A Context Ontology Development Process for Construction Safety

Han-Hsiang Wang
University of Illinois at Urbana-Champaign, USA. (e-mail: hwang49@uiuc.edu)
Frank Boukamp
University of Illinois at Urbana-Champaign, USA. (e-mail: boukamp@uiuc.edu)

Abstract

Raising project participants’ awareness of safety requirements through the automated identification of applicable safety specifications is one of the approaches to create a safe construction site. The authors have previously proposed a context ontology-based framework to support the automated process of identifying safety requirements. In this paper, the authors focus on the context ontology development work, and propose a systematic process to help develop and represent a context ontology by applying two common tools. The advantages and disadvantages of current and proposed ontology development approaches are discussed, as well as the implications of the proposed approach. The proposed process is believed to benefit the knowledge development in the building and construction industry, and as a result, is also expected to be utilized in future systems to benefit the automated identification of and reasoning about context-related information and knowledge, such as requirements imposed by specifications.

Keyword: Context Ontology, Construction Safety, Knowledge Management, Specifications, Construction Industry

1. Introduction

Construction safety is one of the most important issues in the construction industry. Diverse qualitative or quantitative approaches have been proposed in order to improve safety on site for construction activities. One of the approaches is to focus on raising project participants’ awareness of safety requirements through the automated identification of applicable safety specifications [1]. Safety requirements regulate preempted decisions and actions that direct which requirements an employee should comply with. The purpose of such direct guidance is to educate employees and to enable them to avoid improper behavior which may lead to safety hazards. Therefore, raising the awareness of safety requirements could prevent potential accidents resulting from lack of understanding or ignorance of these requirements and is believed to improve construction site safety [2, 3, 4].

To create a tool that can achieve the goal of raising the awareness of safety requirements in given construction contexts, we have previously proposed a system framework [1]. It integrates information from building information models (BIM), construction method models (CMM) and
a construction context ontology. Based on this information, a system will be able to identify safety requirements imposed by specifications that are applicable to given contexts [1].

Contexts are defined as specific project conditions on site, such as components built, activities performed and resources used. A context ontology, hence, is an ontology that models domain context knowledge in order to reuse that knowledge for context reasoning. To achieve automated identification of context-specific safety requirements, a context ontology for the construction safety domain is required; and in this case, words used in specifications to describe specific applicability conditions, such as “precast concrete” or “masonry wall,” become the sources of contexts. In other words, the process we followed to develop a context ontology that represents context knowledge for safety requirements is actually a process of identifying and formalizing all contexts from the specifications. However, implementing the process manually cumbersome. According to our experience, the reasons are twofold: the number of contexts in safety specifications is huge and there are no appropriate classifications or topologies that construction safety-related contexts could build upon.

Therefore, the experience that we gained in trying to develop a context ontology for construction safety specifications helped us identify the need for an efficient context ontology development approach, which should take into account all the issue of the huge number of specifications and the issue of identifying and applying appropriate classifications of contexts and configurations for relationships between contexts. The proposed approach is discussed in this paper. While the approach was derived from experience gained specifically in the domain of construction safety specifications, general lessons learned can be derived and applied to specification-based context ontology development for other domains.

In the following sections, the current practice of ontology development will be discussed first. Then, the proposed process for how to efficiently develop a context ontology from safety specifications is illustrated. Last, we conclude with a discussion on how the proposed process changes can benefit the knowledge development in the building and construction industry.

2. Current Practice of Ontology Development and Ontology Application

2.1 Current Ontology Application in the Construction Industry

The term “ontology” has been used by the Artificial Intelligence community since the 1980’s and relevant research efforts in Computer Science have started to flourish since the early 1990’s [5]. In the construction domain, the interest in ontologies arose even later. One of the major applications in construction is e-COGNOS, a knowledge representation and management framework [6, 7]. Other application domains include cost estimation [8], design assistance [9, 10] and business process modeling [11].

While context ontologies have been widely applied in the Computer Science domain [12, 13, 14], there were no applications developing context ontologies for construction. However, the
2.2 Current Practice of Ontology Development

An ontology models a set of concepts within a knowledge domain and relationships between these concepts. In general, the knowledge can be extracted from many different sources, such as documents and observations; nevertheless, the ontology development still strongly relies on intensive involvement of domain experts [6, 15]. In other words, the process to develop an ontology can be deemed as a process to explore and exploit domain experts’ valuable knowledge and to structure it in a formalized way. Domain experts have abundant knowledge about the domain concepts and their relationships, have experience in diagnosing and preventing potential problems, and have capabilities to analyze and address existing ones in the domain. Their experience and knowledge have been accumulated and verified practically and/or academically over time. Therefore, the major advantage of developing an ontology by acquiring experts’ knowledge evidently is that the ontology developed based on such experience and knowledge is arguably the most representative.

However, this approach also has a disadvantage: in case that no expert for a certain domain is available to help develop an ontology throughout, the ontology development process will become difficult. Acquiring experts’ knowledge for developing an ontology is a cyclic process: every set of concepts and relationships in an ontology needs to be checked for its suitability for the purpose of the ontology. Depending on the result of this check, the set needs to be updated by experts. This process has to be repeated until no further update is needed. If experts can’t join in developing ontologies from the beginning, the whole process will require more time.

Furthermore, if the ontology to be developed is based on specifications, the absence of experts who are familiar with the documents will make it hard to fulfill the development process. Creating an ontology that represents knowledge from specifications isn’t as easy as directly acquiring knowledge from experts because it is cumbersome to organize the knowledge in the form of concepts and relationships. In the case of developing a context ontology based on construction safety specifications, experts are those who wrote or used the specifications frequently; but it is often impractical to find such experts to participate in the entire ontology development process due to its time consuming and tedious nature. Based on our experience, we find that both the large number of regulations and the vocabulary in specifications make it tedious to extract knowledge in the form of concepts and relationships.

3. Proposed Process for Developing a Context Ontology

To develop a context ontology from a construction safety specification, the first step of the proposed process is to utilize a text analysis tool, such as Text2Onto, to quickly identify an
initial set of ontological concepts and relationships. In the second step, an ontology editing tool, such as Protégé, is then adopted to represent and edit the identified concepts and relationships. The herein described process gives specific guidance on how these tools should be used.

Text2Onto is “a framework for ontology learning and ontology evolution” [16], which was developed by the Institute of Applied Informatics and Formal Description Methods (AIFB), University of Karlsruhe, Germany. Protégé, on the other hand, is an open-source platform developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine [17]. Protégé provides many tools to develop ontology models and knowledge-based applications and has become a popular ontology editing tool for different research domains. In our research we used and evaluated Text2Onto [version 130607] and Protégé [version 3.3 beta] for developing context ontologies for safety specifications. While both Text2Onto and Protégé are equipped with many functions, only some of their functionality is required in the proposed process.

To show how Text2Onto was used to identify ontology concepts and relationships in our proposed framework, we focus on a specific section of a construction safety specification, Subpart Q: Construction and Masonry Construction of Title 29, Part 1926 of the Code of Federal Regulations (29 CFR 1926) [18].

3.1 Concept and Relationship Identification for a Context Ontology

Our approach consisted of three straightforward steps of using Text2Onto to extract concept and relationship terms from a specification:

**Step 1: read in an electronic file to be analyzed.** The file can be in txt, pdf, html or htm format. When the text file containing the safety specification is read into Text2Onto, the file can be seen in the rectangle 1 in Figure 1.

**Step 2: determine modeling primitives needed and algorithms for selected primitives.** In Text2Onto, users can select modeling primitives that Text2Onto should try to identify from the given text, such as “Concept” representing classes and “Subclass-Of” representing class hierarchy relationships [16]. For each primitive, there are different algorithms users can select from to analyze the text for the given primitive, such as a Term Frequency Inverted Document Frequency (TFIDF) algorithm or a Relative Term Frequency (RTF) algorithm for Class primitives [16]. Since our goal is to identify context concepts and relationships between them, the only modeling primitives needed are “Concept”, “SubclassOf” and “Relation.” For the primitive “Concept”, the “TFIDFConceptExtraction” algorithm was selected because it seems best suited for extracting the most significant concepts regarding a given domain [19]. For the primitive “SubclassOf”, the “VerticalRelationsConceptClassification” algorithm was chosen, since we found it to be most suitable for identifying potentially relevant concept hierarchies. For the primitive “Relation”, we used the “SubcatRelationExtraction” algorithm, which is the only algorithm available in Text2Onto in this primitive category. The selection of primitives and algorithms is shown in the rectangle 2 in Figure 1.
Step 3: start the knowledge acquisition process. Once the primitives and respective algorithms have been determined, users click the “Run” button (see rectangle 3) to start the knowledge acquisition process. The extracted and analyzed results appear on the right of the Text2Onto window (Figure 1). In the illustrative example, the extracted results of the three primitives “Concept”, “SubclassOf” and “Relation” that we selected will be listed in separate panes with each pane carrying the name of the respective primitive. In the bottom right of the window, the collective analysis results are shown.

Although Text2Onto can identify concepts and relationships from a textual document and can also help automatically combining them into an ontology, the resulting ontology was inappropriate. It classified all identified concepts as instances, rather than classes, which contradicted our presumption that the algorithm for the primitive “Concept” should help identify “classes”. As a result, the concept did not have hierarchical relationships, since these can only be established between classes, not between instances. In addition, the usefulness of and rationality behind identified concepts and relationships also needs to be checked. The identified concepts and relationships are analyzed for their rationality and usefulness by comparing each term to that appearing in the specification and by determining if the identified term fully represents a context relevant for the specification. Our findings were:

1) Identification of concepts and concept hierarchies: Text2Onto can satisfactorily identify most context concepts from the specifications and most of the identified subclass-of relationships are also reasonable. However, there are certain limitations. First, an identified concept may be an incorrect combination of two terms and hence be classified incorrectly. For instance, “percent overload” is identified and claimed as a subclass of “overload,” but the original sentence regulates that “…provided with pipe supports designed for 100 percent
overload.” in which the meaningful phrase should be “100 percent” while “percent overload” is an irrational combined term. Moreover, Text2Onto could not identify synonymous concepts in some cases. For instance, the concept “building column” is claimed to be a subclass of the concept “column” but they stand for the same thing in the specification. Last, identified concepts could not reflect the real meaning of a concept that has multiple constituents. For example, the concept “water mixture” is claimed as a subclass of the concept “mixture” but the original regulation specifies “a cement, sand, and water mixture.”

(2) Identification of semantic relationships: For our example case, Text2Onto only extracted four relationships, three of which were relevant. For instance, a relationship “meet” is identified to link concepts “equipment” and “requirement,” and another relationship “include” is identified to link concepts “equipment” and “load bearing component.” However, Text2Onto failed to provide more relationships for these concepts and only four are quite insufficient to represent the relations between all 277 identified concepts in the example.

Since relationships can’t be thoroughly generated by Text2Onto, developers have to manually determine relationships between identified concepts and give these relationships semantically meaningful names to facilitate understanding. For example, a relationship “isPouredInto” is a relevant relationship between the context concepts “concrete” and “form.” However, overall it can be said that Text2Onto can facilitate the process of extracting ontology concepts and extracting concept hierarchies from a construction safety specification. The resulting preliminary ontology can serve as a basis for further work by ontology developers and domain experts and thus shorten the time and efforts required to identify concepts.

3.2 Concept and Relationship Representation

Although concepts and concept topologies can be identified through the processes described in the previous sections, the resulting ontology requires additional manual editing. Better classification is required to structure the context ontology in order to facilitate the process of navigating concepts in the ontology. To develop a context ontology, we proposed a classification structure which classifies all contextual concepts into three primary concepts: component, activity, and resource (CAR) because CAR can sufficiently represent required context information on sites and classify all concepts in the context ontology [1]. Also, since safety requirements in the specifications are related to components, activities and resources [1], CAR is believed as a suitable classification for processing and modeling contexts in safety specifications. If the ontology developers or users require a more detailed or a different context classification, they can add additional primary concepts to the classification. For example, a class “Documentation” should be supplemented if users want to represent and classify document related concepts such as “Design Drawing” or “Bill of Quantity.”

Moreover, in order to systematically structure all relationships between the identified concepts, we also need a classification for those relationships. In our research, a classification
using three generic relationships, i.e. association, aggregation and composition [7], is adopted for ontology developers to categorize relationships. In ontologies, the relationships between concepts are defined as properties of those concepts; therefore, the classification for the relationships is expressed in the “property” hierarchy, not the “class” hierarchy which organizes the concepts and their super-class/sub-class relationships.

To establish these classifications within the ontology and to manually add and alter concepts, concept hierarchies and other relationships, an ontology editor tool should be used. In our research, we chose Protégé for this purpose. The following lists the steps that have to be performed within an ontology editor to adjust the context ontology according to our needs:

**Step 1: create classes for the primary concepts of classification.** Protégé uses class as the basic unit to stand for the contextual concepts. Therefore, the process of representing contextual concepts can’t be started until the primary concepts have been created as classes in Protégé. These primary concepts are used to properly classify each concept and to develop its class hierarchy that latter significantly affects the ontology reasoning. In the illustrative example, we use CAR triple to fully represent all contextual concepts extracted from the sample safety specification. All these three primary concepts are classified as subclasses of a class context that is the root concept in the context ontology (see rectangles 1 & 2 in Figure 2).

**Step 2: classify concepts into primary classes.** Once primary classes have been created, the other concept classes extracted by Text2Onto can then be assigned to the appropriate primary classes. At this point, ontology developers should also check the automatically created concept hierarchies, as the concept hierarchies acquired through Text2Onto may not be appropriate or complete. For example, Text2Onto identified that “storage facility” is a subclass of “facility” and “storage bin” is a subclass of “bin.” Developers should classify “storage bin” as a subclass of “storage facility” instead of adding a new class “bin” under the class equipment, since the single term “bin” is not used anywhere in the sample specification. Figure 2 shows the concept classification editor in Protégé.

**Step 3: define disjoint relationships between classes.** In this step, concept classes should be checked for mutual exclusiveness with other concept classes. Disjoint relationships are useful because they can help filter out irrelevant concepts based on given concepts and therefore facilitate the ontological reasoning process. In most cases, sibling classes should be disjointed with each other. For example, in Figure 2 (rectangle 3) the highlighted class “Precast Concrete Wall” is declared to be disjoint with the class “Masonry Wall”. However, there are exceptions: for example, class “Rod” has two subclasses: “Lifting Rod” and “Steel Rod” but they can’t be declared as disjointed classes until we can make sure the “Lifting Rod” is not made out of steel. Moreover, classes in different class hierarchies may not be disjointed in a few cases while most of the time they are disjointed, e.g., component class “Precast Concrete Wall” is relevant to resource class “Concrete” and activity class “Precast” but should be disjointed with all equipment classes.
Step 4: create properties for relationships between concepts. Protégé classifies a relationship as a property of related concepts. Therefore, similarly to Step 1, three generic properties should be first created to allow a basic classification of relationships. Other relationships with more specific semantic meaning can be later added as needed. The properties representing relationships defined in this step are general properties, which are available to all classes that are assigned as the domain and range classes of the properties. For example, if we want to represent the composition relationship between class component and class material, e.g. a column is composed of concrete and steel rebar, a relationship IsComposedOf is required. Component and material should then be assigned as the domain class and range class respectively for the IsComposedOf relationship in Protégé (rectangle 4 in Figure 3). In addition, the relationship Comprise is created as an inverse relationship of the former one. That is, each property defined for a context ontology must have a symmetric property as its counter-part to form a property pair. Another key issue in this step is that the property name of a relationship should be meaningful enough to enable ontology users to easily comprehend its semantic meaning and use it in the future. For instance, IsComposedOf clearly expresses the semantic meaning of the combination relationships between two classes as well as the roles that each of the two classes play in this relationship. Figure 3 shows examples of the defined properties for representing relationships.

Step 5: link classes using properties of relationships. After the types of properties have been defined in Step 4, relationships between the concept classes can now be established using these semantically rich properties. In Protégé, a class to be linked is highlighted first, and a property is then selected as well as a proper quantifier restriction (the existential quantifier ∃ or the universal quantifier ∀) and cardinality restriction. For instance, IsComposedOf clearly expresses the semantic meaning of the combination relationships between two classes as well as the roles that each of the two classes play in this relationship. Figure 3 shows examples of the defined properties for representing relationships.
existential quantifier $\exists$ restriction and a cardinality restriction with number 1 (rectangle 5 in Figure 2).

It is quite straightforward to develop a context ontology in Protégé following the five-step procedure. The functionality of creating new classes and properties is well established and ontology developers can easily manipulate such functionality through a user-friendly interface, even if the ontology is complex and consists of many concepts and relationships. However, if developers want to develop a well organized and semantically meaningful ontology in Protégé, they must have knowledge of domain semantics and logic in order to represent relationships between concepts in terms of logical expression, such as using quantifiers and conditional operators. While logic is a significant and powerful technique in knowledge representation [20], Protégé’s logical expression editor could be designed to be more user-friendly, such as providing a translator which can translate a logic-based description of two concepts and a relationship into a brief natural language description, thereby allowing users to experience less difficulty in understanding how logic works in Protégé.

### 3.3 Important Considerations

The following lists some findings and thoughts that we deem important when considering deployment of the proposed context ontology development process:

1. Using the proposed approach, the ontology development starts with a text file containing the desired safety specification section. After an initial ontology has been developed for a safety specification section, new concepts may be introduced by other safety specification sections in other text documents. Accordingly, the ontological concepts should be updated if any new concept that complements the initial ontology is identified. In our example, if new concepts
have been identified from another Subpart of the safety specification through Text2Onto, we should first filter the new concepts by checking their usefulness and rationality (as mentioned in section 3.1) and incorporate those qualified concepts in the existing context ontology by going through Step 2 to 5 in section 3.2. In other words, expanding the current ontology is a cyclic process which repeats the steps until one comprehensive ontology is formed that satisfies the requirements of the targeted ontology reasoning.

(2) We discussed how to use a knowledge acquisition tool, Text2Onto, in section 3.1 to identify concepts from a specification. However, after the concepts were identified, Text2Onto did not provide a function for identifying the source location of extracted concepts in the electronic document. From a user’s viewpoints, such a functionality is important to enable efficient checks of the validity of the extracted concepts and their relationships. Specifically, we don’t want to have to establish the link between the text in the document and the concepts displayed in Text2Onto manually using Text2Onto and a separate document viewer. Instead, an integrated environment in which a user can select an identified context and which then automatically shows the related locations in the text document would be preferred.

(3) The proposed process has limitations: the semantically rich relationships need to be determined manually, which degrades the possibility of automating the entire context ontology development process. The main reason why the relationships can’t be automatically determined is because the original document itself doesn’t explicitly describe all semantically rich relationships and it is impossible for knowledge acquisition tools to identify relationships that are not described. Therefore, we envision that there should be an ontology editor, which has access to a library of comprehensive patterns of semantic relationships between certain classes of concepts for a given domain, e.g. the construction domain. Current ontology development efforts, such as e-COGNOS [6] and ASCE’s proposed Civil Engineering Ontology [21] could constitute examples of such libraries. As long as concepts are identified and classified into proper classes, the ontology editor can automatically provide a list of referable semantic relationships from which ontology users can choose appropriate ones. This envisioned ontology editor could benefit the automation of context ontology development and promote more semantically-rich ontologies.

4. Conclusion

We propose a systematic process to help ontology developers efficiently create a context ontology. The process includes a sub-process of identifying concepts and relationships between concepts and the other sub-process of representing the identified concepts and relationships. For identifying concepts and relationships from specifications, a knowledge acquisition tool, Text2Onto, is utilized. This tool is found to be useful for identifying concepts and their topologies but it fails to provide more relevant, domain specific relationships between concepts. For successive ontology editing work, an ontology editor, Protégé, is adopted and is found to be beneficial for representing an ontology visually and formally.
The process results in a preliminary domain ontology that can be used as a starting point for collaboration with actual domain experts. As a result, the domain experts’ time commitment in the process can be reduced, leading to a less cumbersome experience. The advantages and disadvantages, limitations and important considerations for the tools and the proposed process are discussed. In light of the discussed findings in the paper, we expect that the proposed process for developing context ontologies for construction will benefit the knowledge development and representation from specifications in the building and construction industry.

Reference


