In Proceedings of the ITS'98 Workshop 3 - Efficient ITS Development, San Antonio, TX, 1998.

Efficient Development of Intelligent Test Generation Agents with the Disciple Learning Agent Shell

Gheorghe Tecuci and Harry Keeling
Department of Computer Science, MSN 4A5, George Mason University
4400 University Dr., Fairfax, VA 22030-4444
phone: (703) 993-1722, fax: (703) 993-1710, tecuci@gmu.edu, hkeeling@osf1.gmu.edu

Abstract

Disciple is a shell for developing intelligent agents where an expert teaches the agent how to perform domain-specific tasks in a way that resembles the way the expert would teach an apprentice. This paper discusses the application of Disciple to the development of educational agents that generate test questions for students, together with hints, answers and explanations, and assist in the assessment of students' understanding and use of higher-order thinking skills. This work illustrates an integration of machine learning and intelligent-tutoring systems, in the context of computer-based assessment involving multimedia documents.

1. Introduction

Disciple is an apprenticeship, multistrategy learning approach for developing intelligent agents (Bradshaw, 1997) where an expert teaches the agent how to perform domain-specific tasks in a way that resembles the way the expert would teach an apprentice, by giving the agent examples and explanations as well as by supervising and correcting its behavior (Tecuci, 1998). This approach in which the agent learns its behavior from its teacher, integrates many machine learning and knowledge acquisition techniques (such as inductive learning from examples, explanation-based learning, learning by analogy, learning by experimentation) taking advantage of their complementary strengths to compensate for their weaknesses (Buchanan and Wilkins, 1993, Michalski and Tecuci, 1994; Tecuci and Kodratoff, 1995). As a consequence, the Disciple approach significantly reduces the involvement of the knowledge engineer in the process of building an intelligent agent.

A type of agent that can be built naturally with Disciple is an educational agent. By educational agent we refer to a class of agents that assist an educator in an education-related task. Indeed, an educator can teach a Disciple agent and then this agent can tutor students in the same way it was taught by the educator. In such a case, the Disciple agent would act as an indirect communication channel between the educator and the students. Therefore, such an application of Disciple illustrates an approach to the integration of machine learning and intelligent tutoring systems, a problem that is receiving increasing attention due to its significant potential benefits (Aïmeur and Frasson, 1995; Hamburger and Tecuci, 1998).

This paper discusses the application of Disciple to the development of educational agents that generate test questions for students, together with hints, answers and explanations, and assist in the assessment of students' understanding and use of higher-order thinking skills (Beyer, 1988).

We first introduce the general architecture of the Disciple Learning Agent Shell that implements the current version of the Disciple approach. Then we briefly present the test generation agents built with the Disciple shell, and the development methodology.

2 Disciple Learning Agent Shell

The current version of the Disciple approach is implemented in the Disciple Learning Agent Shell, the architecture of which is presented in Figure 1.

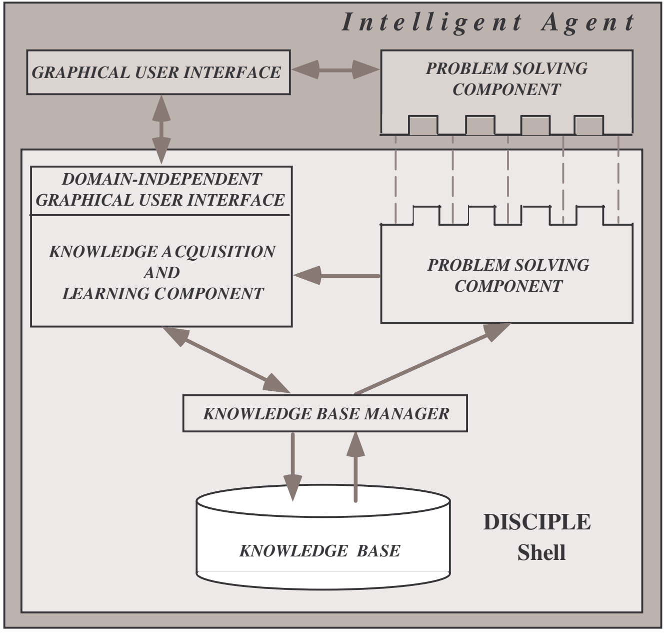

Intelligent Agent
GRAPHICAL USER INTERFACE
PROBLEM SOLVING COMPONENT
DOMAIN-INDEPENDENT GRAPHICAL USER INTERFACE
KNOWLEDGE ACQUISITION AND LEARNING COMPONENT
PROBLEM SOLVING COMPONENT
KNOWLEDGE BASE MANAGER
KNOWLEDGE BASE
DISCIPLE Shell

Figure 1: The architecture of the Disciple shell.

The Disciple shell was designed to facilitate the development of intelligent agents for a large variety of domains. It consists of four main domain independent components shown in the light gray area of Figure 1. They are:

- a *knowledge acquisition and learning component* for developing and improving the knowledge base, with a domain-independent graphical user interface;
- a *basic problem solving component* which provides basic problem solving operations;
- *a knowledge base manager* which controls access and updates to the knowledge base;
- an empty *knowledge base* to be developed for the specific application domain.

The two components in the dark gray area are the domain dependent components that need to be developed and integrated with the Disciple shell to form a customized agent that performs specific tasks in an application domain. They are:

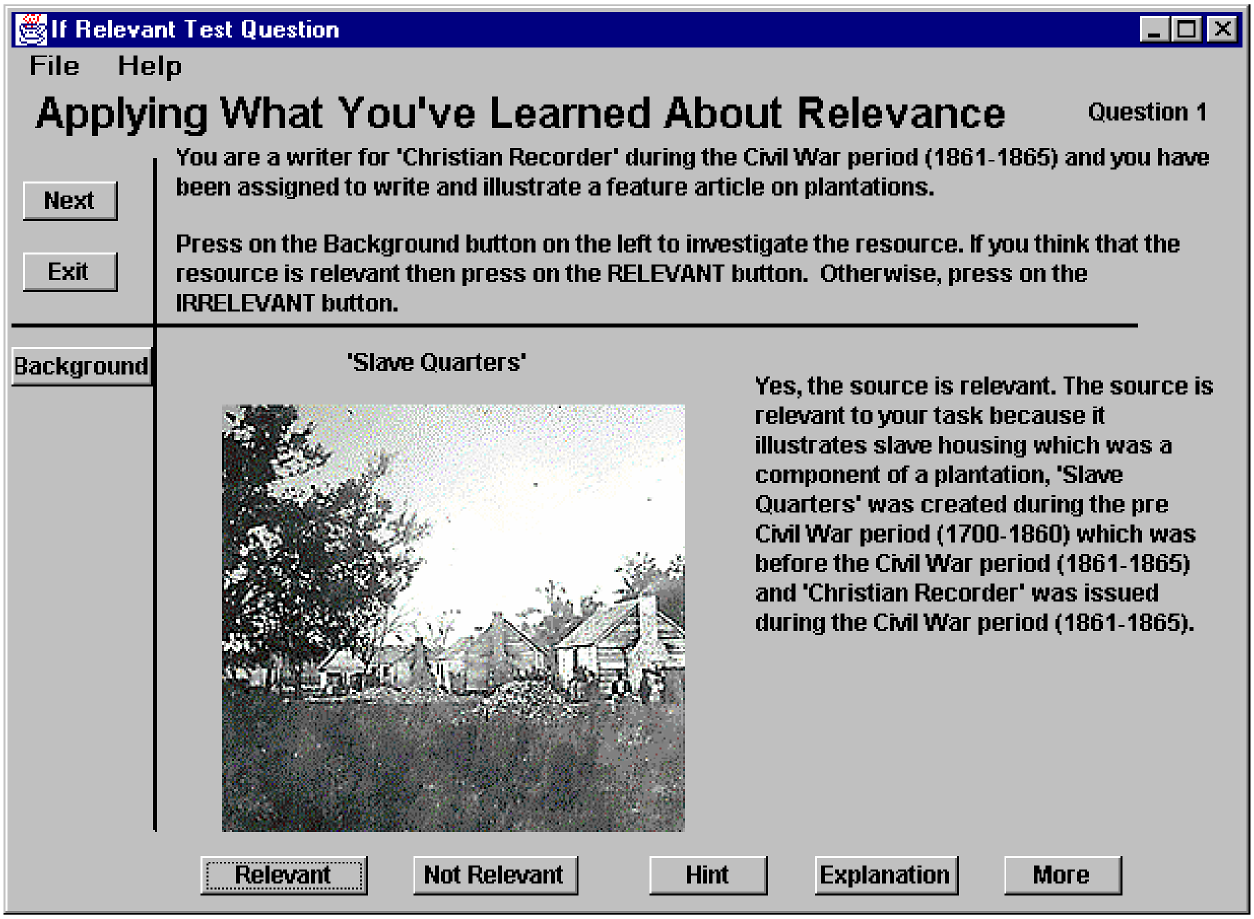
- a *specialized problem solver* which provides the specific functionality of the agent;
- a domain-specific *graphical user interface*.

In the case of the test generation agents discussed in this paper, the specialized problem solver is the test generator that also builds and maintains a student model. Two domain specific interfaces were built. One was built to facilitate the communication between the expert/teacher and the agent. The other was built to facilitate communication between the agent and the students.

3 Test generation agents

We have experimentally applied Disciple to develop three test generation agents. Two of the agents generate history test questions to assist in the assessment of students' understanding and use of higher-order thinking skills. One of the agents is integrated with the MMTS (Multimedia and Thinking Skills) educational system (Fontana et al. 1993), creating a system with expanded capabilities, called Intelligent MMTS (IMMTS). Inside IMMTS, the agent has the role of generating an exam consisting of a set of test questions of different levels of difficulty. The student has to answer one test question at a time and, after each question, he or she receives the correct answer and an explanation of the answer. This type of application of a Disciple-generated agent illustrates a possible role for these agents, that of enhancing the capabilities, generality, and usefulness of non-KB educational software. The IMMTS system has been field-tested in American history classes in several middle schools on American installations in Germany and Italy.

The other test generation agent (Tecuci and Keeling, 1998) is a stand-alone agent that can be used independently of the MMTS software. The student interacts directly with this agent to assess him/herself. He or she chooses the type of test question to solve, and will receive, on request, feedback in the form of hints to answer the question, the correct answer, and some or all the explanations of the answer. That is, this agent also tutors the student. Figure 2 shows a test question generated by the agent. The student is asked to imagine that he or she is a reporter and has been assigned the task to write an article for Christian Recorder, during the Civil War period, on plantations. The student has to analyze the historical source “Slave Quarters” in order to determine whether it is relevant to this task. In the situation illustrated in Figure 2 the student answered correctly. Therefore, the agent confirmed the answer and provided an explanation for it, as indicated in the lower right pane of the window.



Applying What You've Learned About Relevance

Question 1

You are a writer for 'Christian Recorder' during the Civil War period (1861-1865) and you have been assigned to write and illustrate a feature article on plantations.

Press on the Background button on the left to investigate the resource. If you think that the resource is relevant then press on the RELEVANT button. Otherwise, press on the IRRELEVANT button.

'Slave Quarters'

Yes, the source is relevant. The source is relevant to your task because it illustrates slave housing which was a component of a plantation, 'Slave Quarters' was created during the pre Civil War period (1700-1860) which was before the Civil War period (1861-1865) and 'Christian Recorder' was issued during the Civil War period (1861-1865).

Figure 2: A test question, answer and explanation generated by the agent*

The student could have requested a hint to answer the question which, in the case of the test question in Figure 2, would have been the following one:

> "To determine if the source is relevant to your task investigate if it illustrates some component of a plantation, check when it was created and when Christian Recorder was issued."

In general, there may be more than one reason why a source is relevant to a task. By pushing the More button, the student can receive the hints and explanations corresponding to these additional reasons.

Another agent, the Statistical Analysis Assessment and Support Agent, is developed to be integrated in a university-level introductory science course and to be accessed on the Internet through a web browser. The course "The Natural World" introduces students to the world of science using collaborative assignments and problem-centered group projects that look at scientific issues which underlie public policy making and stimulate the development of students' analytic skills. The agent supports two aspects of students' learning in this course: students' knowledge and understanding of statistics, and students' analyses of issues related to statistics. It

* Picture reproduced from LC-USZ62-67818, Library of Congress, Prints & Photographs Division, Civil War Photographs

does this in several ways. It can also be used as an assistant by the students as they work through their assigned projects.

4 The methodology for building the test generation agents

An overview of the Disciple methodology is given in Figure 3.

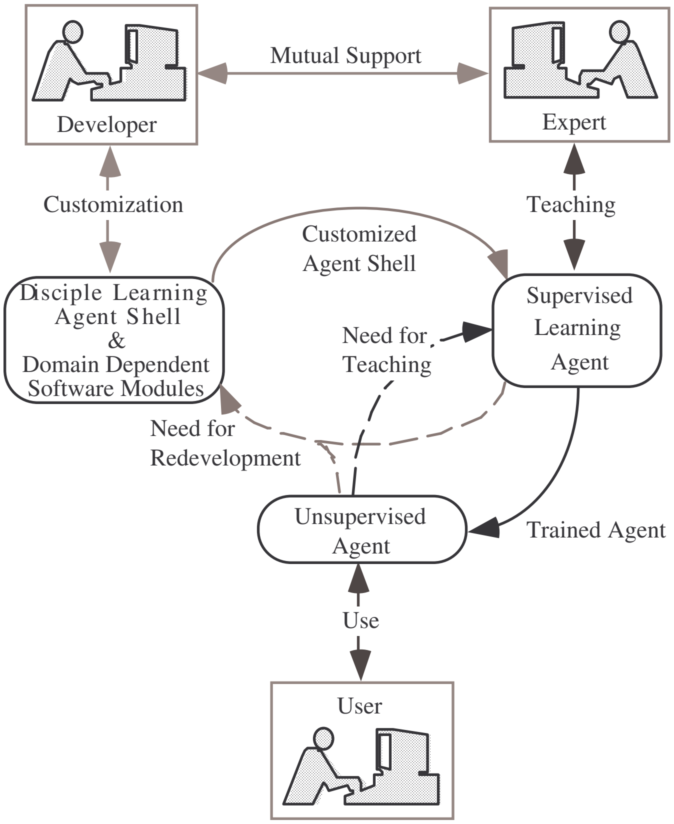


Figure 3 Disciple's agent-building methodology

First, the developer (knowledge engineer) customizes the general Disciple learning agent shell into a specific learning shell for test generation, by defining several interface modules. One is the Source Viewer that allows the agent to access and display historical sources from a multimedia database. Another is a Customized Example Editor that allows the history expert or teacher to give examples of test questions using natural language templates. Yet another is a test interface that allows the agent to display test questions. The knowledge engineer also builds a test generation engine that relies on example generation, a basic problem solving operation supported by the Disciple shell (see Figure 1).

After the learning shell for test generation has been built, the history expert or teacher interacts with it to develop its initial knowledge base and to teach it to generate test questions. The expert starts by choosing a historical theme, such as Slavery in America. Then the expert identifies a set of historical concepts that are appropriate and necessary to be learned by the students. The expert also identifies a set of historical sources that illustrate these concepts and will be used in test questions. All these concepts and the historical sources are represented by the history expert/teacher in the knowledge base of the agent, by using the various editors and browsers of the customized learning agent shell (see Figure 1). This knowledge base includes the description of 252 historical concepts, 80 historical sources, and 6 publications.

A basic relevancy test question consists in judging the relevancy of a historical source to a given reporter's task. To teach the agent to generate and answer such questions, the history expert/teacher gives it an example consisting of a task and a historical source relevant to that task. Then the educator has to explain the agent why the historical source is relevant to the task. Based on this explanation and on the initial example the agent automatically generates an initial test generation rule. Next, the agent generates examples analogous to the initial example. These will be similar tasks and sources that the agent assumes to be relevant. Each generated example is shown to the teacher who is asked to accept it as correct or to reject it, thus characterizing it as a positive or a negative example of the rule. These examples are used to refine the rule. Following this procedure, the agent has learned 54 rules for judging the relevance of historical sources to four types of tasks. These rules have been learned from an average of 2.17 explanations (standard deviation 0.91) and 5.4 examples (standard deviation 1.37), which indicates a very efficient training process. Using these rules the agent can generate more than 10^5 different test questions of the following types: IF-RELEVANT (see Figure 2), WHY-RELEVANT, WHICH-RELEVANT and WHICH-IRRELEVANT.

We have performed four types of experiments with the stand-alone test generation agent, and the results were very encouraging. The first experiment tested the correctness of the agent's knowledge base, as judged by the domain expert who developed the agent. This was intended to clarify how well the developed agent represents the expertise of the expert who taught the agent. The second experiment tested the correctness of the knowledge base, as judged by a domain expert who was not involved in its development. This was intended to test the generality of the agent. The third and the fourth experiments tested the quality of the test generation agent, as judged by students and by teachers.

Conclusions

In this paper we have presented the Disciple approach and its natural and efficient application to the development of educational agents that generates test questions to assess student's understanding and use of higher-order thinking skills. Because each such agent is taught by the educator through examples and explanations, and then it is able to provide similar examples and explanations to the students (as part of the generated tests), it could be considered as being a preliminary example of a new type of educational agent that can be taught by an educator to teach the students (Hamburger and Tecuci, 1998).

Acknowledgments

Tomasz Dybala, Kathryn Wright and Philippe Loustaunau contributed to this research that was supported by the DARPA contract N66001-95-D-8653, as part of the Computer-Aided Education and Training Initiative, directed by Kirstie Bellman. Partial support was also provided by the NSF grant No. CDA-9616478, as part of the program Collaborative Research on Learning Technologies, directed by Caroline Wardle.

References

Aïmeur, E. and Frasson, C. (1995). Eliciting the learning context in co-operative tutoring systems. IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning.

Beyer, B. (1988). *Developing a Thinking Skills Program*. Allyn and Bacon, Inc. Boston, MA.

Bradshaw, J. M. (editor), (1997). *Software Agents*. Menlo Park, CA. AAAI Press.

Buchanan, B. G. and Wilkins, D. C. (editors), (1993). *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems,* Morgan Kaufmann, San Mateo, CA.

Fontana, L., Debe, C., White, C. and Cates, W. (1993). Multimedia: Gateway to Higher-Order Thinking Skills in Progress. In *Proceedings of the National Convention of the Association for Educational Communications and Technology*.

Hamburger H. and Tecuci G. (1998). Toward a Unification of Human-Computer Learning and Tutoring, submitted to ITS'98.

Michalski, R.S. and Tecuci, G., Eds. (1994). *Machine Learning: A Multistrategy Approach Volume 4,* Morgan Kaufmann Publishers, San Mateo, CA.

Tecuci, G. and Kodratoff, Y. (editors), (1995). *Machine Learning and Knowledge Acquisition: Integrated Approaches,* Academic Press.

Tecuci G. and Keeling H. (1998). Developing Intelligent Educational Agents with the Disciple Learning Agent Shell, submitted to ITS'98.

Tecuci G. (1988). *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies,* Academic Press.