DESIGN COMMUNICATION (3B)

OPEN CASE STUDY

An Ontology-Based Case Encoding Tool for Architecture Design Case Study

CHIEH-JEN LIN
Department of Interior Design, Tainan University of Technology
t60011@mail.tut.edu.tw

AND

MAO-LIN CHIU
Department of Architecture, National Cheng Kung University
mc2p@ms21.hinet.net

Abstract. The aim of the paper is to establish an ontology-based case encoding tool with sufficient formalization and expansibility to assist users for organizing the case information and increasing the feasibility of the design knowledge in a case library. The tool is named Open Case Study (OCS). OCS is a formalized and expandable tool for authoring metadata of a case library and organizing them by their semantic ontology. By using the templates constructed by design experts, such as design teachers or experienced architects, OCS provides the user with explicit but adaptable guidelines for case analysis and encoding. OCS then performs the searching and mapping function provided by Open Ontology. Thus, when the user is encoding the information segments of cases, relevant knowledge chunks in the case library can then be immediately provided, such as relevant senses in similar cases, all atoms of a relevant sense, and known value ranges of a relevant property. This assists users to avoid data mistake and duplication in encoding design cases.

Keywords. Case library; design knowledge; knowledge representation; semantic ontology; and metadata.

1. Introduction

Design precedents are considered the containers of the required knowledge for past design problems, experience and solutions (Kolodner and Leake, 1996).

Hence, for architectural design practice or education, “case study” is an important task in the design process. By collecting and analyzing design precedents, architects and students will be able to recognize design problems, learn relevant knowledge and the solution experience. A purpose for developing a design case library is to accumulate the design knowledge and experience obtained from case studies. However, most of the design case libraries focus on the storage, indexing and retrieval of case information and ignore the way to assist users in converting raw case data into useful design information.

On one hand, what constitutes useful design knowledge varies with the design problems and situations. On the other hand, the knowledge representation of a case library must be sufficiently formalized for computers to process data as well as for human beings to read. However, the formats that people can easily read are texts in nature language, photographs, and drawing, etc., in which the computer usually cannot easily organize or automatically build an effective index. Formalized formats, such as XML or OWL, are usually complicated and difficult to read without appropriate visualization. Moreover, such formalized formats are limited to the verified knowledge domain so the system loses its expansibility of other design situations (Antoniou and van Harmelen, 2004).

To efficiently store, index and retrieve design cases, the classification of design case features is critical for developing a case library. Yet, researchers have argued that: (1) classification of case features is meaningful and useful only with respect to a specific purpose at hand; (2) how a designer recognizes the features of a case is a function of their personal experiences; (3) the database of a case library may need complete update or modification when a new classification is proposed; and (4) useful characteristics of cases cannot be known in advance (Dave et al., 1994). Therefore, the priori and static classifications and indexing features of design cases may obstruct personal interpretations acquired from designers and reduce the function of a case library for helping users to study existing or new cases.

2. The Framework of Open Case Study

This aim of the paper is to establish an ontology-based case-encoding tool with sufficient formalization and expansibility, which assists users to organize the case information and increase the feasibility of the design knowledge in a case library. The tool is named Open Case Study (OCS). OCS is based on a previous study named Open Ontology (Lin and Chiu, 2008). Open Ontology is a formalized and expandable tool for authoring metadata in a case library and organizing them by their semantic ontology. By using the templates constructed by the design experts, OCS provides the user with explicit but adaptable guidelines for case analysis and encoding.

2.1. CONCEPTUAL STRUCTURE OF CASE FEATURES

As the database method presents all data formats by several types of data, such as texts, numbers, and time etc., the ontology method extends the ideas of object-oriented approaches and present the elements of a concept or object by (1) classes: a set of relevant concepts or objects composing a conceptualization; (2) attributes/properties: data attached to a class that describes its features; (3) relationships: special attributes to describe semantic relations among relevant classes (Fensel, 2003; Guarino, 1998). Therefore, there are four basic classes, i.e. the supper classes of Open Ontology: (1) "case" class that collects all features of a same design cases; (2) "attribute" class that indicates attributes or properties which should depend on other classes and should not be isolated; (3) "quantity" class that indicates numeric and measuring data of an attribute class; (4) "semantic" class that indicates the semantic relations between two classes.

OSC therefore applies four classes of Open Ontology to compose a conceptual entity, which is called a "sense," serving as a unit of knowledge chunks representing features of the design cases'. Every sense has its attributes, parts, and quantitative atomic classes similar to an entity in a database. While a database usually dose not allow users to freely modify the structures of data entities, it is very easy to generate and modify a sense in OSC by attaching or removing classes within a sense.

2.2. ASSISTANCES FOR ORGANIZING CASE FEATURES

Based on the principle of Six Ws, there are 5 basic senses in OCS: the (1) Building (what), (2) Participant (who), (3) Event (when and where), (4) Site (where), and (5) Issue (why and how). Every basic sense has its essential composing classes and semantic class to indicate the correlations. Semantic class is a major feature of ontology, which is different from other knowledge representations such as frameworks and semantic networks. Semantic class not only helps to organize the attributes of a class and relationships among different classes but also helps to build a general reasoning engine for assisting users to acquire knowledge from cases.

For example, the "is-a" relation indicates a more generalized concept of a sub-class, and the "hold-by" relation indicates a subject class who holds an attribute class (Fellbaum, 1998). Based on the object-oriented theory, the sub class should inherit all or partial attributes of its super class. Therefore, once a sub-class has been indicated an "is-a" relation to a super-class, then OCS will automatically attach the sub-class with all attribute classes which have been assigned the "hold-by" relations to the supper-class. However, users still can attach new or delete unnecessary attributes of the sub-class later. Based on the composition of attribute classes, OCS can easily rank the similarities of different senses and prompt users to check whether two senses are same or not.

Figure 1. The interface of OCS for organizing case features.

2.3. ASSISTANCES FOR ENCODING CASE FEATURES

By compositing senses and atomic classes, design experts, such as design teachers or experienced architects, can build templates for other beginning users, such as students and assistants, to analyze and encode cases features. OCS then performs the searching and mapping function provided by database technology. Thus, when a user is encoding information segments of cases, relevant knowledge chunks in the case library can be immediately provided, such as relevant senses in similar cases, all atoms of a relevant sense, and known value ranges of a relevant property. This assists users in data encoding to avoid data mistake and duplication. With the help of self-organizing abilities of Open Ontology, OSC will automatically organize all encoded features to satisfy the structure of knowledge template in the case library.

3. Implementation and Primary Evaluation of Open Case Study

The implementation of OCS applies MySQL for data storing, PHP script language for data accessing, and JavaScript for OCS's interface. To store expandable metadata of cases' features by a rational database, Open Ontology has concluded and normalized them into three tables of database: (1) a "class" table for all terms applied to represent the case 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. OCS then extends a "template" table, i.e. a set of relevant senses composed and arranged by semantic relations, for experts to build their own templates to guide themselves or other users to analyze cases.

To store encoded facts of case features, Open Ontology has also concluded and normalized them into three tables: (1) a "quantifier" table for different types of quantities; (2) a "unit" table for dimensional or measuring units applied in quantifiers and their converting factors; and (3) a "fact" table for textural factors of case features. Therefore, OCS stores encoded textural and quantitative facts by liking the template tables to these three tables.

To help users reduce repeated inputs of instances, OCS can automatically perform the mapping function when a user encodes a textural data of the case features, then store instances of classes into the "lexicon" table, and index textural facts of case features into the "thesaurus" table. By applying rational database technology, OCS can easily retrieve and conclude the range of quantitative facts associating with a class of case features. For examples, OCS can retrieve a building type class, such as "row house," associating with the stories of building so as to find the maximum or average stories of the row house type cases in the case library. OCS can automatically perform the searching function when a user encodes a quantitative fact of a case feature and then prompt the user by known quantitative range of the associating class.

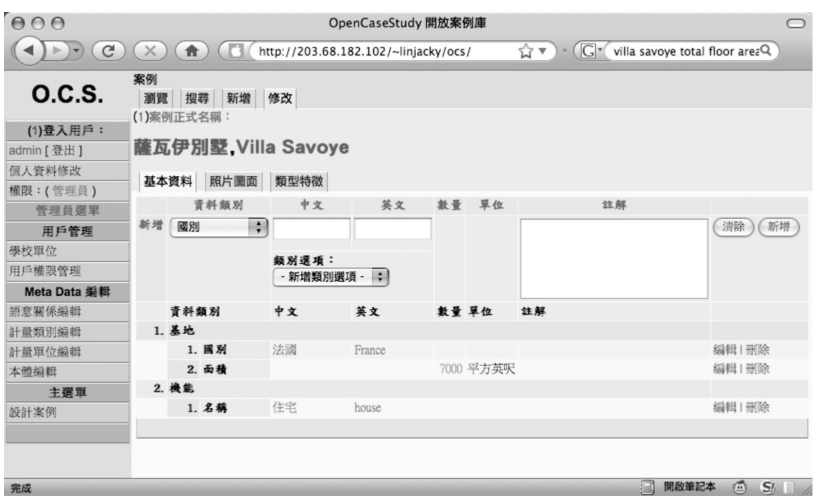

Figure 2. A sample template in OCS for encoding case features.

This function not only helps users reduce mistakes but also helps experts find extraordinary data within the case library.

4. Discussion

Although OCS applies database technologies to abstract case features into a database, a major purpose for developing OCS is to overcome the restrictions of database methods. Even though database technologies can keep data correct and consistent in the system, the database methods must restrict the types and numbers of entities in systems and usually lose semantic relations among entities for the sake of consistency and efficiency. Since what the useful features are varies with design problems and situations, and how users encode those features varies with personal experiences and design purposes, an effective yet static structure of case features will become an obstacle to case study. Therefore, the expandable structure of OCS based on Open Ontology can help design experts to store their personal experiences in studying design cases and help beginning users to encode case features for further use.

The expandability of OCS is based on the ontology method. Note that the meaning of an ontology in the field of artificial intelligence or philosophy is referred to "what is what" and implies that contents and structures of a concept are static to keep them consistent in a knowledge domain. This constraint may satisfy most of domains, but it may not satisfy the purpose of case study. When a new design problem or issue arises, an architect may reinterpret his/her past experience and known knowledge acquired from cases. Take the example of the pilotis and roof garden is two features of Villa Savoye by Le Corbusier; since the green building issue arouse, architects may associate these two features with pile-supported dwelling and energy-saving function, and then reinterpret Villa Savoye as a green building precedent. Although it is easy for a case library to attach a "green building" tag to this case to indicate its feature, this approach cannot help users find out why and where the case is associated with the "green building" issue. For OCS, users can indicate that the pile-supported structure and energy saving are two features of a green building. Since the pilotis is a sub-class of pile-supported structure, the roof garden has the feature of energy saving. As a result, it is easy for OCS to reinterpret existing design cases to help users really reuse the knowledge in the case library.

5. Conclusions

Most case libraries fail to help design experts for representing their experiences and help beginners for encoding data of design cases into the libraries. This paper therefore tries to implement a self-organizing metadata system, which is based on a previous study named Open Ontology, and applies database technology to assist users for composing the encoding template of design cases, which encodes case facts into the case library.

Unlike other more complete knowledge acquiring tools, such as Protégé, which utilizes a top-down approach and focuses on the validation and consistency of a knowledge domain, OCS can start from the facts of cases collected by users, and construct the "fact base" and their "ontology base" at once during the data encoding. Based the expansible and adaptable semantic structures of Open Ontology, OCS has the advantages of convenience and flexibility, which provide users with various selections on how to organize collected case information. In terms of the validation of knowledge, OCS still lacks an authoring tool of more ontological axioms and a powerful reasoning machine for validating the ontology. Data mining technique based on the searching and mapping function of database is currently used to provide classification and comparison with the existing data range and organization of patterns among knowledge atoms and senses, which thereby hints possible abnormality or exceptional situation. In the previous study, we found that even though exceptional information may not be useful for acquiring generalized knowledge, it can identify unique precedents and inspire users to think creatively. In the future, with the implement of an authoring tool of ontological axioms and a reasoning machine for validating axioms, OCS will be able to automatically verify the encoded fact information and learn from the encoded case facts.

References

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

Dave, B. Schmitt, G. Faltings, B. and Smith, I.: 1994, Case-based design in architecture, *The Proceedings of Artificial Intelligence in Design '94,* Kluwer, Dordrecht, pp. 145–162.

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.

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

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

Lin, C. J. and Chiu, M. L.: 2008, Open ontology: A self-organizing tool for knowledge acquisition in a case library, *Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia* (CAADRIA 2008), Pimniyom Press, Chiang Mai, Thailand, pp. 328–334.