OPEN ONTOLOGY: A Self-Organizing Tool for Knowledge Acquisition in a Case Library

CHIEH-JEN LIN[1], MAO-LIN CHIU[2]
[1] *Department of Interior Design, Tainan University of Technology.*
Yung-Kang City, Tainan, Taiwan.
T60011@mail.tut.edu.tw
[2] *Department of Architecture, National Cheng Kung University,*
Tainan City, Taiwan.
Mc2p@ms21.hinet.net

Abstract. This paper is aimed to establish a sufficiently formalized, expandable metadata authoring tool of a case library based on ontology methodologies of AI. This tool cannot only self-organize facts of cases' features, but can also guide users to arrange those facts into a formalized structure to facilitate design reasoning. Then knowledge experts, such as design instructors or project managers, can apply this tool to author and organize metadata of cases' features in order to build knowledge templates for special situations. Beginners, such as students or assistants, can apply those templates to indicate and analyze facts of collected cases' features and acquire essential knowledge in solving problems. Finally, through easily recognized semantic relations of metadata, facts of cases' features can be self-organized, and the system can apply data mining techniques to detect and discover hidden knowledge patterns among facts of cases' features.

Keywords. Case Library; Design Knowledge; Knowledge Representation; Semantic Ontology; and Data Mining.

1. Introduction

This paper proposes a framework to build a self-organizing metadata system of a case library based on ontology approach. "Ontology" is an old term, which is borrowed from philosophy by computer scientists who give it a new definition. In artificial intelligent domain, ontology is a data model developed to facilitate knowledge sharing and reuse (Fensel, 2003). Gruber (1993) gave "ontology" the clearest definition: "formal, explicit specification of a shared conceptualization." Therefore building an ontology is to provide a machine-processable semantics of information sources that can be communicated between different agents including machine and human. Thus, it provides a feasible approach for solving the semantic problems of information.

1.1. ONTOLOGY BASED KNOWLEDGE REPRESENTATION

Ontology essentially is extended from two knowledge representations: (1) knowledge frame and (2) semantic networks (Figure 1). An ontology is similar to a knowledge frame, which is a collection of atoms applied to present aspects of a concept and is easily converted into a table within a database. An ontology can also present semantic relations using directed graphs which can be applied in automated logical deduction as in a semantic network. Therefore an ontology has greater expressive power in evoking human recognition and creativity through associations.

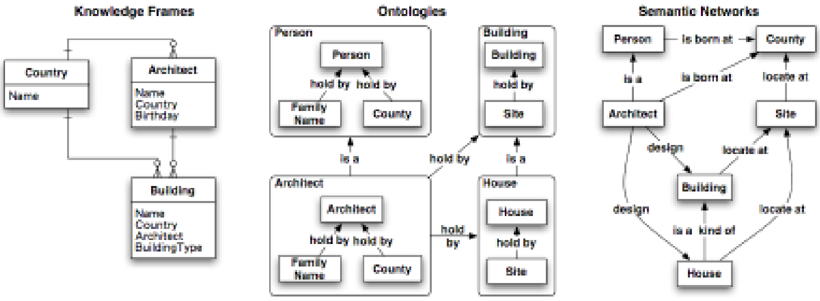

Figure 1. Three knowledge representations: frames, ontologies and semantic networks.

1.2. DISADVANTAGES OF ONTOLOGY-BASED REPRESENTATION

Whether for AI or philosophy, the meaning of an ontology is referred to "what is what" and implies contents and structures of a concept are fixed and static to keep them correct and consisting in a knowledge domain. This constraint may satisfy most of domains except design industry. Creativity to propose novel solutions for design problems is an important issue either for design practice or education. In architectural domain, architects and students are often encouraged to challenge existed specifications of conceptualizations in their domain. Even though genuine creative concepts still might be very rare, but many important precedents often went beyond normal specifications at that time. Precedents for this reason are collected by case libraries, but it seems that none could explicitly reveal why these precedents are important and how unique they are.

1.3. CASE BASE FOR ARCHITECTURE

This research is based on a previous study, "Case Base for Architecture" (CBA), an architectural design case library and a learning environment implemented on the web. CBA was developed based on database technology like other systems. Database is an important technology for preserving data, and it is also a methodology for abstracting information into formalized formats. However, database method must restrict numbers and types of data entities for efficiency reasons, and usually disregard semantic relations among entities. Although human can recognize semantic relations among cases' features by annotations of metadata, but knowledge represented by restricted metadata is always too shallow for designers to make effective design reasoning.

In previous studies, we found many useful clues were hidden in ill-structured non-textural contents (Lin & Chiu, 2003). Non-textural contents, such as images and diagrams, are more favored by designers than overly simple metadata, but it is very difficult to convert them into an index-able format without losing potential information. An alternative method of indexing them is to attach textural annotations, such as "tags" applied in popular web sites like Flickr and YouTube. Tags are so easy to be attached and recognized by humans, but they don't have any clue for machine to organize them. Therefore, it inspires us that a case library should also open metadata to users for representing which and what knowledge is interesting or useful, but it also needs a device to formalize them for machine processing.

2. Open Ontology of CBA

To represent commonality and uniqueness of creative concepts within cases, a case library actually needs an adaptable and extensible annotation system rather than any fixed specification of restricted conceptualizations. Therefore this paper proposes “Open Ontology”, an open metadata system with self-organizing ability based on ontology approach, to improve knowledge representing and organizing abilities of CBA.

2.1 ESSENTIAL ELEMENTS OF CONCEPTS IN THE OPEN ONTOLOGY

As the database method concludes all data formats into several types such as texts, numbers, and time etc., ontology method extends the ideas of object-oriented approaches and concludes the elements of concepts into (1) classes: a set of relevant abstract concept or objects composing a conceptualization; (2) attributes or properties: data attached on a concept to describe its features in order to distinct different classes; (3) relationships: special attributes to describe semantic relations among relevant classes.

There are four basic classes, i.e. the super classes of Open Ontology: (1) “case” class to collect all features of a same design cases; (2) “attribute” class to indicate attributes or properties which should depend on other class and should not be isolated, such as positions, material, and constructions etc.; (3) “quantity” class to indicate numeric data of an attribute class, such as time, length, height, area, and other dimensions; (4) “semantic” class to indicate the semantic relations between two classes.

Semantic relation is the major feature of ontology, so semantic class should be most critical. There are four instances of semantic classes based on WordNet for now: (1) “same-as” relation to indicate synonymous classes; (2) “is-a” relation to indicate a more generalized concept of a sub-class, such as “window” class “is-a” sub-class of the “opening” class; (3) “hold-by” relation to indicate a subject who holds an attribute class, such as “construction” class is “hold-by” the “building” class; (4) “part-of” relation to indicate the subject who consists of a partial class, such as “window” is a “part-of” of the “wall” class. Open Ontology further presents conceptual entities of design cases based on those classes and semantic relations.

2.2. ESSENTIAL CONCEPTUAL ENTITIES IN THE OPEN ONTOLOGY

A conceptual entity in Open Ontology is called a “sense.” Every sense has its attributes, parts, and quantitative atoms similar to an entity in a database. A database usually does not allow users to freely modify structures of entities, but it is possible for Open Ontology to generate and modify a sense.

There are five basic senses built first in Open Ontology: (1) “Building”, (2) “Participant”, (3) “Event”, (4) “Site” and (5) “Issue” senses. The first four senses seem to be similar to the essential entities in most case libraries, but an expert can still extend, combine and modify several existed senses to build a new sense in order to adapt in new situations. However, the “issue” sense seems unable to give it a well-structured and sufficiently formalized definition, and there are many esthetic and visual issues, such as “rhythm,” “proportion” or “style” etc., which are very difficult to represent by ontology. Therefore Open Ontology focused first on the explicit and well-definable issues such as “materials,” “constructions,” “components” etc for now.

2.3. SELF-ORGANIZING ABILITIES OF OPEN ONTOLOGY

Ontology can solve chaotic problems of semantic networks in efficient reasoning by hierarchical semantics. Therefore it is possible to implement a self-organizing metadata than to acquire knowledge from this hierarchy. The self-organizing ability of Open Ontology is based on two features of an ontology: (1) inheritable atoms from parents; (2) generalizing classification by classes' or senses' hierarchy.

For example, an expert can extend "house" class to "rowhouse" class by "is-a" relation. Since "house" class is a subclass of "building type" class of the "building" sense, Open Ontology can automatically classify "rowhouse" class into a subclass of "building type" and "rowhouse" class will inherit all subclasses of "house" class. Therefore Open Ontology can retrieve "rowhouse" cases when a user requests the "house" building type. Furthermore, another expert can extend "rowhouse" class to "townhouse" class by "same-as" relation and attach a "fa ade" class which is an attribute class extended form "elevation" class. Then Open Ontology will automatically attach the "fa ade" subclass to "rowhouse" class as well.

3. Implementation and Primary Evaluation of Open Ontology

Some ontology authoring and knowledge acquisition tools, such as Prot g , have very limited ability for processing Chinese language or merging bilingual data, and only focus on validation and consistency of knowledge rather than adaptable structures for different situations. So they are very difficult to integrate with an existing case library like CBA. Therefore, we developed Open Ontology to convert CBA into Open-CBA.

3.1. THE IMPLEMENTATIONS OF OPEN ONTOLOGY

The database of CBA applies MySQL and its interface uses PHP and Flash. Then there are two tasks for implementing Open Ontology in the database of CBA: (1) to separate metadata from the database, and (2) to separate data types from facts of cases features.

For storing the metadata of cases' features, Open Ontology applies design method of rational database to normalize them into three tables: (1) a "class" table for all terms applied to represent cases' features; (2) a "semantic" table for all semantic relation applied to organize classes; (3) an "sense" table for actual linkages of semantic relations among classes within senses.

For storing facts of cases features, Open Ontology normalizes them into three tables: (1) a "quantifier" table for different types of quantities such as "length," "area," "story" and "year"; (2) a "unit" table for dimensional units applied in quantifiers and their converting factors; and (3) a "fact" table for actual factors of cases' features. Database management systems (DBMS) like MySQL provide several data types in order to save memories and to improve efficiencies. As a prototype, Open Ontology only provides float point numbers for quantities and text for instances of metadata, but focuses on the necessary units and their convertibility of quantifiers.

Finally, Open Ontology uses two tables to help users to reduce repeated inputs of instances: (1) a "lexicon" table for instances of classes and (2) a "thesaurus" table for textural facts of cases' features. By applying rational database technology, Open Ontology converts CBA into Open-CBA and keeps contents and structures of CBA flexible but sufficiently formalized.

3.3. PRIMARY EVALUATION OF OPEN ONTOLOGY

There are two basic approaches for data mining: (1) hypothesis testing: top-down approaches for examining whether a hypothesis is supported by data or not; and (2) knowledge discovery: bottom-up approaches for discovering unknown patterns within data (Berry & Linoff, 1997). Open Ontology also provided two approaches: (1) knowledge templates, and (2) online analytic processing (OLAP) based on data queries.

An expert, such as a teacher or an architect, can build a template to represent his knowledge for a special issue, he can then use the template to collect and encode cases' features by himself or assistants. For example, we built a simple template of “row house,” which consists of two sub-senses: “space” and “fa ade.” Then we chose 10 rowhouses including two famous precedents: “Azuma House” by Tadao Ando and “Town House” by Philip Johnson. Next, we asked students to encode those cases by this template.

Finally, we could inquire every features associating with others of the same or different senses to discover hidden patterns based on the template. For example, when we inquired: “Which rowhouse’s fa ade has window?” Only 9 cases would be retrieved. The exception is “Azuma House,” whose fa ade has no window at all. Therefore the “confidence level” about the first hypothesis: “if (*rowhouse* has *fa ade*) then (*fa ade* has *window*)” is only 90%. Furthermore, if we inquired: “Which rowhouse’s fa ade has window at the first floor?” only 8 cases would be retrieved. Another exception is “Town House,” whose fa ade has no window at the first story. Therefore the “confidence level” about the second hypothesis: “if (*rowhouse* has *fa ade*) then (*fa ade* has *window*(*story* (*1st*)))” is only 80%.

The fa ade of a row house is supposed to be the only elevation that can receive natural light then should have some windows on it. Therefore the question is emerging: how do these row houses receive natural light? To answer the question, we further inquired cases whose fa ade has no “window” class at first story, and then ranked all the retrieved subclasses. As a result, we found that “courtyard” within “space” sense appeared in both cases. Therefore we found a pattern: “if not (*fa ade* has *window* (*story* (*1st*))) *then* (*rowhouse* has *courtyard*)” and the “support level” of this pattern is 100%. This example demonstrated a simple but typical process of OLAP and revealed the abilities of Open Ontology in discovering hidden patterns within a case library.

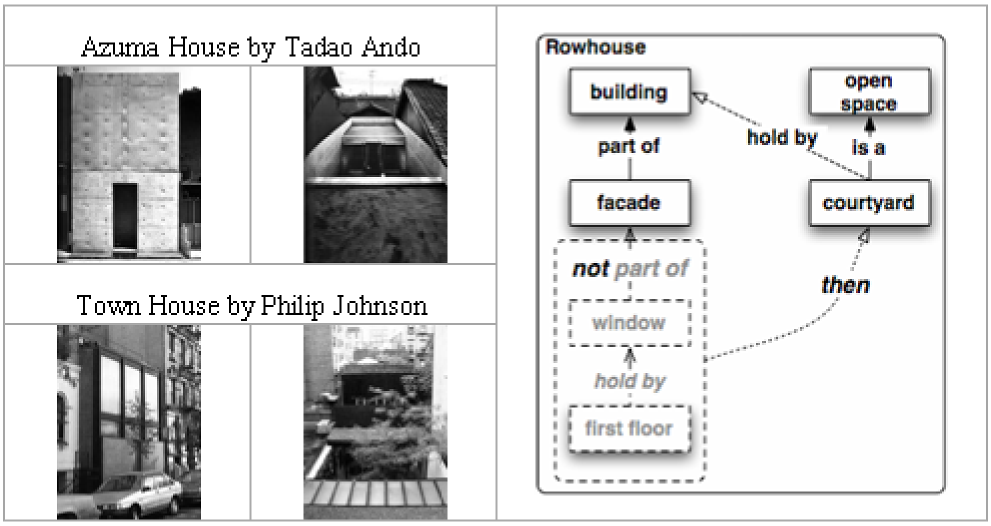

Figure 2. A pattern of rowhouse building type: Fa ade has no window at first story then building has a courtyard.

4. Discussions

Some scholars argued that the convincing breakthrough of case-based systems has yet to come for now, and suggested putting the learning topic on the top of research agenda (Heylighen & Neuckermans, 2001). The learning topic of the case-based systems provided ground for the following discussion about our studies of Open Ontology in Open-CBA.

4.1. OPEN ONTOLOGY VS. CLOSE ONTOLOGY

The conventional method of building “a domain ontology” must rely upon domain experts and applies top-down approaches to acquire knowledge, thus the ontology inevitably must be close to keep them stable, valid and consistent. However, cases within a library cannot and should not be restricted in limited domains, because they were instances of the “application ontology” which might cross over more than one domain or task ontology.

Design situations restrict the valid domain of design knowledge within cases. Therefore, top-down approaches usually encounter difficulties regarding the validity and consistency of design knowledge if cases crossed over different domains or situations. In such a circumstance, the system requires a bottom-up approach such as data mining to distinguish the different sub-domains of knowledge. For preserving different sub-domains into a system, a formalized but elastic and extensible structure is necessary, in other words, it needs a system like the Open Ontology rather than a monotonic and fixed metadata system within most of case libraries.

4.2. DATA ENCODING VS. DATA MINING

A commercial system could collect raw data from transactions by machine, but all data within a case library needs humans to encode then to preserve in the system. If the encoding processes were correct and efficiency, the procedures to clean and convert data into mine-able resources might be unnecessary. However, mistakes, inconsistence, and losing important information could not be avoided in encoding processes by humans. Therefore, most of the data in a case library is either too shallow to mine out any useful pattern, or not formalized enough to be processed by machine.

Open Ontology can be considered as an easy and instinctive assistant of encoding cases’ features. Knowledge templates of Open Ontology not only encode factors of cases’ features, but also encode semantic relations among those features, which were lost by knowledge frame and database method. Semantic relations are critical factors for humans to recognize, learn and remember knowledge. The lack of semantic relations should be one of the reasons for causing case-based systems to fail on learning purposes.

4.3. ASSOCIATIONS VS. ISOLATIONS

Alexander said: “a city is not a tree,” then a design case should not be either. Tree structure can keep the hierarchy of an ontology valid and consistent along vertical levels, but it inevitably will lose semantic associations along horizontal connections. Therefore concepts within different levels of the hierarchy might become isolated and lose important associations for design reasoning. Since every class or sense can associate with his parents, Open Ontology can preserve associations crossing different levels of the hierarchy. Therefore, freely associative reasoning is possible in Open-CBA.

Open Ontology has only four semantic instances, and has less representing ability than other tools such as the Web Ontology Language (OWL) (Antoniou & van Harmelen, 2004). For example, OWL can easily build a “face” class to describe the relation between “courtyard” and “window” class, and then it will be easier to retrieve their relations with in the Azuma House. But Open Ontology can provide more semantic relations along the horizontal associations than hierarchies of the OWL, therefore user may associate “window” with “courtyard” by natural light and privacy, which are two features within both “window” class and “open space” sense. Associative reasoning makes Open-CBA more inspiring and helpful for learning than other systems based on analogy reasoning. But how to control the levels of associations for effectively design reasoning still needs further investigations.

5. Conclusions

Database technology is still the most reliable and effective means of preserving and organizing cases' contents by machine. Consequently, the knowledge discovery techniques in database, such as data mining, should be a feasible means to acquire knowledge from databases of a case library. All data mining methods use induction-based learning, which is a process of forming general concepts by observing special example of concepts to be learned (Roiger and Geatz, 2003). However, data in most of case libraries, which were abstracted based on database method and knowledge frame, usually lost important semantic associations among cases' features. Therefore, concepts represented by cases' features became isolated and lost the mine-able clues such as semantic associations for machine or human to induce general concepts. This paper therefore tried to implement a self-organizing metadata system based on the ontology approach of AI domain named Open Ontology. Open Ontology can help expert to organize concepts within cases, and help user to encode data into formalized formats. Then, through inquiring semantic associations among classes within knowledge templates, the correlations among cases' features would be easy to retrieve in order to discover hidden patterns. However, creative designers might not only hope to learn ordinary solutions but also discover unique creations from the case library. Open Ontology can also help users to detect those cases by retrieving exceptions of ordinary patterns then revealing their correlations. Those exceptions may not be important for data mining approaches, but may be good edifications for creative design.

Regardless, a satisfied retrieval is always concerned with user's searching intensions, and the useful design knowledge is always concerned with design situations. Future works shall include the analysis of necessary semantic relations for representing relations among cases' features beyond the four basic instances, and the software agents for providing more efficient retrieving and detecting for known or unknown patterns, then to make the case library more sensitive for users' searching intensions.

References

Antoniou, G. & van Harmelen, F.: 2004, *A Semantic Web Primer*, MIT Press, Cambridge.

Fellbaum, C. ed.: 1998, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge.

Fensel, D.: 2003, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, New York.

Gruber, T. R.: 1993, A translation approach to portable ontology specifications, *Knowledge Acquisition*, **5**, pp.199-200.

Guarino, N.: 1998, Formal ontology and information systems, *FOIS '98*, Trento, Italy, pp.3-15.

Heylighen, A. & Neuckermans, H.: 2001, A case base of case-based deign tools for architecture, *Computer Aided Design*, **33**(14), pp.1111-1122.

Kolodner, J.L. & Leake, D.B.: 1996, A tutorial introduction to case-based reasoning, in Leake, D.B. (ed.), *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, MIT Press, Cambridge, pp.31-65.

Lin, C.J. and Chiu, M.L.: 2003, Smart semantic query of design Information in a case library, *CAADFutures 2003*, Kluwer Academic Publishers, Tainan, Taiwan, pp.125-135.

Roiger, R. J. and Geatz, M.W.: 2003, *Data Mining: A Tutorial-based Primer*, Addison Wesley, Boston.