Visual Architectural Topology: An Ontology-Based Topological Tool for Use in an Architectural Case Library

Chieh-Jen Lin

Tainan University of Technology, t60011@mail.tut.edu.tw

ABSTRACT

This paper aims to develop a tool entitled "Visual Architectural Topology (VAT)" for encoding topological information within a case library. By applying previous research results, such as interactive spatial topology encoding and retrieval tools, VAT can annotate design objects and their topological information within unstructured information, such as pictures or plan drawings of a design case. By applying an ontology-based topological validation mechanism, VAT aims to establish a visual language for representing the "topological knowledge" of architectural design objects in a case library. The purpose of VAT is to extend the knowledge representation ability of a design case library, and to provide a foundation for the development of a design-assistance tool performing the conversion and processing of semantic and geometric design information.

Keywords: case-based design, case library, architectural topology, semantic ontology, and visual language.
DOI: 10.3722/cadaps.2013.929-937

1 INTRODUCTION

Building information modeling (BIM) has been implemented by many commercial CAAD tools, and has been widely adopted in architecture, engineering, and construction practice. The BIM concept was originally developed via research on building product modeling (BPM) [6]. According to initial suggestions, BPM involves three types of information: semantic, topological, and geometric information concerning building components. Among the three type of building information, topological information is critical to BIM, because topology describes the spatial connections among building components, and is a fundamental aspect of parametric design [5]. Unfortunately, since there is insufficient research suggesting necessary topology for architectural design applications, BIM currently only supports topological information for fabrication-level applications [5], which aim to support communitarians of architects, constructers, and other stakeholders when process detail and construction design in middle and late design stage. Without implementation of supporting architectural-level topology such as spatial topology [19], which are important for architects to select a case then to develop proposals in concept and schematic design stage, BIM therefore is not suitable for early architectural design stage.

From the perspective of an information-driven approach, architectural design employing BIM can be regarded as the processing and inter-conversion of three types of design information. While experienced architects have meta-knowledge concerning conversion among the three types of design information, students and novice designer must learn this meta-knowledge by doing. In other words, they must study past design cases, and try to reconstruct the conversion process. Although design case libraries were originally considered to play the role of knowledge depositories in the learning process, case libraries are usually implemented employing database technology, which only extracts semantic information of case features as an index mechanism, and therefore most clues about the conversion process among the three types of design information are lost.

This paper is a follow-up study to the two previous projects "STR: Spatial Topology Retrieval [18]" and "SSO: Smart Spatial Ontology [16]." This paper aims to develop a visual tool entitled "Visual Architectural Topology" (VAT) for the purpose of encoding architectural topologies from design cases in a house case library termed "OCS: Open Case Study [17]." By applying previous research results, VAT aims to establish a visual language for representing topological information concerning architectural design objects, and also seeks to extend the knowledge representation ability of the OCS case library. Section 2 of this paper firstly explains the approach for why and how to develop VAT in order to encode architectural topology within unstructured information of design cases. Section 3 then introduces relevant technologies and primary evaluations for the implementation of VAT on the Internet. Section 4 finally discusses the experiences after developing VAT from the view of design information processing.

2 THE VISUAL ARCHITECTURAL TOPOLOGY APPROACH

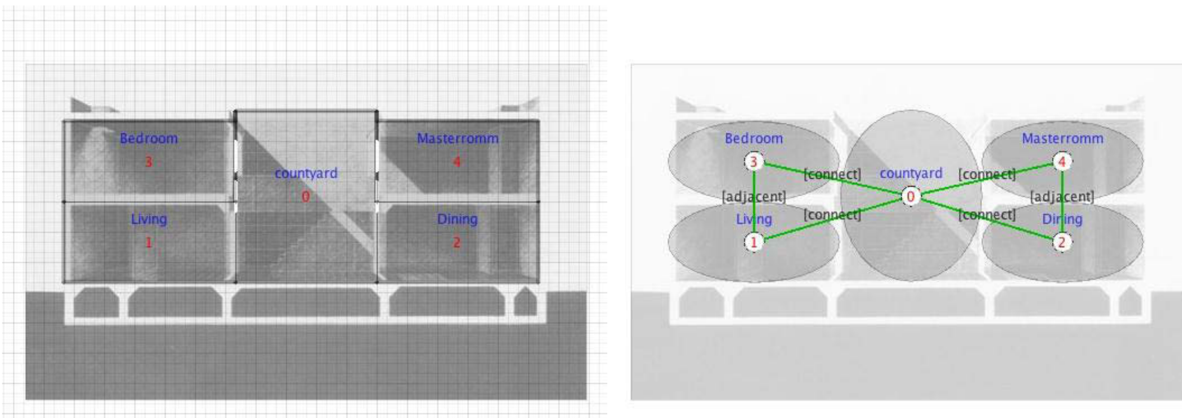
The initial idea of developing a case-based design system was to provide design cases as a knowledge source for design problems and solutions. 2D drawings or 3D models of design cases can explicitly represent constructional knowledge, which involves topological relations of building components in fabrication-level for what and how to build the building. However, abstract architectural knowledge, which usually involves correlations of abstract subjects in conceptual level, are usually implicit within unstructured information such as graphics and pictures, which are preferred by architects to represent architectural-level topology. For example, design concept for why and how to compose spatial topology usually is ill-defined and varies with different architects. It is therefore difficult to formalize this kind of design knowledge into machine-processable formats, which causes major bottlenecks for case-based design (CBD) and case-based reasoning (CBR) research for architectural applications. As more BIM implementations are used in practical work, BIM may give rise to a rich depository of BIM-based case libraries in the future. However, for important precedents outside of present practice or before the initiation of BIM, it is usually difficult to collect detailed information in order to construct BIM files. In addition, since abstract topology for architectural design applications is absent in most implementations of BIM, it is still difficult to represent abstract architectural design concepts by applying BIM [11]. In contrast, unstructured information concerning design cases, such as diagrams and images of important precedents, are easily collected and stored in case libraries, and more "information-rich" than BIM files as far as further analysis of architectural concepts is concerned.

The current method for indexing unstructured information in design cases usually relies on the attachment of semantic tags. The advantages of this method are that it is open, flexible, and easy to implement, but simple semantic tags cannot adequately represent the relationships among objects within the design information. Therefore, employing the open-annotation strategy of the OCS library, this paper proposes an open and adequately-formalized tool that can assist users to visually represent their interpretations of topological knowledge. Rather than providing a rigid representation framework of cases' common features, the VAT approach is based on (1) graphic annotations attached to unstructured information, (2) the bridging of semantic ontology and visual topology, and (3) visual validation when encoding topology.

2.1 Graphic Annotations on Unstructured Information

Appropriate interfaces, which can deal with non-textual information and go beyond textual search, were usually absent from early case libraries due to technical reasons. Since architects are educated to think and reason through visual information such as sketches and diagrams [20], graphic interfaces such as "SpaceScope [12]" and sketch-based interfaces [13] have been proposed in order to improve retrieval in a case library. Visual approaches must rely on preceding mechanics, regardless of the topologies or ontologies of a case's contents. However, declaration of an ontology is usually the job of knowledge engineers, and involves the lengthy representation of the semantic features of knowledge chunks. And while burdensome parametric inputs in BIM inevitably confuse architects, ontology-authoring tools such as Protégé perplex designers as well. Although visual alignment tools for ontologies have been proposed [14], alignment tools for semantic ontologies with visual knowledge resources are still absent.

In the previous studies, a visual tool named "smart spatial ontology" (SSO) was therefore used to integrate graphic declarations with a search interface in order to deal with visual information in design cases [16]. SSO can easily draw schematic spatial layouts while attaching graphic annotations of spatial topologies to the house plan images. VAT extends the graphic interface of SSO to the translation of more topology types than just the spatial topologies of plan drawings. For example, VAT can attach simple diagrams as graphic annotations to a section drawing of Azuma House Ando (Fig. 1.a), which is a masterpiece by the Japanese architect Tadao Ando. Based on the schematic diagram, VAT can automatically encode the initial topological relations of relevant objects (Fig. 1.b). In addition, VAT allows users to define more topological relations than SSO.



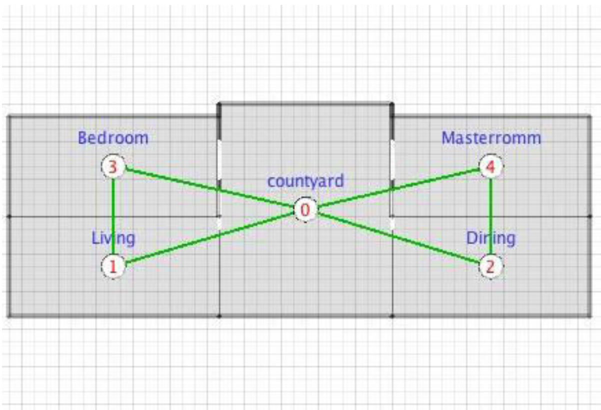
(a): Schematic diagram of a section (b): Spatial topologies of spaces

Fig. 1: Graphic annotations on a section drawing of Azuma House.

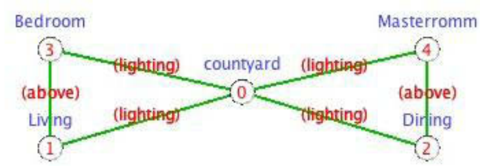
2.2 Bridging Semantic Ontology and Visual Topology

Ontology is a knowledge engineering technique and a data model facilitating the sharing and reuse of knowledge. A knowledge chunk in an ontology can be represented by a triple set of "subject," "predicate," and "object [7]." As a topology represents spatial relations between design objects, the "predicate" represents the semantic or causal relationship among knowledge objects, and determines how an intelligent agent reasons with and validates an ontology by first-order logic or other formal logic. Development of an ontology therefore consists of the establishment of a formal language to describe design knowledge. For example, a spatial ontology has been proposed [3] to represent three aspects of spatial knowledge: connectivity, proximity and orientation. However, most proposed ontologies can only represent what topologies are contained in a case, but cannot represent why and how a topology is composed in a certain way within the case, let alone allow users to state different interpretations or new types of topologies.

To assist users to associate semantic ontologies with spatial topologies, SSO allows users to annotate semantic predicates on spatial topologies, and therefore provides a bridge between the semantic ontologies and spatial topologies of architectural spaces [16]. However, the functions of SSO focus on spatial topology, and are restricted to the same declaration level of an ontology. For improving knowledge representation ability, VAT adds the cross-level function of declaring the predicates associated with different levels in a semantic ontology. For example, VAT can assign a "part-of" predicate with "a suite" to a connective topology of a master bedroom and a bathroom, and assign a "serve" predicate to declare the bathroom "serves" the bedroom in order to represent the concept of "a suite." VAT therefore can not only associate topologies of design objects assigned by users with existing semantic features in a library, but also associate topologies with abstract concepts interpreted by users. By attaching semantic features to diagrams and associating diagrams with semantic predicates (Fig. 2.b), VAT can not only align an ontology with visual information when a knowledge chunk is acquired, but also provide a visual interface for the alignment of graphic topologies and the semantic ontology of design objects (Fig 2.).



(a): Topological diagram of the house section



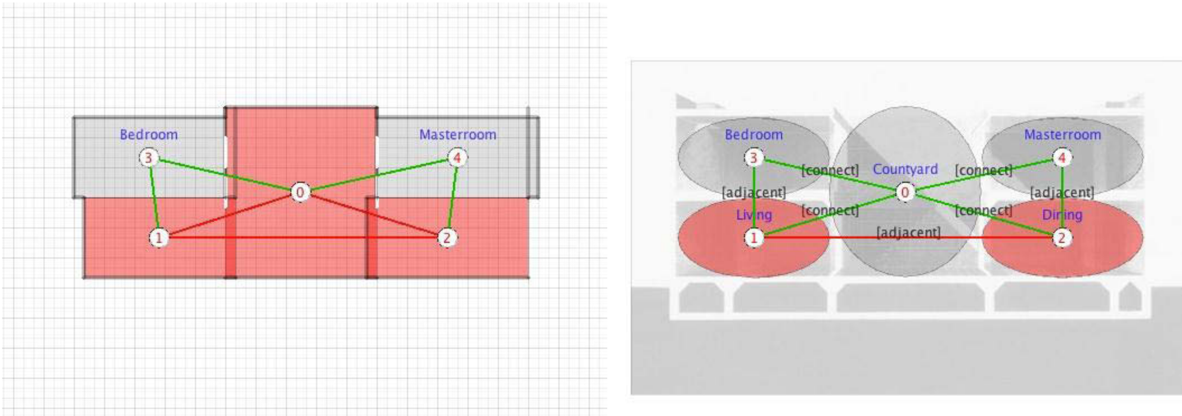
(b): Semantic ontology of the topology

Fig. 2: Bridging semantic ontology with topological diagrams of Azuma House.

2.3 Visual Validation when Encoding Topology

In the case of both AI and philosophy, an ontology usually implies that the content and structures of a shared concept are fixed and static, which ensures that they are correct and consistent. The knowledge-acquisition process therefore usually employs a top-down approach and focuses on the validation and consistency of an ontology within a particular domain of knowledge. However, while this constraint may be adequate for most domains, it may not be able to satisfy the needs of the architectural design domain. By definition, an ontology should be a "formal, explicit specification of a shared conceptualization [9]." However, architects are typically educated to be creative, and are therefore encouraged to challenge pre-existing specifications of conceptualizations in order to win design competitions. Important precedents are therefore collected based on the basis of their unique concepts. But the more distinctive the concept of a design case, the smaller the valid domain of its design knowledge. Since architects are usually asked to propose unique design concepts for every project, no wonder Eastman claims that there are no common topologies that should be supported in BIM for architectural applications [5].

While a scholar has declared that "designers reason from cases, not from principles [2]," experienced architects can transform and interpret cases to derive new solutions, rather than to directly imitate precedents. Although most fabrication-level information should be fixed and static within built precedents, designers' interpretations of topologies and ontologies when analyzing built precedents may still vary with the times and technology. The valid domain of an ontology can therefore be restricted to the relevant information within a case, rather than a larger scope, such as same building types in the case library. VAT can therefore interactively validate the encoded topology and its ontology within a single case through use of the collision-test based algorithm [15], and provides visual clues to help correct conflicts by modifying geometric features within schematic layouts. When users encode a topology on unstructured information, VAT can immediately check whether features of the diagrams meet the definitions of a known ontology. For example, when an adjacent topology has been assigned to two objects within the diagram, but one object of the topology is moved by the user or is far away from the other, VAT will then detect a conflict in the diagram (Fig 3). When a conflict is detected, VAT will highlight the conflict and attempt to automatically modify the diagram configuration based on the known definition of topology or ontology (Fig 3.a). Although VAT cannot automatically solve all possible conflicts, visual clues to invalidated relations can help users to recognize and solve conflicts by themselves (Fig 3.b).

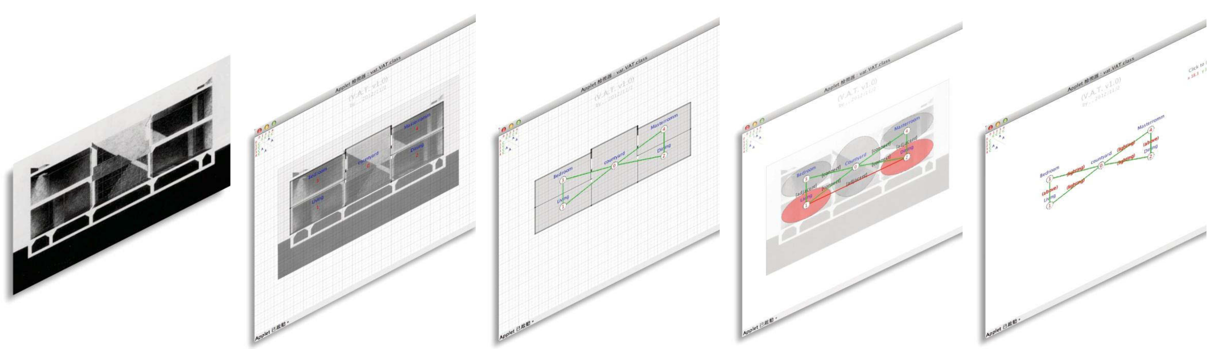


(a): Highlighting conflicts in the schema diagram (b): Prompting invalidated topology
Fig. 3: Visual validation of topology and ontology within VAT.

2.4 Summary

Figure 4 demonstrates an exemplary procedure for encoding topologies and their semantic ontology of the section image of Azuma Houses. The major procedures are: (a) Retrieval: to select a case, such as a section image from a case in the case library (Fig 4.a); (b) Identification: to identify design objects, such as schematic spaces in the selected image (Fig 4.b); (c) Encoding: to assign topological relations among identified design objects, such as topologies among identified spaces (Fig 4.c); (d) Validation: to validate topologies by STR's algorithm, which automatically validate assigned topologies and then to prompt conflicts (Fig 4.d); (e) Ontology: to assign semantic annotations, which is opened to users to explain the meanings of topologies (Fig 4.b).

Based on the prior knowledge consisting of the ontological techniques in SSO and predefined topologies of STR, the VAT project is devoted to developing a visual language tool assisting users to represent the topological knowledge in design cases. VAT improves the topological knowledge representation in the OCS case library from the spatial topologies of house plans to free and open interpretations of other unstructured information in the library. Via the semantic association of ontologies with topologies, VAT can improve on SSO's graphic-based search mechanism, and assists users to retrieve and learn topological design knowledge more efficiently from unstructured information in design cases.



(a) Retrieval (b) Identification (c) Encoding (d) Validation (e) Ontology
Fig. 4: An exemplary procedure for encoding topologies within the section of Azuma House.

3 THE IMPLEMENTATION AND INITIAL EVALUATION OF VAT

The initial version of OCS consisted of a web-based application applying MySQL database software, and its access interface is implemented employing PHP and Flash. However, thanks to the continuing development of information technology, VAT is implemented by a new database system and a web-based interface, improving the usability and accessibility of the OCS library.

3.1 NoSQL Database for Flexible Storage of Design Knowledge

As a rational database management system, MySQL cannot easily be used to implement flexible metadata-authoring tools allowing users to develop different interpretations of case content. VAT therefore applies the MongoDB database software, which is a scalable, high-performance, open source, document-oriented NoSQL database [1]. Unlike a rational database, "Not only SQL" (NoSQL) database does not need a predefined schema of data tables. The NoSQL database is therefore better at storing open-structured metadata of design information. Since MongoDB is also a key-value store database, there is no need for a fixed data model to store information. MongoDB is therefore applied to store semantic ontology and graphic topological annotations within unstructured information. Through the use of MongoDB, VAT can thus easily store topological knowledge as objects or XML, and can retrieve graphic annotations attached to unstructured information as documents from the OCS library.

3.2 VAT's Up-to-Date Web-Based Interface

The initial version of SSO applied PROCESSING to implement an interactive interface for encoding spatial topology. PROCESSING is a simplified version of the Java program language, and offers a lightweight integrated development environment (IDE) in which visual design students can learn programing [21]. In keeping with emerging modern web-based interface technologies, VAT integrates Processing.js to improve the compatibility of its interface with modern web browsers, which may not be compatible with Java plugins for safety and efficiency reasons. Processing.js is a JavaScript version of PROCESSING [8], which aims to simplify data visualization and interactive animations, and help users to execute Java-based PROCESSING works on Java-incompatible modern web browsers.

3.3 Primary Evaluation of VAT

The knowledge representation bottlenecks in CBD and CBR research for architectural applications are not only caused by the unstructured content of architectural design cases, but also the quoting contexts of design knowledge. Since the design contexts usually cannot be completely predictable, design problems may go beyond the predefined classifications of a case library. Furthermore, since how designers recognize the features of a case is a function of their personal experiences [4], when in testing VAT, users unsurprisingly preferred to explore and browse by themselves in order to discover new ideas from the content of cases, rather than rapidly retrieving relevant cases by applying the system's search mechanisms. A representing tool, which can assist designers to represent their personal recognitions and interpretations, should therefore be more useful than a prior and static classification mechanism in helping users to study and reuse existing or new cases.

The Web 2.0 technology of VAT's interface provides users with a simple, efficient, and user-friendly means of storing, indexing, retrieving, and sharing unstructured information, including images and pictures of design cases. However, the presentation of visual knowledge, such as the topological knowledge contained in unstructured information, still faces challenges in connection with both technological and user-experience aspects. The VAT prototype is still rough and lacks 3D annotation ability, which was the most sought-after function when students experimented with the annotation of pictures and photos, and was sometimes not intelligent enough with regard to user-experience aspects when tested by our students. However, VAT has some advantages over other approaches: (1) Semantic ontology can be attached to topologies, providing explanations facilitating learning and reasoning. (2) The integration of semantic ontology, graphic topology, and their visual sources of unstructured information allow the alignment of different abstract forms of design knowledge. (3) Simple diagrams can be added as graphic annotations to visually represent topological knowledge. (4) VAT adds more use-defined topologies than SSO, providing a validation mechanism helping users to correct conflicts in knowledge representation.

4 DISCUSSIONS

An architectural design information processing cycle has emerged from previous studies: (1) First, to declare the "semantic ontology" of design objects based on the building program. (2) Second, to translate "semantic ontology" into appropriate "topological relations" of design objects in response to the design context. (3) Third, to interpret "topological relations" by "geometric propensities" of design objects in order to represent the chosen solutions. (4) Finally, to validate the required "semantic ontology" based on the given "geometric propensities." This perspective on the information-driven approach provides a basis for the following discussion.

4.1 Topological Knowledge within Design Cases

Topology is the representation of relations among design objects, and is therefore the key to parametric design in BIM. However, because of technical difficulties and the lack of consensus in the architectural design domain, most BIM implementations overlook topological information in architectural applications. From a mathematical point of view, the complexity of topological information should increase exponentially with the number of design objects and topological definitions. This may make it difficult to implement a system, and the ill-defined problem in architectural topology may even be insoluble. In addition, there seems to be no consensus in the academic and practical communities concerning what topologies should be integrated into a CAAD system. Unsurprisingly, most practical workers do not apply topology-based tools such as generating or automatic spatial layout software [19]. And even if BIM can be expanded with more topological information in the future, the implementation of BIM may only be able to represent what and how the topologies of design objects are, rather than design concepts of why the topologies should be this way.

By extracting memories from past experiences or design cases, architects quote abstracted patterns to reduce the complexity of topological problems. This is clearly different from how a machine deals with topological problems. Therefore, how architects can retrieval topological patterns, which usually are deduced from semantic information in given design problems, should be the key to CBD and CBR. Since VAT has grafted semantic ontology to graphic topology, it should be more useful for designers wishing retrieve, learn, and apply topological knowledge within design cases.

4.2 Visual Representation of Topological Knowledge

A picture is said to be worth a thousand words. But most case libraries are implemented by applying database technology, and focus on the semantic features of design cases. Because of the technical difficulties entailed by unstructured information, knowledge representations within the visual and graphic information of design cases are usually ignored. However, the original visual media within a design case usually cannot explicitly represent topological knowledge, and textual tags are not the best means of retrieving visual topology.

Traditional methods of abstracting topological knowledge, such as topological matrices, graphs, bubble diagrams, and schematic layouts, can represent different degrees of abstraction for different purposes. However, as architects may deal with several different abstract levels of design information at one time, including mass sketches, schematic diagrams, and details of components, the one-time abstraction of topologies often cannot satisfy the needs of architects, and is not helpful to novices wishing to learn relevant knowledge. Since VAT can simultaneously represent topological information by matrices, graphs, and schematic layouts, and allows users to exclude unnecessary information from view, it should be easier for users to learn and use than systems presenting overly abstract mathematics or more primitive visual media.

4.3 Interface for Encoding and Retrieval of Topological Knowledge

While one of the major purposes of developing a case library is to provide learning resources, some scholars claim that important learning functions are absent from CBD systems proposed in previous studies, and note that most such systems have no functions for acquiring and re-indexing design knowledge from cases [10]. To perform acquisition and re-indexing functions, it is necessary to have a user-friendly interface encouraging users to participate in acquisition actions, especially when acquiring unstructured information.

Regardless of whether textual tags or other visual approaches are employed, preprocessing is necessary to convert unstructured information into a machine-processable format. Since it is still difficult for machines to automatically recognize topologies within unstructured design case information, VAT provides a simple but intuitive interface assisting users in encoding and retrieving their perceptions. VAT can not only provide the results of analysis by users, which is performed through associations with semantic ontology, but also provide more clues about correlations among different cases in the library.

5 CONCLUSIONS

Since unstructured visual media can more easily represent architectural topologies, it is necessary to develop a visual tool in order to solve the problem of encoding, indexing, and retrieval of architectural topology meta-knowledge within a case library. This tool should provide necessary architectural topology functions, which can bridge the semantic and geometric information of design objects, and assist users in encoding and representing information conversion meta-knowledge among semantic, topological, and geometric forms of information.

The VAT project in this paper illustrates our methods of improving the representation of topological knowledge within the OCS case library. The recognition results of users applying VAT provide a visual language for communication between different agents in the system, including human designers and reasoning machines. VAT not only encompasses more topological knowledge within the design case library than spatial topologies of plan drawings in SSO, but also provides a foundation for development of the next generation of design assistance tools, which should be based on the conversion of the three types of design information used in BIM systems.

ACKNOWLEDGEMENT

The National Science Council of Taiwan supports this paper under the grant number NSC 101-2221-E-165-004.

REFERENCES

[1] MongoDB: Aglie and Scalabe, http://www.mongodb.org/, 10gen, Inc.
[2] Boling, E.: The Need for Design Cases: Disseminating Design Knowledge, International Journal of Designs for Learning, 1(1), 2010, 1-8.
[3] Brennan, J.; Martin, E.A.; Kim, M.: Developing an ontology of spatial relations, in: J.S. Gero, B. Tversky, T. Knight (Eds.) 3rd International Conference on Visual and Spatial Reasoning in Design, Key Centre of Design Computing and Cognition, Cambridge, USA, 2004, 163-182.
[4] Dave, B.; Schmitt, G.; Faltings, B.; Smith, I.: Case-based design in Architecture, in: J.S. Gero, F. Sudweeks (Eds.) Artificial Intelligence in Design '94, 1994, 145-162.
[5] Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K.: BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd ed., John Wiley & Sons Inc, Hoboken, N.J., 2011.
[6] Eastman, C.M.: Building Product Models: Computer Environments, Supporting Design and Construction, CRC Press, Boca Raton, Fla., 1999.
[7] Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, New York, 2003.
[8] Fry, B.; Reas, C., Processing.js - a port of the Processing Visualization Language, http://processingjs.org/.
[9] Gruber, T.R.: A translation approach to portable ontology specifications, Knowledge Acquisition, 5, 1993, 199-200.
[10] Heylighen, A.; Neuckermans, H.: A case base of Case-Based Design tools for architecture, Computer-Aided Design, 33(14), 2001, 1111-1122.
[11] Holzer, D.: BIM's Seven Deadly Sins, International Journal of Architectural Computing, 8(4), 2011, 463-480.

[12] Hwang, J.-E.; Choi, J.-W.: SpaceScope: Spatial Content-Based Retrieval of Architectural Floor Plans, in: Digital Design - Research and Practice - Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures, Tainan, Taiwan, 2003, 53-62.
[13] Langenhan, C.; Weber, M.; Petzold, F.; Liwicki, M.; Dengel, A.: Sketch-based Methods for Researching Building Layouts through the Semantic Fingerprint of Architecture, in: Proceedings of the CAADfutures 2011, Liege, 2011, 85-102.
[14] Lanzenberger, M.; Sampson, J.J.; Rester, M.; Naudet, Y.; Latour, T.: Visual ontology alignment for knowledge sharing and reuse, Journal of Knowledge Management, 12(6), 2008, 102-120.
[15] Lin, C.-J.: Spatial Layout Game-An Interactive Tool for Spatial Layout of Architectural Design, Journal of Architecture(Taiwan), 68s, 2009, 71-88.
[16] Lin, C.-J.: Smart Spatial Ontology: Bridging Semantic Ontology to Spatial Topology, Computer-Aided Design and Applications, 10(3), 2013, 489-497.
[17] Lin, C.-J.; Chiu, M.-L.: Open Case Study, in: CAADRIA 2009, Yunlin, Taiwan, 2009, 393-399.
[18] Lin, C.-J.; Chiu, M.-L.: Spatial topology retrieval: a visual approach for representing and retrieving spatial topology in a case library, in: CAADRIA 2010, Hong Kong, 2010, 147-154.
[19] Lobos, D.; Donath, D.: The problem of space layout in architecture: A survey and reflections, Arquitetura Revista, 6(2), 2010, 136-161.
[20] Oh, Y.; Do, E.; Gross, M.: Critiquing freehand sketching: A computational tool for design evaluation, in: J.S. Gero, B. Tversky, T. Knight (Eds.) 3rd International Conference on Visual and Spatial Reasoning in Design, Cambridge, USA, 2004, 105-120.
[21] Reas, C.; Fry, B.: Processing: A Programming Handbook for Visual Designers and Artists, MIT Press, Cambridge, 2007.