Abstract

The project is to design and implement a ‘wizard’ with the following functionality to support a traditional CAL learning environment. To continually monitor and collect data on individual and group student performance, to perform statistical analysis, to provide feedback for teachers, to enable students to contribute to the wizard’s experience, and to provide advice to the individual student.

Motivation

Based on my experience, I believe in student imagination. If I give them a clear and simple example program, most students have fun with adding more functions into my original. Their programs considerably enhance my own.

This is my inspiration of wanting an easier CAL tool for teachers to build on-line lessons. Allowing students to get basic knowledge on their own would let more able students to get more advance knowledge as soon as they can. Moreover, average students can repeat a lesson as many times as they want. What students need is a highly flexible, customisable, supportive CAL tool.

The Learning Process

Separating one lesson into several topics is a typical method of teaching. Students can focus on each topic that has predefined objectives and take some exercises at the end of topic. Topics should have some relative points to connect them. It might be ‘the next step of doing something’, ‘another part of something’, ‘why this step is like this’, etc. A person walking along a path needs to know where he or she is going and his current progress.

During this process, teachers cannot predefine everything students should know. My idea is that teachers can learn from students’ obstacles as we all learn from our mistakes. It would be wonderful if this process could be automatic improved using its own experience. This means arrangement of students, teacher, topics and history of user interaction. The teacher predefines an initial learning track, and then students follow or repeat the track. Topics in the track could be moved, added, deleted or changed in content. This may be done either by the teacher, or by a wizard learning from students experience of the track.
Students start at the entry component and study until the end of each subsequent component that might have a small formative assessment exercise, and then allows students to make a decision to move on or to guide them to the next topic if they prefer otherwise. Moreover, students can go back to repeat the same component as often as they wish.

All information since students have logged in the system would be recorded – where and when they log in, which topic they choose to start learning and which topic they continue, which topic they repeat and how many times repeated. Until the end of learning, students can suspend a lesson and continue a later date.

Moreover, students can raise their problems or suggest new steps to be added into the system. During the learning process, gaming-like themes would apply to each lesson – fighting with devils, collecting golden medals – these themes would be automatically created by the system.

**The Learning Hierarchy**

As students execute each learning module (driven by JSP and MySQL) observation and analysis module would produce and leave information in the right place. This hierarchy shows working relevance between humans and machine. The more repetition in a module, the more attention the teacher should pay.

**My Questions**

My objective is to build a web-based tool including all functions for teachers and students to get involved with basic knowledge, in particular as many subjects as they want. This action leaves a lot of useful raw data. Inside the learning hierarchy, neat analysis methods would be used to interpret students’ behaviour and represent their behaviour in an understandable way. Hopefully, a learning pattern could be generated. Analysis of student progress is a major challenge.

**Possibilities**

The mechanics of this model could be supported by current web technologies, for instance, getting IP addresses that automatically sent to the server allowing the system to know where the student is and to verify the same student who is learning at that time. Obviously, JSP (Java Server Page) is the most convenience language for web-based application. Moreover, MySQL is a well-known database engine and it is free. Any operating systems are available to be server or client. Nice and easy is the main feature of all applications.

Guiding the system for users both student and teacher would be built-in – how to create lessons and examinations for the teacher, how to follow lessons for students. Generally, students can learn from anywhere-anytime-on any system.

**Bibliography**