Relatório Técnico

Núcleo de Computação Eletrônica

Remote Learning of Design Patterns

H. Gandra
C. Lima

NCE - 04/02

Universidade Federal do Rio de Janeiro

Remote learning of design patterns

Henrique GANDRA
IBCCF/IM/NCE/UFRJ
Rua Siqueira Campos, 238 / 705
22031-070 – Copacabana
Rio de Janeiro, RJ, Brasil
T: 55 21 2548-5878
hgandra@ufrj.br

Cabral LIMA
DCC/IM/UFRJ
Av. Brigadeiro Trompowsky s/n, CP 2324
Cidade Universitária / Ilha do Fundão
20001-970 Rio de Janeiro, RJ, Brasil
T: 55 21 2598-3168 / F: 55 21 2598-3156
clima@dcc.ufrj.br

Abstract
Since the early 1990's, the software community focuses on the significant benefits of software reuse. Nowadays, the design patterns in object-oriented systems have been considered one of the best strategies in order to improve software design, based on the idea of reusable solutions to design problems. A pattern addresses a recurring design problem that arises in specific design situations and presents an abstract solution to it. Although its application is not simple (since they have a complexity cost), it describes the problem, the solution (when it applies the solution) and its consequences. So, it is necessary to let important learning period to help the software designers in the learning process of using correctly design patterns.

Our research team is actually designing and developing a flexible platform model aimed to computer-based distance learning, which is going to support plug-in of intelligent learning systems.

In this paper we describe a tutoring system to support presence and distance learning of design patterns in object-oriented systems - LeSOOP (Learning System of Object-Oriented Patterns). This system uses uniform and contextual diagnosis processes in order to help the software designers in the correct application of design patterns. We also discuss about the plug-in and integration features of our system in a distance-learning platform.

Keywords
Software engineering; distance learning; tutoring system; design of computer-based distance learning platforms.

Introduction
Human's learning is considered as a slow, complex and large process by most of the authors (Michalski et al, 1983) because it involves cognitive processes not all understood. In distance-learning system, the complexity is still bigger due to the differences in space and time between teaching and learning processes. Computer-based distance learning, using new information and communication technology, has an additional complexity to students and teachers, who sometimes have little or no technological background. This makes its usage tougher.

The technological tools developed to computer-based distance learning, are usually projected to work in educational platforms built-up in order to group them. The use of these platforms usually

has shown that the emphasis is rather in technological aspects, than in cognitive aspects or learning strategies. Furthermore, platforms' users should never get the feeling of being lost, they should always be in control of the interaction with the application and the application should guard against possible misuse by the users. Based on it, researchers have been focus the most elaborated development of platforms with friendly interfaces and intuitive, easy usage, which ones can guide the users (students and teachers) in their teaching and learning tasks, dealing with contextual diagnostic techniques of artificial intelligence, etc.

In this article, first of all we detail the importance of design patterns in object-oriented systems to reusable software's elaboration, which is a goal even more persecuted by the Software's Engineering Community. Following, we present the LeSOOP system (Learning System of Object-Oriented Patterns) a computer-based system aimed to support presence or distance learning of design patterns in object-oriented systems.

This System uses uniform and contextual processes of diagnostic aimed to help designers of software in the correct application of design patterns. LeSOOP will be connected to the Acadia system (Correa and Martins, 2002) - a learning-teaching environment with friendly graphic interface - and to the Dedalus platform (Paiva and Faissal, 2002) – a platform that uses peer-to-peer technology and shared resources, with desirable characteristics named above.

Design Patterns

Design patterns began to be recognized more formally in the early 1990s by Richard Helm, in 1990, and Erich Gamma, in 1992 (Cooper, 1998), and they came to assure themselves in 1995 with the publication of the book "Design Patterns – elements of reusable object-oriented software". During the 90's the Software Engineering Community recognized the benefits of the use of design patterns and its application has been rapidly generalized. Nowadays design patterns are too spread and, consequently, too known by the community, however, their application is not ordinary because there is a cost of complexity indeed in it: knowing how to use correctly the design patterns in software projects, the ones which are a little more elaborated require a significant period of learning by the designers, who must know how to evaluate either the right use of these design pattern or the real necessity of their usage in that specific application.

The definition of (Gamma et al, 1995) about design patterns is: "*The design patterns are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context*". Therefore, a design pattern reach abstract reusable strategies by interactions among objects and it describes them as a possible solution of a problem included in a specific context: "*Each design pattern focuses on a particular object-oriented design problem or issue. It describes when it applies, whether it can be applied in view of other design constraints, and the consequences and trade-offs of its use*" (Gamma et al, 1995). Design patterns are a robust way of reuse of solutions of problems, which show up constantly in several applications. Being a reuse of ideas, one of the strongest aspects about object-oriented, it is easy to realize the intense growth of the interest and the usage of design patterns.

To apply a design pattern, the designer needs to understand pretty well the pattern's abstract structure, its benefits and consequences, as well as the communication's rules among objects to,

afterwards, adapt this pattern to his problem. This is one of the mainly reasons of necessity there is a learning tool been consistent and friendly able to aid a learner in the task of develop object-oriented systems, including the right utilization of design patterns. Some features (which help designers in use and implementation of design patterns available in the market) don't seem to assist effectively the designers in the task (Motelet, 2000). In this way, the choice and implementation of a design pattern, in order to solve a program problem, unfortunately still depend on the analyst's or programmer's skill and sensitivity.

A tutorial system described in the theses called "*A contextual help system for assisting OO software designers in using design patterns*" (Motelet, 2000), which was implemented in Java, focus on offering an on-line help system to the designer in the learning task about conceptions of six design patterns from the catalogue presented in (Gamma et al, 1995): the clue is help the designer choose one of the patterns to solve a specific problem.

LeSOOP system

The described system, in this paper, is destined to distance learning and it is based on one side (Motelet, 2000) – in what it concerns the design patterns presentation, their consequences and relationships - on another side a system in development, which focuses on the remote learning of design patterns. This system is united to the Dedalus platform through Acadia system.
The system could use also in presence learning, extending its possibility of use.

We have been spread out this prototype's extension developed in (Motelet, 2000) implementing others seventeen design patterns, which compose a suggested collection in (Gamma et al, 1995). This new system presents a graphic interface where a window introduces a menu bar in which we can choose one of the 23 design patterns (Picture 3).
In the panel, there is a visualization of each design pattern and these are introduced as well as the most important information in order to learn about the respective pattern. These information were objects of study from several researchers, including (Motelet, 2000) such as:

- *Name*: The pattern's name conveys the essence of the pattern succinctly.
- *Purpose*: describes when to apply the pattern, it explains the problem and its context.
- *Result*: The main advantage of using the pattern.
- *Variation*: Another viewpoint on the design pattern's intent.
- *Mean*: The mean by which the design pattern attempt to its purpose.

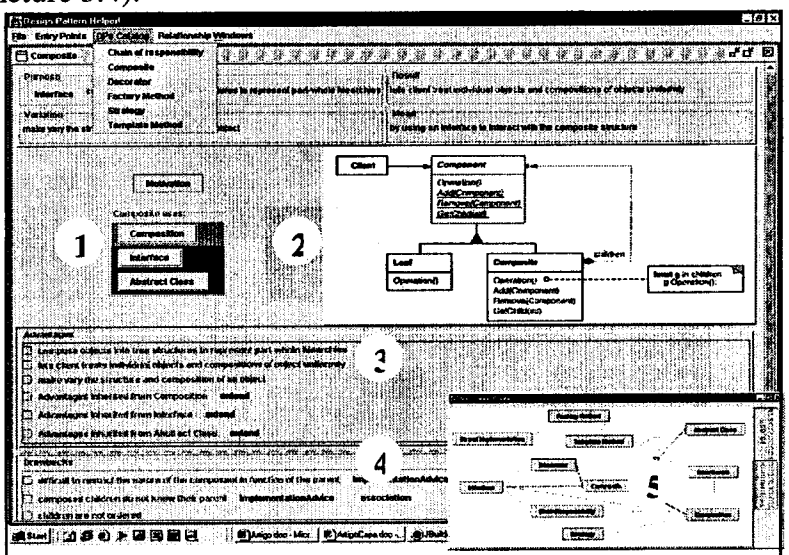
Defining in which principle of object-oriented project this pattern is based on is another information considered as a big utility to designers by several authors. The detailed principles in the system are:
Inheritance: a relationship that defines one entity in terms of another. Class inheritance defines a new class in terms of one or more parent classes. The new class inherits its interfaces and implementation from its parents. **Composition**: assembling or composing objects to get more complex behavior. **Interface**: The set of all signatures by an object's operations. The interface describes the set of requests to which an object can respond. **Abstract Class**: a class whose primary purpose is to define an interface. An abstract class defers some (in opposition to an interface) or all (like an interface) of its implementation to subclasses. An abstract class cannot be

instantiated. **Direct Implementation**: in opposition to an interface and abstract class, which defer the implementation to other classes.

The **LeSOOP** system presents the group of principles in which the pattern is based on. This information contains links, which guides the users to a more formal description from each of one, these principles (Picture 3.1). At the right side of the principles, a design pattern's structure is presented, graphic representation of standard classes, using an appointment based on Object Modeling Technique (OMT) (Rumbaugh et al, 1991) (Picture 3.2). In another panel, below, advantages of application of the pattern and its alternatives are detailed (Picture 3.3). In the last panel of this frame, the disadvantages of application of the pattern, information of implementations, alternatives, associations with other patterns, which will work as an antidote of these disadvantages, are presented (Picture 3.4).

Another aspect, which eases the comprehension of these design patterns, is perceived the relationships among them, their differences and similarities (Motelet, 2000).

Looking forward an importance of comprehension of these relationships, the system illustrates, in its interface, a window where existent relations among patterns are underlined: it makes things easy to the users in navigation, such as consulting/ checking up these patterns and related principles (Picture 3.5).



Picture 3 – System's Graphic Interface (apud Motelet, 2000)

The system has also a range of examples of applications of design patterns, all implemented in Java, with explanation step by step in how to use the conceptions of each pattern in the solution of a contextual problem: the advantages and disadvantages (from the chosen broach) are shown too. This range serves also to take doubts, which perhaps go on after the navigation through the theoretical conceptions. It eases the comprehension of them by the designer and contributes to the learning process become more efficient and realistic.

The LeSOOP system also has a basis of proposed exercises and a key exercises one. During the learning process, dealt by LeSOOP, the user can require a self-exercise about the design pattern's use. The system chooses an exercise from the basis unsuitable and sends it to be solved by the student (in case there are no information about the knowledge level of the student). The solutions associated are linked, but not presented, in order to compare them to a developed solution by the user (by the overlapping method). A subsystem of intelligent diagnostic is acting to try to identify the stage of knowledge the student is on, during an exercise's solution: it serves to classify the

student related to the knowledge profile about design patterns. Following, exercises proposed to that student will be chosen according to the respective profile established by the system.

The Platform

We are interacting with a research group of Software Engineering, which has been developing two projects applicable in distance learning: Acadia and Dedalus projects.

The Acadia project's propose is offer a learning/teaching environment with graphic interface which motivates the collaboration, creativity and learning of the candidates, making a human-computer interaction easier (Correa and Martins, 2002). The system is composed by web pages. The sites simulate virtual environments as a classroom, library and secretary, for instance. Another goal from it is increase collaboration among students and teacher/students, opening lines to discussions about doubts, suggestions and criticisms.

The Acadia project is developed about a graphic platform called Dedalus. Dedalus (Paiva and Faissal, 2002) is a platform, used as a basis to other applications involved to peer-to-peer technology and shared resources. One of the Dedalus platform's applications is the usage of tools aimed to distance learning. In terms of architecture, Dedalus is federated, which brings benefits according to the performance, once the customers will be able to communicate themselves independently of a server. Dedalus has a great advantage: it is ease to use. People with little technological knowledge are able to build their own pages up, setting rid of sophisticated components easily done by a friendly interface.

Conclusion

In this paper, we have detailed an importance of the design patterns to an object-oriented developed system to objects as an important step in the reuse software's project's generation. We have also introduced the LeSOOP system, an on-line system which eases the comprehension and knowledge of conceptions, aiming the correct application of design patterns, using contextual and uniform processes of diagnostics through a bank of proposed exercises, whose solutions, developed by the student, serve as a basis to a specification of his knowledge level. We have already focus on the importance of connecting the LeSOOP to the Acadia in the Dedalus platform, developed to distance learning system application, using a peer-to-peer technology and friendly interfaces able to guide the user during these systems' usage. This connection avoids lost users during a learning/teaching process.

Bibliography

Cooper, J. W. (1988); The Design Patterns: Java Companion; Addison Wesley Design Patterns Series.

Correa, B. O. B. and Martins C. G. (2002); Ambiente de ensino-aprendizagem com ênfase em comunicação gráfica, desenvolvido em plataforma distribuída; Núcleo de Computação Eletrônica / Universidade Federal do Rio de Janeiro – Brasil.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995); Design Patterns. Elements of Reusable Software; Addison-Wesley, Reading, MA.

Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (1983); Machine Learning: An Artificial Intelligence Approach; Morgan Kaufmann Publishers, Inc.; pp 26-27.

Motelet, O. (2000); A contextual help system for assisting OO software designers in using design patterns; MSc thesis. EMOOSE project. Vrije Universiteit Brussels – Belgium / École des Mines de Nantes – France / Universidade Federal do Rio de Janeiro – UFRJ.

Paiva, D. M. and Faissal, G. M. (2002); Uma Plataforma Java para Aplicações Peer-to-Peer; Núcleo de Computação Eletrônica / Universidade Federal do Rio de Janeiro – Brasil.

Romanczuck-Requile, A., Lima, C., Kaestner, C., and Scalabrin, E. (1998); A contextual help system based on intelligent diagnosis processing aiming to design and maintain object-oriented package; Lecture Notes in Computer Science; issn 0302-9743; Springer-Verlag, pp. 64-65.

Rumbaugh, I.; Blaha, M.; Premerlani, W.; Eddy, F. & Lorenson, W. (1991); Object-Oriented Modeling and Design; Prentice Hall, Englewood Cliffs, NJ.