In B.P. Goettl, H.M. Halff, C.L. Redfield, V.J. Shute (Eds.), Intelligent Tutoring Systems, 4th International Conference, ITS'98, San Antonio, Texas, USA, Springer Verlag, 1998.

Best Paper Award at ITS'98

Developing Intelligent Educational 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, USA
{tecuci, hkeeling}@gmu.edu

Abstract. Disciple is an apprenticeship, multistrategy learning approach for developing intelligent agents where an expert teaches the agent how to perform domain-specific tasks in a way that resembles how the expert would teach an apprentice. We claim that Disciple can naturally be used to build certain types of educational agents. Indeed, an educator can teach a Disciple agent which in turn can tutor students in the same way it was taught by the educator. This paper presents the Disciple approach and its application to developing an educational agent that generates history test questions. The agent provides intelligent feedback to the student in the form of hints, answer and explanations, and assists in the assessment of student's understanding and use of higher-order thinking skills.

1 Introduction

Disciple is an apprenticeship, multistrategy learning approach for developing intelligent agents where an expert teaches the agent how to perform domain-specific tasks in a way that resembles how the expert would teach an apprentice, by giving the agent examples and explanations, as well as by supervising and correcting its behavior [11]. It integrates many machine learning and knowledge acquisition strategies taking advantage of their complementary strengths to compensate for their weaknesses [5, 9, 10]. As a consequence, it 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 (i.e. an agent that assists 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. Therefore, such an application of Disciple illustrates an approach to the integration of machine learning and intelligent tutoring systems [1, 8].

This paper is organized as follows. Section 2 introduces the architecture of Disciple. Section 3 presents the test generation agent built with Disciple. Section 4 describes the process of building the agent. Section 5 presents experimental results and Section 6 summarizes the evidence in support of the claims of the Disciple approach.

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 Fig. 1. The Disciple shell has four main domain independent components shown in the light gray area of Fig. 1:

- a knowledge acquisition and learning component for developing the knowledge base (KB), with a domain-independent graphical user interface;
- a problem solving component that provides basic problem solving operations;
- a knowledge base manager which controls access and updates to the KB;
- an empty KB 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 that provides the specific functionality of the agent;
- a domain-specific graphical user interface.

In the case of the test generation agent that is presented 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 to facilitate the communication between the history expert/teacher and the agent, and between the agent and the students.

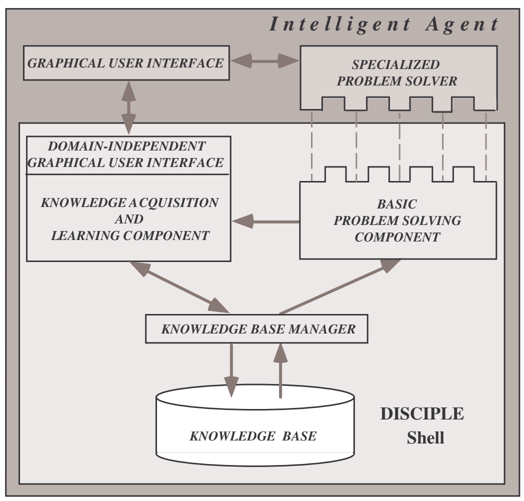


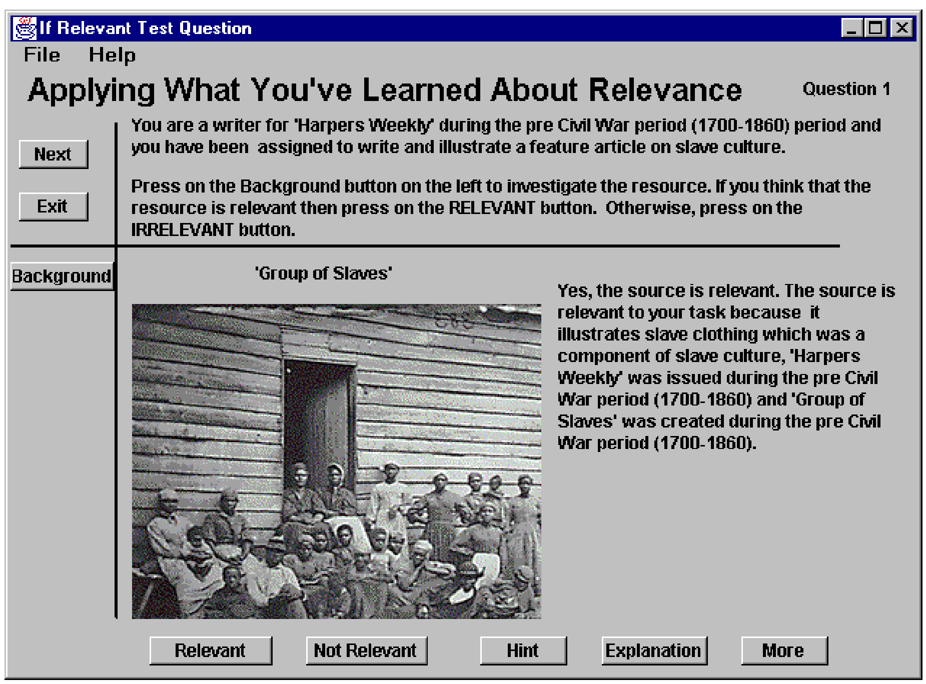
Fig. 1. The architecture of the Disciple shell

3 A Test Generation Agent for Higher-Order Thinking Skills

The developed Disciple agent generates history tests to assist in the assessment of students' understanding and use of higher-order thinking skills. An example of specific higher-order thinking skill is the evaluation of historical sources for relevance, credibility, consistency, ambiguity, bias, and fact vs. opinion [2,3,4,6]. To motivate the middle school students, for which this agent was developed, and to provide an element of game playing, the agent employs a journalist metaphor. It asks the students to assume the role of a journalist who has to complete assignments from the Editor. One assignment could be to write an article on the experience of African American women during the Civil War. Within this context, the students are given source material and asked various questions that would require them to exercise the skill of evaluation. In these assignments, students are asked to apply higher-order thinking skills in much the way journalists do when they complete their assignments and prepare stories for publication.

The agent dynamically generates a test question, based on a student model, together with the answer, hints and explanations. An example of a test question is shown in Fig. 2. The student is asked to imagine that he or she is a reporter and has been assigned the task to write an article for Harper's Weekly during the pre Civil War period on slave culture. He or she has to analyze the historical source "Group of Slaves" in order to determine whether it is relevant to this task. In the situation from Fig. 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. The student can also request a hint, which in this case is the following one: "To determine if the source is relevant to your task investigate if it illustrates some component of slave culture, check when it was created and when Harper's Weekly was issued."

The agent has two modes of operation: final exam mode and self-assessment mode. In the final exam mode, it generates an exam consisting of a set of test questions. 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. In the self-assessment mode, the student chooses the type of test question to answer, and may request a hint to answer the question, the correct answer, and the explanation of the answer.



In B.P. Goettl, H.M. Halff, C.L. Redfield, V.J. Shute (Eds.), Intelligent Tutoring Systems, 4th International Conference, ITS'98, San Antonio, Texas, USA, Springer Verlag, 1998.

Fig. 2. A test question, answer and explanation generated by the agent[1]

4 Building the Test Generation Agent

To build the test generation agent the teacher (possibly assisted by a knowledge engineer) first develops the knowledge base of the agent. Then, the knowledge engineer builds the test generation engine and the student's interface.

The KB of any Disciple agent should contain an ontology [7] and a set of rules. The ontology contains descriptions of historical concepts (such as "slave culture"), historical sources (such as "Group of Slaves" in Fig. 2), and templates for reporter tasks (such as "You are a writer for a PUBLICATION during a HISTORICAL-PERIOD and you have been assigned to write and illustrate a feature article on a SLAVERY-TOPIC."). Using these descriptions, the agent communicates with the students through a stylized natural language, as shown in Fig. 2.

Fig. 3 shows an example of a relevancy rule. It is an IF-THEN rule where the condition specifies a general reporter task and the conclusion specifies a source relevant to that task. The condition also incorporates the explanation of why the source is relevant to the task. Associated with the rule are the natural language templates corresponding to the task, explanation and conclusion of the rule. These templates are automatically created from the natural language descriptions of the elements in the rule. One should notice that each rule corresponds to a certain type of task (WRITE-DURING-PERIOD, in this case). Other types of tasks are WRITE-ON-TOPIC, WRITE-FOR-AUDIENCE, and WRITE-FOR-OCCASION. Therefore, for each type of reporter task there will be a family of related relevancy rules. The rules corresponding to the other evaluation criteria, such as credibility, accuracy, or bias, will have a similar form.

```
IF
    ?W1  IS   WRITE-DURING-PERIOD,  FOR ?S1,  DURING ?P1,  ON ?S2
    ?P1  IS   HISTORICAL-PERIOD
    ?S1  IS   PUBLICATION,  ISSUED-DURING ?P1
    ?S2  IS   SLAVERY-TOPIC
    ?S3  IS   SOURCE,  ILLUSTRATES ?S4,  CREATED-DURING ?P1
    ?S4  IS   HISTORICAL-CONCEPT,   COMPONENT-OF ?S2
THEN
    RELEVANT  HIST-SOURCE  ?S3
```

Task Description: You are a writer for ?S1 during ?P1 and you have been assigned to write and illustrate a feature article on ?S2.

Explanation: ?S3 illustrates ?S4 which was a component of ?S2, ?S1 was issued during ?P1 and ?S3 was created during ?P1.

Operation Description: ?S3 is relevant

Fig. 3. A relevancy rule

[1] Picture reproduced from LC-B8171-383, Library of Congress, Prints & Photographs Div.

4.1 Building the Agent's Ontology

The process of building the agent's ontology starts with choosing a module in a history curriculum (such as Slavery in America) for which the agent will generate test questions. Then the teacher identifies a set of historical concepts that are appropriate and necessary to be learned by the students. The teacher also identifies a set of historical sources that will enhance the student's understanding of these concepts and that will be used in test questions. All these concepts and the historical sources are represented by the history teacher in the knowledge base of the agent, by using the various editors and browsers of Disciple. One is the Source Viewer that displays the historical sources. Another is the Concept Editor that is used to describe the historical sources. The historical sources have to be defined in terms of features that are necessary for applying the higher-order thinking skill of evaluation. For instance, a source is relevant to some topic if it identifies, illustrates or explains the topic or some of its components. In particular, "Group of Slaves" in Fig. 2 is defined as being a photo. It illustrates the concepts slave clothing, child slave, female slave, male slave and field slave. Other information is also represented, such as the audience for which this historical source is appropriate and when it was created.

4.2 Teaching the Agent to Judge the Relevance of a Source to a Task

After the semantic network is defined, the teacher has to teach the agent how to judge the relevancy (as well as the credibility, accuracy, and the other evaluation criteria) of a source with respect to various reporter tasks. Fig. 4 presents the phases of this teaching process.

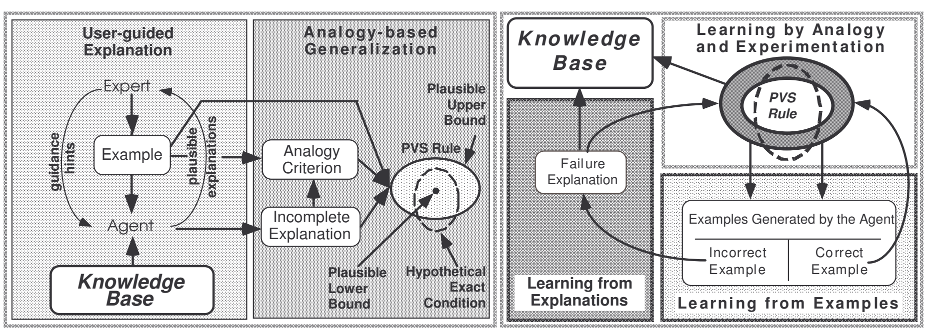


Fig. 4. (a) Rule Learning (b) Rule Refinement

First, the teacher gives to the agent an example consisting of a task and a historical source relevant to that task, as illustrated in Fig. 5. Next, the teacher has to help the agent understand why the source is relevant to the task. Rather than giving an explanation to the agent, the teacher will guide the agent to propose explanations and will select the correct ones. For instance, the teacher may point to the most relevant objects from the input example and may specify the types of plausible explanations to be searched for. From these interactions, it was concluded that the source HUMAN-

In B.P. Goettl, H.M. Halff, C.L. Redfield, V.J. Shute (Eds.), Intelligent Tutoring Systems, 4th International Conference, ITS'98, San Antonio, Texas, USA, Springer Verlag, 1998.

FLESH-AT-AUCTION is relevant to the task of writing an article for CHRISTIAN-RECORDER, during POST-CIVIL-WAR, on SLAVE-LIFE because:

- HUMAN-FLESH-AT-AUCTION illustrates SLAVE-SELLING which was a component of SLAVE-LIFE
- CHRISTIAN-RECORDER was issued during POST-CIVIL-WAR
- HUMAN-FLESH-AT-AUCTION was created during POST-CIVIL-WAR

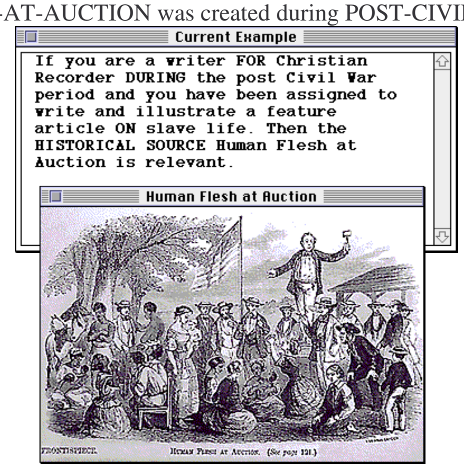


Fig. 5. Initial example given by the teacher

One may notice that this explanation is similar to the explanation from the test question in Fig. 2. This illustrates a significant benefit to be derived from using the Disciple approach to build educational agents. That is, the kind of explanations that the agent gives to the students are similar to the explanations that the agent itself has received from the teacher. Therefore, the agent acts as an indirect communication medium between the teacher and the students.

The found explanation is generalized by the agent to an analogy criterion. Then the analogy criterion and the example are used to generate a rule with two conditions: a plausible lower bound condition which is very specific, covering only the example in Fig. 5, and a plausible upper bound condition which is very general, covering examples that are analogous with the initial example. To improve this rule, the teacher will invoke the rule refinement process represented in Fig. 4b, asking the agent to generate examples similar with the one in Fig. 5. Each example generated by the agent is covered by the plausible upper bound and is not covered by the plausible lower bound of the rule. The example (which looks like the one in Fig. 5) 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. A positive example is used to generalize the plausible lower bound of the rule's condition. A negative example is used to elicit additional explanations from the expert and to specialize both bounds, or only the plausible upper bound. This process will continue until either the two bounds of the rule become identical or until no further examples can be generated. The final learned rule is the one from Fig. 3.

4.3 Developing the Test Generation Engine

One of the agent's requirements was that it also generates hints and feedback for right and wrong answers (see Fig. 6). The Hint is the part of the Explanation that refers only to the variables used in Task Description. The Right Answer is generated from the Operation Description and the Explanation, and the Wrong Answer is a fixed text.

Hint: To determine if the source is relevant to your task investigate if it illustrates some component of ?S2, check when it was created and when ?S1 was issued.

Right Answer: Yes, the source is relevant. The source is relevant to your task because it illustrates ?S4 which was a component of ?S2, ?S1 was issued during ?P1 and ?S3 was created during ?P1.

Wrong Answer: No, the source is not relevant. Investigate this source further and analyze the hints and explanations to improve your understanding of relevance. You may consider reviewing the material on relevance. Then continue testing yourself.

Fig. 6. Additional templates associated with the rule in Fig. 3

We have developed a test generation engine that generates four types of test questions:

- IF RELEVANT: Show the student a writing assignment and ask whether a particular historical source is relevant to that assignment;
- WHICH RELEVANT: Show the student an assignment and three historical sources and ask the student to identify the relevant one;
- WHICH IRRELEVANT: Show the student an assignment and three historical sources and ask the student to identify the irrelevant one; and
- WHY RELEVANT: Show the student an assignment, a source and three possible reasons why the source is relevant, and ask the student to select the right reason.

To generate an IF RELEVANT test question with a relevant source, the agent simply needs to generate an example of a relevancy rule. This rule example will contain a task T and a source S relevant to it, together with a hint and an explanation. However, if the student requires all the possible reasons for why the source S is relevant to the task T, then the agent will need to find all the examples containing the source S and the task T of all the relevancy rules from the family of rules corresponding to the task T. To generate an IF RELEVANT test question with an irrelevant source the agent has to first generate a valid task T by finding an example of a relevancy rule R. Then it has to find a historical source S such that the task T and the source S are not part of any example of any rule from the family of rules corresponding to the task T.

The methods for generating WHICH RELEVANT and WHICH IRRELEVANT test questions are based on the methods for generating IF RELEVANT test questions. First an IF RELEVANT test question with a task T and relevant source S is generated. Then additional relevant or irrelevant sources are looked for, as described above.

The method for generating WHY RELEVANT test questions starts with generating an example E_1 of a relevancy rule R_1. This example provides a correct task description T, a source S relevant to T, and a correct explanation EX_1 of why the source S is relevant to T. Then the agent chooses another rule that is not from the family of the relevancy rules corresponding to T. Let us suppose that the agent chooses a credibility rule R_2. It then generates an example E_2 of R_2, based on E_1 (that is, E_2 and E_1 share as many parts as possible, including the source S). The agent also generates an explanation EX_2 of why S is credible. While this explanation is correct, it has nothing to do with why S is relevant to T. Then, the agent repeats this process to find another explanation that is true but explains something else, not why S is relevant to T.
Similar test questions could be generated for each evaluation skill such as, IF CREDIBLE test questions or WHY CREDIBLE test questions.

5 Experimental Results

The ontology of the test generation agent includes the description of 252 historical concepts, 80 historical sources, and 6 publications. The KB also contains 54 relevancy rules grouped in four families. 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.

We have performed five experiments with the test generation agent. The first three experiments tested the correctness of the knowledge base, as judged by the domain expert who developed the agent, and by a domain expert who was not involved in its development. The fourth and the fifth experiments tested the quality of the test generation agent, as judged by students and by teachers.

The results of the first three experiments are summarized in Table 1. IF RELEVANT test questions were randomly generated by the agent and answered by the developing expert (in the first experiment) and by the independent expert (in the second and the third experiment). The agreements or the disagreements between the expert and the agent were recorded and the percentage of the correct answers of the agent (the accuracy) was computed. These experiments have revealed a much higher predictive accuracy in the case of IF RELEVANT test questions where the source was relevant. We have analyzed each case where both the developing expert and the independent expert agreed that the agent failed to recognize that a source was relevant or irrelevant to a certain task. In most cases it was concluded that the representation of the source was incomplete. This analysis suggested that the representation of the sources should be guided by the following "projection" principle which, if followed, would have avoided many of the agent's errors: *Any historical source must be completely described in terms of the concepts from the KB.* This means that if the knowledge base contains a certain historical concept, then any historical source referring to that concept should contain the concept in the description of its content.

Table 1. Evaluation results

Reviewer	Total number of reviewed questions	Number of IF questions with relevant sources	Number of IF questions with irrelevant sources	Time spent to review all the questions	Accuracy on IF questions with relevant sources	Accuracy on IF questions with irrelevant sources	Total accuracy
Developing expert	406	202	204	5 hours	96.53%	81.86%	89.16%
Independent expert	401	198	203	10 hours over 2 days	95.45%	76.35%	85.76%
Independent expert	1,524	198+1,326	–	22 hours for 1,326 questions	96.19%	–	–

We have also conducted an experiment with a class of 21 students from the 8th grade at The Bridges Academy in Washington D.C. The students were first given a lecture on relevance and then were asked to answer 25 test questions that were dynamically generated by the agent. Students were also asked to investigate the hints and the explanations. To record their impressions, they were asked to respond to a set of 18 survey questions with one of the following phrases: very strongly agree, strongly agree, agree, indifferent, disagree, strongly disagree, and very strongly disagree. Fig. 7 presents the results from 7 of the most informative survey questions.

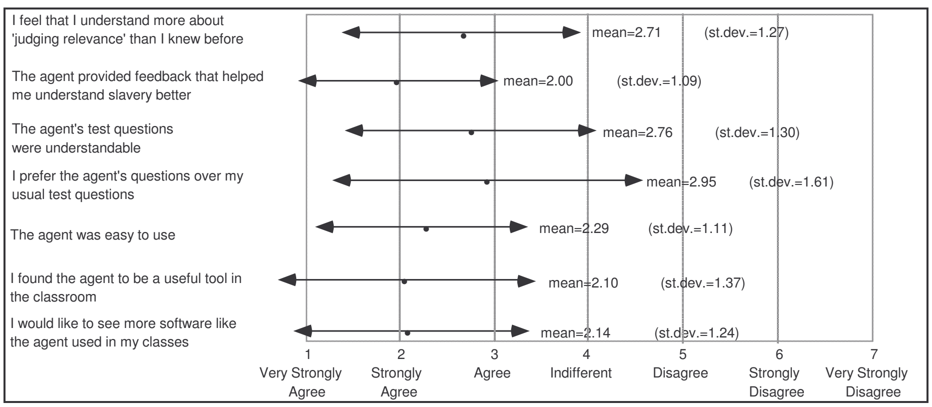


Fig. 7. Student survey results

Finally, a user group experiment was conducted with 8 teachers at The Public School 330 in the Bronx, New York City. This group of teachers had the opportunity to review the performance of the agent and was then asked to complete a questionnaire. Several of the most informative questions and a summary of the teachers' responses are presented in Fig. 8.

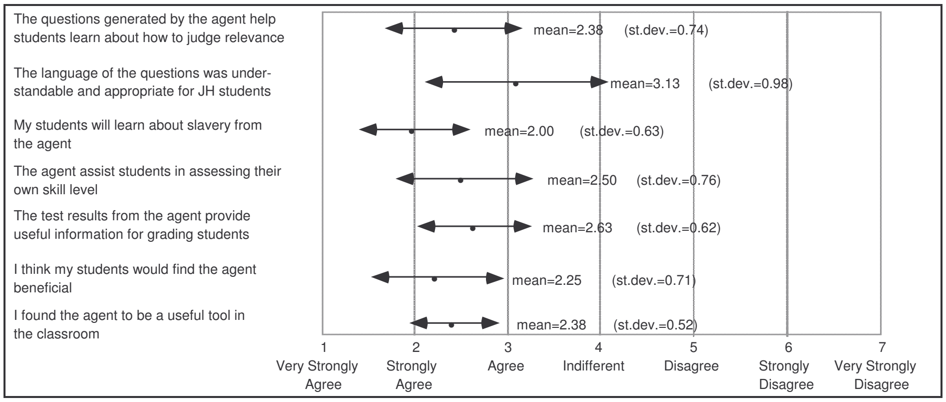


Fig. 8. Teacher survey results

6 Conclusions

We have presented the Disciple approach and its application to the development of an educational agent. We have provided experimental evidence that the process of teaching the agent is natural and efficient, and that it results in a knowledge base of high quality and in a useful educational agent. The agent provides the educator with a flexible tool that lifts the burden of generating tests for large classes, tests that do not repeat themselves and take into account the instruction received by each student. Because the 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 [8]. Although not discussed in detail in this paper, this work also shows an automated computer-based approach to the assessment of higher-order thinking skills [2,3], as well as an assessment that involves multimedia documents [6]. Both of these represent very important goals in the current education research. We are currently using Disciple to develop a statistical analysis assessment and support agent.

Acknowledgments

This research was done in the Learning Agents Laboratory at George Mason University. Tomasz Dybala and Kathryn Wright contributed to the development of Disciple. Lynn Fontana, Rich Rosser, and David Webster assisted in the initial development of the KB. Lawrence Young, who holds an MS in History Education, acted as the independent expert. The research of the Learning Agents Laboratory is supported by the AFOSR grant F49620-97-1-0188, as part of the DARPA's High Performance Knowledge Bases Program, by the DARPA contract N66001-95-D-8653, and by the NSF grant No. CDA-9616478.

References

1. Aïmeur, E., Frasson, C.: Eliciting the learning context in co-operative tutoring systems. IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning (1995)
2. Beyer, B.: Practical Strategies for the Teaching of Thinking. Allyn and Bacon, Inc. Boston, MA. (1987)
3. Beyer, B.: Developing a Thinking Skills Program. Allyn and Bacon, Inc. Boston, MA. (1988)
4. Bloom, B.: Taxonomy of Educational Objectives. David McKay Co., Inc. New York. (1956)
5. Buchanan, B. G. and Wilkins, D. C. (eds.): Readings in Knowledge Acquisition and Learning, Morgan Kaufmann, San Mateo, CA. (1993)
6. Fontana, L., Debe, C., White, C., Cates, W.: Multimedia: Gateway to Higher-Order Thinking Skills in Progress. In Proceedings of the National Convention of the Association for Educational Communications and Technology (1993)
7. Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R. (eds.): Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic (1993)
8. Hamburger H. and Tecuci G.: Toward a Unification of Human-Computer Learning and Tutoring, in this volume (1998)
9. Michalski, R.S. and Tecuci, G. (eds.): Machine Learning: A Multistrategy Approach Volume 4, Morgan Kaufmann Publishers, San Mateo, CA. (1994)
10. Tecuci, G. and Kodratoff, Y. (eds.): Machine Learning and Knowledge Acquisition: Integrated Approaches, Academic Press, London (1995)
11. Tecuci G.: Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies, Academic Press, London (1998)