Assessing Object-Oriented Technology Skills Using an Internet-Based System

Ahmed Seffah
Computer Research Institute of Montreal
550, Sherbrooke West, H3A 1B9
Montreal (Quebec) Canada
Phone: +1(514) 840-1235
ahmed.seffah@crim.ca

Moncef Bari
University of Quebec at Montreal
PO 8888, suc. Centre-Ville Montreal
(Quebec) Canada - H3C 3P8
Phone: +1(514) 987-3000
bari.moncef@uqam.ca

Michel Desmarais
Computer Research Institute of Montreal
550, Sherbrooke West, H3A 1B9
Montreal (Quebec) Canada
Phone: (514) 840-1235
michel.desmarais@crim.ca

ABSTRACT

In this paper, we describe a Web-based system that defines training needs for object-oriented developers by identifying the strong and the weak areas of their knowledge and skills. The system is based on the use of two tools, GAA [8] and UKAT [3], developed at the Computer Research Institute of Montreal (CRIM). UKAT (User Knowledge Assessment Tool) uses a state-of-the-art knowledge assessment method to create a user profile of the proficiency in a subject domain. GAA (Intelligent Guide) is a Web-based training system that uses the UKAT to personalize a training course and facilitate self-learning.

Keywords

Internet-based training, object-oriented technology, knowledge and skills assessment, self-learning resources.

1. DIFFICULTIES IN LEARNING OBJECT-ORIENTED TECHNOLOGY

Training is a major challenge facing software development organizations in their effort to migrate towards object-oriented technology [4]. Theoretical classroom presentations introduce concepts and present simple case studies in an attempt to facilitate understanding these concepts. Textbook and case studies are also limited and restricted in their scope to reflect true-life experiences. According to [9], what must be learned are not only concepts and skills, but also the mindset that will facilitate an effective and efficient transition towards object-oriented technology and its practice.

To achieve this, the following difficulties must be overcome:

- Difficulties in learning OO fundamental principles. Hohmann [7] indicated that students find it easy to differentiate conceptually between inheritance, aggregation and, association relationships. They (learners), however, find it extremely difficult to create solutions that will correctly combine these relationships.
- Difficulties in learning OO design and techniques. Rosson [11] observed that the problem of learning object-oriented design and programming techniques is compounded by the fact that expert developers are highly confident in their ability to learn new programming languages and techniques.
- Difficulties in learning an OO programming language. Evidence gathered from our day-to-day experiences demonstrates that the effectiveness in OO programming depends directly on the user's level of skills regarding the language and the acquired experience of OO programming principles.
- Difficulties in using OO development tools. In most cases, an OO development environment offers a wide array of powerful, but complex tools and services. An effective teaching strategy should demonstrate how to use the services and explicitly highlight their relationships with OT concepts and OO design techniques.

The following observations make these difficulties even more critical. According to Lato [5], taking an OOA/OOD course and an OOP course does not appear to be recommendable. Teaching programming before design will lead to bad design practices, which would require effort to unlearn. Teaching design before programming could result in a sterile course where students are unable to fully apply what they learn. Whitelaw [13] indicated that at least some of the problems associated with learning object-oriented programming occur because we do not usually think of everything as an object and objects are not all of the same type.

The goal of the system that we developed is to offer an environment in which concepts and skills can be self-assessed and self-learned in order to overcome the difficulties listed above.

The system combines the two following approaches:

- Intelligent Training Systems (ITS). ITS can indicate the best way to solve a problem and/or organize course content [12] and/or motivate and encourage the learner during a learning session. ITS is also able to establish a review of the learner's interaction and/or criticize the learner's solutions and/or cooperate and compete with the learner during problem solving [2].
- Internet technology has the potential to be used either internally by companies developing their own corporate Intranet [1, 10] or externally with an Extranet targeting

ITiCSE '99 6/99 Cracow, Poland
© 1999 ACM 1-58113-087-2/99/0005...$5.00

their client base. Internet technology is also presented as an ideal vehicle for helping software developers perform their jobs, share expertise on how to develop well and at the same time learn through several kinds of resources, such as real-life, "best-use" case studies, demonstrations, courseware, etc.

2. A BRIEF OVERVIEW OF THE GAA ARCHITECTURE

The long-term goal of GAA has been to develop a generic self-learning guide that could provide user knowledge assessment and pedagogical feedback for a number of domains. We especially focussed on domains that require the user to master a number of concepts to achieve a satisfactory level of skills in the domain. Such domains are object-oriented technology and software engineering.

The GAA architecture is based on the typical Internet client/server infrastructure and the Intelligent Training System architecture. It comprises the following four components:

- The Intelligent Training System
- Web Server (HTTP)
- The Web manager generates an HTML page based on the pedagogical engine decisions and the information contained in the subject database. The Web manager is a set of Common Gateway Interface scripts
- A Web Browser-Based User (Learner) Interface

The ITS used in GAA includes three main components [8]:

- The knowledge network contains Knowledge Units (KU) or domain entities such as OO principles and Java programming techniques. KU are connected via multiple domain relationships (e.g. prerequisite). The next section describes the structure of the knowledge network.
- The user knowledge assessment tool is the central component of the GAA. It is responsible for identifying strong and weak points.
- The pedagogical engine is responsible for interpreting the knowledge state of the student and, depending upon the currently selected goals and the state of interaction with the user, it determines what should come next in a session.

3. KNOWLEDGE NETWORK

In this section, we describe the knowledge network organization by using the *Building a software application using Java* (J++_BApp) course as an example.

In order to detect the strong and weak areas of a developer's skills about Java programming, the fundamental concepts (that we called knowledge units) are identified. Examples of concepts in the J++_BApp course are:

1- Definition of Applets
2- Creating an Applet User Interface
3- Communicating with Other Programs
4- Understanding Applet Capabilities and Restrictions
5- Completing an Applet
6- Common Applet Problems and Their Solutions

Concepts in the knowledge network can be connected through different kinds of relationships (i.e. prerequisite). For instance, the *definition of applets* concept is a prerequisite to concept 4 to 6.

For each concept, a list of questions that are used during the assessment process are also part of the knowledge network. A question is always linked to at least one concept. A concept can have multiple questions linked to it. Table 1 shows a question about *applet capabilities and restrictions* concept

Question #1: Current browsers impose some restrictions on applets that are loaded over the network. Which of the following is not a restriction (An applet can:)
a) Load libraries or define native methods
b) Read or write files on the host that is executing it
c) Make network connections to hosts other than the one it originated from
d) Start any program on the host that is executing it
e) Read certain system properties
f) Invoke public methods of other applets on the same page
g) I do not know

Table 1: an example of question

Besides concept and question, solution objects are attached to questions in the knowledge network. Each solution has a description of the solution and further information (comments, advice, etc.) about the question. Further information are displayed depending the user profile. Table 2 shows a possible solution to question #1.

Solution: question #1
Description
The correct answer to this question is f.
An applet can find another applet either by looking it up by name (using the AppletContext and GetApplet() methods) or by finding all the applets on the page (using the AppletContext and the GetApplets() methods). Both methods, if successful, give the caller one or more Applet objects. Once the caller finds an Applet object, the caller can invoke methods on the object.
Further information
The java.applet package provides an API that gives applets some capabilities that applications do not have. For example, applets can play sounds, which other programs cannot yet do. However, every browser implements security policies to keep applets from doing damage. The implementation of the security policies differs from browser to browser. Also, security policies are subject to change.

Table 2: an example of solution

The GAA displays a solution for a question after a test/quiz. Our experience with GAA shows us that the solution for a question is sometimes incomplete, ambiguous and/or incomprehensible to the learner.

For this reason, we have attached Note-Box objects to any object (concept, question, and solution) in the knowledge network. A Note-Box helps to achieve a greater understanding by dispensing further information. A Note-box can point to a book, a journal, a URL, and/or provide a mail address of an expert that can be contacted for help. Table 3 shows a Note-Box attached to the Understanding applets capabilities and restrictions concept.

Note-box: Learn more about applets restriction and restrictions, consult:
- The Java Tutorial: Object-Oriented Programming for the Internet Mary Campione and Kathy Walrath, Adisson-Wesley Press, 1996
- Email to news_java@crim.ca
- Read the following FAQ on Java restrictions http ://www.coopere.com/discussion

Table 3: an example of Note-box

4. ASSESSMENT METHOD AND PROCESS

Based upon the information it receives from the pedagogical engine about which concept (knowledge unit or KU) is mastered by the user, the knowledge assessment tool will infer the likelihood (i.e. probability) that every other KU is mastered.

The knowledge assessment tool adopts the overlay approach to defining the whole domain knowledge. It assigns a state, which is a collection of numerical attribute values attached to the each knowledge network (KU). Each value indicates the likelihood of a user knowing a specific KU. In the knowledge network, implication (precedence) relationships connect also KU. An implication relationship is a gradation constraint that expresses whether a certain KU must be understood before another difficult one. The implication relationship is what enables the inferences about the mastery of KU.

The KU state is built and updated as soon as a few observations are made (e.g., questions are answered). This new bit of information may be propagated to other KU in compliance with the gradation constraints (inference structure). Further details about knowledge assessment methods are given in [3].

The GAA interacts with the learner for multiple sessions. Each login with the GAA invokes a new session. At the first login, the system recommends that the learner take a pre test. An initial user profile is thereby created. A profile is a sub network of the knowledge network. Each test/quiz taken by the learner causes a change in the user profile and drive the training and assessment process.

The GAA provides two main modes of interaction: free learning and Test/Diagnosis. No restriction is placed in interacting between these modes of interaction.

The free-learning mode makes available the knowledge network contents through a hierarchical browser (such as the file manager in Windows). Student can browse freely through the hierarchy.

The test/diagnosis mode provides learners with opportunities to obtain an evaluation of their knowledge. This diagnosis is presented in a variety of ways so that an overall and a detailed view of the progress learning the course material are made clear to the learner. The test phase is what utilizes the strength of the assessment tool to effectively and efficiently provides, through a set of question, knowledge assessment to the learner.

5. CONCLUSION

In this paper, we presented a self-assessment approach to deal with the difficulties of providing effective object-oriented training. Our system has the capability to assess the learner's knowledge state and then suggest resources, which can promote learning by OO developers.

We note the following benefits of the system. First, the system is able to identify the real aptitudes and difficulties of software developers by identifying concepts that they master and those they do not master. This can be very helpful for a project manager when identifying the best developer for a certain task requiring specific skills. Second, by using this system, developers can assess their skills periodically.

Compared to paper manuals and others training approaches [6] the system holds the opportunity to implement a distributed approach for distance continuing training that support the integration of:

- Multimedia material (image, video, sound)
- Material available on the World Wide Web, such as online bibliographies, technical reports and remote courseware
- Others training tools such as classrooms and tutorials.

However, we noted the following limits:

- The power of the system dependent on the quality of the knowledge network content and its organization.
- The feedback given by the system after each quiz is not quantitative. It is very difficult to understand some of the decisions made by the system (question selection and its relationship with the concept).

6. REFERENCES

[1] Capell, P. Distance Learning Technologies. CMU/SEI-95-TR-004. Software Engineering Institute, Carnegie Mellon University, 1995.

[2] Chan, T.W. Baskin, A.B. Learning Companion Systems. In Intelligent Tutoring Systems: at the crossroads of AI and Education. Edited by C. Frasson and G. Gauthier. Montreal, Canada, 1990.

[3] Desmarais, M. C., Maluf, A. and Liu, J. User-expertise modeling with empirically derived probabilistic implication networks. User Modeling and User Adaptive Interaction, 1996.

[4] Fayad, M., Tsai., W.T., Fulghum, I. Transition to Object-Oriented Software Development. Communications of the ACM. 39(2), 1996.

[5] Gehringer, E.T. Manns, M.L. OOA/ÓOD/OOP: What programmers and managers believe we should teach. Journal of Object-Oriented Programming. 9(6), 1996.

[6] Hermann, H. Metz, I. Teaching OO Software Engineering by Example." ECOOP'96 Educator's Symposium, 1996.

[7] Hohmann L. The First Step in Training: Analysis and Design or Implementation Language. Journal Of Object-Oriented Programming." 9(6) pp. 61-63, 1996.

[8] Khuwaja, R., Desmarais, M & Cheng, R. Intelligent Guide: Combining User Knowledge Assessment with Pedagogical Guidance. Proceedings of the ITS'96. New York: Springer-Verlag, 1996.

[9] Lato, C. Learn to Learn: Training on new Technology. Journal of Object-Oriented Programming. 9(6), 1997.

[10] Kurt Malay, K., Abdel Wahab, H., Overstreet, M.C., and all. Interactive Distance Learning over Intranets. IEEE Internet Computing, Volume. 1(1), 1997.

[11] Rosson, B. Carroll, J.M. Scaffolded Examples for Learning Object-Oriented Design. Communications of the ACM. 39(4), 1996.

[12] Seffah, A., Bouchard, R. A Discussion of Pedagogical Strategies and Resources for Software Engineering Online Training. ED-MEDIA 98 — World Conference on Educational Multimedia and Hypermedia, Germany, 1998.

[13] Whitelaw M, Weckert J. The Humanness of Object-Oriented Programming. First International Cognitive Technology Conference. August, Hong Kong, 1995.