

An Object-Oriented Design Support Tool Based On Design Patterns

6 S - 1

Bing Zhang† Yoshiaki Fukazawa‡

†Academia Sinica & Centre for Informatics, Waseda University

‡School of Science and Engineering, Waseda University

1 Introduction

With the development of object-oriented design technique, some experts record design experience in a succinct form that people can use effectively, that is so-called Object-Oriented Design Patterns[1]. Utilization of design Patterns is a promising technique for achieving widespread reuse of software architectures. These patterns capture the static and dynamic structures and collaborations of components in successful solution to problems that arise when building software in domain applications (but they are not domain-specific). "Design Patterns" makes object-oriented design more flexible, elegant and ultimately reusable.

Thus, it is a significant task to let more and more developers of new systems comprehend and master the well-proven object-oriented design patterns. Therefore an Object-Oriented Design Support Tool(OODeST) has been developed. Our objective is not only to help users have a better understanding of various design patterns with different application background, but more important, to help promote good object-oriented design, and choose more reusable design alternatives from design pattern base.

2 Characteristics and Structure

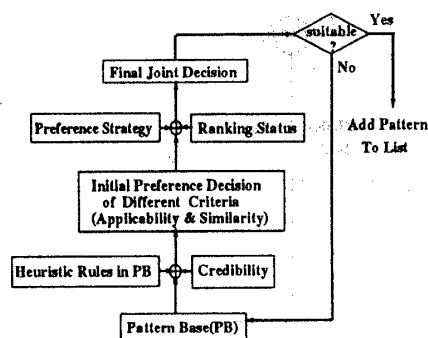
OODeST has a friendly graphical user interface to assist occasional and inexperienced users. It consists of three main modules which can be run concurrently.

First is a pattern-based diagram editor. Though acting as an accessory, it is the foundation stone of OODeST. Not only all supporting graphs of OODeST are drawn upon it, but also it allows skillful developers to fulfill their own object-oriented graphical representations.

The second is a 23-design-pattern overview module. You may have an oversight of their relationships and navigate anywhere you like to get further information.

The third is mainly a decision-making module. Combining with heuristics and system-guided user interaction, users may retrieve and select well-satisfied design patterns. Figure 1 shows the selection process scheme. In this figure, the arrows indicate the flow of information and the symbol \oplus means that two or more knowledge modules together derive one or more outcomes directly.

Figure 1: The information flow diagram of decision making



3 Diagram Editor Module

Based on OMT(Object-Model-Technique), OODeST adopt the notation of Class Diagram which depicts classes, their structure and the static relationships between them. For example, a class is denoted by a box with the class name in bold type at the top. The key operations of the class appear below the class name. Any instance variables appear below the operations. An object reference representing a part-of or aggregation relationship is indicated by an arrow-headed line with a diamond at the base.

OODeST provides a palette of graphical tools to represent the components of class diagram. There is no need for users to draw manually. Users can conveniently select these tools, interact with the system, then get an automatically generated object-oriented design drawing.

4 Overview Module

This module acts like a tutorial. It records the detailed information about a design pattern, such as Intent, Motivation, Applicability, Structure, Participants, Collaborations, Sample Code etc.. For example, Intent is a short statement that answers the following questions: what does the design pattern do? what is its rationale and intent? what particular design issue or problem does it address.

However, with 23 design patterns in the catalog to

choose from, it might be hard to find the one that addresses a particular design problems, especially if the catalog is new and unfamiliar to users. Hence it is necessary to organize design patterns by some ways which help users to learn and concentrate on patterns faster.

One way is to classify design patterns by Purpose as Creational, Structural and Behavioral[1]. Creational patterns(5) concern the process of object creation, Structural patterns(7) deal with the composition of classes or objects. Behavioral patterns(11) characterize the ways in which classes or objects interact and distribute responsibility. First users need to read through each pattern's intent to find relevant patterns and may use Purpose classification scheme to narrow search scopes later.

Another way to organize design patterns is according to how they reference each other[1]. This module present users with an overview relationship diagram. You will have an overall concept of how they compare and relate. Studying these relationships can help direct users to the right pattern or group of patterns. By pressing buttons of interested patterns, you may also navigate and have an insight into what they do and when to apply them.

5 Decision Making Module

At first, we distinguish users by their experience in object-oriented design with level of Expert, Medium or Novice. They are assigned with credibility, expert is 100%, medium is 90% and novice is 80%.

Then we adopt two criteria to make a joint decision whether an individual pattern suits users' requirement, namely in Applicability and in Motivation Similarity.

5.1 Applicability and Heuristic rules

Applicability is the summary of those like "what are the situations in which the design pattern can be applied"? "what are examples of poor designs that the pattern can address"? and "how can you recognize these situations"?

Considering each design pattern, there are generally several items which function together and describe the aspects of applicability. Not all of items are of same consequence. As a result, we use a heuristic method to assign different weight to every item, which must ensure only when the most crucial items are selected, may the corresponding design pattern be considered for next step. Every design pattern has a heuristic rule respectively.

5.2 Motivation Similarity

Motivation is about application background, a scenario that illustrates a design problem and how the class and object structures in the pattern solve the problem. The scenario will help users understand the more abstract description of the pattern.

Considering users' ranking status in object-oriented design, and abstract description of applicability as well, we think "Motivation" might be a big hand to users' comprehension of design patterns, especially for those unexpertised. Only users know their requirement best, therefore it is easier for them to affirm what similarity is between their needs and the background motivation of the design pattern.

Motivation Similarity comes into being as a supplementary criterion to Applicability decision.

5.3 Preference strategy

Based on a scale of 10, and considering experience status, we adopt the preference strategies as follows:

Strategy 1: if the users rank in Expert, then the outcome decision of Applicability will occupy much more proportion than that of Similarity(say 7 to 3).

Strategy 2: if the users rank in Medium, then the outcome decision of Applicability will occupy some more proportion than that of Similarity(say 6 to 4).

Strategy 3: if the users rank in Novice, then the outcome decision of Applicability will occupy much less proportion than that of Similarity(say 2 to 8).

Once the joint decision value is over the threshold, the relevant design pattern may be a design candidate of the target system.

6 Conclusion

Our ODeST was built on SUN workstation by Motif. Now ODeST is being evaluated by some designers. By the feedback of their evaluation, ODeST will be appended some complementary facilities.

Furthermore, the current set of design patterns is only a fraction of those experts know. There are still many domain-specific patterns to be supplied to ODeST's Pattern Base. We wish soon to build a larger knowledge base, and combine more experts' experience to perfect the retrieval strategy and heuristic rules. Moreover, it should be capable of supporting straight answers without pre-understanding and of supporting collaborative decision intelligently.

References

- [1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns-Elements of Reusable Object-Oriented Software", Addison-Wesley, Mass. 1994
- [2] S. T.C. Wong, "Preference-Based Decision Making for Cooperative Knowledge-Based Systems", ACM Trans. on Information Systems, Vol.12, No.4, Oct. 1994, pp407-435
- [3] D. C. Schmidt, "Using Design Patterns to Develop Reusable Object-Oriented Communication Software", Comm. of the ACM, Oct. 1995, Vol.38, No.10, pp65-74