A Practical Agent-Based Approach for Pattern Layout Design

Cunhao Fang[1], Song Cao[2]

[1] Tsinghua National Laboratory of Information Science and Technology
[2] School of Software, Tsinghua University
100084 Beijing, P. R. China
fangch@tsinghua.edu.cn

Abstract. This paper explores and discusses the application of Software Agent in pattern layout design. First we introduce Pattern Decomposition Representation Model (PDM). By combining Agent technology with the PDM, we also propose and analyze an Agent-Based Pattern Design Model (ABPDM) and present the implementation of the Agents and their cooperation mechanism. Results show that ABPDM is an effective approach for the design of serial and regular pattern.

1 Introduction

In some fields of application in pattern design, such as pattern design of fabric and carpets, there are some principles that are valuable in computer-aided design. First, the elements of pattern, i.e. pattern primitives, are reused frequently. Second, different pattern types contrast sharply and their structures are regular. Thus, the design activities can be formularized clearly. All these would be helpful in automatic and intelligent design.

This paper explores and discusses the application of Software Agent in pattern layout design. First We introduces Pattern Decomposition Representation Model (PDM). Then the Agent-Based Pattern Design Approcah is proposed and the implementation of Agents is presented at the end. We propose an Agent-Based Pattern Design Model (ABPDM), five types of Agents are defined according to the Pattern Decomposition Model (PDM). Each Agent takes charge of specific work automatically. With the cooperation of the Agents, pattern design can be accomplished more quickly. Results show that Agent technology is an effective resolution to the pattern design.

---

* This paper is sponsored by National Natural Science Foundation of China (NO. 90604027)

2 Pattern Decomposing Model

By taking the advantages mentioned above, we introduce a model named "Pattern Decomposing Model" (PDM). In this model, the reusable parts of a pattern are extracted as pattern primitives and organized in a Primitive Database. In the meantime, a module, which is separated from specific data, is defined with the abstract structure of the pattern. Therefore, the procedure of pattern design is reduced to two steps: module adjusting and primitive selecting and editing, which enable the changing of the operated object from single line or stroke to the whole primitive. This PDM model also improves the reusability of primitives and reduces the interactive complexity counted in traditional pattern design, so that the speed and quality of pattern generation can be enhanced significantly. PDM is illustrated in **Fig. 1.**

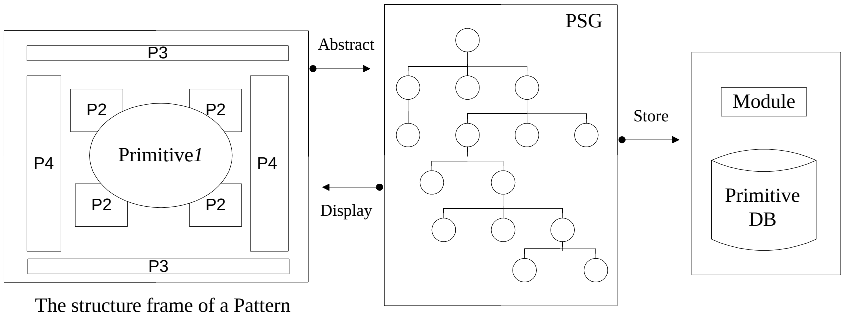

Fig. 1. Pattern Decomposing Model

Based on the model, the pattern can be described with a two-dimensional hierarchical structure, called Pattern Structure Graph (PSG), which usually is acyclic. With this structure, the primitive objects and their relationship can be depicted dynamically. Each node of the PSG represents an object, while the leaf nodes are elementary objects in which the attributes and display information are specified. The non-leaf nodes are managing objects. They indicate the display's layer of the sons and determine the traverse strategy to the descendant. By traversing the son-nodes, the manage object sends the control information to the relative elementary primitive objects, and the receptors trigger some actions according to the current state and the message they get. Therefore, with the PDM model, the interaction between primitive objects and the control mechanism can be implemented practically.

3 Agent-based Pattern Design Model

Based on the PDM model and Agent theory analyzed above, we construct an Agent-Based Pattern Design Model (ABPDM), in order to support higher reusability, reduce interactive complexity and improve intelligence of design. The architecture of ABPDM is illustrated in **Fig. 2**.

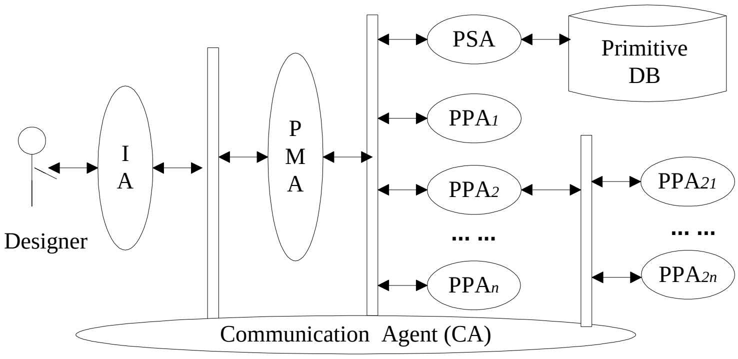

Fig. 2. Architecture of ABPDM

The key component of ABPDM is Agents, which can be divided into five types:

- ***Pattern Primitive Agent (PPA)*** maps the leaf node in PSG and stands for pattern primitives.
- ***Primitive Manage Agent(PMA)*** maps the root in PSG, and is responsible to maintain and harmonize all PPAs.
- ***Interface Agent(IA)*** is responsible to receive the users' input and convert it to PMA to process. IA also serves to PPA's adaption process.
- ***Primitive Search Agent(PSA)*** manages the Primitive Database and automatically selects the primitives according to user's design requirements.
- ***Communication Agent(CA)*** establishes a route and communication between Agents, meanwhile, CA also filters the messages and controls their priorities.

4 Implementation of Agents

4.1 Pattern Primitive Agent: PPA

PPA corresponds to the leaf node in PSG. This kind of Agent only has the representative information of primitives, including some relative computing operations. PPA mainly provides service for the upper Agents.

In pattern design systems, there are three types of primitives used frequently: vector primitives, point-array primitives and literal primitives. According to the type of primitives, PPA manages different primitive data and realizes atomic operations respectively, such as displaying, rotation and translation.

4.2 Primitive Manage Agent: PMA

PMA corresponds to the root in PSG. This kind of Agent is a control Agent, responsible for maintaining and harmonizing all PPAs. By organizing and managing them based on pattern distribution knowledge and the interactive requests submitted by designer, PMA get PPAs to cooperate in order to accomplish some design tasks.

PMA maintains and manages all the PPAs. Its functions fall on the following two aspects:

Task Disassembly: PMA acknowledges the design task received from IA and then disassembles it. With Message Mechanism, the subtasks are submitted for processing.

Coherence Maintenance: PMA monitors all the state changes of PPAs. Based on the internal constraints of Agents, PMA maintains the whole coherence of pattern.

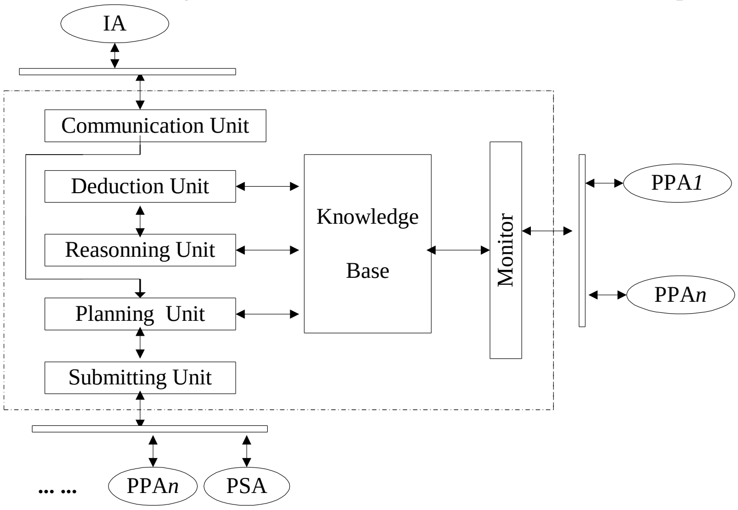

Fig. 3. Implementation of PMA

The workflow of PMA is illustrated in **Fig. 3**.

Communication Unit: It receives the messages from IA.

Deduction Unit: Based on the Knowledge Base, this unit deduces the messages and assembles the design tasks.

Knowledge Base: It stores the requisite knowledge used in deduction, and mainly includes the following:

1) System's Working Status: the statuses of all the registered IA, PPA and CA, such as their position and dimensions.

2) System's Abilities: every Agent's action abilities (operations' prototype) and the history log recording their accomplished tasks.

3) A Set of Predefined Constraints of Pattern Coherence: some constraints of pattern construction, such as the central radiation constraint contained in some pattern; also can be some constraints of field knowledge, for example, the square continuity constraint of pattern in textile industry.

4) Rule Set: It is a set of generation rules. They indicate a series of operations under different prerequisites in order to satisfy the Always constraint. For example:

By defining the internal constraints, the whole coherence of the pattern can be maintained automatically. This method is useful in supplying the gap of messy produced by PDM.

Reasoning Unit: On the basis of the coherent constraints defined in Knowledge Base and the generation rules in Rule Set, this unit maintains the coherence of pattern automatically by reasoning.

Planning Unit: It disassembles the tasks according to the Knowledge Base, and then plans and arranges those subtasks.

Submitting Unit: Using the message mechanism, the unit submits the subtasks to relevant PPAs and PSAs.

Monitor: Snoop on every PPA's state and update the System's Working Status in Knowledge Base in real time.

4.3 Interface Agent: IA

Collaborating with the operating system that manages the human-machine interactive hardware, IA takes the responsibility of receiving the users' inputs and converting them into design tasks. Then, these tasks would be passed on to PMA for processing. IA also serves PPAs and displays the graphs represented in them on the hardware.

4.4 Primitive Search Agent: PSA

PSA manages the Primitive Database and realizes the search strategy accordingly. Based on the user's searching requests, PSA automatically selects the primitives that satisfy the design demands. With the results, PPA is assembled and provided to PMA for pattern distribution. If necessary, PSA would give a candidate set for user to choose.

4.5 Communication Agent: CA

ABPDM does not implement the point-to-point communication between Agents. Instead, all messages are pasted on a blackboard managed by CA. At first, CA establishes a route between Agents, then delivers or broadcasts messages through it. Thus, CA also filters the messages and controls their priorities.

5 Conclusion

In this paper, some research work in the field of applying the Agent theory in the pattern CAD domain is introduced and discussed, which is a new topic. Utilizing the automation and intelligence of Agent, and combining them with the regularity and reusability of pattern design, we propose the Agent-Based Pattern Design Model (ABPDM). In this model, five types of Agents are defined according to the Pattern Decomposing Model (PDM): Pattern Primitive Agent (PPA), Primitive Manage Agent (PMA), Interface Agent (IA), Primitive Search Agent (PSA) and Communication Agent (CA). These Agents construct a hierarchical structure that is similar to PDM's, and each Agent takes charge of some specific work automatically. With the cooperation of the Agents, pattern design is accomplished more quickly, while keeping the quality of opus. Results show that Agent technology is an effective method for pattern design.

References

1. Polzleitner W,et al . Invariant pattern location using unsupervised color based perceptual organization and graph2based matching. 2001 International Joint Conference on Neural Networks Proceedings [C] .Washington DC ,IEEE Press ,2001. 594 – 599;
2. Marculescu, D.; Marculescu, R.; Khosla, P.K. "Challenges and opportunities in electronic textiles modeling and optimization" Design Automation Conference, Proceedings. 39th , 2002 Page(s): 175 –180
3. Rantanen J ,et al . Smart clothing for the arctic environment. the Fourth International Symposium on Wearable Computers Proceedings [C] . Atlanta ,GA ,IEEE Press2000. 15 – 23;
4. Li Sheng ,et al . Application of pattern emulation on weave CAD automatization, the 3rd World Congress on Intelligent Control and Automation Proceedings [C] . Hefei ,China , IEEE Press ,2000. 2412 -2416;
5. Fang Cun-hao, " The Textile Oriented Pattern Database Design" the bachelor degree thesis of Zhejiang University, 1998(in Chinese);
6. Pan Yunhe, "Computer Art", Science Press, 1985(in Chinese).