Work in Progress - Development of a 3-Tier Architecture C++ STL ITS

Christine Lee and Mohd Sapiyan Baba
Faculty of Computer Science and Information Technology, University of Malaya
50603 Kuala Lumpur, Malaysia. christin@kbu.edu.my, pian@um.edu.my

Abstract - Most of the current Intelligent Tutoring Systems (ITSs) for programming focus on teaching students the syntax of a programming language as opposed to application. The main tutoring approach is to present a problem specification for the student to solve and followed by analysis of the solution with various feedbacks. Existing ITSs suffer from static domain knowledge and are restricted to the tutoring session. Therefore, this research proposes the development of a web-based ITS to teach the application of C++ Standard Template Library (STL) to problem solving. It is discovered that students find the C++ STL difficult due to their weaknesses in understanding various object-oriented concepts. Subsequently, the dynamic domain knowledge of the C++ STL Tutor is modeled in a 2-level hierarchical structure based on pre-requisites. Bayesian network is applied to model the student's knowledge and direct the tutoring intelligently. The ongoing development of the C++ STL ITS applies practices from eXtreme Programming methodology and J2EE technologies.

Index Terms – Bayesian Networks, Dynamic Domain Knowledge, Intelligent Tutoring Systems, Standard Template Library.

ITS FOR PROGRAMMING

Intelligent Tutoring System (ITS) is an advanced training software that mimics a human tutor by adapting its instructional approach to each individual student. It was developed to overcome the deficiencies of traditional Computer Aided Instruction systems. Users interact with the system by answering simple questions about themselves and their understanding of the domain knowledge. The information is then used to find out the user’s knowledge. This is achieved via an expert system.

ITSs to teach various programming languages from the 1980s to the current millennium have matured in the areas of the choice of programming languages, level of programming, tutoring goals and application of Artificial Intelligence (AI) techniques.

Domain knowledge modeling has progressed from predefined rules and bug catalogues to hierarchical structure and dynamic bug library entries. The earliest work on the domain knowledge falls in the field of AI. As AI has matured and demonstrated its applicability to a range of practical problems, its almost exclusive reliance on LISP and PROLOG has diminished. Consequently, ITSs for programming shift to object-oriented programming as their domain knowledge.

The level of programming taught in the current ITSs covered elementary topics that are typically found in an introductory course to Computer Programming. Undoubtedly, there is a need for tutoring materials that target the second level of programming for a course. These include application of the programming language to create data structures and solve more complex problems.

Most of the earlier ITSs have focused on teaching students the syntax of the programming language as opposed to application. The common teaching strategy adopted is to provide a program specification to solve, followed by intelligent analysis of the students’ solutions.

MOTIVATION

The review of prior work motivated this research to shift the tutoring of programming language from teaching elementary programming syntax to application. The interest in C++ is most obvious as it is widely taught in the first year of a Computing degree course. It is considered as one of the de facto programming languages to computer scientists and engineers.

Stroustrup [1] pointed out that there is a need to adopt a paradigm shift in the way we write and teach C++ programs. Using some simple examples, he argued his preferred approach to teaching C++ along with its Standard Library. Consequently, the aim of the proposed ITS is to teach students the application of the C++ Standard Template Library (STL) to problem solving.

STANDARD TEMPLATE LIBRARY

The STL is a framework collection of data structures (called containers) and algorithms designed by Alexander Stephanov and Meng Lee at Hewlett-Packard based on their research on generic programming [2].

It is discovered that students find the C++ STL difficult due to their weaknesses in understanding various object-oriented concepts. These concepts form the prerequisites in learning the C++ STL. Examples of such concepts are function overloading and parametisation. The other common
problem in applying the STL is the difficulty to determine the most appropriate STL algorithm to apply.

C++ STL is covered in a higher level which is after students have acquired the foundation in programming. Tutors who are teaching the C++ STL cannot assume that all students have the same understanding in the elementary topics, and these topics form the prerequisites in learning the STL. Some may have already forgotten and just need hints to prompt them to recall. Others may have lack of understanding, and hence require exercises to drill the concepts into them again. Besides imparting skills on applying the STL, it is necessary for tutors to put in effort to understand the needs of individual student.

BAYESIAN THEOREM

Considering the problem in the prerequisites knowledge of individual student, a method that is able to model a cause-effect relationship with uncertainty management is required. Therefore, Bayesian Networks which encodes a corresponding set of conditional probability distributions in directed graphical models are highly appropriate for the student modeling. The Bayesian approach also supports a transparent student model as it has a strongly proven derivation probability theory basis, and the probabilities associated are intuitively understandable.

To model the student’s competence in the prerequisites after taking a pretest, the desired conditional probability is that the student understands the prerequisite given that he or she answered it correctly. Based on an example given by Ross [3], the conditional probability is obtained by

\[ P(U|C) = \frac{P(U \cap C)}{P(C)} = \frac{P(C|U) \times P(U)}{P(C|U) \times P(U) + P(C|\neg U) \times (1 - P(U))} \]  

(1)

where

\( U \) is the event that the student understands the prerequisite  
\( C \) is the event that the student answers correctly.

Equation (1) can be simplified to yield:

\[ P(U|C) = \frac{P}{p + (1/m) \times (1 - p)} \]  

(2)

where

\( p \) is probability that student understands prerequisite  
\( m \) is the number of multiple choice alternatives.

PROPOSED DEVELOPMENT

The proposed 3-tier architecture ITS has 6 main modules – Administrator, Tutor, Student, Tutorial, Pre-Test and Post-Test, providing the authoring of the domain knowledge dynamically (Figure 1). The Administration Module includes the management of various users of the system. The Pre-Test Module includes the setting of questions to test the prerequisites concepts of a student. The Tutorial Module allows the implementer tutor to design the tutorial sessions and employ different teaching strategies. The Bayesian theorem and Fuzzy Logic is applied here to direct the tutorial for individual student. The purpose of the Post-Test module is to test the effectiveness of the tutorial sessions. This includes questions related to the C++ STL.

The modularity of the implementation allows more effective participation of both the curriculum planner and implementer tutor. eXtreme programming methodology and J2EE technologies are applied in the project to reduce the development time and promote scalability.

FIGURE 1
COMONENTS OF C++ STL ITS

PROGRESS AND EVALUATION

Currently, the initial development that includes the Administration, Tutor, Student and Pre-Test are in the completion stage. The Tutorial module that employs the Bayesian is in progress. Research on the application of fuzzy logic to monitor the student’s progress and understanding is being carried out. On completion of the Tutorial and Post-Test Modules, an evaluation will be conducted to test the effectiveness of the Pre-Test and the applicability of the Bayesian theorem to model the student’s understanding. Finally, the students will be put through the Post-Test to evaluate the tutorial sessions.

CONTRIBUTION

The main contribution of this research work is the development of an ITS that targets application level. The novelty of the student modeling is in the adjustment of the prerequisite subskill threshold during the tutoring session using fuzzy logic. Integrating the Bayesian Theorem and fuzzy logic will produce a dynamic student model that changes smoothly as the conditional probability is updated.

REFERENCES

