, 131(5), 591–603.

El-Diraby, T. E., Lima, C., and Fies, B. (2005). "A domain taxonomy for construction concepts: Toward a formal ontology for construction knowledge." *J. Comput. Civ. Eng.*, 19(4), 394–406.

El-Diraby, T. E., and Wang, B. (2005). "E-society portal: Integrating urban highway construction projects into the knowledge city." *J. Constr. Eng. Manage.*, 131(11), 1196–1211.

El-Diraby, T. E., and Zhang, J. (2006). "A semantic framework for corporate memory management in building construction." *J. Autom. Constr.*, 15(4), 504–521.

El-Gohary, N., and El-Diraby, T. (2010a). "Domain ontology for processes in infrastructure and construction." *J. Constr. Eng. Manage.*, 136(7), 730–744.

El-Gohary, N., and El-Diraby, T. E. (2010b). "A dynamic knowledge-based process integration portal for collaborative construction." *J. Constr. Eng. Manage.*, 136(3), 316–328.

El-Gohary, N., Osman, H., and El-Diraby, T. E. (2006). "Stakeholder management for public private partnership." *Int. J. Proj. Manage.*, 24(7), 595–604.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*, MIT, Cambridge, Massachusetts.

Fox, M. S., and Grüninger, M. (1997). "On ontologies and enterprise modeling." *Proc., Conf. on Enterprise Integration*, Springer, Torino, Italy.

Gehre, A., Katranuschkov, P., Wix, J., and Beetz, J. (2006). "Interoperability of virtual organizations on a complex semantic grid. InteliGrid deliverable D31: Ontology specification." *IST-004664*, InteliGrid Consortium, Electric Power Research Institute, Palo Alto, CA.

Gómez-Pérez, A., Fernández, M., and de Vicente, A. (1996). "Towards a method to conceptualize domain ontologies." *Proc., Workshop on Ontological Engineering (ECAI 96)*, Budapest, Hungary.

Gruber, T. R. (1993). "Toward principles for the design of ontologies used for knowledge sharing." *Formal ontology in conceptual analysis and knowledge representation*, N. Guarino and R. Poli, eds., Kluwer Academic, Amsterdam, Netherlands, 907–928.

Guarino, N. (1995). "Formal ontology, conceptual analysis and knowledge representation." *Int. J. Hum. Comput. Stud.*, 43(5–6), 625–640.

Guesgen, H. W. (1989). *Spatial reasoning based on Allen's temporal logic*, Int. Computer Science Institute, Berkeley, CA.

Hobbs, J. R., et al. (2002). "A DAML ontology of time." ⟨http://www.cs.rochester.edu/~ferguson/daml/daml-time-nov2002.txt⟩ (Nov. 20, 2010).

International Framework for Dictionaries (IFD). (2012). "IFD Library white paper." *IFD Library for BuildingSmart*, ⟨http://www.ifd-library.org⟩ (Jul. 4, 2012).

Jureta, I. J., Mylopoulos, J., and Faulkner, S. (2009). "A core ontology for requirements." *Appl. Ontol.*, 4(3–4), 169–244.

Katanuschkov, P., Gehre, A., and Scherer, R. J. (2002). "An engineering ontology framework as advanced user gateway to IFC model data." *Proc., ECPPM*, Prortoroz, Slovenia.

Koskela, L. (2000). "We need a theory of construction." *Proc., Berkeley-Stanford CE&M Workshop: Defining a Research Agenda for AEC Process/Product Development in 2000 and Beyond*, Stanford, CA.

Lenat, D. B. (1995). "Cyc: A large-scale investment in knowledge infrastructure." *Comm. ACM*, 38(11), 33–38.

Mazzocchi, S. (2010). "Closed world vs. open world: The first semantic web battle." ⟨http://www.betaversion.org/~stefano/linotype/news/91/⟩ (Mar. 20, 2010).

Noy, N. F., and McGuinness, D. (2001). "Ontology development 101: A guide to creating your first ontology." *Stanford Knowledge Systems Laboratory Technical Rep. KSL-01-05, Stanford Medical Informatics Technical Rep. SMI-2001-0880*, Stanford University, Stanford, CA.

Osman, H., and El-Diraby, T. E. (2011a). "A knowledge-enabled system for routing utility infrastructure in urban areas." *J. Constr. Eng. Manage.*, 137(3), 198–213.

Osman, H., and El-Diraby, T. E. (2011b). "An ontology for construction terms in utility products." *Autom. Constr.*, 20(8), 1120–1132.

Racer Systems GmbH & Co. KG. (2005). "RacerPro user's guide, version 1.9." ⟨http://www.racer-systems.com⟩ (Dec. 20, 2010).

Sowa, J. F. (2000). *Ontology*, ⟨http://www.jfsowa.com/ontology/⟩ (Mar. 20, 2005).

Staab, S., and Maedche, A. (2000). "Axioms are objects, too: Ontology engineering beyond the modeling of concepts and relations." *Proc., 14th European Conf. on Artificial Intelligence, ECAI 2000*, Berlin, Germany.

Staub-French, S., Fischer, M., Kunz, J., and Paulson, B. (2003). "An ontology for relating features with activities to calculate costs." *J. Comput. Civ. Eng.*, 17(4), 243–254.

J. Constr. Eng. Manage. 2013.139:768-784.

Steimann, F. (2000). "On the representation of roles in object-oriented and conceptual modelling." *Data Knowl. Eng.*, 35(1), 83–106.

Swindler, J. K. (1991). *Weaving: An analysis of the constitution of objects*, Rowman & Littlefield, Lanham, MD.

Tamma, V., and Bench Capon, T. (2002). "Attribute meta-properties for formal ontological analysis." *Proc., 13th Int. Conf. on Knowledge Engineering and Knowledge Management, EKAW 2002*, Springer, New York, 301–316.

Tham, K. D., Fox, M. S., and Gruninger, M. (1994). "A cost ontology for enterprise modeling." *Proc., 3rd Workshop on Enabling Technologies-Infrastructure for Collaborative Enterprises*, West Virginia University, Morgantown, WV.

von Neumann, J. (1955). "Methods in the physical sciences." *The unity of knowledge*, L. Leary, ed., Doubleday Publishing, New York.

Woestenenk, K., van Rees, R., Lima, C., Stephens, J., and Bonsma, P. (2000). "bcTaxonomy." *Rep. IST-1999-10303-D501*, European Commission, Brussels, Belgium.

Zhang, J., and El-Diraby, T. E. (2012). "Social semantic portal for coordinating construction communication." *J. Comput. Civ. Eng.*, 26(1), 90–104.

Domain ontology for construction knowledge

El-Diraby, Tamer E.

01 Minguk Kim Page 1

1/11/2019 8:11

온톨로지는 담론 영역에 대해 알려진 것을 모델링하려는 지식에 대한 주장입니다. 도메인 온톨로지는 도메인의 모든 개념을 철저하게 나열하는 것이 아니라 도메인의 지식의 본질에 대한 추상적 (아직 확장 가능) 철학적 (아직 실용적) 개념화를 구축하는 것을 목표로합니다. 온톨로지의 핵심에는 온톨로지 모델이 있습니다 (도메인 내에서 세계가 어떻게 행동하는지)에 대한 아키텍처입니다. 온톨로지는 개념, 양식 및 맥락이라는 세 가지 주요 차원에 걸쳐 건축 지식을 분류합니다. 개념은 5 가지 핵심 용어 : 개체 (일반 및 보조로 더 세분화 됨), 환경 요소, 추상 개념, 속성 및 시스템 (이전 4 가지 유형의 조합)을 포함합니다. 양식은 설명 된 각 개념에 대해 다양한 유형을 생성하는 수단입니다. 컨텍스트를 사용하면 개념을 다양한 방식으로 연결하여 다양한 세계를 만들 수 있습니다.

02 Minguk Kim Page 1

1/11/2019 8:11

1/11/2019 8:11

[소개]

정보학 온톨로지는 담론 영역에 대해 알려진 것에 대한 공식적인 설명입니다. 여기에는 개념의 계층 구조 (택 소노 미), 관계 및 동작을 설명하는 공리 (제한)가 포함됩니다. 정보학 온톨로지는 데이터 교환 표준이 아닙니다. 그럼에도 불구하고, 그들은 상호 운용 가능한 컴퓨터 프로그램을 지원하기 위해 데이터 표준과 상당히 잘 상호 작용할 수 있습니다. 온톨로지는 추론보다는 지식 표현과 관련이 있지만 의사 결정 지원 기능을 향상시키기 위해 인공 지능 (AI) 도구로 보완 될 수 있습니다. 상위 온톨로지는 가장 추상적 인 형태의 온톨로지이며, 일반적으로 철학적 수준에서 보편적 인 도메인 간 개념화를 포착하는 것을 목표로한다. 도메인 온톨로지는 철학적이며 보편적 인 범주화에 관심이 있지만 특정 담론 영역 내에 있습니다. 응용 온톨로지는 모든 유형 중에서 가장 철학적이며 컴퓨터 소프트웨어 시스템에서 사용하기 위해 도메인 온톨로지를 인스턴스화하여 일반적으로 용어의 의미 론적 표현을 풍부하게합니다.

개념을 나열하고 연결하고 그들의 행동을 설명하는 것이 중요하지만 온톨로지는 근본적으로 지식에 대한 주장입니다. 정보학 온톨로지의 핵심에는 온톨로지 모델이있다. 개념이 어떻게 구성되어 있는지에 대한 철학적 견해이다. 이러한 모델의 주요 질문은 (개념적) 보편성에 관한 것입니다. 그들은 무엇인가? 그들은 서로 어떻게 관련이 있습니까? 그들은 변화 / 변형합니까? 그렇다면 어떤 조건 하에서? 온톨로지를 구축하는 것의 장점은 두 가지라고 주장 할 수있다 : 기계 해석이 가능한 방식으로 인간의 지식을 공식적으로 표현하고 공유하는 것 (정보학 기여); 그것은 존재 론적 모델, 즉 지식 (도메인에있는 것)과 그것을 제시하는 방법 (철학적 기여)에 대해 사용자들 간의 대화를 촉진하기 위해 사용됩니다. 철학 영역에서 온톨로지 구축의 풍부한 역사는 두 번째 장점이 더 유익하고 오래 지속될 수 있음을 보여주었습니다.

이 논문은 구성 지식 (DOCK 1.0)을위한 도메인 온톨로지에 대한 반복을 제시한다. 목표는 개념을 철저히 나열하는 것이 아니라 구성, 관계 및 동작에서 핵심 용어의 개념적 아키텍처를 구축하는 것입니다. 이 온톨로지는 구성 론자와 실용주의 인식론을 혼합하여 개발되었습니다. 구성주의는 절대 온톨로지의 존재를 거부하고 (현실은 사회적으로 구성됨) 상향식 모델링 접근법을 옹호합니다. 실용주의는 실용성과 그러한 지식을 통해 얻을 수있는 이점을 기반으로 모델링 지식에 중점을 둡니다 (자세한 내용은 El-Diraby 2012 참조). 이를 바탕으로 완벽한 모델의 불가능 성을 인식하는 DOCK의 목표는 건축 지식과 관련된 주요 개념을 설명하기에 충분한 골격을 찾는 것입니다. 제안 된 범주와 그 구조가 그러한 지식의 지각에 대한 적절한 관점에 기초하여 건축 영역에서 지식에 대한 가능한 개념적 이해 / 대부분을 대표 할 수 있는가?

많은 지식에 대한 주장과 마찬가지로, 제안 된 모델의 일관성 (특히 DOCK의 구성 주의적 실용적 기초가 주어진)에 대해 많은 환영을받는 주장이 반드시 있어야한다. 일관성은 중요하지만 ":: : 온톨로지는 개념적 계층 구조라는 것을 기억해야합니다. 가능할지라도 반드시 공리 시스템 일 필요는 없습니다. 일관성을 테스트하기 위해 온톨로지를 공리적으로 공식화하는 것이 유행이다. :: : 일관성은 온톨로지의 무제한 공급이 있기 때문에 수용성 테스트가 너무 약합니다. Quine는 오래 전에 충분한 독창성과 유연성으로 모든 아이디어를 공식화 할 수 있다고 제안했습니다. 형식화는 일관된 여부에 상관없이 모든 믿음의 사소한 속성입니다. :: :"(Swindler 1991).

시공 지식 모델링과 관련된 몇 가지 노력에 대한 검토가 먼저 제시됩니다. 그 다음에는 DOCK의 범위와 한계에 대한 설명과 그 개발 방법론이 설명됩니다. DOCK의 개념을 제시하기 전에, 존재 론적 모델의 요약이 정의되어 문맥에 대한 정의를 제시합니다. 이 요약 다음에 DOCK의 주요 개념, 관계 및 공리가 제시됩니다.

04 Minguk Kim Page 1

1/11/2019 8:11

온톨로지는 추론보다는 지식 표현과 관련이 있지만 의사 결정 지원 기능을 향상시키기 위해 인공 지능 (AI) 도구로 보완 될 수 있습니다.

05 Minguk Kim Page 1

1/11/2019 8:11

06 Minguk Kim Page 1

1/11/2019 8:11

응용 온톨로지는 모든 유형 중에서 가장 철학적이며 컴퓨터 소프트웨어 시스템에서 사용하기 위해 도메인 온톨로지를 인스턴스화하여 일반적으로 용어의 의미 론적 표현을 풍부하게합니다.

07 Minguk Kim Page 1

1/11/2019 8:11

08 Minguk Kim Page 1

1/11/2019 8:11

온톨로지를 구축하는 것의 장점은 두 가지라고 주장 할 수있다 : 기계 해석이 가능한 방식으로 인간의 지식을 공식적으로 표현하고 공유하는 것 (정보학 기여); 그것은 존재 론적 모델, 즉 지식 (도메인에있는 것)과 그것을 제시하는 방법 (철학적 기여)에 대해 사용자들 간의 대화를 촉진하기 위해 사용됩니다. 철학 영역에서 온톨로지 구축의 풍부한 역사는 두 번째 장점이 더 유익하고 오래 지속될 수 있음을 보여주었습니다.

09 Minguk Kim Page 1

1/11/2019 8:11

10 Minguk Kim Page 2

1/11/2019 8:11

건축 지식 모델링

건축 지식의 모델은 분류 시스템 및 시사, 제품 및 프로세스 모델, 온톨로지의 세 가지 범주로 구성됩니다. 첫 번째 카테고리는 가장 유명하고 오래된 카테고리입니다. 지금까지 이러한 분류 시스템 (스웨덴 건축 용어 분류 : sfb, Uniclass 및 MasterFormat)은 온톨로지 모델링에주의를 기울이지 않고 제품 분류에 중점을 두었습니다. IFC (Industry Foundation Classes)와 같은 제품 모델은 의미 론적 지식과 달리 상호 운용 가능한 제품 데이터 교환을 보장하기 위해 온톨로지 기능이 제한적이었습니다. 그러나 GenCOM (General Construction Object Model)과 같은 일부 모델은 시맨틱 기울기를 사용하여 건설 제품 및 프로세스의 주요 용어를 모델링하는 것을 목표로합니다. 국제 사전 체계 (IFD)는 IFC 및 건물 정보 모델링 (BIM)과 밀접한 관련이 있습니다. 다국어 사전 또는 온톨로지를 만드는 것을 목표로하여 건축 용어의 동의어 사전에 더 가깝습니다. 이는 건축 및 건설 산업 (IFD 2012)에서 향상된 상호 운용성을 지원하기위한 참조 라이브러리를 의미합니다.

최근에는 건축 / 엔지니어링 / 건설 분야에서 온톨로지 관련 작업이 등장했습니다. 예를 들어 분류법 (Woestenenk et al. 2000) 또는 하위 도메인 수준 온톨로지 개발 (Katanuschkov et al. 2002; Staub-French et al. 2003; Gehre et al. 2006). 거의 예외없이 Ekholm (Ekholm 1996 참조)과 Eir (Eir 2004 참조)의 대부분의 건축 온톨로지는 개념화 (진짜 언어 적 또는 철학적 방식)보다는 공식화에 초점을 맞추고 하위 영역에 집중되었다 (또는 응용 프로그램) 수준. 흥미롭게도 가장 관련성이 높은 철학적 공헌은 희박한 건설 및 조직 관리에서 찾을 수 있습니다 (Koskola 2000; Abdelhamid 2004 참조).

11 Minguk Kim Page 2

1/11/2019 8:11

지금까지 이러한 분류 시스템 (스웨덴 건축 용어 분류 : sfb, Uniclass 및 MasterFormat)은 온톨로지 모델링에주의를 기울이지 않고 제품 분류에 중점을 두었습니다.

12 Minguk Kim Page 2

1/11/2019 8:11

13 Minguk Kim Page 2

1/11/2019 8:11

14 Minguk Kim Page 2

1/11/2019 8:11

IFD는 다국어 사전 또는 온톨로지를 만드는 것을 목표로하여 건축 용어의 동의어 사전에 더 가깝습니다. 이는 건축 및 건설 산업에서 향상된 상호 운용성을 지원하기위한 참조 라이브러리를 의미합니다.

15 Minguk Kim Page 2

1/11/2019 8:11

16 Minguk Kim Page 2

1/11/2019 8:11

거의 예외없이 Ekholm (Ekholm 1996 참조)과 Eir (Eir 2004 참조)의 대부분의 건축 온톨로지는 개념화 (진짜 언어 적 또는 철학적 방식)보다는 공식화에 초점을 맞추고 하위 영역에 집중되었다 (또는 응용 프로그램) 수준. 흥미롭게도 가장 관련성이 높은 철학적 공헌은 희박한 건설 및 조직 관리에서 찾을 수 있습니다 (Koskola 2000; Abdelhamid 2004 참조).

-> 대부분의 건축온톨로지는 하위영역에 집중되며, 가장 관련성 높은 공헌은 lean construction에서 실행되었다.

17 Minguk Kim Page 2

1/11/2019 8:11

18 Minguk Kim Page 2

1/11/2019 8:11

1/11/2019 8:11

[방법론]

온톨로지는 많은 지식 영역 (예 : 철학,인지 과학, 언어 및 정보 검색)에서 개발되었습니다. 이러한 각 도메인의 모델링 스타일은 크게 다릅니다. 대부분의 정보학 온톨로지는 인공 지능에 의해 유도되는 (또는 기반이되는) 직관적이고 직관적 인 접근법을 따르는 것으로 보인다. 예를 들어, WordNet (언어 및 정보 검색 스키마)은인지 과학이 구부러진 순진한 의미론의 원칙에 기초하여 구축됩니다 (Fellbaum 1998). 같은 줄을 따르는 추가 주요 온톨로지에는 CYC (백과 사전에서 발췌) 온톨로지 (Lenat 1995)와 토론토 가상 엔터프라이즈 온톨로지 (TOVE)가 있으며 Fox와 Gruninger (1997)가 개발했습니다.

TOVE 접근법은 순수한 AI 사고 방식을 따릅니다. 온톨로지는 공식적인 모델입니다. 연구원은 일련의 역량 질문 (모델 사양)을 설정하고 모델을 개발하고 (전술 법 미적분학 및 1 차 또는 설명 논리를 광범위하게 사용) 온톨로지가 역량 질문을 어떻게 만족시키는 지 테스트해야합니다. 1990 년대 온톨로지의 주요 기능은 자동화 된 프로세스 또는 가상 (온라인) 트랜잭션 (Guarino 1995). 일부 연구자들은 응용 프로그램의 일부로 온톨로지의 유용성을 테스트하는 것과 관련된 기능을 추가했습니다. Gomez-Perez et al. (1996)은 온톨로지의 수명주기 관리의 일환으로 최종 사용자와 포커스 그룹의 의견을 사용하여 주장했다. 이러한 접근 방식과 다른 접근 방식을 잘 비교하려면 Almeida (2009)를 참조하십시오. 최근에는 소셜 웹의 발전과 함께 언어학의 언어 및 의사 소통 측면이 공식적 사유와는 반대로 강조되었다. 이는 검색 엔진 (예 : Google)에서 온톨로지를 채택하고 생의학 분야에서 완전한 도메인 별 언어를 만들려는 시도에서 볼 수 있습니다 (Bodenreider 2004). 이것은 언어 적으로 풍부한 철학적 / 추상적 온톨로지의 발달로 이어졌다. 이러한 움직임의 예는 표준 상부 온톨로지 (SUO), 언어 및인지 엔지니어링 (DOLCE)을위한 기술 온톨로지 및 CYC (Jureta et al. 2009)의 세 가지 주요 온톨로지에서 볼 수 있습니다.

DOCK은이 세 가지 상위 온톨로지를 분석하고 벤치마킹하려고 시도했습니다. 상위 온톨로지는 구체적으로 타겟팅되지 않았지만 개발 후 분석은 DOLCE 구조와 CYC 용어 및 공리의 영향에 더 밀접한 관련이 있음을 보여줍니다. IFC와 기존 분류 시스템은 DOCK에 더 많은 영향을 미쳤다 (특히 사용 된 용어와 관련하여). TOVE 역시 주요한 영향과 영감 (특히 개념 구조와 공리)에 영향을 미쳤습니다. 많은 측면에서 건축과 제조 사이에는 명확한 유사점이 있으며, 이는 많은 TOVE의 구조와 주장이 건축과 직관적으로 관련되어 있습니다. 그러나 추론에 대한 개념화와 관련하여 DOCK은 TOVE보다 형식이 훨씬 적습니다.

1/11/2019 8:11
[범위와 한계]

이름과 모델링의 목표 (즉, 존재) 때문에 온톨로지는 그 범위에서 포괄적이고 그 용어에 포함되어야한다는 다소 잘못된 기대가있다.
또한 머신 러닝에 대한 온톨로지 도입은 이와 유사한 잘못된 기대를 추가했다. 추론을위한 정확한 장소를 제공하기위한 보편적 공리가 존재할 수 있다는 것이다. 진실에서 더 이상 갈 수있는 것은 없습니다.
온톨로지는 지식에 대한 주장입니다. 질문에 대한 답변만큼 새로운 질문을 시작합니다. 결과적으로,이 논문에서 제시된 영역 온톨로지는 구성 지식의 보편적 표현이 아니라 가능한 많은 일반적인 표현 중 하나를 생성하려는 시도를 의미한다.

이를 위해, 가치 (및 온톨로지의 기본 기여)는 온톨로지 모델의 일관성, 일반적인 (일반적이지 않은) 상황을 지원하기위한 적합성 / 적응성 및 확장성에 있습니다. 이를 위해 존재 론적 약속을 최소화하는 것은 (특히 공리 측면에서) 근본적인 교훈이다 (Gruber 1993; Guarino 1995; Noy and McGuinness 2001). 물론 이것은 명확한 범위를 채택하고 온톨로지의 깊이 나 엄격 성을 최소화하지 않고 충돌을 줄이는 약한 도메인 이론을 사용하는 것을 말합니다. 다음 하위 섹션에 제시된 문제는 DOCK의 범위를 형성했습니다.

1/11/2019 8:11
[인식론]

DOCK은 현대의 실용주의와 구성주의 인식론 사이의 균형에 기초하여 만들어졌습니다. 모든 개념에는 가능성 (다양한 모양 또는 존재 모드)과 가능성 (특정 맥락에서 다양한 모양 또는 존재 모드)이 있습니다. 개념과 모델은 주로 도메인 전문가의 사회적 집단 이해에 대한 상향식 모델링을 기반으로 (또는 적어도 초기에) 구축해야합니다. 개념화는 기계적 프로세스가 아닙니다 (위에서 아래로 결정적). 사용자 간의 합의를 통해 구축해야합니다. 이러한 합의는 제한된 사용자 그룹에 의해 개발된다는 점에서 상대적입니다. 궁극적으로 개념화는 첫 번째 시도에서 충분한 수준에 도달 할 수 없습니다. 동의 그룹의 이해 관계자에 대한 대표 성과 재 균형에 대한 적절한 테스트를 위해서는 반복적 인 개발을 거쳐야한다. 이것은 하향식 접근법이 온톨로지 개발에 아무런 영향을 미치지 않는다는 것은 아닙니다. 항상 그렇듯이 균형에 도달해야합니다 (자세한 내용은 El-Diraby 2012 참조). 보편성과 완전성 제안 된 온톨로지는 구성 용어에 대한 보편적 인 설명이 아니다. 현실적으로 분류 딜레마에 대한 충분한 해결책이 수용되어야한다. 이것은 아리스토텔레스가 그의 원래 분류법 (속에 초점을 맞춘)이 완전하지 않다는 것을 인정한 후, 우연한 속성 (미분이라고 함)이 추가되어야 함을 시사합니다. 개념화의 지속적인 문제에 중점을 둔 최초의 건설 중 하나 인이 온톨로지는 건축 개념에 대한 초기의 충분한 온톨로지 모델을 제시하는 것을 목표로한다. DOCK은 개방형 온톨로지입니다. 닫힌 세계 모형에서 모형에 포함되지 않은 모든 것은 거짓으로 간주됩니다 (즉, 모형은 가능한 모든 주장을 포함합니다). 반면, 개방형 모델에서는 모델에 포함되지 않은 모든 항목을 알 수 없거나 정의되어 있지 않습니다. 모델에 나열되지 않은 항목에 대한 가정은 없습니다 (Mazzocchi 2010).

보편성과 완전성 제안 된 온톨로지는 구성 용어에 대한 보편적 인 설명이 아니다. 현실적으로 분류 딜레마에 대한 충분한 해결책이 수용되어야한다. 이것은 아리스토텔레스가 그의 원래 분류법 (속에 초점을 맞춘)이 완전하지 않다는 것을 인정한 후, 우연한 속성 (미분이라고 함)이 추가되어야 함을 시사합니다. 개념화의 지속적인 문제에 중점을 둔 최초의 건설 중 하나 인이 온톨로지는 건축 개념에 대한 초기의 충분한 온톨로지 모델을 제시하는 것을 목표로한다. DOCK은 개방형 온톨로지입니다. 닫힌 세계 모형에서 모형에 포함되지 않은 모든 것은 거짓으로 간주됩니다 (즉, 모형은 가능한 모든 주장을 포함합니다). 반면, 개방형 모델에서는 모델에 포함되지 않은 모든 항목을 알 수 없거나 정의되어 있지 않습니다. 모델에 나열되지 않은 항목에 대한 가정은 없습니다 (Mazzocchi 2010).

1/11/2019 8:11

정보학 온톨로지의 검증은 어려운 작업입니다.
실제로 저자의 관점에서 온톨로지와 함께 유효성 검사라는 단어를 사용하는 것은 적합하지 않을 수 있습니다. 대신 평가가 더 적절합니다. 이에 대한 이론적 근거는 다음과 같은 이유 / 인수에 근거합니다.

• 도메인 온톨로지의 철학적 특성 : 응용 온톨로지와 달리 도메인 온톨로지는 도메인 지식의 기초에 대한 높은 수준의 주장입니다. 온톨로지는 영역 지식에 대한 이해의 매우 철학적 측면을 제시 (및 도전)함으로써 답변만큼 많은 질문 (및 반대)을 만듭니다. 실제로 온톨로지를 개발하면 새로운 질문에 답하기 위해 약간의 수정이 필요합니다. 따라서 응용 온톨로지의 개발은 두 가지 주요 단계 사이에서 순환한다 : 이론화와 실천에 대한 이론화. 안정적이고 적절한 온톨로지에 도달하기 전에 반복적 인 방식으로 개발 단계와 기본 가정을 자주 재검토하고 교정해야합니다.

• 추상 모델의 상대성 : 온톨로지는 개념, 관계 및 공리의 분류법의 세 가지 가시적 요소로 구성됩니다. 이것 뒤에는 온톨로지의 가장 중요한 요소 인 인식 론적 가정과 존재 론적 모델이있다. 인식 론적 모델 / 가정은 지식 발견 모드 (즉, 우리가 알고있는 것을 어떻게 아는가)를 정의합니다. 온톨로지 모델은 주요 개념 범주의 기본 레이아웃 (와이어 프레임)입니다. 두 모델의 추상적 인 특성은 본질적으로 상대적이며, 그들에 대한 보편적 인 합의를 찾기가 어렵습니다. 이 두 가지 요소에 대한 토론은 지난 2 천년 동안 철학자들 사이에서 끊임없는 논쟁 (그리고 쓰라린 불일치)의 주제였습니다.

• DOCK의 반 형식적 특성 : DOCK는 강성이 감소 된 개방형 온톨로지로 설계되었습니다. DOCK는 웹 온톨로지 언어 (OWL)로 프로그래밍되었지만 상황 미적분학 또는 1 차 논리로 완전히 표현되지는 않습니다. 이것은 건설 지식의 주관적 성격과 DOCK의 텍스트 / 의미 적 명료성을 향상시키려는 욕구 때문에 주어진 사례입니다. 이론 건물의 품질 관리에서 가장 중요한 단계 중 하나는 작업 품질을 테스트하기위한 도구를 삼각 측량하는 연구 방법론을 설계하는 것입니다. DOCK는 각 개발 단계 후에 중간 평가를 수행하는 반복적 인 방식으로 개발되었으며, 다음 섹션에 설명 된대로 다양한 도구를 사용하여 이러한 평가를 수행했습니다.

1/11/2019 8:11
결론 및 향후 작업

DOCK은 건축 개념에 대한 지식에 대한 구성 주의자 및 실용적 주장입니다. 그 목적은 추가적인 하위 도메인 및 응용 프로그램 수준 온톨로지를 개발할 수있는 공유 (기본) 플랫폼을 제공하는 것입니다. 건설 정보 시스템을 다른 영역에 이미 존재하는 온톨로지와 연결하는 것을 용이하게한다. 그리고 기본적으로 건축 지식을 표현하는 방법에 대한 논의를 촉진 할 수있는 개념적 모델을 대표합니다. 인간의 지식은 다면적이며 동일한 개념에 대한 많은 견해가 공존 할 수 있습니다. 지식은 역동적이며 동일한 개념의 표현은 상황에 따라 다를 수 있습니다. 결과적으로, 특정 지식 영역에서 동일한 지식을 기술하기 위해 다양한 온톨로지가있을 것으로 기대된다 (그리고 정당화 됨). 온톨로지 보편성을 달성 할 수는 없지만 온톨로지를 개발하고 미세 조정하는 작업을 중단해서는 안되며 온톨로지 통합 영역 (통합이 아닌)에서 가장 어려운 작업을 수행해야합니다. 더 중요한 것은 이러한 다양한 온톨로지의 존재 론적 모델의 타당성, 관련성 및 적절성을 분석하여 인간의 지식과 그것을 표현하는 방법을 더 잘 이해하고 이해해야합니다.

DOCK 접근법의 핵심은 지식 표현이 맥락 적이라는 인식입니다. 개념, 관계 및 공리에는 가능성이 있습니다 (다양한 존재 모드). 문맥, 양식 및 개념은 DOCK의 주요 차원으로 사용되었습니다. 양식은 개념 유형을 생성합니다. 문맥은 상호 관계와 공리를 변경 / 수정 (양식)하여 다양한 세계에 이러한 개념을 짜 넣습니다.

DOCK에 대한 향후 연구는 세 가지 주요 추진력에 초점을 맞출 것이다. 첫째, DOCK를 CYC 및 DOLCE에 연결하여 일관성 공리의 품질을 향상시키고 DOCK를 다른 영역의 온톨로지에 연결할 수있는 더 나은 기회를 제공합니다. 둘째, 건축 관련 공리를 풍부하게하기 위해 관련 제품 및 프로세스 모델의 직접 및 간접 가정 (축)을 분석합니다. 마지막으로 서술적인 논리 도구를 사용하여 컨텍스트를 생성하는 공식적인 수단을 제공합니다.