Modeling Complex Architectures Based on Granular Computing on Ontology

Yong Liu, Yunliang Jiang, and Lican Huang

***Abstract*—We propose granular computing (GrC) on ontology as a solution to the problem of modeling complex architectures. We expressed the architectures formally as ontology domains, which include two components: the *set of basic vocabularies* and a *knowledge library of rules*. The set of basic vocabularies contains elements or basic architecture components. The knowledge library comprises rules that control the combination and construction of the basic elements. As the rules are often given by architectural experts subjectively, they may contain redundant, conflicting, and overlapping rules, especially in certain styles of ancient southeast Chinese architecture. It is difficult to distinguish or identify these rules; therefore, we apply the multilevel approach on ontology [Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "Smart architect: Scalable ontology-based modeling for ancient chinese architecture," *IEEE Intell. Syst.*, vol. 23, no. 1, pp. 49–56, Jan./Feb. 2008] and approximation theory of GrC. In this process, we present a measurement that is based on roughness functions to evaluate the degrees of approximation between the selected set and certain architecture domains. With the monotonicity characteristic of roughness functions, we can design a heuristic algorithm to select a suitable knowledge base (rule set) to assist in integrating the parts into final architectures, via several levels. Experiments with a real architectural project, i.e., modeling ancient southeast Chinese architectures, show that our method is effective and may simplify the design of the automodeling system and enhance its performance.**

***Index Terms*—Complex-architectures modeling, granules, hierarchical ontology design, roughness function.**

I. Introduction

To MANIPULATE knowledge indirectly to solve practical problems is the key idea of this paper. In many practical applications, knowledge often overlaps, and boundaries are vague. It is quite difficult to classify knowledge directly. However, the implementation of the knowledge instances may suggest clearer categories. Consequently, we can distinguish the knowledge by its implementations. Since this approach uses granular classes of instances to reflect and distinguish knowledge, we call it as "manipulating knowledge by instances" or "manipulating knowledge indirectly."

Based on such an idea, in this paper, we address the problem of modeling complex architectures that share a common style and structure, for example, ancient Chinese architecture, automatically and efficiently. Our automodeling system [2] features a knowledge library, which is called "U," which contains many general construction rules. We characterize the rules in U as a recursive-production grammar system, called the L-system [3], which is similar to first-order logic. As all of these rules are manually extracted by architects, most of them are empirical and imprecise. The knowledge library U may be highly redundant, which is a major drawback, especially when we generate a single style of architecture, for example, ancient Chinese or ancient Indian styles. The redundancy in U can be summarized as follows.

1) The rules in U may define several styles of architecture. These multiple styles of architecture are redundant for a specific architecture style. This is called *other-style redundancy*.
2) Some general rules should be shared by multiple architecture styles. For example, gate and wall combination rules may be the same in ancient-Chinese architecture and in ancient-Indian architecture. This is called *share redundancy*.
3) Architects may introduce some wrong rules erroneously. If the architecture generation includes these rules, this will yield incorrect results. This is called *wrong-rule redundancy*.
4) More than one rule may refer to the same combination or topological relationship for a specific style. This is called *repeating redundancy*.

The drawbacks of redundancy are as follows.

1) Incorrect rules in U produce unreasonable architectures, for example, a gate may lie upon the window, and the roof may lie below a column.
2) As the knowledge library contains rules for multiple styles and some rules are shared by different styles, the library generates "ununified" styles and unexpected structures. It will generate architectures that are hybrids of multiple styles, for example, some strange architectures may contain an Indian roof, while the house body is in Chinese style.
3) Redundancy and incorrect rules hinder the performance of the modeling system. In such a system, users must generate a large number of instances and can see only a few

Manuscript received July 14, 2009; revised November 24, 2009; accepted January 25, 2010. Date of publication February 22, 2010; date of current version May 25, 2010. This work was supported by the National Natural Science Foundation Project of China under Grant 60803053 and Grant 60872057, by the Science Foundation of Chinese University under Grant 2009QNA5004, and by the Natural Science Foundation Project of Zhejiang Province under Grant R1090244 and Grant Y1080212.

Y. Liu is with the Institute of Cyber-Systems and Control, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: cckaffe@gmail.com).

Y. Jiang is with the School of Information and Engineering, Huzhou Teachers College, Huzhou 313000, China (e-mail: jylsy@hutc.zj.cn).

L. Huang is with the Institute of Network and Distributed Computing, Zhejiang Sci-Tech University, Hangzhou 310018, China (e-mail: huang_lican@yahoo.co.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2010.2043848

satisfactory results. For example, in the worst case, if users use the raw knowledge library U to generate some ancient Chinese-style architectures, over half of the returns will be incorrect architecture instances due to the redundancy of knowledge library.

Fortunately, the architectonics provide a perfect categorization for distinct architectures or, more precisely, a natural granulation of the domain. Therefore, we can granulate (categorize) specific styles of rules.

We adopt the ontology and granular-computing (GrC) techniques. We describe the hierarchical architecture styles formally with ontologies. We characterize each style as a domain of ontology that contains both the instances and the constructing knowledge of the corresponding concept domain.

We use the upper and lower approximations to represent the maximal and minimal boundaries of the knowledge, they are subsets of U, with respect to the corresponding concept domain and then present a global critical measurement to obtain the optimal knowledge set from U with minimal roughness. We also introduce a heuristic-search algorithm that is based on the hierarchical ontology domain. This algorithm can significantly increase the efficiency to search the optimal knowledge set with minimal roughness.

The paper is organized as follows. Section II gives a brief introduction to our target, task, and related work on architecture modeling. Section III gives an overview of GrC and presents the granular models, which are used in our complex-architecture-modeling problem. Section IV presents our ontology-based representation of the modeling problem. In this representation, architecture modeling is considered to be an ontology-design process. We present the GrC-based knowledge-distinguishing method in Section V, as well as a heuristic rule-selection algorithm. Section VI gives implementation details of the modeling system and knowledge-distinguishing process. Section VII gives experimental results, and Section VIII gives the conclusion.

II. Problem Description and Related Work

A. Background and Problem Description

There are many beautiful ancient Chinese buildings in Southeast China, such as the Hefang Street in Hangzhou, Zhou Village in nearby Shanghai, and Nanxun town and Xitang town in Zhejiang Province, as shown in Fig. 1. These ancient examples vary in appearance, style, and structures. Modeling them manually is an overwhelming burden; it requires intensive labor and in-depth domain knowledge.

Our research aims to increase the efficiency of modeling complex architecture in our digital heritage project for ancient Southeast Chinese buildings.

Further analysis of the topologies and structures of these architectures reveals that they have taken similar basic components and like constructions due to the regional characteristics. Hence, some automodeling techniques can be applied.

It would be beneficial if the modeling process worked automatically at the semantic level by using accumulated domain knowledge directly extracted from the architectures. To achieve this goal, we adapt the methodology of *GrC on ontology* to the architecture-modeling process.

Fig. 1. Ancient architecture in southeast China. (Top left) Hefang Street in Hangzhou. (Top right) Zhou Village in Jiangsu Province. (Bottom left) Nanxun town in Zhejiang Province. (Bottom right) Xitang town in Zhejiang Province.

B. Related Work on Automated Modeling of Architectures

There are four primary categories of automated modeling technologies: scanner modeling, image-based modeling, text-to-scene modeling, and procedural modeling.

The scanner-modeling technique uses a laser scanner to scan the architectures and then obtains points to regenerate the models of buildings. Typical examples include the digital heritage project for large-scale architectures, which are carried out by the University of California, Berkeley [4], and the vehicle-platform-based scanning method for architecture modeling [5]. However, these methods are restricted due to the high cost of scanners, complex lighting conditions, the denseness of buildings, etc.

The image-based modeling technique rebuilds architectural models from multiple photos. Traditional image-based architecture-modeling methods require a large number of photos for the same building. They calculate the spatial information about corresponding points in the photos and then generate the 3-D models. An example is the camera-array-based approach of Antone and Teller [6]. Another approach combines geometric information with image information to reconstruct architecture models [7]. This technique is much cheaper than the scanner methods, but it still needs a lot of interaction in modeling, and it cannot generate large numbers of architecture models.

The third automated modeling technique is the text-to-scene technique [8], [9]. This technique takes natural language as input, parses sentences that describe scenes, and rebuilds them based on these sentences. Currently, it can only process simple scenes and architecture.

The procedural-modeling methods, which we used, can generate large number of models from only a few input parameters [1], [2], [10]. Parish and Müller [3] introduced a stochastic, parametric L-system to generate geometries for buildings. A set of rules control the transformation, scaling, extrusion, and branching of buildings' geometry. Another approach [11] is to construct new buildings by combining the basic units of buildings, such as a roof, a wall, a window, and a gate. However, the two aforementioned approaches suffered with the randomicity of the grammar system.

As we can see, all of the aforementioned four automated modeling techniques have their shortcomings in practice. Therefore, automated-architecture model techniques should be chosen on a case-by-case basis. In our digital heritage project, we must generate architecture models cheaply and quickly; therefore, we use the procedural-modeling technique. We propose corresponding improved techniques via GrC on ontology to overcome the weakness of the procedural-modeling methods, e.g., uncontrollable combination generated by recursive-grammar system.

III. Granular Computing

Granulation seems to be a natural methodology, which is deeply rooted in human thinking. Many daily "things" are routinely granulated into sub"things." For example, the human body is granulated into the head, neck, etc. The concept of granulation might start from the creation of Leibniz's calculi; however, the notion is intrinsically fuzzy, vague, and imprecise. As an attempt to deal with vague concepts, the fuzzy set of Zadeh and the Rough Sets of Pawlak have been introduced as a basis on which to build such calculi. To formalize the concept of "thing" is difficult, and mathematicians have idealized and simplified it into the notion of partitions (equivalence relations) and have developed it into a fundamental part of mathematics, for example, congruence in Euclidean geometry, quotient groups and rings, and the field of algebra, in general. In computer science, it has been formalized into a programming discipline called structured programming, which was quite popular in the 1960s and 1970s [12], although it has lost its popularity in recent years.

Nevertheless, the notion of partitions, which does not permit any overlapping among its granules, seems too restrictive for real-world problems.

Therefore, a more general theory, namely, GrC [13], is developed.

The term GrC was coined by Lin and Zadeh [14], [15]: In the fall of 1996, Lin derived the term "GrC" from Zadeh's granular mathematics to his research area. Over the past ten years or so, the concept has gained substantial momentum and has become a viable field.

A few words about the pre-GrC period (before the term was proposed) are emphasized here. The roots of GrC can be found by Leibniz in his work on "Calculus of Thoughts." At that time, it was not recognized that in the calculi of thoughts, noncrip concepts should be used.

In 1979, Zadeh introduced the concept of information granulation in the context of fuzzy sets [16], and Hobbs presented his granularity framework [17] in 1985. In 1981, Ginsburg and Hull studied ordered binary relations in databases [18]. Note that a binary relation naturally introduces a granulation [19].

During 1988–1989, in his research on approximate retrieval, Lin derived the notion of neighborhood systems (NSs) from topological spaces and binary relations [20]. He used the idea to develop the first nonpartitioning GrC models [21]. Many more papers (nonpartition cases) were developed by the rough-set community and others. For this paper, the following three areas are the most relevant ones:

1) rough mereology [22];
2) relation/logic-based GrC models [13], in which granules are ordered tuples and not the classical sets;
3) real-world vague and uncertain applications: Many complex real-life problems are vague, uncertain, and imprecise, such as image background and foreground classification [23] with the pixel-level- based granules, to simulate dangerous road situations with hierarchical structures of granules [24] and rough-rule extracting from inconsistent information systems [25].

Finally, we note that although there is some related work on spatial reasoning with hierarchical structures of complex granules [24], [26], as far as we know, our work is the first to adapt the GrC method to complex-architecture-knowledge modeling.

To solve the complex-architecture-modeling problem, we introduce two granular models that our architecture-modeling application implements. These two models aim to model complex-architecture elements or granules to construct a specific instance of architectural style.

A. *Concepts and Granular-Computing Models Used in Our Approach*

Our granular approach for architecture-modeling system is mainly based on the *3th Grc Model* [27], and the related concepts are defined as follows.

Binary NS (BNS) [27]: Let V and W be two classical sets; each $w \in W$ is assigned a subset $B(w)$, such a set is called a binary neighborhood, and the collection $\{B(w)|\forall w \in W\}$ is called the BNS.

The three-tuple (W, V, β), where β is a BNS, is called a *Binary Grc Model*; it is also named *3th Grc Model* [27].

The definition of w-neighborhood is given as follows.

1) v and v' are said to be directly related, if v and v' are in the same tuple (of a relation in β), where v' could be an element of V or W.
2) If v and v' are indirectly related and if there is a finite sequence $v_i, i = 1, 2, \ldots, t$ such that a) v_i and v_{i+1} are directly related for every i and b) $v = v_1$, and $v' = v_{t+1}$, then the indirect relation is valid across the different relations of a granular model.
3) An element v in V is said to be w-related if v and w are directly or indirectly related.
4) The w-neighborhood V_w consists of all the $v \in V$ that are w-related.

A granular model on multiple V with unique W induces a BNS as follows [19]:

$$B_W : W \longrightarrow 2^V; \qquad w \longrightarrow V_w$$

where V_w is a w-neighborhood in V and, hence, induces a *binary granular model* (W, V, B_W).

In our approach, we extend the BNS as *sequence BNS*, which is defined as follows.

Letting V and W be two classical sets, where each $w \in W$ is assigned a tuple sequence $T(w)$, such a tuple set is called a sequence binary neighborhood, and the collection $\{T(w)|\forall w \in W\}$ is called the sequence BNS (SBNS), which is denoted as (W, V, T_w).

For example, there are two sets $W = \{w, w', w''\}$ and $V = \{v_1, v_2, \ldots, v_8\}$, and we can construct a SBNS that is a mapping (map or function) $T_W : W \rightarrow \text{seq}(V)$ as follows:

$T_W(w) = \{v_1, v_2, v_3, v_2, v_1\}$ [also presented as $r_1(w : v_1, v_2, v_3, v_2, v_1)$], which indicates that v_1, v_2, and v_3 are w-related. The full SBNS is presented as follows:

$$r_1(w : v_1, v_2, v_3, v_2, v_1), \qquad r_2(w' : v_1, v_4, v_7), \quad r_3(w'' : v_8).$$

Here, v_1 and v_2, v_2 and v_3, and v_1 and v_3 are directly related, according to the relationship r_1. In the same way, v_2 and v_7 are indirectly related, v_7 and w' are directly related, and v_2 and w' are indirectly related. Therefore, v_2 and v_7 are said to be w'-related. We can then obtain the $V_{w'} = \{v_1, v_2, v_3, v_4, v_7\}$, and $V_{w''} = \{v_8\}$.

In our modeling system, W is the architecture instance set, e.g, the house models of architecture, V is the component set, which include windows, roofs, gates, etc., and the SBNS $T_W(W \rightarrow \text{seq}(V))$ is the knowledge set (the R in the following ontology definition).

IV. Representing Architecture as Ontology Domains

Our goal is to implement an architecture-modeling system that can distinguish different elements and styles in a variety of buildings. In addition, the system should be able to generate many similarly structured or similarly styled architecture examples, which are based on semantic knowledge extracted from existing buildings. We believe that an ontology-based approach is one of the best ways to achieve these goals.

A. *Ontology of Architecture*

Gruber gave a general definition of ontology [28]. Ontologies are typically used as to specify a representational vocabulary for a shared domain of discourse. This may include definitions of classes, relations, functions, and other objects. Moreover, vagueness and imprecision are quite common in practice; therefore, many fuzzy terms have been introduced into the ontology literature [29], [30], and it is also mentioned that “ontologies are only approximate specifications of conceptualizations” [31]. The rough-set-based hierarchical learning approaches from the ontology concept [32]–[34] have also found real applications to solve complex problems. Our architecture-modeling problem that was based on ontology can be formalized as follows.

The ontology of the architecture can be viewed as a specific conceptualization of the standard ontology [35]. The intuitive understanding of the architecture ontology includes a series of architecture categories, components, and relations, which are intended to be systematic descriptions that covers all instances of the specific style. For example, the concept of ancient Chinese architecture can be defined by the answers to the following three questions. Which kinds of architecture can be classified into the ancient Chinese category? Which components constitute the ancient-Chinese architectures? What kind of relationships are maintained in these components? A good ontology for ancient Chinese architecture should cover all the instances of that style.

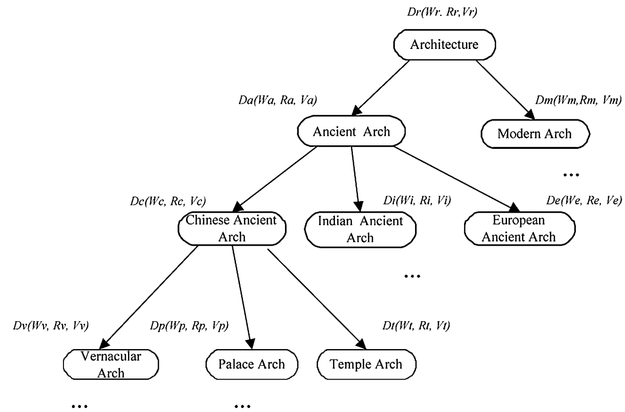


Fig. 2. Domain concept tree for architecture ontology. It is a plus-category tree, compared with the one in [1] and [10], and contains more details in the ancient Chinese domain. Here, our modeling target, i.e., the ancient Southeast Chinese architecture, belongs to the vernacular architecture domain.

Formally, the ontology of architecture is a three-tuple $D = \langle W, R, V\rangle$, where D represents a label for a domain of architecture, for example, the southeast Chinese architecture domain (D_c). It induces a classification of architectures, which is shown in Fig. 2. This is related to the first question. Therefore, in a certain architecture ontology $D = \langle W, R, V\rangle$, having

$$D \in \{D_r, D_a, D_m, D_c, \ldots\}$$

V is defined as

$$V = \{v|v \text{ is a basic architecture component of the corresponding domain } D\}.$$

V is the set of all related subconcepts (entities or vocabularies) in the architecture ontology. It should include all the subcomponents of represented architectures. This is related to the second question. For example, the southeast Chinese ancient-architecture domain's V could include the components shown in Fig. 3.

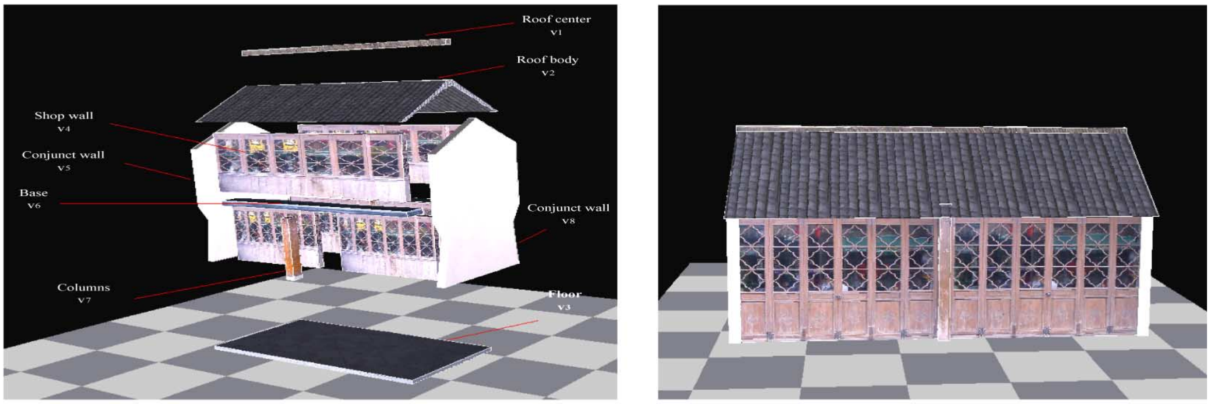
W is defined as

$$W = \{w|w \text{ is the instance belonging to the domain of } D\}.$$

W is the domain space (or instance set) that includes all the instances (known examples) within the domain. For example, W_c in the ontology of ancient Chinese architecture $D_c = \langle W_c, R_c, V_c\rangle$ should include all the ancient Chinese architecture instances.

R is defined as follows:

$$R = \{r|r \text{ is an element of SBNS } T_w : W \rightarrow \text{seq}(V). \text{ Such a relation defines an appropriate combination of architecture components in } D\}.$$



w_1 w_2

Fig. 3. Two instances in the sample ontology domain. We decompose the w_1 to demonstrate the V set of the ontology.

Mathematically, R is an SBNS. These relations define how various architecture components can be combined appropriately and smoothly to an instance w. However, in the real case, the relations or sequences in R are not enumerable, and they are always presented by some enumerable approximate inductive representations. In our architecture-modeling system, we use the L-system-based recursive-grammar rules to represent the approximate knowledge set. The recursive rules consist of an inductive relation set for the combination of basic elements in V, for example, windows, gates, and walls shown in Fig. 3, and the recursive rule examples are shown in Fig. 6.

B. Architecture-Instance Function

At a high level, the architecture-model-generating-software system combines various ingredients into a model [2], [36]. Mathematically, it is a function, which is called the architecture-instance function, that takes a w-neighborhood, i.e., subset V_w of V, and forms an instance w. The process of realizing this function for each instance is called designing an ontology for a specific architecture.

Roughly, the process starts with a V, which are architecture components, and then selects appropriate relationships, which are tuples in relations and decision tables, in R to form an instance, i.e., building, in W. The collection of the components that are actually used is denoted by $V_w \subseteq V$. This transformation of V_w to w is called an *architecture-instance function* and is denoted by F.[1]

Recall that $D = (W, R, V)$ induces a sequence-binary-granular model (W, V, T_W). With these, the formal definition of an architecture-instance function can be described by[2] $w = F_\rho(V)$, which is a function that maps a w-neighborhood V_w of V into an instance $w \in W$ using an appropriate subset ρ of R. In fact, it is the inverse sequence binary relation of T_W. Note that each member of V_w plays active roles to form the instance w.

This modeling process can be described by the following:

$$W = \{w | w = F_\rho(V), \rho \subseteq R\} \tag{1}$$

The aforementioned formula means that the domain space W contains all the instances that belong to the domain, and these instances are generated by the instance function with a specific rule set ρ.[3] Then, the maximal rule set that can satisfy the aforementioned formula is called the knowledge set R.

In practice, the model-generating system starts with the two basic elements V and R. V contains the basic components, such as the window, gate, wall, and house. W is the set of all instances, i.e., buildings, which comprises all the *correct* combinations of components. R presents the combination relationships and topological relationships among those components in V into an instance in W; D is the set of all category labels for those buildings that belong to W, for example, the ancient Chinese architectures, the ancient Indian architectures, etc. The architecture-instance function is an architecture-model-generating system [2], [36] that chooses some basic architecture components, i.e., a subset V_w of V, and then assembles them with some combinations and topology-based guides, i.e., R, to form the architecture model (instances $w, w \in W$) that belongs to a certain architecture style, i.e., domain, D.

A sample example of the ontology definition is also shown in Fig. 3.

D is a sample Chinese ancient southeast architecture domain only with two instances. In real case, there may be numberless instances; here, we construct the following example for demonstration only.[4]

$$\begin{aligned} V &= \{v_1, v_2, \ldots, v_8\} \\ W &= \{w_1, w_2\} \\ R &= \{r_1(w_1 = v_1 v_2 v_4 v_5 v_4 v_5 v_6 v_4 v_7 v_4 v_8 v_4 v_7 v_4 v_8 v_3) \\ &\quad r_2(w_2 = v_1 v_2 v_4 v_7 v_4 v_5 v_4 v_7 v_4 v_5 v_3)\}. \end{aligned}$$

[1]It used to be called *generalized function* [10], but it has a different meaning in mathematics; therefore, we have changed it to the architecture instance function.

[2]It may be presented as $w = F_\rho(V_w)$, since only the elements in V_w are actually used in the construction process; to avoid misunderstanding of the circular definition, we use the aforementioned definition.

[3]In our implementation, ρ consisted of several rules that can be certainly classified into R. We need to assign these rules in ρ manually.

[4]In this example, we have omitted the complex location-control terms for the combination of components.

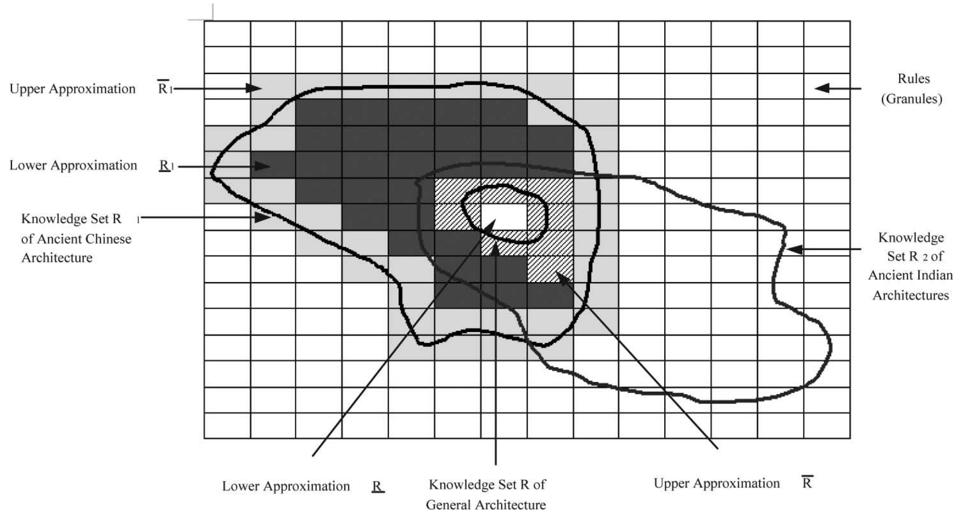


Fig. 4. Granular presentation of multiple domains with knowledge overlap, which is similar to the condition in [1].

Obviously, the R set cannot be enumerated when there are numberless instances, which is quite normal in the real case. Therefore, we use some L-system-based grammar rules to generate the sequences of R, and these recursive L-system-based grammar rules consist of the knowledge library U (or U_R).

An example of U is presented as follows:

$$\begin{aligned} U = \{ & L_1(v_{\text{window-wall}} \to v_{\text{window-wall}} | v_4) \\ & L_2(v_{\text{window-wall}} \to v_{\text{window-wall}} | v_4 | v_7) \\ & L_3(v_{\text{roof}} \to v_1 | v_2) \\ & L_4(v_{\text{house}} \to v_{\text{house}} | v_{\text{roof}} | v_{\text{window-wall}} \\ & | v_4 | v_{\text{window-wall}} | v_4) \\ & \ldots \}. \end{aligned}$$

Here, $v_{\text{window-wall}}$, v_{house}, and v_{roof} are temporary components in logic, and after the recursive generating, they will generate a final sequences, which are all constituted by the terminal components $v_1, \ldots, v_8$.

The aforementioned recursive rules are subjectively obtained from examples; therefore, they could not describe the Chinese ancient-architecture ontology domain precisely, and they also contain redundancies, which are mentioned in Section I. For this reason, we use the granular model, especially the instance function F, to represent the modeling process and try to obtain the most proper rule set from U that can describe the corresponding domain knowledge R.

V. Granular Computing on Ontology

In the previous section, we defined the ontology. Specifically, we have built four spaces, i.e., D, W, R, and V, and an architecture-instance function that converts appropriate data into an architecture instance, i.e., a building model. Equivalently, we have defined a sequence-binary-granular model (W, V, T_W). Observe that T_W is the inverse of the architecture-instance function, where $T_W : W \longrightarrow \text{seq}(V)$. Next, we will restructure this granular model so that we can select a most proper subset from U with respect to T_W. Observe that T_W is derived from R.

In real applications, the knowledge set R, which can describe the relationships between entities and domains in ideal world, cannot be usually obtained or enumerated. As a substitute, we often choose an proper subset from the recursive rule set U_R, where the subscript R will be suppressed with redundancy that has been mentioned in previous sections.

For each instance w, we associate a maximal subset[5] $\rho(\rho \subseteq U_R)$ such that $w = F_\rho(V)$. This association gives us a map

$$B : W \to 2^{U_R} : w \to \rho. \tag{2}$$

This leads to a new binary granular model (W, U_R, B). The rule set ρ is a binary neighborhood or granule in U_R. This binary granular model has a well-defined notion of lower and upper approximations that is based on neighborhood systems.

In the processing of real architecture modeling, U (or subsets in U) does not match the conceptual version of the knowledge very well. The situation is illustrated in Fig. 4. The real knowledge set of a specific domain will lie in an irregular shape, as shown in Fig. 4, each grid represents a rule in the Universal U, and the real knowledge set will cross some grids, which means that these crossed rules will be classified either as that knowledge set, or some other knowledge set, and within vague boundaries in the rule-based granule. According to Fig. 4, we can see that none of the rule subsets of U could exactly match the boundary of the real knowledge set R. However, R can be approximated by choosing the best-matched subset in U.

P is a subset of U: It contains rules that may not match well with the boundary of the knowledge set R.

Although R in $D = (W, R, V)$, which gives (W, U_R, B), cannot be obtained directly, a small rule set K in R can be determined manually. In the implementation, K is initialized by several basic architecture-modeling rules that are necessary for architecture generation. Then, one rule at a time, it tests whether a rule r in P can be mapped by $F_{K \cup \{r\}}(V)$ into an

[5]Here, ρ can be viewed as the most likely knowledge set compared with R.

instance $w \in W$. By this method, we can find the upper and lower approximations of P with respect to the ontology domain D. In the same way, the $\underline{NP}_D$ and $\overline{NP}_D$ can also be defined by testing the rule r not in P, i.e., $r \in U - P$, whether the $w = F_{K\cup\{r\}}(V)$ belongs to the instances set W. We proceed as follows.

Let the universe U be the whole rule library (knowledge) in our automodeling system [10]. Letting P be a subset of U, then approximate sets within the context of $D = \langle W, R, V\rangle$ are[6]

$$\overline{P}_D = \bigcup \left\{ r \mid \exists w, \ \text{ having } w \in W \right. \\ \left. w = F_{K\cup\{r\}}(V), \text{and } r \in P, K \subseteq R \right\}. \tag{3}$$

The lower approximation of P is

$$\underline{P}_D = \bigcup \left\{ r \mid \forall w, \ \text{ having } w \in W \right. \\ \left. w = F_{K\cup\{r\}}(V), \text{and } r \in P, K \subseteq R \right\} \tag{4}$$

Here, $F_{K\cup\{r\}}(V)$ is the architecture-instance function of the ontology domain $D = \langle W, R, V\rangle$. In our modeling system, the architecture-instance function refers to the process of using the rule set $K \cup \{r\}$ to combine the vocabularies V into a single architecture instance.

A intuition explanation for the upper and lower approximations could be found in Fig. 4, the upper approximation $\overline{R}_1$ consists of all the granules related with the R_1, and the lower $\underline{R}_1$ consists of the granules that are all directly within the R_1.

In the same way, we can also define the approximate sets of the knowledge set $\overline{P}$ ($\overline{P} = U - P$) with respect to the concept domain D.

The knowledge set $\overline{P}$'s upper approximate set with respect to the concept domain D is

$$\overline{NP}_D = \bigcup \left\{ r \mid \exists w, \ \text{ having } w \in W \right. \\ \left. w = F_{K\cup\{r\}}(V), \text{and } r \in U - P, K \subseteq R \right\}. \tag{5}$$

The knowledge set $\overline{P}$'s lower approximate set with respect to the concept domain D is

$$\underline{NP}_D = \bigcup \left\{ r \mid \forall w, \ \text{ having } w \in W \right. \\ \left. w = F_{K\cup\{r\}}(V), \text{and } r \in U - P, K \subseteq R \right\}. \tag{6}$$

Although the notion of roughness [37] is a rough-set notion, it is easy to generalize to GrC. Therefore, the roughness of the knowledge set P with respect to the domain D can be calculated as follows:

$$\chi_{P_D} = \frac{|\overline{P}_D| - |\underline{P}_D|}{|\overline{P}_D|}. \tag{7}$$

In addition, the corresponding roughness of knowledge set $\overline{P}$ with respect to the domain D can be calculated as follows:

$$\chi_{NP_D} = \frac{|\overline{NP}_D| - |\underline{NP}_D|}{|\overline{NP}_D|}. \tag{8}$$

A. Measurement on Minimal Knowledge Boundary

After defining the upper and lower approximations and the roughness of the rule set P with respect to a certain ontology domain $D = \langle W, R, V\rangle$, we need to select the best rule set for the ontology domain, i.e., to find the rule set P that is closest to the knowledge set R in the ontology domain. To determine the closeness between the rule set P and the knowledge set R in an ontology domain, we need a critical metric for closeness. The measurement should be able to determine the *fitness* of the rule set P with respect to the real domain knowledge R.

Why not use the roughness χ_{P_D} as the metric for knowledge selection? The roughness χ_{P_D} could well represent the vague degree of the selected rule set P with respect to R: Less roughness will be less vague to the P, and once the χ_{P_D} equals zero, it will reach the state of being the least vague. However, χ_{P_D} always approaches zero when P is a subset of R. Obviously, P is not the optimal rule set to match the real knowledge set R in this case. Therefore, roughness only represents the vagueness of P with respect to R. It does not represent the degree of which P matches R.

To keep P from shrinking into the true subset of R, another parameter χ_{NP_D} is desired. It will be used to constrain the P and ensure that P will not be a subset of R.

In our solution, we adopt a roughness-function-based metric to identify the minimal knowledge boundary between the selected rule set P and the ontology domain knowledge R. Intuitively, the roughness-function definition can be understood as the selected rule set P that contains, at most, the true knowledge, which is represented by χ_{P_D}, and the $U - P$ that contains at least the true knowledge, which is represented by χ_{NP_D}.

The roughness-based measurement has been adopted in many cases [38], [39] and applications [23]. In this paper, we need a roughness-function metric that can minimize the roughness of the knowledge set P with respect to the desired ontology domain and maximize the roughness of knowledge set $\overline{P}$ with respect to the desired ontology domain. The roughness-function measurement is defined as follows:

$$E_{P_D} = \chi_{NP_D} - \chi_{P_D}. \tag{9}$$

Then, the task of finding the most-suitable knowledge set with respect to the ontology domain is equivalent to calculating each knowledge rule set's roughness-function measurement and to choose the knowledge set, whose roughness function measurement is maximal as the best domain-knowledge set. The appropriate knowledge set P with respect to the ontology domain D can be calculated by the following:

$$P^* = \arg\max\,(E_{P_D}). \tag{10}$$

[6] $\forall w \in W$ can be determined by enumerating a large number N of instances. We believe that the negative condition will occur when the N is large enough. Note that K is a small rule set in R.

B. Heuristic Knowledge-Selection Algorithm Based on a Hierarchical-Concept Domain

Selection of appropriate rules from the large architecture-modeling knowledge library is a time-consuming process. We would like a more efficient method than the full-search policy.

Before presenting the algorithm, we first state the *monotonicity theorem*. The theorem follows from the assumption that the modeling system's rule units are not a good fit for the knowledge granules.

Monotonicity Theorem. Assume that K and K' are two knowledge rule sets in U ($K \subseteq U$, $K' \subseteq U$) and that $K' = K \cup \{k\}$, $k \notin K$, $D(W, R, V)$ is an ontology domain concept. We then have

$$k \in R \leftrightarrow E_{K_D} < E_{K'_D}.$$

We prove the theorem in two parts: $k \in R \rightarrow E_{K_D} < E_{K'_D}$, and $E_{K_D} < E_{K'_D} \rightarrow k \in R$.

Proof. $k \in R \rightarrow E_{K_D} < E_{K'_D}$.

First, calculate the following formula:

$$\begin{aligned} E_{K_D} - E_{K'_D} &= (\chi_{NK_D} - \chi_{K_D}) - (\chi_{NK'_D} - \chi_{K'_D}) \\ &= (\chi_{NK_D} - \chi_{NK'_D}) + (\chi_{K'_D} - \chi_{K_D}). \end{aligned}$$

Then, consider the following two formulas: $\chi_{K'_D} - \chi_{K_D}$, and $\chi_{NK_D} - \chi_{NK'_D}$. Based on their definitions, we get

$$\chi_{K'_D} - \chi_{K_D} = \left(1 - \frac{|\underline{K'}_D|}{|\overline{K}'_D|}\right) - \left(1 - \frac{|\underline{K}_D|}{|\overline{K}_D|}\right)$$

$$\chi_{NK_D} - \chi_{NK'_D} = \left(1 - \frac{|\underline{NK}_D|}{|\overline{NK}_D|}\right) - \left(1 - \frac{|\underline{NK'}_D|}{|\overline{NK}'_D|}\right)$$

i.e.,

$$\chi_{K'_D} - \chi_{K_D} = \frac{|\underline{K}_D|}{|\overline{K}_D|} - \frac{|\underline{K'}_D|}{|\overline{K}'_D|}$$

$$\chi_{NK_D} - \chi_{NK'_D} = \frac{|\underline{NK'}_D|}{|\overline{NK}'_D|} - \frac{|\underline{NK}_D|}{|\overline{NK}_D|}.$$

If $k \in R$, then k is in the lower approximation of K', which means $|\underline{K'}_D| = |\underline{K}_D| + 1$. So do $\underline{NK'}_D$ and $\underline{NK}_D$, and therefore, $|\underline{NK'}_D| + 1 = |\underline{NK}_D|$.

Similarly, k is contained in both the upper approximation of K' and NK ($NK = U - K$). Therefore, the upper approximations have the following relationships:

$$|\overline{K}'_D| = |\overline{K}_D| + 1$$

$$|\overline{NK}_D| = |\overline{NK}'_D| + 1.$$

In addition

$$\chi_{K'_D} - \chi_{K_D} = \frac{|\underline{K}_D|}{|\overline{K}_D|} - \frac{|\underline{K}_D| + 1}{|\overline{K}_D| + 1} = \frac{|\underline{K}_D| - |\overline{K}_D|}{|\overline{K}_D|(|\overline{K}_D| + 1)}$$

$$\begin{aligned} \chi_{NK_D} - \chi_{NK'_D} &= \frac{|\underline{NK'}_D|}{|\overline{NK}'_D|} - \frac{|\underline{NK'}_D| + 1}{|\overline{NK}'_D| + 1} \\ &= \frac{|\underline{NK'}_D| - |\overline{NK}'_D|}{|\overline{NK}'_D|(|\overline{NK}'_D| + 1)}. \end{aligned}$$

We then have[7]

$$\chi_{K_D} - \chi_{K'_D} \leq 0 \quad \text{and} \quad \chi_{NK_D} - \chi_{NK'_D} < 0$$

or

$$\chi_{K_D} - \chi_{K'_D} < 0 \quad \text{and} \quad \chi_{NK_D} - \chi_{NK'_D} \leq 0$$

which leads to

$$E_{K_D} < E_{K'_D}.$$

Proof. $E_{K_D} < E_{K'_D} \rightarrow k \in R$.

First, assuming $k \notin R$, there may be the following two conditions for k.

1) k is not the boundary element of the knowledge set R: k will not generate any instances in the domain $D(W, R, V)$, which does not affect the lower and upper approximations of K and K'; therefore, we have $\chi_{K_D} = \chi_{K'_D}$, and $\chi_{NK_D} = \chi_{NK'_D}$, i.e., $E_{K_D} = E_{K'_D}$.
2) k is the boundary element of the knowledge set R: k generates at least one instance in the domain $D(W, R, V)$; therefore, we have $\overline{K}_D + 1 = \overline{K}'_D$, $\overline{NK'}_D + 1 = \overline{NK}_D$, and $\underline{K}_D = \underline{K}'_D$, $\underline{NK}_D = \underline{NK}'_D$.

Therefore, we have

$$\chi_{K'_D} - \chi_{K_D} = \frac{|\underline{K}_D|}{|\overline{K}_D|} - \frac{|\underline{K'}_D|}{|\overline{K}'_D|} > 0$$

$$\chi_{NK_D} - \chi_{NK'_D} = \frac{|\underline{NK'}_D|}{|\overline{NK}'_D|} - \frac{|\underline{NK}_D|}{|\overline{NK}_D|} > 0$$

i.e.,

$$E_{K_D} > E_{K'_D}.$$

At last, we have the following conclusion based on our assumption ($k \notin R$):

$$E_{K_D} \geq E_{K'_D}$$

This contradicts $E_{K_D} < E_{K'_D}$; therefore, the assumption is incorrect, and we have $k \in R$. QED.

The *monotonicity theorem* can be understood as follows: Adding only the rules that belong to the knowledge set R into the selected rule set P increases the value of roughness function. Once there are no more rules that can increase the roughness function, the rule set is optimal with respect to the knowledge set.

Next, we present an effective heuristic knowledge-selection algorithm to compute the best knowledge set P^* from a certain knowledge set. We call it the Domain-Knowledge-based Heuristic-Selection (DKHS) algorithm. The heuristic pruning of the spatial search tree is based on the *monotonicity theorem* given previously. The algorithm is as follows.

[7]Here, both $\chi_{K_D} - \chi_{K'_D} = 0$, and $\chi_{NK_D} - \chi_{NK'_D} = 0$, iff all the elements in U that are crisp belonged to R or did not belong to R, i.e., $\overline{U}_D = \underline{U}_D$. Obviously, in the architecture-modeling case, there must be some relation elements that are fuzzy that belonged to R; then, $\overline{U}_D \neq \underline{U}_D$.

Algorithm 1: DKHS

Data: Ontology concept domain $D =< W, R, V >$, knowledge set S.
Result: Best suitable knowledge set P, with respect to concept domain D.
begin
/* calculate the roughness function*/
1. $\vartheta = E(S, D)$
2. $P = S$

/* construction the knowledge sub set*/
3. For all rules r in S
4. $\quad S' = S - r$

/*the branches that are not less than ϑ will not be processed*/
5. $\quad$ If S' is reasonable and $E(S', D) \geq \vartheta$

/* recursively generate the knowledge subset with maximized measurement from current set*/
6. $\quad\quad Q = DKHS(S', D)$
7. $\quad\quad$ If $E(Q, D) > E(P, D)$
8. $\quad\quad\quad P = Q$
9. $\quad\quad$ End If
10. $\quad$ End If
11. End For
12. return P

Note that the function $E(S, D)$ denotes the roughness-function measurement of the knowledge set S with respect to domain D. The DKHS algorithm recursively searches the input rule set S and produces the best knowledge set with respect to domain D.

The algorithm starts with the full set U and repeatedly removes one rule from S_i^{l-1} in turn to generate subsets S_i^l, where l is the current level, and i is the sequence of different subsets. If $E(S_i^{l-1}, D) > E(S_i^l, D)$, the branch of S_i^{l-1} will be pruned; otherwise, it will recursively search the next level.

Normally, the input knowledge set S is initialized to U: the entire architecture-modeling rules library of our modeling system. As Fig. 4 shows, the knowledge rules overlap in both their upper and lower approximations, e.g., the upper and lower approximations of R_1 may overlap with the R_2. They may even be subsets of each other. For example, the rule set of general architecture R is a subset of both the ancient Chinese architectures' R_1 and the Indian architectures' R_2. We can use the hierarchical category of these domains, as shown in Fig. 2, i.e., we can begin the search from the appropriate child domain-knowledge-rule set rather than searching the whole universe U. In addition, use of the parent domains can increase the search efficiency, since they can help to validate S' in step 5.

VI. Implementation

A. General Implementation of the Modeling System

A model of ancient southeast Chinese architecture should first design the basic architecture components with a specific style, such as a window, wall, and roof, in ancient southeast Chinese style, as shown in Fig. 3. They consist of the vocabulary set V in our approach. Then, the architect should define the combination relations and topology relations for these basic components, i.e., the knowledge library U in our approach.

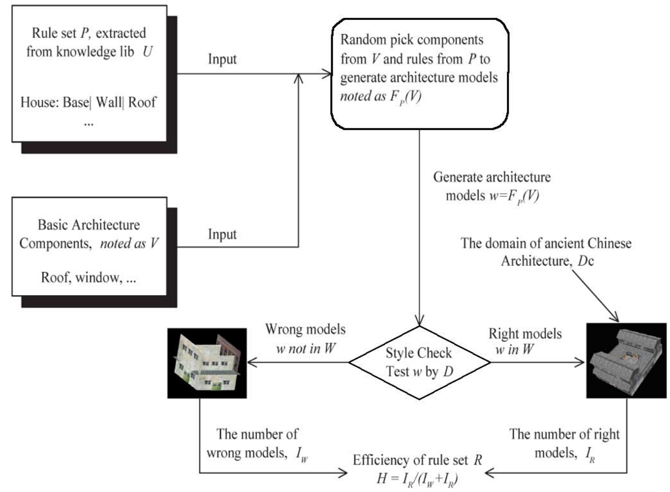


Fig. 5. Implementation diagram of the automodeling system with ontology approach.

Therefore, the goal of our approach is to select the best rule set P (from U), which can be mostly alike with the knowledge R in ancient southeast Chinese architecture ontology $D(W, R, V)$. To achieve this goal, we need to solve two problems. The first one follows: By which measure can we evaluate the quality of the selected rule set P? The other one is as follows: When should we determine the best P, according to that measure of quality? In our approach, the roughness function E_{P_D} with respect to the ancient southeast Chinese architecture ontology domain D is used as the measure of quality. It can also determine that the optimal condition occurs when the degree to which P satisfies E_{P_D} is maximal, according to the monotonicity theorem applied to the roughness function.

The implementation of our architecture-modeling system is shown in Fig. 5. It has two parameters: The architecture-modeling knowledge rules, which corresponds to the knowledge library U in the ontology domain, and the basic architecture elements, i.e., components, which corresponds to vocabulary set V in the ontology domain. From these parameters, which are guided by the input rules, the architecture generator combines the architecture components and generates the architecture instances. A style-checking module, which is used to verify whether the instance's style is desirable, is as follows. The detailed system implementation of our architecture-modeling system can be found in [2] and [10].

B. Knowledge Presentation in Architecture Modeling

The architecture-modeling knowledge is presented in terms of recursive rules about basic architecture elements. The basic architecture elements are natural architectural components or vocabularies in the ontology domain, as shown in Fig. 3. Each recursive rule adopts the improved L-System grammar with components' ratio, as shown in Fig. 6. In Fig. 6, the rule defines both the combination relationship and the ratio relationship. Here, the ratio includes four parameters: width, length, height, and number. The first three parameters describe the outline ratio relationship between components, and the numeric

*{**CITY** :: (ROAD), (BLOCK);*
[R$_{road}$ (Width, Length, Height, Number)]: [R$_{block}$(Width, Length, Height, Number)]
}

*{**BLOCK** :: (PATCH), (BLOCK) | ;*
[R$_{patch}$ (Width, Length, Height, Number)]: [R$_{block}$ (Width, Length, Height, Number)]
}

*{ **PATCH** :(CONJUNCT_WALLl), (HOUSE), (PATCH) |;*
[R$_{conjunct_wall}$ (Width, Length, Height, Number)]: R$_{house}$ (Width, Length, Height, Number)]: [R$_{patch}$(Width, Length, Height, Number)]
}

*{**HOUSE** :: (BASE), (WALL), (ROOF) ;*
[R$_{base}$(Width, Length, Height, Number)]: R$_{wall}$(Width, Length, Height, Number)]: [R$_{roof}$(Width, Length, Height, Number)]
}

Fig. 6. Recursive production rules for modeling architectures defined in our previous work [2]. The rule defines both the combination and the ratio relationships. The ratio includes four parameters: width, length, height, and number. The first three parameters describe the outline ratio relationship among components, and the number parameter describes the numeric ratio among components.

parameter describes the number ratio between components. The recursive-production knowledge-based modeling system can be viewed as a kind of language system. The architecture components are the basic vocabularies, and the recursive rules on topology and combination define the language grammars. The right vocabularies assembled randomly cannot produce the right sentence; only the right vocabularies properly assembled under the specified grammars can produce the right sentence. Therefore, using the correct rules on topology and combination is quite important in order to describe and model the vernacular houses' styles and structures correctly.

C. Ontology-Domain Verification

The ontology-domain verification is implemented in the style-check module in Fig. 5. Its purposes are to check the generated architecture instance and to check whether the instance belongs to the desired ontology domain. In our system, the generated house is established by a production of the recursive grammar, and each house can be represented individually as a sequence of grammar terms. This way, the style-judging work can be reduced to grammar checking. In addition, the verification system only needs to check the sequences generated by the production engine and match them with the predefined style-term sequences. In our modeling system, the components and control rules are all described in Extensible Markup Language (XML); we then verify them with a document-type definition (DTD)-based technique [10].

VII. Experiments and Results

In our digital heritage project, we have taken the rough-set-based domain-knowledge-selection approach. In this section, we present three comparable experiments within the ancient southeast Chinese architecture domain.

In experiments 1 and 2, we use several knowledge rule sets as input parameters to generate architecture instances and then evaluate the results. The detailed experiments' process is shown in Fig. 5.

All the input rule sets use similar vocabularies over the ancient Chinese architecture domain, which means that they all use the same basic architecture components, as the generator's input parameters. We also use the same southeast Chinese architecture domain style DTD [2] to verify the instances generated by the input rule sets.

We use the input knowledge rule set to generate 2000 architecture instances and record the number of instances that belong to the ancient southeast Chinese architecture, which are noted as I_c, and the number of instances that do not belong to the ancient southeast Chinese architecture, which are denoted by $I_{\overline{c}}$. I is the total number of instances: $I = I_c + I_{\overline{c}} = 2000$. Then, we calculate the domain hit ratio, which is noted as H, with the following:

$$H = \frac{I_c}{I} = \frac{I_c}{I_c + I_{\overline{c}}}. \tag{11}$$

Similar to the error-rate measurement used in the classifier, the domain hit ratio describes the performances of the input knowledge rule set fitted for the ontology domain.

In experiment 1, we adopt two evaluative metrics and two search policies to generate four knowledge rule subsets (R_1, R_2, R_3, R_4) from the whole knowledge rule library U. We use these four sets as input parameters and then evaluate them with their hit ratios; the hit ratios of their corresponding complement sets ($U - R_i$) are also presented in experiment 1. These two evaluative metrics are rough function (RF), which is defined in our approach, and rough entropy (RE) [1], [23], which is proposed by Pal and used to extract image objects. The RE in our experiment is calculated as follows:

$$E_{P_D} = -[\chi_{P_D}\log_2(\chi_{P_D}) + (1-\chi_{NP_D})\log_2(1-\chi_{P_D})]. \tag{12}$$

Besides the recursive backward-search policy, which is DKHS algorithm in our approach, we also implement the QUICKREDUCT algorithm (QR) [40], it is a forward-search algorithm and adopts a greedy increasing policy. The results of experiment 1 are shown in Table I.

In our automodeling system, there are total 244 rules ($|U| = 244$), all of which are subjectively given by the architectural experts, and to further evaluate the performance of our approach, we also carry out the experiment 2, which randomly selects several rule sets (T_i) from U. Among them, the rule numbers in T_2, T_3, T_4, and T_5 are same with the four rule sets in experiment 1, respectively. The experimental results are shown in Table II

Experiments 1 and 2 show that the rule sets selected with evaluative metrics will be much fitter than the randomly-selected rule sets with respect to the knowledge domain because their domain hit ratios are all much higher than the randomly selected rule sets. The domain hit ratio of T_6 (also U) is 0.332 under 2000 instances generated by our automodeling system, which indicates that the whole knowledge library is highly redundant

TABLE I
EXPERIMENTAL RESULTS ON DIFFERENT EVALUATION MATRICES AND SEARCH POLICIES

	R_1(QR+RE)	R_2(QR+RF)	R_3(DKHS+RE)	R_4(DKHS+RF)
Selected rules	131	93	107	105
Hit Ratio of R_i	0.6015	0.813	0.995	0.997
Hit Ratio of $U - R_i$	0.163	0.218	0.015	0.009
Test Instance I	2000	2000	2000	2000

TABLE II
EXPERIMENTAL RESULTS ON SEVERAL RANDOM SELECTED RULE SETS FROM U

T_i	T_1	T_2	T_3	T_4	T_5	T_6
$\|T_i\|$	50	131	93	107	105	224
H_{T_i}	0.463	0.423	0.313	0.218	0.288	0.332
$\frac{\|T_i\|}{\|U\|}$	20.49%	53.69%	38.11%	43.85%	46.875%	100%
H_{U-T_i}	0.381	0.415	0.434	0.512	0.489	—
I	2000	2000	2000	2000	2000	2000

with respect to the knowledge of the ancient southeast Chinese architecture.

By implementing the DKHS + RF algorithm on the Universal U (the whole knowledge library), we get the most suitable knowledge rule set R_4 (containing 105 rules) with respect to the ancient southeast Chinese architecture domain, and the domain hit ratios of the knowledge set R_1 and R_3 are much greater than the ratios of the $T_1, T_2, \ldots, T_6$ and almost reach the levels of R_2 and R_4. This suggests that RF may be a better-evaluated metric in the case of selecting approximation for the architecture-modeling knowledge set, although RE is also able to evaluate the tightest of the selected rule set, compared with the knowledge set.

Although, the greedy search policy in QR can reduce the search time much, both the RE and RF metrics under the QR cannot reach the global peak. The reason may be that the monotonicity of RF can only guarantee the maximal RF is optimal, while increasing along the direction of maximal grads in the RF may not reach the global peak. Therefore, the QR + RF obtains a much small rule set, while QR + RE obtains too many rules.

As there are four types of redundancy cases, which are described in Section I, we further analyze the relationship between the redundancy conditions and the selected knowledge set. After considering the instances generated by R_4, the wrong instances in this condition are all caused by the combination of multiple rules and not introduced by only one rule. This means the selection algorithm has successfully removed the *wrong-rule redundancy* and *other-style redundancy*. However, there may be some *share redundancy* or *repeating redundancy* remaining in the selected rule set R_4. Within $U - R_4$, there still are several satisfied instances, which indicate that the raw knowledge

TABLE III
USER CASE EXPERIMENTAL RESULTS FOR THE INSTANCES GENERATED BY DIFFERENT RULE SETS

Rule Sets	Average score	Standard deviation
QR+RE	6.21	2.07
QR+RF	5.92	1.02
DKHS+RE	7.82	1.33
DKHS+RF	8.79	1.21
T_1	3.13	1.13
T_2	2.94	0.97
T_3	2.07	1.05
T_4	1.76	0.77
T_5	1.83	1.45
$T_6(U)$	2.24	1.52

set U must contain *repeating redundancy*, and the selection algorithm can remove some of this redundancy. Based on the vague knowledge-boundary assumption, the DKHS + RF algorithm can do a good job of distinguishing both the *wrong-rule redundancy* and *other-style redundancy* and partially distinguishing the *repeating redundancy*. However, the *share redundancy*

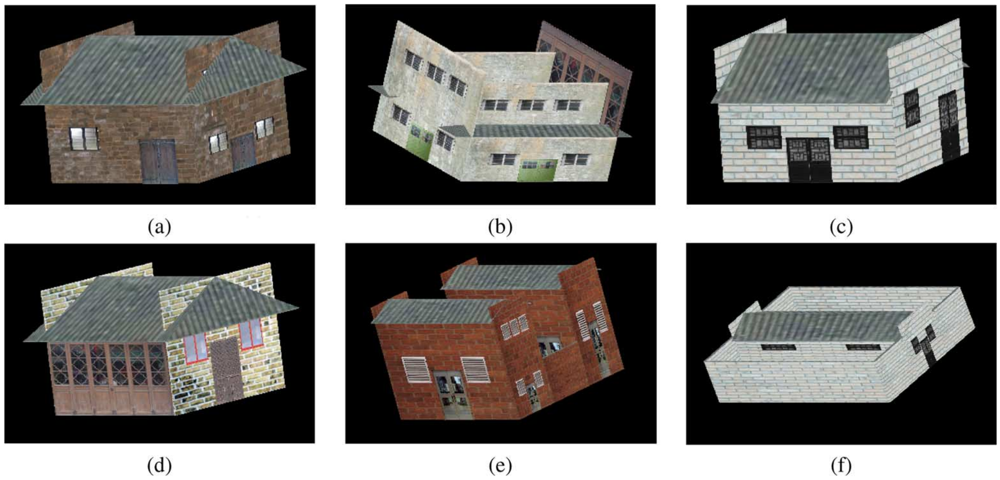


Fig. 7. Experimental results generated by unselected knowledge rules, including the whole knowledge library U and the randomly selected knowledge rule set T. (a)–(c) are houses generated by U, and (d)–(f) are houses generated by T_1.

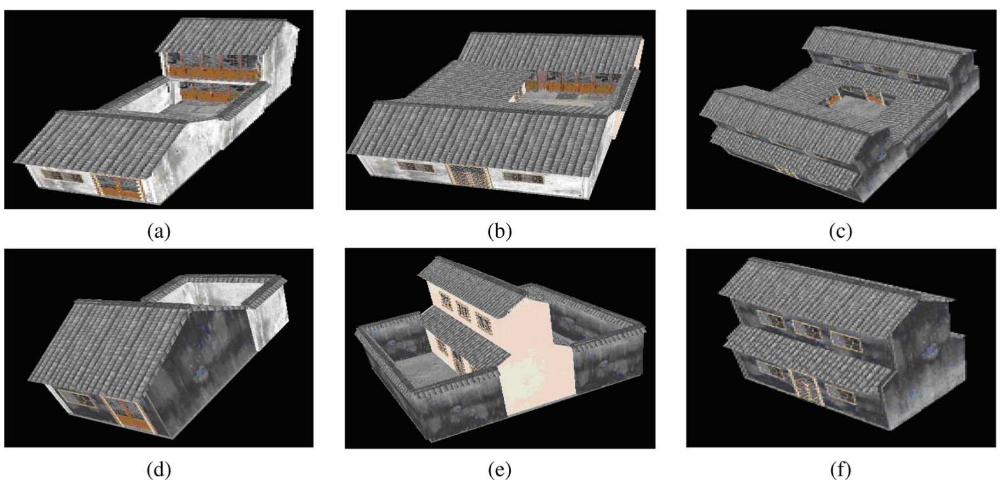


Fig. 8. Experimental results generated by the rule set R, which is selected by the DKHS algorithm with respect to the ancient Chinese architecture domain. (a)–(f) Six typical topologies of houses in southeast Chinese ancient architecture.

condition may be more difficult, and our future research will address this problem.

We also carry out experiment 3, which is a user-study experiment, to evaluate the effectiveness and validity of our approach to generate buildings with correct style. We invite 50 graduate students from different background to evaluate the results that are generated by our architecture-modeling system with different rule-sets input. Each input rule set generates ten architecture instances.

In our user study, the instances are randomly presented to the testers, and the participants grade the quality of the instances, whether it is a correct Chinese ancient-architecture instance, using the following scores: 0 for totaly wrong, 5 for moderate between wrong and right, 10 for perfect, the value between 0 and 5 for less correct, and the value between 5 and 10 for more correct.

We then record the scores of each instances by rule sets and then calculate the average scores and standard deviation of each rule set; results are shown in Table III. The average scores of R_4 (DKHR + RF) is 8.79, which is the rule set that achieves the best performance. The average scores of $T_1, \ldots, T_6$ are all much lower than the previous four test sets, which suggests that the rule selection with a proper metric is quite necessary and effective.

Finally, several generated instances are shown in Figs. 7 and 8. Fig. 7 shows the architectural models generated by the knowledge set U and T_1. Fig. 8 shows the architectural models generated by the knowledge set R_4, which is filtered by the rough-set-based algorithm.

VIII. Conclusion

We have introduced a new approach that combines the ontology-design method with the GrC method to enhance the accuracy and performance of modeling complex architectures. The main unique features worth noting include the following:

1) We have presented an ontology-based description for different classes of architectures, using formal ontologies to define the complex-architecture problem.
2) We describe the knowledge-overlapped condition from both the ontology and granular approaches. The

architecture ontology domains are organized as a hierarchical tree structure, which can simplify the complex-architecture-modeling problem by reusing and distinguishing these overlaps.

3) We construct a granular model for the architecture-modeling knowledge about a certain ontology domain. To distinguish the rough boundaries of a knowledge set is difficult; therefore, our granular model distinguishes them indirectly, i.e., by defining the upper and lower approximations by distinguishing the instances that are generated by the knowledge rules.
4) We present a roughness-function-based measurement to evaluate the degree of a knowledge set with respect to a certain ontology domain. Based on the *Monotonicity Theorem* of the roughness function, we also present a heuristic rule-selection algorithm, i.e., DKHS, which can increase the efficiency of the algorithm by pruning the searching tree.

The method introduced in this paper has been implemented in our ancient southeast Chinese architecture digital heritage project. It can simplify the design of the automodeling system and increase the hit rate when generating complex architectures with similar style structure. Additionally, the highlighted features of our approach could be used in other similar applications that cannot directly distinguish the objects.

Acknowledgment

The author would like to thank Prof. T. Y. Lin for kind help and support in making this paper possible.

References

[1] Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "Smart architect: Scalable ontology-based modeling for ancient chinese architecture," *IEEE Intell. Syst.*, vol. 23, no. 1, pp. 49–56, Jan/Feb. 2008.

[2] Y. Liu, C. Xu, Z. Pan, and Y. Pan, "Semantic modeling for ancient architecture of digital heritage," *Comput. Graph.*, vol. 30, no. 5, pp. 800–814, 2006.

[3] Y. I. H. Parish and P. Müller, "Procedural modeling of cities," in *Proc. 28th Annu. Conf. Comput. Graph.*. Los Angeles, CA: ACM, 2001, pp. 301–308.

[4] A. C. Addison and M. Gaiani, "Virtualized architectural heritage: New tools and techniques," *IEEE MultiMedia*, vol. 7, no. 2, pp. 26–31, Apr.–Jun. 2000.

[5] C. Früh and A. Zakhor, "3d model generation for cities using aerial photographs and ground level laser scans," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, Kauai, HI, Dec. 8–14, 2001, pp. 31–38.

[6] M. E. Antone and S. J. Teller, "Automatic recovery of relative camera rotations for urban scenes," in *Proc. CVPR*, 2000, pp. 2282–2289.

[7] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. SIGGRAPH*, 1996, pp. 11–20.

[8] R. Johansson, A. Berglund, M. Danielsson, and P. Nugues, "Automatic text-to-scene conversion in the traffic accident domain," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, Edinburgh, U.K., Jul. 30–Aug. 5, 2005, pp. 1073–1078.

[9] R. Coyne and R. Sproat, "Wordseye: An automatic text-to-scene conversion system," in *Proc. 28th Annu. Conf. Comput. Graph.,* Los Angeles, CA: ACM, 2001, pp. 487–496.

[10] Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "Ontology based semantic modeling for chinese ancient architectures," in *Proc. 21st Nat. Conf. Artif. Intell./18th Conf. Innovative Appl. Artif. Intell.*, Boston, MA, 2006, pp. 1808–1813.

[11] P. J. Birch, V. J. Jennings, A. M. Day, and D. B. Arnold, "Rapid procedural-modelling of architectural structures," in *Proc. 19th Eurograph. U.K. Conf.*, Apr. 3–5, 2001, pp. 187–196.

[12] E. Yourdon and L. L. Constantine, *Structure Design*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[13] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of Granular Computing*. New York: Wiley, 2008.

[14] T. Y. Lin, "Granular computing,," presented at the Announcement BISC Spec. Interest Group Granular Comput., Berkeley, CA, 1997.

[15] L. A. Zadeh, "Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems," *Soft Comput.*, vol. 2, no. 1, pp. 23–25, 1998.

[16] L. Zadeh, "Fuzzy sets information granulation," in *Advances in Fuzzy Set Theory and Applications*. Amsterdam, The Netherlands: North-Holland, 1979.

[17] J. R. Hobbs, "Granularity," in *Proc. IJCAI*, 1985, pp. 432–435.

[18] S. Ginsburg and R. Hull, "Ordered attribute domains in the relational model," in *XP2 Workshop on Relational Database Theory*. University Park, PA: Penn State Univ. Press, 1981.

[19] T. Y. Lin, "Granular computing on binary relations. I: Data mining and neighborhood systems," in *Rough Sets In Knowledge Discovery*, A. Skoworn and L. Polkowski, Eds. Berlin, Germany: Physica-Verlag, 1998, pp. 107–121.

[20] T. Y. Lin, "Neighborhood systems and approximation in database and knowledge base systems," in *Proc. 4th Int. Symp. Methodol. Intell. Syst. (Poster Session)*, 1989, pp. 75–86.

[21] T. Y. Lin, "Granular computing: From rough sets and neighborhood systems to information granulation and computing in words," in *Proc. Eur. Congr. Intell. Tech. Soft Comput.*, Sep. 8–12, 1997, pp. 1602–1606.

[22] L. Polkowski and A. Skowron, "Rough mereology: A new paradigm for approximate reasoning," *Int. J. Approximate Reasoning*, vol. 15, no. 4, pp. 333–365, 1996.

[23] S. K. Pal, B. U. Shankar, and P. Mitra, "Granular computing, rough entropy and object extraction," *Pattern Recogn. Lett.*, vol. 26, no. 16, pp. 2509–2517, 2005.

[24] H. S. Nguyen, A. Skowron, and M. S. Szczuka, "Situation identification by unmanned aerial vehicle," in *Proc. Rough Sets Curr. Trends Comput.*, 2000, pp. 49–56.

[25] Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "Rough rule extracting from various conditions: Incremental and approximate approaches for inconsistent data," *Fundam. Inf.*, vol. 84, no. 3–4, pp. 403–427, 2008.

[26] L. Polkowski and A. Skowron, "Rough mereology in information systems with applications to qualitative spatial reasoning," *Fundam. Inf.*, vol. 43, no. 1–4, pp. 291–320, 2000.

[27] T. Y. Lin, "Granular computing: Ancient practices theories, future directions," in *Encyclopedia of Complexity and Systems Science*. A. R. Meyers, Ed. Berlin, Germany: Springer-Verlag, 2008.

[28] T. Gruber, "A translation approach to portable ontolgoy specifications," *Int. J. Knowl. Acquistion Knowl.-Based Syst.*, vol. 5, no. 2, pp. 199–220, Jul. 1993.

[29] C.-S. Lee, Z.-W. Jian, and L.-K. Huang, "A fuzzy ontology and its application to news summarization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 5, pp. 859–880, Oct. 2005.

[30] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic fuzzy ontology generation for semantic web," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 6, pp. 842–856, Jun. 2006.

[31] N. Guarino, "Toward a formal evaluation of ontology quality," *IEEE Intell. Syst.*, vol. 19, no. 4, pp. 74–81, Jul./Aug. 2004.

[32] J. G. Bazan and A. Skowron, "On-line elimination of non-relevant parts of complex objects in behavioral pattern identification," in *Proc. 1st Int. Conf. Pattern Recogn. Mach. Intell.*, Kolkata, India, Dec. 20–22, 2005, pp. 720–725.

[33] J. Bazan, N. H. Son, and M. Szczuka, "A view on rough set concept approximations," *Fundam. Inf.*, vol. 59, no. 2–3, pp. 107–118, 2004.

[34] J. Bazan, "Hierarchical classifiers for complex spatio-temporal concepts," in *Transaction Rough Sets IX* (Lecture Notes in Computer Science), vol. 5390. Berlin, Germany: Springer-Verlag, 2008, pp. 474–450.

[35] R. Guarino, "Toward a formal evaluation of ontology quality," *IEEE Intell. Syst.*, vol. 19, no. 4, pp. 78–79, Jul./Aug. 2004.

[36] P. Wonka, M. Wimmer, F. X. Sillion, and W. Ribarsky, "Instant architecture," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 669–677, 2003.

[37] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Dordrecht, The Netherlands: Kluwer, 1991.

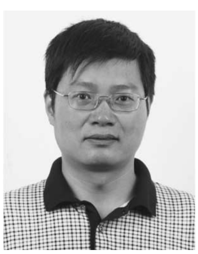
[38] V.-N. Huynh and Y. Nakamori, "A roughness measure for fuzzy sets," *Inf. Sci. Inf. Comput. Sci.*, vol. 173, no. 1–3, pp. 255–275, 2005.

[39] Y. Yang and R. John, "Roughness bounds in set-oriented rough set operations," in *Proc. FUZZ IEEE*, 2006, pp. 1461–1468.

[40] R. Jensen and Q. Shen, "Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 12, pp. 1457–1471, Dec. 2004.

Yong Liu received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2001 and 2007, respectively.

He is currently a Lecturer with the Institute of Cyber-Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has authored or coauthored more than 30 research papers in machine learning, computer vision, geographic information system, intelligent computer-aided design system, and granular computing. His research interests include machine learning, robotic vision, information processing, and granular computing.



Yunliang Jiang received the B.S. degree in mathematics from Zhejiang Normal University, Jinhua, China, in 1989 and the M.E. degree in computer science and technology and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 1997 and 2006, respectively.

He is currently a Professor with the School of Information and Engineering, Huzhou Teachers College, Huzhou, China. His research interests include geographic information systems, artificial intelligence, and information fusion. He has authored or coauthored more than 20 papers in journals, such as *Lecture Notes in Artificial Intelligence*, *Lecture Notes in Computer Science*, *Pattern Recognition*, *Artificial Intelligence* (in Chinese), etc.

Lican Huang received the Bachelor's degree in biology from Nanchang University, Nanchang, China, in 1982, the Master's degree in modular biology from Hangzhou University, Hangzhou, China, in 1984, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, in 2003.

He is currently a Professor and the Director of Network and Distributed Computing, Zhejiang Sci-Tech University (ZSTU), Hangzhou, and a Guest Professor with Beijing University of Posts and Telecommunications, Beijing, China. Prior to ZSTU, he was a Senior Research Associate with the School of Computer Science, Cardiff University, Cardiff, U.K. He was a Technical Leader or Department Manager with several companies before joining Cardiff University, where he was engaged in developing many large software systems. His research has been focused on challenges about Grid and peer-to-peer computing. He has worked on e-Science and Grid computing since the beginning of the 21st century. He has authored or coauthored more than 70 technical papers in various conferences and refereed journals.

Dr. Huang has been a Chair or Program Committee Member of many international conferences. He was honored in Marquis' *Whos Who in the World 2006*, Marquis' *Whos Who in the Science and Engineering 2006–2007*, and Marquis' *Whōs Who in Asia 2006–2007* due to his achievement of proposing Virtual and Dynamic Hierarchical Architecture for e-Science and Grid and Virtual Hierarchical Tree Grid Organizations (VIRGO) protocols.