

A Java Framework for Developing Intelligent Tutoring Systems

Julian Arthur and Colin Depradine
Department of Computer Science, Mathematics
and Physics
The University of the West Indies
Bridgetown, Barbados

julian.arthur@hotmail.com
colin.depradine@cavehill.uwi.edu

ABSTRACT

An intelligent tutoring system (ITS) is software that provides an interactive learning environment and can adapt its instruction depending on user needs. The development of these systems is expensive due to significant resource requirements. To ease the development process of these systems, technologies such as authoring tools and shells are sometimes used. In this paper a new java framework is proposed to be implemented as the foundation of a new authoring tool for intelligent tutoring systems (ITSs).

1 Introduction

The development of ITSs is expensive due to the large amount of expertise needed and the long development times [1]. The development team for a single ITS can consist of numerous experienced programmers as well as experts in number of varying research areas, such as educational theory and domain knowledge design.

The potential benefits when using ITSs are great but unfortunately the difficulty in producing these systems has limited both access and acceptance of them. Therefore, efficient and effective authoring technologies are required to assist in creating ITSs.

This paper presents the motivation for creating this new java framework and introduces the fundamental architecture of the framework.

2 Problem

The conceptual designs of ITSs are modular but unfortunately, typically the final implementations of these systems are quite monolithic in structure [2] which can be attributed to the functionality required for effective instruction of specific material. The difficulty in efficiently and effectively creating ITSs has also reduced their accessibility and acceptance in the traditional classroom.

3 Motivation

To provide a java-based framework that will encourage both modular design of the ITS and reuse of ITS components. Another reason for this work is to provide a sensible foundation for an authoring tool that can be used to simplify and improve ITS development productivity.

4 Goals and Objectives

The first goal is to reduce the development cost and implementation times of ITS software by providing a comprehensive ITS authoring tool (includes framework). The secondary goal is to establish a public test bed for research into ITS design and implementation by providing the framework as an open source tool.

The objectives are that participating developers using this framework and authoring tool will:

- Act as a catalyst to improve both the framework and tool.
- Create an active developer community in this research area thereby benefitting from peer interaction.
- Create an industry standard for the design and implementation of ITSs.

5 ITS System Design

The java language was chosen as the primary language for this project due to its object-oriented paradigm and due to the java community's open source nature.

The architecture for this framework consists of seven primary modules as shown in figure 1. The primary modules in the ITS architecture are (1) the distributed learning module, (2) the pedagogical module, (3) the domain knowledge module, (4) the student model, (5) the expert model, (6) the communications module and (7) the control module.

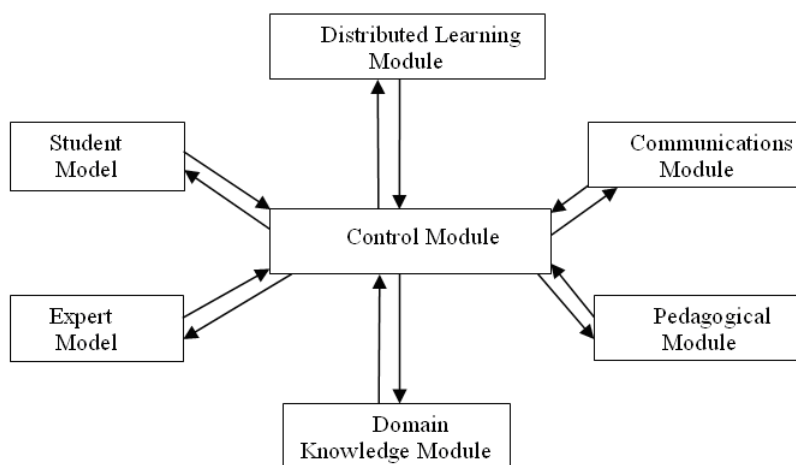


Figure 1: The fundamental architecture representing the framework

The conceptual flow of information between modules within an ITS can be seen below in figure 2.

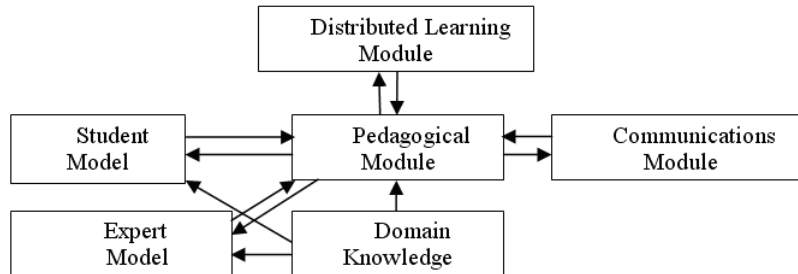


Figure 2: Conceptual flow of communication between components

5.1 Student Model

The student model holds unique user information which represents the user's understanding of the material taught by the ITS. A user's actions are monitored by the system and the model is updated appropriately to track a student's performance.

5.2 Expert Model

The expert model contains the information that is taught to the user. However, it is more than just a representation of the data; it is a model of how someone skilled in a particular domain uses the knowledge. An expert model can be used to pinpoint user difficulties by comparing the learner's solution to the expert's solution.

5.3 Domain Knowledge

This component contains all the information that the tutoring system will teach to an ITS user. The representation of the domain knowledge is critical since the other components which must access this module should be able to do so efficiently.

5.4 Pedagogical Module

The pedagogical module performs a number of tasks such as determining when to review a topic and deciding when to present a new topic. This module uses input provided by the student model to adapt to a user's specific needs.

5.5 Communications Module

The communications module is the user interface of the ITS. All interactions between the user and the ITS are handled by this component. This module handles the presentation of the taught material ensuring it is presented in an appropriate way.

5.6 Distributed Learning Module

The distributed learning module is a nonconventional component which has been added and it is used to provide various networking functionality in the ITS environment.

5.7 Control Module

This module acts as the central control for the primary modules of the ITS. All communication between modules is relayed via this component.

6 Conclusion

We have provided a brief explanation of what research area we are exploring, the limitations that currently exist in ITS implementations, our goals for the future of this research and present the architecture currently adopted in the framework and authoring tool.

References

- [1] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10:98 – 129, 1999.
- [2] T. Murray. An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*, pages 493 – 546. Kluwer Academic Publishers, Dordrecht, 2003.
- [3] V. Devedzic and A. Harrer. Common patterns in its architectures. *Kunstliche Intelligenz*, 18:17 – 21, 2004.