Using Adaptability to Create an Automatically Adaptive Course Presentation on the Web

Mohammed A. Razek¹, Claude Frasson¹, Marc Kaltenbach²

¹Département d’informatique et de recherche opérationnelle
Université de Montréal C.P. 6128, Succ. Centre-ville Montréal, Québec Canada H3C 3J7
²Williams School of Business and Economics, Bishop’s University
Email :{ abdelram, frasson, kaltenba}@iro.umontreal.ca

Abstract
This paper describes our approach to building an automatically adaptive course presentation on the Web. Our method aims to combine adaptability with the learner-driven course. To achieve this goal, we design and implement an agent called a confidence agent. This agent analyzes the learner’s behavior during a learner-driven stage and then not only adapts the next presentation but also updates the domain model accordingly.

1. Introduction
As the information on the web is plentiful and constantly evolving, the Confidence Intelligent Tutoring System (CITS) [1] gains large amounts of information for each concept. Although this abundance of information strengthens the course content, it can also overwhelm and confuse the learner. To overcome this drawback, when presenting a course, this information should be filtered according to its pertinence. Our idea is to give the learners some extra features built in the CITS by which they can modify the content of the course presentation.

We designed an adaptive agent called confidence agent. This agent combines adaptability with learner-driven software. The agent works on the server side. Based on the CITS domain model (the granularity level of materials) and the user model (i.e. learning style, goals, etc.), the agent can observe the learners’ behaviors, allow them to insert and remove parts, and update the course materials. In this technique, each course consists of a header and 5 fragments (Figure 1): background, definition, problems, examples, and exercises. Each fragment contains different chunks of text grouped with a checkbox, images, and media. For each chunk there is a learner rating, which is used to measure the granularity of the chunk.

To measure the granularity of the chunk, suppose that \( C_h \) is a concept that is educated and \( \Gamma \) is an extracted chunk. And suppose also that a set of dominant meanings of the concept \( C_h \) is \( \{ w_1, ..., w_T \} \) [1]. Based on these, we compute the distance space between the chunk \( \Gamma \) and the concept \( C_h \) as follows:

\[
P(\Gamma \mid C_h) = \frac{1}{T} \sum_{j=1}^{T} F(\Gamma \mid w_j)
\]

(1)

Where the functions \( F(\Gamma \mid w_j) \) and \( F(\Gamma \mid C_h) \) represent the frequency of occurrence of the two words \( w_j \) and \( C_h \) in the chunk \( \Gamma \). To evaluate the particular view of specific learners on the chunk \( \Gamma \), assume that the number of times the chunk \( \Gamma \) is erased is \( E \), and the number of times the concept \( C_h \) is visited is \( C \). We consider that a chunk that is not erased by the user is read and recommended. Then we can calculate the importance value of the chunk \( \Gamma \) for the concept \( C_h \) as follows:

\[
I(\Gamma \mid C_h) = \frac{C - E}{C}
\]

(2)

Finally, from formulas (1) and (2), we can represent the granularity level \( G(\Gamma \mid C_h) \) of the chunk \( \Gamma \) for the concept \( C_h \) as follows:

\[
G(\Gamma \mid C_h) = P(\Gamma \mid C_h) + I(\Gamma 
\mid C_h)
\]

(3)

Accordingly, our adaptation algorithm sorts the chunks \( \{ \Gamma_1, ..., \Gamma_n \} \) in decreasing order related to the value of \( G(\Gamma_i \mid C_h) \) \( \forall i \) and then returns the first three chunks to be presented with the current fragment. The agent was implemented using Java-Servlets, the database of course contents using XML, and dynamic course presentation using JSP. We used Apache Tomcat 3.2.1 to launch the CITS system (including the confidence agent) and to provide it with an environment to excite the JSP file.

References