www.cadanda.com Computer-Aided Design AND APPLICATIONS

Architectural Knowledge Modeling: Ontology-Based Modeling of Architectural Topology with the Assistance of an Architectural Case Library

Chieh-Jen Lin

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

ABSTRACT

This paper aims to develop a knowledge-representation tool entitled "Architectural Knowledge Modeling" (AKM) for the purpose of encoding architectural design concepts with the assistance of a house case library. By applying previous results, AKM aims to establish formal representations of design objects and their topologies with the assistance of unpacking and sharing architects' design knowledge in decision-making processes.

Keywords: building information modeling, semantic ontology, architectural topology, design knowledge modeling, architectural case library.

1. INTRODUCTION

Building information modeling (BIM) applications have gradually replaced CAD software as a means of solving complex information integration problems in different disciplines in the architecture, engineering, and construction (AEC) industry. As communication platforms for design knowledge, however, most BIM software cannot achieve formal representation of architectural design knowledge, especially with regard to differences from engineering and construction. In the traditional design-bid-build (DBD) process, architects package knowledge within drawings, which isolates architects' contributions from those of engineers, fabricators, and constructors [5]. Once the knowledge expressed in drawings is lost after being converted to BIM, it is very difficult for architects to protect their contributions in the decision-making process. When architects cannot explicitly share and validate their knowledge during their negotiations with other disciplines, the BIM-based integrated project delivery (IPD) process, which has been promoted by BIM providers as a means of saving costs, inevitably causes architects to be marginalized [10].

Based on the initial proposal of Eastman [6], BIM should be composed of three types of design information: semantic, topological, and geometric. In BIM, topology consists of the mathematical connections between components, along with the functional definitions of parametric modeling [5], and is the key to the conversion of the three types of architectural design information [14]. Unfortunately, while the conversion and processing of design information is implicit and packaged within architects' drawings, topological definitions are usually ambiguous and vary among different architects. As a consequence, the objects and topologies needed for design computation, especially during the early and conceptual design stages, are usually ignored by current BIM applications, and this further impedes the architectural application of BIM.

This paper is a follow-up study to two previous projects, "Smart Spatial Ontology" (SSO) [13] and "Visual Architectural Topology" (VAT) [14], and aims to develop a knowledge-representation tool entitled "Architectural Knowledge Modeling," (AKM) which is used to encode architectural design concepts with the assistance of a house case library termed "Open Case Study" (OCS) [15]. By applying previous results, AKM aims to perform the formal representation of design objects and their topologies, which will assist the unpacking and sharing architects' design knowledge in decision-making processes.

2. THE APPROACH OF ARCHITECTURAL KNOWLEDGE MODELING

BIM is currently used in conceptual design is to reduce the level of details and parameters in models. By reducing the level of details (LOD), BIM can usefully reduce the recognition load during learning and operating. However, the aggregating hierarchies of components in most BIMs are still major obstacles in architectural applications. For example, spaces and zones are generative features of floors and walls in most BIMs, rather than the topological manipulations for generating or controlling the physical components needed in early conceptual design stages [1]. One potential method for explicitly representing design concepts is the use of algorithm-based parametric design tools, such as Rhinoceros' Grasshopper plugin, which can assist architects in exploring geometric possibilities through the visual composition of algorithms [12]. However, similar to complicated input in BIM, the visual compositions of algorithms in tools of this type are still too complex to be intuitively recognized and associated with abstract concepts. The gap between algorithms and design concepts is thus a disadvantage when applying algorithm-based tools as a medium for communicating with other stakeholders. For presenting implicit concepts within drawings and communicating with other stakeholders of a project, architects typically apply other visual media, which usually consist of a series of diagrams or precedents representing their design beliefs and intentions. Since these media will be separated from drawings, they cannot be directly converted into and validated in BIM.

Therefore, employing previous research results and knowledge extracting from the OCS library, such as the thesaurus and the semantic ontology of house cases [15], this paper proposes an ontology-based but moderately formalized tool that can assist architects in associating design concepts with case information and parametric algorithms. Rather than providing a rigid framework of an ontology [1], this approach is based on (1) the open aggregating hierarchies of semantic ontology, (2) graphic linkages among topological relations, and (3) the visual association of geometric features retrieved from design cases.

2.1. Open Aggregating Hierarchies of Semantic Ontology

Ontology is a knowledge engineering technique in artificial intelligence, as well is a data model facilitating the sharing and reuse of conceptualization in the development of knowledge-based systems. An ontology basically refers to a "formal, explicit specification of a shared conceptualization [8]," and implies that the content and structures of concepts in a knowledge domain are must be fixed and static to keep them correct and consistent in that system. But while this constraint of a fixed ontology may satisfy most domains, it is problematic in most areas of design. In architectural design, the conceptual specifications of similar projects typically vary with different design situations, including sites, clients, users, and budgets, let alone different architects who hold different theories and methodologies. For example, "row house" implies a building type in which every house unit is attached to others on two or three sides to form a row-shape building for fully efficient use of urban land. Every row house unit therefore has only a façade of windows for ventilation and natural illumination. As a result, one basic row house design criterion—that the façade of a row house should have some windows in it to provide good ventilation and natural illumination—can be represented as the following reasoning rule in Semantic Web Rule Language (SWRL)[8] (Eqn. 1):

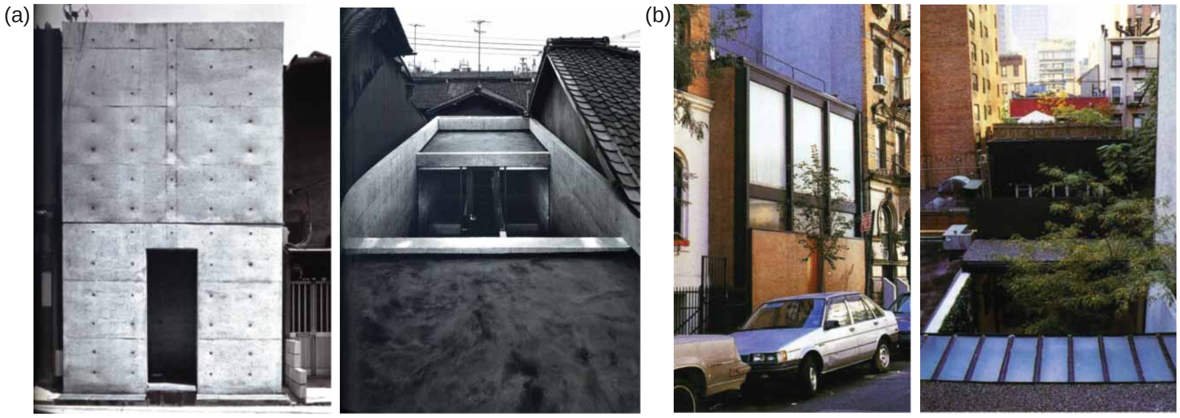
$$\begin{aligned}&\mathrm{RowHouse}(?x) \wedge \mathrm{Façade}(?y) \wedge \mathrm{Window}(?z)\\&\quad \wedge \mathrm{hasFaçade}(?x, ?y) \wedge \mathrm{hasWindow}(?y, ?z) -> \\&\quad \mathrm{hasVentilation}(?x, \text{“Good”})\\&\quad \wedge \mathrm{hasDayLighting}(?x, \text{“Good”})\end{aligned} \tag{1}$$

Although ventilation and natural illumination are very basic requirements for house design, some architects have nevertheless tried to challenge these essential criteria. For example, the "Azuma House," which was an award-winning masterpiece of the Japanese architect Tadao Ando, is a small row house with no windows on its façade (Fig. 1.a). A similar case is the "Rockefeller Guest House," which is a masterpiece by the American architect Philip Johnson, and has only fixed glass curtains on the façade for natural lighting, but no ventilation (Fig. 1b). Tadao Ando declared his intention in designedly making the Azuma House to be an uncomfortable residence for criticizing the indolent tendencies of modernist house design, but he still provided a courtyard allowing clients to contact with natural elements, such as sunshine, rain, and wind. Although the small courtyard can provide the Azuma House with basic ventilation and natural lighting, this house nevertheless violates many cardinal design criteria, while nevertheless revealing the influence of Minimalism. It is desired to assign these two cases "positive" properties complying with known design criteria, a design instructor may assign the rule for the Azuma House that a façade without any window can provide a high level of privacy (Eqn. 2), and an architect can assign the rule for the Rockefeller Guest House that a façade without ventilation can provide higher noise insulation (Eqn. 3).

$$\begin{aligned}&\mathrm{RowHouse}(?x) \wedge \mathrm{Façade}(?y) \wedge \mathrm{hasFaçade}(?x, ?y)\\&\quad \wedge \mathrm{hasNoWindow}(?y) -> \mathrm{hasPrivacy}(?x, \text{“High”})\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathrm{RowHouse}(?x) \wedge \mathrm{Façade}(?y) \wedge \mathrm{Window}(?z)\\&\quad \wedge \mathrm{hasFaçade}(?x, ?y) \wedge \mathrm{hasWindow}(?y, ?z)\\&\quad \wedge \mathrm{hasNoVentilation}(?z) -> \\&\quad \mathrm{hasNoiseInsulation}(?x, \text{“High”})\end{aligned} \tag{3}$$

Since row houses are usually located in high-density urban contexts, with the advancement of building technology, privacy, and noise may therefore be more important than basic ventilation and natural lighting in some design situations. However, the foregoing two rules may violate the basic axiom of



Azuma House by Tadao Ando [3]. Rockefeller Guest House By Philip Johnson [19].

Fig. 1: Two unique row house precedents are collected in the OCS library.

design criteria, which asserts a good house design should have good ventilation and natural illumination (Eqn. 4). An artificial intelligent reasoner, such as FaCT++ and HermiT built in Protégé 4.x[11], cannot automatically classify Azuma House or Rockefeller Guest House as an instance of the good house design case. Even though users can manually assert Azuma House or Rockefeller Guest House as a "GoodHouseDesign" instance, this assertion will make the semantic ontology of design criteria become inconsistency, and cause the artificial intelligent reasoner to crash.

$$\begin{aligned} \text{House}(?x) &\wedge \text{hasVentilation}(?x, \text{"Good"}) \\ &\wedge \text{hasDayLighting}(?x, \text{"Good"}) \\ &= \text{GoodHouseDesign}(?x) \end{aligned} \tag{4}$$

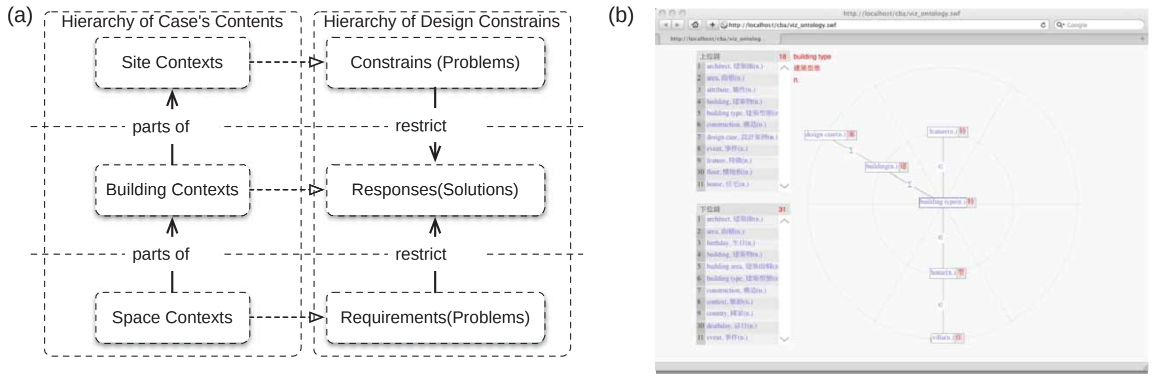
In architectural design practice, the completion and consistency of design criteria are usually not the primary preconditions for an architect's winning a design competition, and architects sometimes are encouraged to challenge known criteria in order to achieve innovation in specific design situations. Architects must often present creative and innovative ideas to their clients if they are to win out among other proposals, which explains the significance of the Azuma House's privacy rule and the Rockefeller Guest House's noise insulation rule.

To assist architects in representing specifications corresponding to a project's requirements and the architects' conceptual intentions, such as privacy in an urban context, this paper therefore proposes an authoring tool employing a partial semantic ontology for house design objects. Unlike other approaches, such as Protégé, which typically focus on the correctness and consistency of an ontology, AKM provides rapid authoring assistance by retrieving and reusing partial ontologies from the OCS library, and allows the temporary use of incomplete and inconsistent ideas, which allows users to explore the possibilities of their concepts in further topological compositions.

With the help of open and flexible aggregating hierarchies, AKM aims to help architects to quickly define and then manipulate conceptual design objects before composing the topological relations of those conceptual objects in order to validate proposed conceptual goals. Based on previous results, the ontology of OCS classifies the conceptual features of house design issues into three levels: (1) site context, (2) building context, and (3) spatial context (Fig. 2.a). For example, users can retrieve partial ontologies of house design issues from OCS, such as privacy issues concerning the urban site context associated with Azuma House, the building form issue of Azuma House associated with minimalist style, and the spatial context issue of Azuma House associated with minimum circulation of spatial functions. The user can then define objects and their conceptual properties (Fig. 2.b), such as the rule of the "hasNoWindow" property implying "hasPrivacy('High')" or the "hasNoVentilation" property implying "hasNoiseInsulation('Good)". The defined ontological classes and properties will become manipulatable objects and their parameters in further topological operations.

2.2. Graphic Linkages of Topological Relations

The ontology of design criteria can only represent semantic descriptions of architectural design knowledge. However, topological relations, such as adjacency, overlapping, separation, and inclusion of spaces, which are the critical issues of data modeling in an geographic information system (GIS) [16], cannot easily be obtained from reasoning axioms or rules of semantic ontology. An artificial intelligent reasoner usually cannot easily determine topological relations based on a simple semantic description. For example, even though a reasoner knows the semantic descriptions of the topological relations of x and y to all other atoms, the reasoner still cannot easily determine



Three levels of design contexts

Visual indexing of conceptual classes of design criteria [13]

Fig. 2: The basic aggregating hierarchies of semantic ontology in OCS.

whether x is adjacent to y without knowledge of their geometric properties. Topology is the mathematical connection of building components in BIM, and one of the fundamental definitions embedded in parametric modeling [5]. Consequently, topology becomes a priori knowledge embedded in building components in BIM, and topological operations are thus ignored and cannot be freely manipulated by architects. Since acceptable topological relation between basic building components, such as doors, windows, walls, and columns, are defined in most BIM applications, BIM can detect an error when a window on a wall overlaps another connecting wall (Fig. 3). However, some obvious errors may be ignored by BIMs when relevant components are not properly connecting. Fig. 4 shows that Autodesk Revit does not identify an error when the left door has exceeded the range of a wall and has overlapped other two walls.

Architectural topologies at the conceptual design stage are usually still ambiguous, especially in the case of abstract objects such as spaces, circulations, axes, views of opening, and other custom objects, and therefore cannot easily be converted into parameters and algorithms. For example, an architect does not need to employ computational fluid dynamics (CFD) software to know that two windows of a room should be installed on opposite walls, and face each other, in order to have better ventilation (Fig. 4.a) than two window installed on different walls of a room (Fig. 4.b), while windows installed on same side of a wall will have the worst ventilation (Fig. 4.c). This knowledge can be imparted easily through such semantic rules as Eqn. 5, but it does not make it easy to rapidly implement a recognition algorithm or rules based on the windows' semantic or geometric properties.

$$\begin{aligned}&\mathrm{Room}(?x) \wedge \mathrm{Window}(?y) \wedge \mathrm{Window}(?z) \wedge \mathrm{Wall}(?a)\\&\wedge \mathrm{Wall}(?b) \wedge \mathrm{hasWall}(?x, ?a) \wedge \mathrm{hasWall}(?x, ?b)\\&\wedge \mathrm{hasWindow}(?a, ?y) \wedge \mathrm{hasWindow}(?b, ?z)\\&\wedge \mathrm{DifferentFrom}(?a, ?b) \wedge \mathrm{DifferentFrom}(?a, ?b)\end{aligned}$$

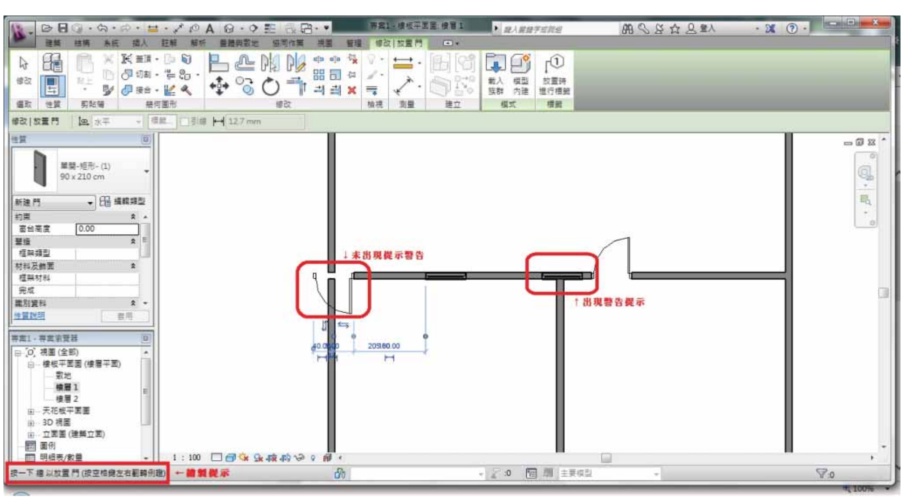
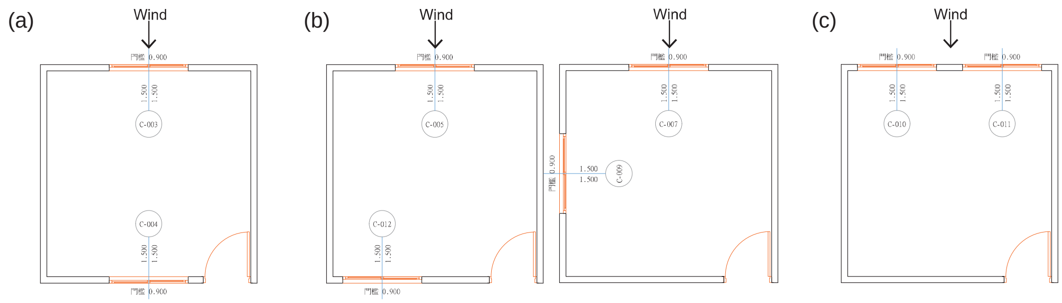


Fig. 3: Autodesk Revit displays an error in the case of the right window, but ignores the overlap of the left door.



Better ventilation Acceptable ventilation Poor ventilation

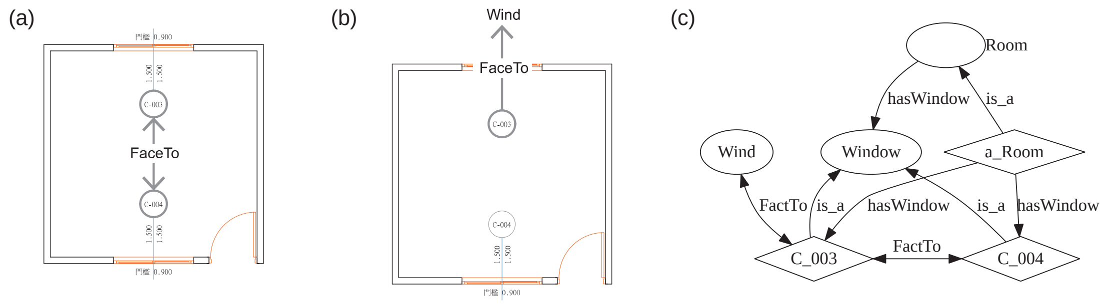
Fig. 4: Installation of two windows in a room yielding different levels of ventilation.

$$\wedge \ \mathrm{FaceTo}(?y.?z) \wedge (\mathrm{FaceTo}(?y, \mathrm{Wind}) \vee \mathrm{FaceTo}(?z, \mathrm{Wind})) -> \mathrm{hasVentilation}(?x, \text{"Better"}) \quad (5)$$

It is easy for an architect to recognize and identify the topological relations of the building components shown above (Eqn. 5). However, it is not so easy for an architect to implement a "FactTo" algorithm to determine whether a window faces to the direction of wind, or faces another window on the opposite side of a wall. Modern graphic algorithm implementation tools, such as the Grasshopper plugin of Rhino, can reduce the learning and recognition load of designers. However, since a programmer usually cannot understand a program written by himself without pre-written annotations, users often forget why and how they originally composed algorithmic components in Grasshopper. Even determining a simple adjacency relationship between two objects may require complex algorithms that depend on the geometric types and parameters of the objects in question, let alone complex topologies involving multiple objects such as enclosure, aggregating, concentration, or deployment [9]. On the other hand, since a single architect may vary his/her design concepts in different situations of a same project, implementing reasoning algorithms for every re-emerging concept may be not a cost-effective strategy. Instead of implementing a complex reasoning algorithm, the assignment of reasoning rules based on the semantic relations of building components, and then reasoning whether an asserted ontology of design criteria is consistent or not using a reliable existing artificial intelligent reasoner, such as FaCT++ or HermiT in Protégé, should be a more economical and effective strategy. Since, for humans, semantic ontological annotations are more easily read than complex composing components in Grasshopper, it may therefore be more helpful for architects to associate abstract topological relations with assigned design criteria, and then to implement the actual reasoning algorithm if it becomes necessary.

To assist architects in associating topological relations of design objects and the criteria that design solutions must comply with, this paper proposes an ontological means for the visualization of assigned topologies. Since a knowledge chunk in an ontology can be represented by the three-fold set of a "subject," "predicate," and "object" [7], topological relations linking redefined objects can play the roles of "predicate" of objects' relations, which is referred to as an "object property" in the Web Ontology Language (OWL) in Protégé[11], and will become a "blank" component of inferring algorithm waiting for user to implement in the latter. For example, an architect could assert a symmetric "FaceTo" property between two instances of "Window" on a plan drawing (Fig. 5.a). The symmetric property would imply two windows facing to each other, and another "FaceTo (Wind)" property to one of the two instances (Fig. 5.b). Therefore AKM would generate a partial ontology of two instances of "Window" class and their "FaceTo" properties to "Wind" class (Fig. 5.c). Based on those properties, AKM could infer the "a_Room" instance of "Room" class to be classified into "hasVentilation('Better')."

Since topological properties in an AKM prototype can be assigned by users, unlike Grasshopper's components, most topological properties assigned by users therefore have no actual computational abilities, and cannot automatically generate possible solutions, except when the simple adjacent topology of rectangular objects is inherited from VAT. However, by exporting an ontology of asserted design criteria into the OWL format, AKM still could validate whether consistency of an asserted criterion was met or not with the help of Protégé's or other logical reasoners. On the other hand, in the same way, which pseudo code was used in programming could help humans understand the algorithmic procedures. The semantic tags of topological properties of building components could help users communicate with others and implement the inferring algorithm for the validation of asserted topological properties in the latter.



"FaceTo" property between two windows.

"FaceTo(Wind)" property of a window.

Partial ontology of two instances of "Window"

Fig. 5: Graphic linkages of topological relations are asserted by user.

2.3. Visual Association of Geometric Features from Design Cases

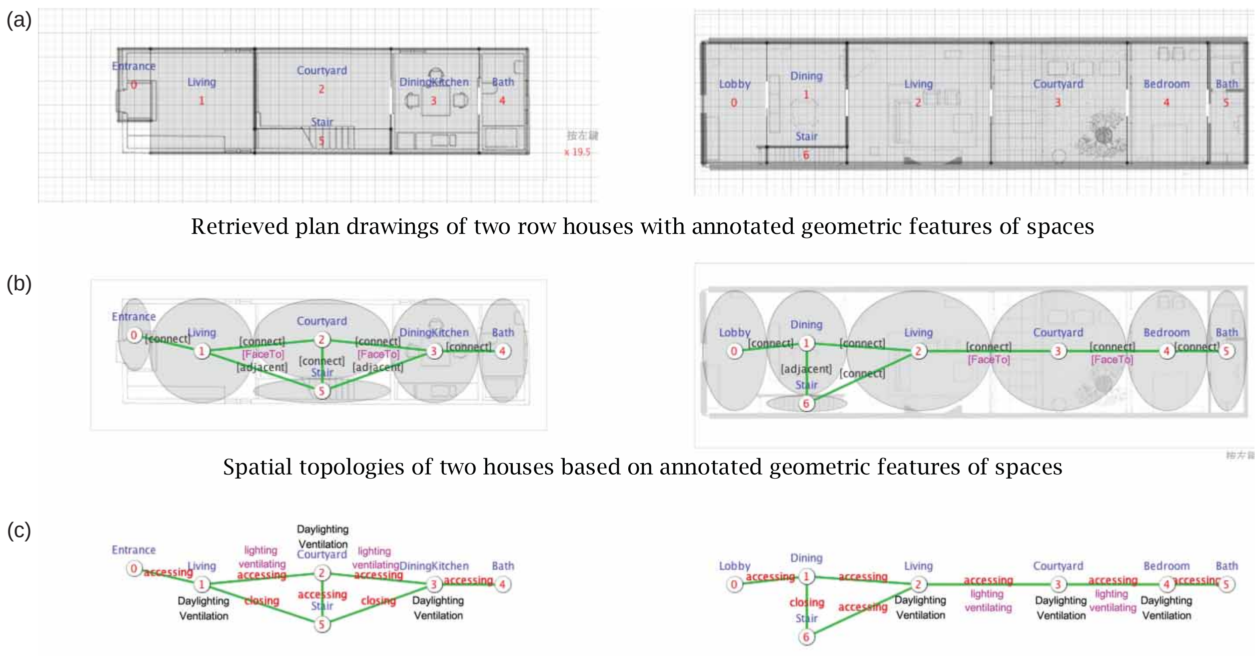
One reason for Grasshopper's popularity among architects is that it can provide real-time feedback concerning the interactions of parametric values and generated geometric forms. Generated geometric forms can not only provide visual validation of algorithms, but also provide geometric alternatives of acceptable design solutions. As mentioned above, however, the task of converting topological relations into composing algorithms was often complicated and time-consuming. Unless validated algorithms were available to generate geometric forms or solutions in response to specific emerging topology, the rapid compositions of algorithms for asserting arbitrary topology were obviously difficult. Furthermore, it might not be economically-effectively to develop algorithms before validating any conceptual topologies of design criteria, or that they could be accepted by clients and other relevant stakeholders. Traditionally, an architect tends to sketch diagrams or to cite persuasive precedents while quickly responding to specific design issues, and validating conceptual topologies for those issues. Those diagrams provided criteria for evaluating and validating the inferring procedures of asserted topologies, while parts of persuasive precedents often provided acceptable geometric solutions of conceptual topology for responding to design criteria.

Instead of implementing generative algorithms of geometric forms based on asserted topologies, as a prototype knowledge-modeling tool, AKM provided visual associations with the geometric features of relevant cases in OCS library. By applying the previous results of VAT, such as textual tags and geometric features attached to non-textual media in house design cases, including spatial forms, their topologies, and other graphic annotations attached by users to drawings or photographs in OCS, AKM retrieves and associates relevant visual media from the OCS library to help users to represent the acceptable geometric outcomes of architectural topologies. For example, when a user searched with keywords, such as "RowHouse," "UrbanContext," "Privacy," or "Ventilation" issues in OSC, there would come up with relevant cases such as "Azuma House" and "Rockefeller Guest House" in the library (Fig. 6.a). However, AKM not only could it retrieve case media from the OCS library with the help of VAT visual interface, which allowed users to attach and retrieve graphic topological annotations to those media, but also reveal the associated design criteria, inferred from the topological relations of graphic annotations (Fig. 6.b). For example, AKM could therefore infer "hasDaylighting ('Livingroom')" and "lighting ('Courtyard', 'Living')" properties from two asserted properties: "hasDaylighting ('Courtyard')" and FaceTo('Courtyard', 'Living')" (Fig. 6.c), based on the inferring rules as defined in the ontology of design criteria in AKM.

The graphic annotations of topologies and their geometric features, such as spatial dimensions and relative positions, could easily assist architects in figuring out bases of their ideas about retrieved cases such as privacy issue of Azuma House and noise issue of Rockefeller Guest House. Subsequently, the retrieved media of cases allowed architects to reuse or demonstrate how their ideas could be accomplished. An example was that the courtyard in both cases could solve the conflicts between client's privacy and the requirements for basic ventilation and day lighting. Therefore, even though AKM still could not infer the geometric solutions of design criteria in the meanwhile, the graphic annotations and their geometric features of retrieved case could provide rapid solutions for asserting design criteria.

2.4. The Implementation of an AKM Prototype

The AKM prototype was developed on the foundation of previous results, which included: (1) a web-based application server based on the Apache-MySQL-PHP stacks for accessing OCS data over the Internet; (2) the MogoDB for storing flexible hierarchies of semantic ontology and architectural topologies; and (3) Processing.js for implementing a JavaScript-based visual interface for interactively authoring



Retrieved plan drawings of two row houses with annotated geometric features of spaces

Spatial topologies of two houses based on annotated geometric features of spaces

Partial ontology of design criteria inferred from spatial topology of two houses.

Fig. 6: Visual association of geometric features of “Azuma House (Left)” and “Rockefeller Guest House (Right)” retrieved from the OCS library.

semantic ontologies, modifying topological relations, and retrieving visual media from the OCS library.

The initial version of OCS consisted of a web-based application applying the MySQL database, and using PHP to implement the interface. However, relational database management systems (RDBMS) such as MySQL require predefined data schema, which can reduce flexibility when users wish to reinterpret the content of existing design cases. A “Not only SQL” (NoSQL) database, such as the MongoDB applied in WebProtégé for storing OWL files, which does not need predefined schemas and stores data via key-value pairs, is better at storing open and flexible hierarchies of semantic ontology, topological relations, and other custom-designed information. Furthermore, in keeping with emerging modern web technologies, AKM implements its interface via the application of Processing.js, which is a JavaScript version of the PROCESSING visual programming language. Processing.js helped us to convert the old Java-based PROCESSING codes of SSO so that they would work on modern Java-incompatible web browsers.

2.5. Initial Evaluation of AKM

Although experienced architects can manipulate design concepts using pencil and paper alone, they still need appropriate media to communicate their ideas with other project stakeholders in order to obtain agreement. Unfortunately, most of the analog or digital media used by architects to represent their beliefs and intentions, such as text, sketches, diagrams, and PowerPoint files, cannot be directly validated by machines, and cannot be easily converted into operable components in BIM. This is why our students cannot easily find obvious contradictions in their proposed design concepts, and then lose track of their previous beliefs and intentions when they begin to convert their abstract concepts into the concrete geometric forms and physical components of a building. It was found that with the help of AKM, students were able to trace the evolving process of their designing beliefs and intentions, and clearly present their initially-vague ideas in machine-processable format. They did not need to struggle with ontology authoring software like Protégé. Via the assistance of semantic ontology, design course teachers could easily present their criteria about how to assess students’ work, and then to guide students to recognize their intentions, and improve or correct their vague concepts in convincing proposals. However, some teachers might refuse to clearly express their criteria for not restricting students’ creativity.

Based on the SSO ontology, the VAT graphic annotation interface, and visual information concerning design cases in the OCS library, the AKM project aimed to develop an easy tool for architects to present their knowledge of design criteria. AKM improved the open interpretive tool for ontology and topology in OSC as a handleable communicating media. It was able to present, explain, and validate architects’ design beliefs and intentions. Through the semantics of the SSO ontology, VAT’s visual annotations, and associations among rich media in OCS, using AKM, architects could communicate with developers who wish to apply algorithm-based tools for geometric exploration, and with co-workers in other AEC disciplines who use BIM or other tools in detailed design and development.

3. DISCUSSIONS

Based on the information-driven approach emerging from previous studies, architectural design at the conceptual stage can also be regarded as the interconversion of three types of design information in order to perform the following tasks: (1) To acquire the "semantic ontology" of design criteria from client's requirements or the specifications of a building project. (2) To compose appropriate "topological relations" among design objects in response the semantic ontology. (3) To generate the geometric properties of a design solution in order to represent the topological relations. (4) And finally to validate the generated geometric solutions on the basis of the proposed semantic ontology. This information-driven approach provides the basis for the following discussion.

3.1. Semantic Ontology of Architectural Design Objects

With the rapid development of 3D visualization via BIM, scholars have predicted that once an architect can construct virtual simulations of a building, he/she would no longer need to compose abstract 2D drawings [2], such as plan and section drawings, in order to express design criteria. However, this prediction remains as far from fulfillment as paperless offices, or the replacement of traditional books with e-books. To date, the use of 2D CAD and 3D simulation has not reduced the amount of paper used in architectural design offices, and the heavy computational requirements of BIM's simulations make considerable demands on hardware and software. Furthermore, the complexity of BIM operations often forces students and employees to work long hours, and makes teachers and architects to long for the use of pencil drawings on tracing paper. Yet another obstacle to the application of BIM in conceptual design is that the representations of 2D drawings to simulate final building modeling results must not be too abstract, and there are no sufficiently abstract representations to express the criteria embodied in architects' proposals. No wonder some BIMs will require the further development of abstract design objects, such as the "mass" object in Autodesk Revit, or "space" object in VirtualBuilder GongBuilder [17], if they are to be used effectively for developing design criteria in conceptual design stage.

Abstraction is a critical skill in architectural design, and essential if architects wish to communicate with project stakeholders, especially clients and users who have no background in design, engineering, or construction. The task of an architect is not only to propose reasonable solutions for a building, but also is to raise appropriate issues concerning the project in order to elicit responses. In the era of 2D analog media and no computers, traditional design education trained architects how to package their design concepts in the form of 2D architectural drawings, and then to employ sketches to unpack their concepts. This separation and the necessary translation of drawing symbols caused the overly-abstract arguments in 2D drawings to constitute a jargon accessible only to professionals. However, since architects' drawings and diagrams actually constitute a visual and graphic system of linguistic representation, it is therefore possible and necessary to represent these drawings via semantic ontologies, which can facilitate communication between human and machine. At a time when building projects are becoming increasingly complex, the requirements for abstracting features of a project have also become more critical. For example, while the classification and aggregation of spatial functions in order to allocate floor areas or building mass can be relatively easily represented by semantic information, architects cannot readily check whether 2D drawings or 3D models meet predefined criteria, let alone when design criteria may still be developing during the conceptual design stage. An authoring tool involving semantic ontology, such as AKM, will therefore be useful in rapidly representing, validating, and sharing architects' beliefs and ideas with other stakeholders.

3.2. Composition of Architectural Topology

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 not only ignore the abstraction of design objects, but also overlook topological information concerning those abstracted objects in architectural applications. Although an ontology of spatial topologies has been proposed [4], how to compute and validate complex topological relations still faces technological challenges. Ill-defined problems concerning abstract architectural design objects require an authoring tool employing a semantic ontology, and the manipulations of architectural topologies need an open tool allowing architects to define their own computational and validating procedures, rather than relying on predefined parameters. It is clear, however, that the task of actually programming architectural topology computation and validation procedures usually exceeds the abilities of architects.

Rhinoceros' Grasshopper plugin demonstrates a relatively ideal method by which designers can visually and easily compose algorithms for generating complex geometric forms. When designers try to explore new concepts that go beyond their past experiences, nevertheless, Grasshopper also reveals how difficult and time-consuming those tasks can be, which are similar to and by no means easier than composing 3D building components in BIM. Inspired by Unified Modeling Language (UML), it seeks to model the requirements of software development employing a visual and uniform approach. Through graphic visualization of linking semantic ontology and architectural topology, AKM also attempts to model design criteria of architects' beliefs and intentions in response to clients' requirements,. Nevertheless since the blank inferring rule of "predicate" class connecting "object" and "subject" semantic ontological classes still has no actual programming code, except simple adjacent topology of rectangle shapes, AKM cannot yet automatically compute and validate topological relations, or generate geometric forms and modify the geometric features of design objects. However, it was found that the predicate class of topologies can facilitate communication with architects and others, and also can run as "pseudo code" allowing designers to identify necessary spatial topologies for developing real programming code by Grasshopper or other algorithmic tools in the latter.

3.3. Generating Geometric Building Forms

Although algorithm-based generating tools such as Grasshopper are increasingly popular among educators and practical workers, their ability to validate whether generating geometric forms are consistent with proposed design criteria still remains doubtful. As algorithmic-based generating tools often face the problem of "garbage-in garbage-out," the generation of geometric forms by algorithms cannot provide a firm basis for acceptance by clients or other stakeholders. Debates among stakeholders commonly involve aesthetic issues, as well as the accompanying construction and funding issues. Even in architectural education, it is becoming more difficult for teachers to assess the results when students apply algorithm-based tools to present his/her concepts. The situation is even more daunting for practicing architects who must constantly communicate with clients, other disciplines, and their own employees. Designers may nevertheless input sketch or images into Grasshopper as visual annotations while seeking to compose algorithms, and images retrieved from a remarkable case may be worth a thousand words.

Since it will require more investigation to determine how an architect interprets an abstract concept as a concrete geometric forms, such via the conversion of a spatial function or a cultural image into a specific geometric form, rather than a rectangle box, the prototype AKM inherits a simple rectangular reorientation of geometries, and does not yet implement more generative geometric algorithm. We do not seek to reinvent a Grasshopper-like generating tool for now, and it should be easier and more efficient to integrate AKM into an algorithm-based generating tool. The visualization in AKM can provide a basis for future integration into Grasshopper or other algorithmic tools. However, as an alternative for representing geometric consensus for the sake of communications, AKM can retrieve relevant design case information from the OCS library in order to associate geometric features within retrieved cases and architectural topology, and architectural topologies can also be used as a reverse index of relevant media allowing other users to subsequently retrieve cases and their concepts.

4. CONCLUSIONS

The MacLeamy curve [5], which provides good publicity for the BIM concept, was shown at the 2007 CURT by the CEO of HOK, and reveals that earlier decisions in the design process have a greater impact on the quality and cost of a project. And of course architects seek to win the design rights to projects or be invited to join an IPD team based on their creative insights. While BIM has gradually become a communication platform for the AEC industry, applying BIM to represent creative design concepts is often not only be costly and time-consuming for architects, but also makes it harder for architects to protect their painstaking work. It is therefore unsurprising that algorithm-based modeling tools like Grasshopper have become popular with architects, who are willing to pay the cost of converting conceptual models into BIMs in order to safeguard their contributions. Although algorithm-based modeling tools are useful for exploring complex and elaborate geometric forms, other forms of conceptual knowledge, which clients and other disciplines can understand and accept, are also essential for architects who wish to survive in their practice.

The AKM project in this paper illustrates our approach to improving representation of design knowledge through associations with the OCS library. Modeling results of design criteria obtained using AKM enable visual communications among stakeholders, and can play an assisting role during the early design stages. Before an architect can devote himself to the algorithmic composition of attractive form or parametric modeling in BIM, it is necessary to represent and validate conceptual knowledge involving design criteria [18]. AKM can not only add critical architectural topology absent in BIM, but also provide a foundation for development of the next generation of design assistance tools, which can associate abstracted design criteria, and their semantic ontology and topology with algorithmic compositions and parametric modeling of BIM.

ACKNOWLEDGEMENT

This work was supported by The National Science Council of Taiwan under the grant number [NSC 102-2221-E-165-002].

REFERENCES

[1] Aksamija, A.; Iordanova, I.: Computational Environments with Multimodal Representations of Architectural Design Knowledge, International Journal of Architectural Computing, 8(4), 2010, 439–460. http://dx.doi.org/10.1260/1478-0771.8.4.439

[2] Ambrose, M.A.: Agent Provocateur – BIM In The Academic Design Studio, International Journal of Architectural Computing, 10(1), 2012, 53–66. http://dx.doi.org/10.1260/1478-0771.10.1.53

[3] Ando, T.: Tadao Ando 1: Houses & Housing, Toto Shuppan, Tokyo, 2007.

[4] 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.

[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] Gruber, T.R.: A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, 5(2), 1993, 199–220. http://dx.doi.org/10.1006/knac.1993.1008

[9] Ho, H.-Y.; Wang, M.-H.: Meta Form as a Parametric Design Language, in: eCAADe 2009, Istanbul, Turkey, 2009, 713–718.

[10] Holzer, D.: BIM's Seven Deadly Sins, International Journal of Architectural Computing, 9(4), 2011, 463–480. http://dx.doi.org/10.1260/1478-0771.9.4.463

[11] Horridge, M.: A Practical Guide To Building OWL Ontologies Using Prot ´eg ´e 4 and CO-ODE Tools, 1.3 ed., University Of Manchester, 2011.

[12] Leitão, A.; Santos, L.; Lopes, J.: Programming Languages For Generative Design: A Comparative Study, International Journal of Architectural Computing, 10(1), 2012, 139–162. http://dx.doi.org/10.1260/1478-0771.10.1.139

[13] Lin, C.-J.: Smart Spatial Ontology: Bridging Semantic Ontology to Spatial Topology, Computer-Aided Design and Applications, 10(3), 2013, 489–497. http://dx.doi.org/10.3722/cadaps.2013.489-497

[14] Lin, C.-J.: Visual Architectural Topology: An Ontology-Based Topological Tool for Use in an Architectural Case Library, Computer-Aided Design and Applications, 10(6), 2013, 929–937. http://dx.doi.org/10.3722/cadaps.2013.929-937

[15] Lin, C.-J.; Chiu, M.-L.: Open Case Study, in: CAADRIA 2009, Yunlin, Taiwan, 2009, 393–399.

[16] Rigaux, P.; Scholl, M.O.; Voisard, A.: Spatial Databases: With Application to GIS, Morgan Kaufmann, 2002.

[17] Virtualbuilders, Gong Builder - 3D Unified Architectural Modeling Software, http://www.vbuilders.co.kr/eng/products_gongbuilder.html, Virtualbuilders Co. Ltd.

[18] Wong, J.F.: The text of free-form architecture: qualitative study of the discourse of four architects, Design Studies, 31, 2010, 237–267. http://dx.doi.org/10.1016/j.destud.2009.11.002

[19] Yoshifumi, N.: Jūtaku Junrei, Shinchosha, Tokyo, 2000.