In International Journal of Artificial Intelligence in Education, vol. 10, no.3-4, 1999.

Developing an Intelligent Educational Agent with Disciple

GHEORGHE TECUCI
Learning Agents Laboratory, Department of Computer Science
MSN 4A5, George Mason University
4400 University Dr., Fairfax, VA 22030-4444, USA
phone: (703) 993-1722, fax: (703) 993-1710, email:tecuci@gmu.edu

HARRY KEELING
Department of Systems and Computer Science
Howard University
2300 6th St., NW, Washington, DC 20059, USA
phone: (202) 806-4830, fax: (202) 806-4531, email:hkeeling@scs.howard.edu

Disciple is an apprenticeship, multistrategy learning approach for developing intelligent agents where an expert teaches the agent to perform domain-specific tasks in a way that resembles how the expert would teach an apprentice, by giving the agent examples and explanations, and by supervising and correcting its behavior. The Disciple approach is currently implemented in the Disciple Learning Agent Shell. We make the claim that Disciple can naturally be used by an educator to build certain types of educational agents. The educator will directly teach the Disciple agent how to perform certain educational tasks and then the agent can interact with the students to perform such tasks. This paper presents the Disciple approach and its application to building an educational agent that generates history tests for students. These tests provide intelligent feedback to the student in the form of hints, answer and explanations, and assist in the assessment of students' understanding and use of higher-order thinking skills.

INTRODUCTION

Disciple is an apprenticeship, multistrategy learning approach for developing intelligent agents (Bradshaw, 1997) where an expert teaches the agent 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 (Tecuci, 1998). We define a *learning agent shell* as consisting of a learning engine and an inference engine that support a representation formalism in which a knowledge base can be encoded, as well as a methodology for building the knowledge base.

The central idea of the Disciple approach is to facilitate the agent building process by the use of synergy at three different levels. First, there is synergy between different learning methods employed by the agent (Michalski and Tecuci, 1994). By integrating complementary learning methods (such as inductive learning from examples, explanation-based learning, learning by analogy, learning by experimentation) in a dynamic way, the Disciple agent is able to learn from the human expert in situations in which no single strategy learning method would be sufficient. Second, there is synergy between the expert teaching the agent, and the agent learning from the expert (Tecuci and Kodratoff, 1995). For instance, the expert may select representative examples to teach the agent, may provide explanations, and may answer the agent's questions. The agent, on the other hand, will learn general rules that are difficult to be defined by the expert, and will consistently integrate them into its knowledge base. Third, there is synergy between the expert and the agent in solving a problem. They form a team in which the agent solves the more routine but labor intensive parts of the problem and the expert solves the more creative ones. In the process, the agent learns from the expert, gradually evolving toward an "intelligent" agent (Mitchell et al., 1985).

We claim that the Disciple approach significantly reduces the involvement of the knowledge engineer in the process of building an intelligent agent, most of the work being done directly by the domain expert. In this respect, the work on Disciple is part of a long term vision where personal computer users will no longer be simply consumers of ready-made software, as they are today, but also developers of their own software assistants.

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 a user in an education-related task. Similar to authoring systems, such as RIDES (Munro et al., 1997) or SimQuest (Veermans and Van Joolingen, 1998), Disciple places more of the task of building intelligent educational software in the hands of the educator. The educator will directly teach the Disciple agent how to perform certain tasks and then the agent can interact with the students to perform such tasks. 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; Frasson and Gouarderes, 1998; Hamburger and Tecuci, 1998; Mengelle et al., 1998).

This paper presents the Disciple approach and its application to building an educational agent that generates history tests to assist in the assessment of students' understanding and use of higher-order thinking skills (Fontana et al., 1993). We first introduce the general architecture of the Disciple Learning Agent Shell that implements the current version of the Disciple approach. Then we present the test generation agent built with the Disciple shell and describe the process of building the agent, with an emphasis on the process of teaching it. After that we briefly present the various methods used to generate test questions. Finally, we present several experimental results and summarize the evidence in support of the claims of the Disciple approach made in this section.

DISCIPLE LEARNING AGENT SHELL AND METHODOLOGY

The current version of the Disciple approach is implemented in the Disciple learning agent shell, the architecture of which is presented in Figure 1.

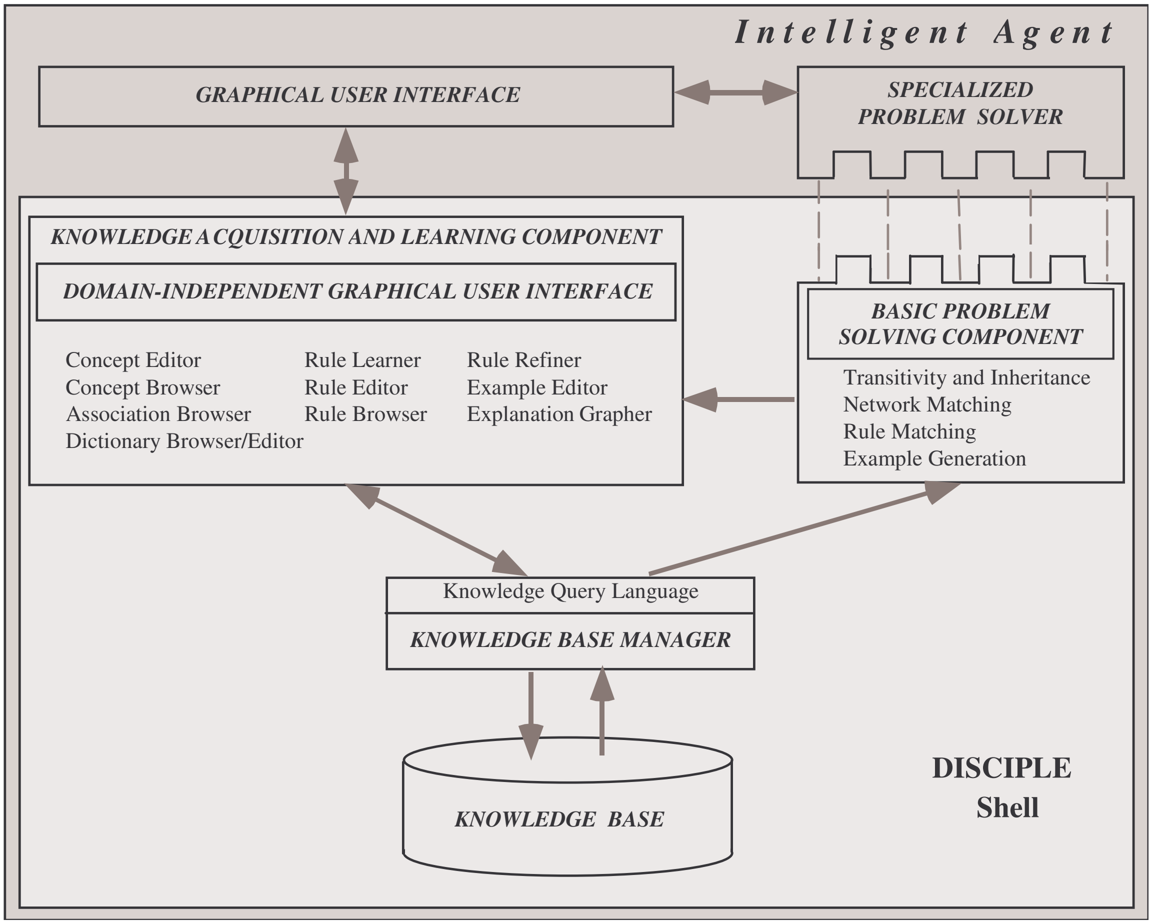


Figure 1: The architecture of the Disciple shell.

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 darker 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** which facilitates training and use by educators and students.

The process of building the test generation agent presented in this paper consisted of the following stages. First, the agent developer (software and knowledge engineer), cooperating with an educator (a history expert and teacher), customized the Disciple shell by developing the specialized problem solver, and two domain specific interfaces. The specialized problem solver is a test generator. One interface was built to facilitate the communication between the history expert/teacher and the agent. The other interface was built to facilitate the communication between the agent and the students. Second, the educator interacted with the customized Disciple agent to develop its initial knowledge base, to teach it to generate test questions, and to ultimately verify and validate its test generation capabilities. In the next section we present the developed test generation agent and in the following sections we illustrate the methodology of building it.

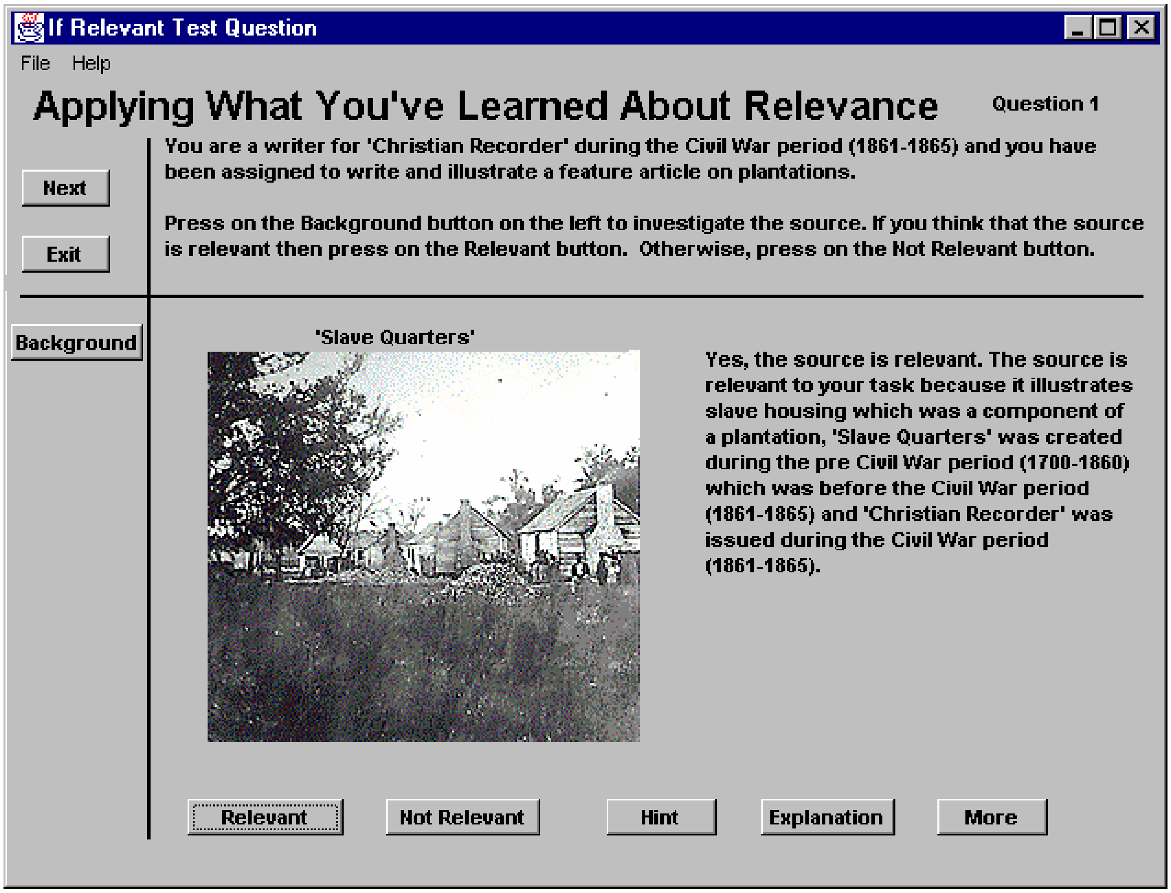
A TEST GENERATION AGENT FOR HIGHER ORDER THINKING SKILLS IN HISTORY

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 (Bloom, 1956; Beyer, 1987, 1988).

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, asking the students to assume the role of a novice journalist. Figure 2, for instance, 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. The student could have requested a hint to answer the question and would have received 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 several reasons 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 example of a test question is shown in Figure 3. The student is given a task, a historical source and three possible reasons why the source is relevant to the task. He or she has to investigate the source and decide which reason(s) account for the fact that the source is relevant to the task. The student is instructed to check the box next to the correct reason(s).

The agent has two modes of operation: final exam mode and self-assessment mode. In the final exam mode, the agent generates 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. In the self-assessment mode, the student 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. The test questions are generated such that all students interacting with the agent are likely to receive different tests even if they follow exactly the same interaction pattern. The agent maintains a list of historical sources that have been investigated by the student, or have been used in previous tests, and uses it to generate test questions that do not involve these sources.



Applying What You've Learned About Relevance

Question 1

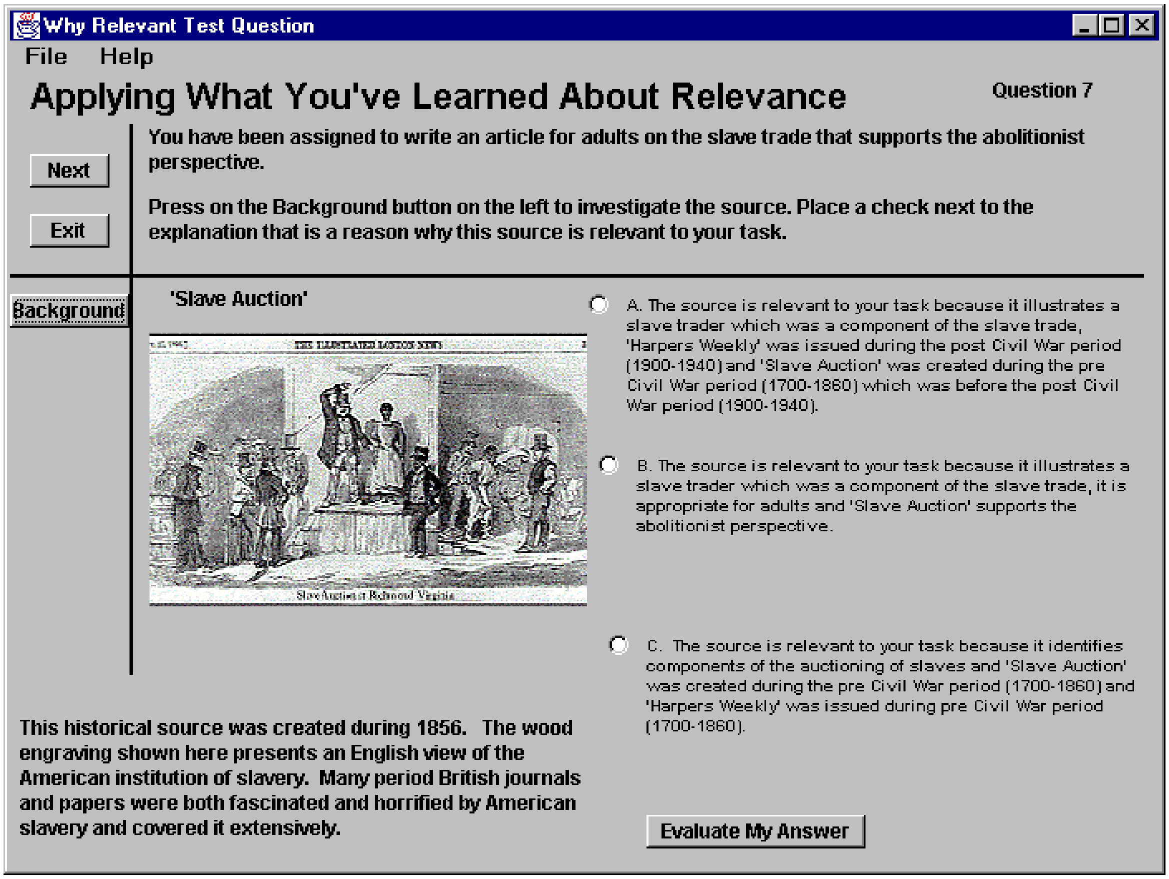
You are a writer for 'Christian Recorder' during the Civil War period (1861-1865) and you have been assigned to write and illustrate a feature article on plantations.

Press on the Background button on the left to investigate the source. If you think that the source is relevant then press on the Relevant button. Otherwise, press on the Not Relevant button.

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[*]

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


Why Relevant Test Question
File
Help
Next
Exit
Background
THE ILLUSTRATED LONDON NEWS
Slave Auction at Richmond Virginia
Evaluate My Answer

Applying What You've Learned About Relevance

Question 7

You have been assigned to write an article for adults on the slave trade that supports the abolitionist perspective.

Press on the Background button on the left to investigate the source. Place a check next to the explanation that is a reason why this source is relevant to your task.

'Slave Auction'

○ A. The source is relevant to your task because it illustrates a slave trader which was a component of the slave trade, 'Harpers Weekly' was issued during the post Civil War period (1900-1940) and 'Slave Auction' was created during the pre Civil War period (1700-1860) which was before the post Civil War period (1900-1940).

○ B. The source is relevant to your task because it illustrates a slave trader which was a component of the slave trade, it is appropriate for adults and 'Slave Auction' supports the abolitionist perspective.

○ C. The source is relevant to your task because it identifies components of the auctioning of slaves and 'Slave Auction' was created during the pre Civil War period (1700-1860) and 'Harpers Weekly' was issued during pre Civil War period (1700-1860).

This historical source was created during 1856. The wood engraving shown here presents an English view of the American institution of slavery. Many period British journals and papers were both fascinated and horrified by American slavery and covered it extensively.

Figure 3: Another test question [*]

BUILDING THE TEST GENERATION AGENT

The knowledge base of any Disciple agent contains an ontology (Gruber, 1993) and a set of rules. To build the test generation agent the educator, assisted by the knowledge engineer, has to develop the agent's ontology and then to teach the agent how to generate test questions. The knowledge engineer has to build the test generation engine and the agent's domain-specific interface.

Building the agent's ontology

The agent's ontology contains descriptions of historical concepts (such as "plantation"), historical sources (such as "Slave Quarters" in Figure 2), and templates for reporter tasks (such as "You are a writer for PUBLICATION during HISTORICAL-PERIOD and you have been assigned to write and illustrate a feature article on SLAVERY-TOPIC."). Using these descriptions and templates, the agent communicates

[*] Picture reproduced from LC-USZ62-15398, Library of Congress, Prints & Photographs Division, Civil War Photographs

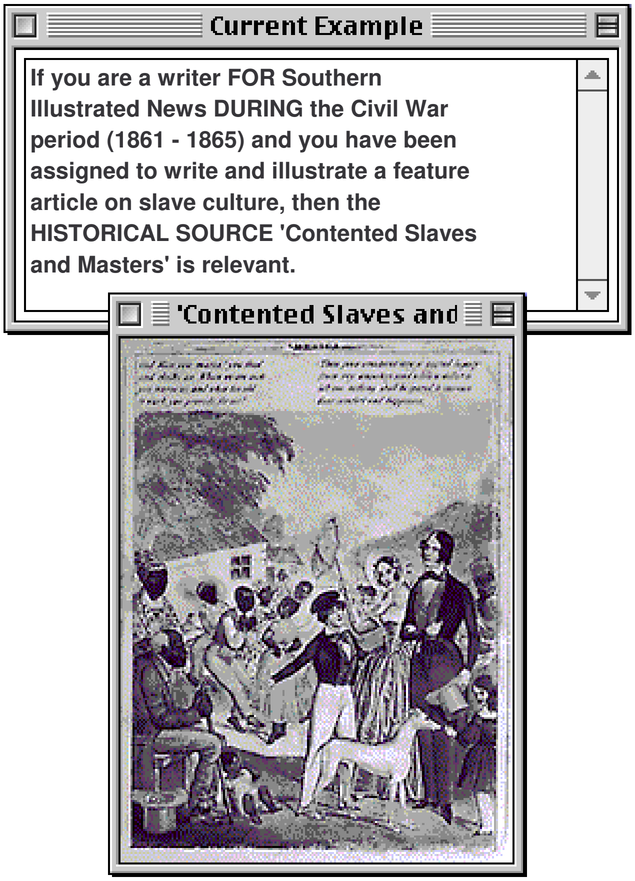
with the students through a stylized natural language, as illustrated in Figure 2 and Figure 3.

The ontology building process starts with choosing a module in a history curriculum (such as Slavery in America) for which the agent will generate test questions. Then the educator identifies a set of historical concepts that are appropriate and necessary to be learned by the students. The educator also identifies a set of historical sources that can enhance the student's understanding of these concepts and will be used in test questions. All these concepts and historical sources are represented by the history educator in the knowledge base, by using the various interfaces 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 skills of relevance, credibility, etc. For instance, a source is relevant to some topic if it identifies, illustrates or explains the topic or some of its components. Let us consider the historical source 'Contented Slaves and Masters', from the bottom of Figure 4. This source is defined as being a LITHOGRAPH that ILLUSTRATES the concepts SLAVE-DANCE, MALE-SLAVE, FEMALE-SLAVE, and SLAVE-MASTER. Other information also has to be represented, such as the audience for which this source is appropriate and when it was created. The concepts from the knowledge base are hierarchically organized in a semantic network (Quillian, 1968; Chaudhri et al. 1998) that can be inspected with the Concept Browser. For instance, SLAVE-DANCE was defined as being a type of SLAVE-RECREATION which, in turn, was a SLAVE-LIFE-ASPECT. This initial knowledge base of the agent was assumed to be incomplete and even possibly partially incorrect, needing to be improved during the next stages of the agent's development.

Teaching the agent to generate basic relevancy test questions

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 educator gives it an example consisting of a task and a historical source relevant to that task, as shown in Figure 4.

Starting from the example in Figure 4, the agent has learned the relevancy rule in Figure 5. This 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. In the following we will briefly present the rule learning and refinement methods employed by Disciple to learn the rule in Figure 5.



If you are a writer FOR Southern Illustrated News DURING the Civil War period (1861 - 1865) and you have been assigned to write and illustrate a feature article on slave culture, then the HISTORICAL SOURCE 'Contented Slaves and Masters' is relevant.

Figure 4: Initial example given by the educator*.

IF

```
?W1   IS    WRITE-DURING-PERIOD, FOR ?S1, DURING ?P1, ON ?S2
?S1   IS    PUBLICATION, ISSUED-DURING ?P1
?P1   IS    HISTORICAL-PERIOD
?S2   IS    HISTORICAL-CONCEPT
?S3   IS    SOURCE, ILLUSTRATES ?S4, CREATED-DURING ?P2
?S4   IS    SLAVE-LIFE-ASPECT, COMPONENT-OF ?S2
?P2   IS    HISTORICAL-PERIOD, BEFORE ?P1
```

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, ?S3 was created during ?P2 which was before ?P1 and ?S1 was issued during ?P1.

Operation Description

?S3 is relevant

Figure 5**:** A learned relevancy rule.

* Picture reproduced from LC-USZ62-89745, Library of Congress, Prints & Photographs Division, Civil War Photographs

Rule learning

The rule learning method is schematically represented in Figure 6. As Explanation-based Learning (DeJong and Mooney, 1986; Mitchell, Keller, Kedar-Cabelli, 1986), it consists of two phases, explanation and generalization. However, in the explanation phase the agent is not building a proof tree, and the generalization is not a deductive one.

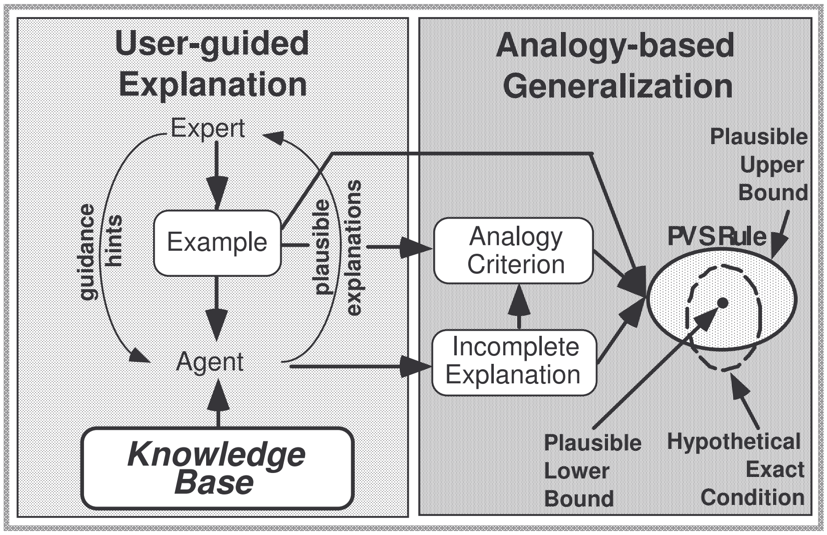


Figure 6: The rule learning method of Disciple.

In the explanation phase, the educator helps the agent to understand why the example in Figure 4 is correct (that is, why the source is relevant to the given task). The explanation of the example has a form that is similar to the one given by a teacher to a student. The source "Contented Slaves and Masters" is relevant to the given task (see Figure 4) because:

"Contented Slaves and Masters" illustrates a slave dance which was a component of slave culture.
"Contented Slaves and Masters" was created during the pre Civil War period which was before the Civil War period.

Each of these sentences corresponds to a path in the agent's ontology, as shown in Figure 7.

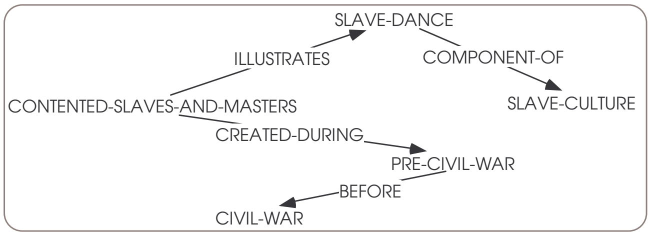


Figure 7: The (incomplete) explanation of the example in Figure 4.

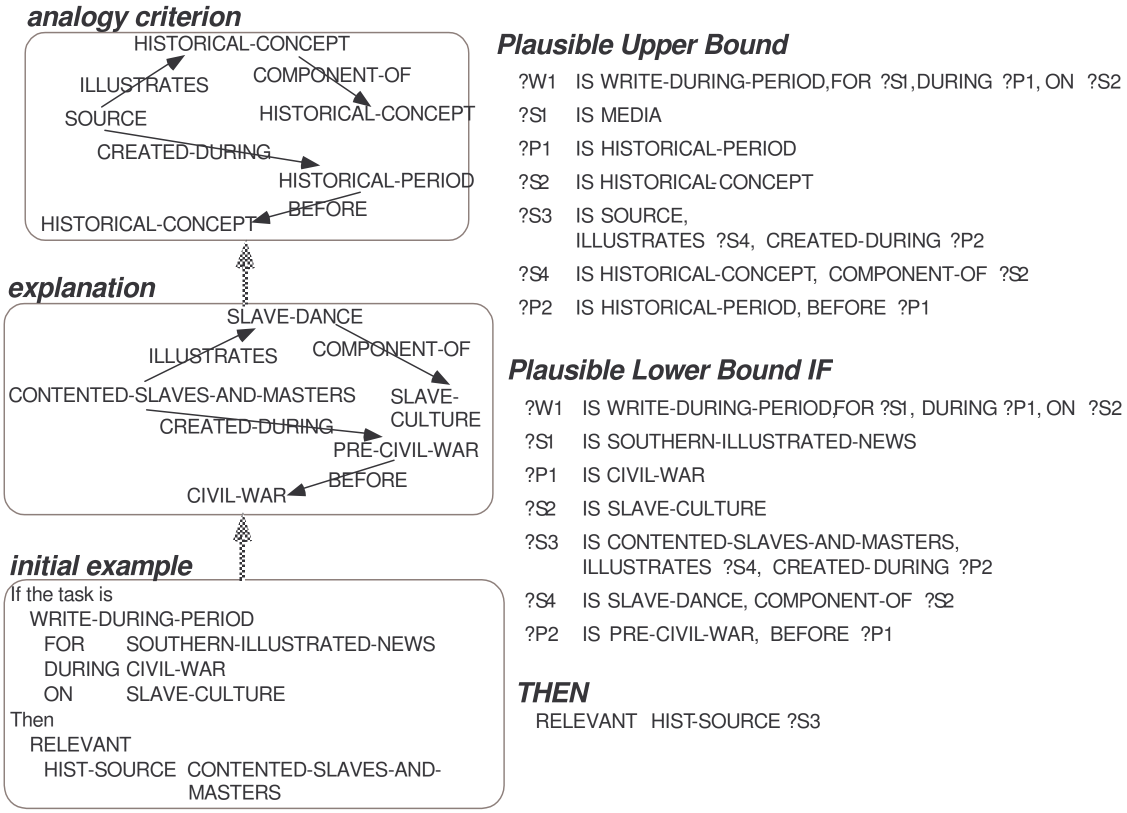
However, rather than giving an explanation to the agent, the educator guides the agent to propose explanations and then selects the correct ones. For instance, the educator may point to the most relevant objects from the input example (such as the source 'Contented Slaves and Masters') and may specify the types of explanations to be generated by the agent (e.g. a correlation between two objects or a property of an object). The agent uses such guidance and specific heuristics to propose plausible explanations to the educator who has to select the correct ones. A particularly useful heuristic is to propose explanations of an example by analogy with the explanations of other examples. One should notice that the explanation generated in this way is likely to be incomplete and will have to be completed during rule refinement.

The above explanation is similar to a part of the explanation from the test question in Figure 2. This illustrates a significant benefit to be derived from using the Disciple approach to build educational agents. That is, the kinds of explanations that the agent gives to the students are similar to the explanations that the agent itself has received from the educator. Therefore, the agent acts as an indirect communication medium between the educator and the students.

In the generalization phase (see Figure 6), the agent performs an analogy-based generalization of the example and its explanation into a plausible version space (PVS) rule. A PVS rule is an IF-THEN rule with two conditions, a plausible upper bound condition that is likely to be more general than the exact condition, and a plausible lower bound condition that is likely to be less general than the exact condition. The generalization process is illustrated in Figure 8. The initial example in Figure 8 is the internal representation of the example in Figure 4. Also, the explanation is the one from Figure 7. First, the explanation is generalized to an analogy criterion by preserving the object features (such as ILLUSTRATES and CREATED-DURING) and by generalizing the objects to more general concepts (e.g. generalizing SLAVE-DANCE to HISTORICAL-CONCEPT). To determine how to generalize an object, Disciple analyzes all the features from the example and the explanation that are connected to that object. Each such feature is defined in Disciple's ontology by a domain (that specifies the set of all the objects from the application domain that may have that feature) and a range (that specifies all the possible values of that feature). The domains and the ranges of these features restrict the generalizations of the objects. For instance, in the explanation from Figure 8, SLAVE-DANCE has the feature COMPONENT-OF and appears as value of the feature ILLUSTRATES. Therefore, the most general generalization of SLAVE-DANCE is the intersection of the domain of COMPONENT-OF and the range of ILLUSTRATES:

MGG(SLAVE-DANCE) = Domain(COMPONENT-OF)$\cap$Range(ILLUSTRATES) =
HISTORICAL-CONCEPT

The analogy criterion and the example are used to generate the plausible upper bound condition of the rule, while the explanation and the example are used to generate the plausible lower bound condition of the rule.



```
If the task is
  WRITE-DURING-PERIOD
    FOR        SOUTHERN-ILLUSTRATED-NEWS
    DURING CIVIL-WAR
    ON         SLAVE-CULTURE
Then
  RELEVANT
    HIST-SOURCE  CONTENTED-SLAVES-AND-
                       MASTERS
```

Plausible Upper Bound

```
?W1   IS WRITE-DURING-PERIOD,FOR  ?S1, DURING  ?P1, ON  ?S2
?S1    IS MEDIA
?P1    IS HISTORICAL-PERIOD
?S2    IS HISTORICAL-CONCEPT
?S3    IS SOURCE,
       ILLUSTRATES  ?S4,  CREATED-DURING  ?P2
?S4    IS HISTORICAL-CONCEPT,  COMPONENT-OF  ?S2
?P2    IS HISTORICAL-PERIOD, BEFORE  ?P1
```

Plausible Lower Bound IF

```
?W1   IS WRITE-DURING-PERIOD,FOR ?S1, DURING ?P1, ON  ?S2
?S1    IS SOUTHERN-ILLUSTRATED-NEWS
?P1    IS CIVIL-WAR
?S2    IS SLAVE-CULTURE
?S3    IS CONTENTED-SLAVES-AND-MASTERS,
       ILLUSTRATES  ?S4,  CREATED- DURING  ?P2
?S4    IS SLAVE-DANCE, COMPONENT-OF  ?S2
?P2    IS  PRE-CIVIL-WAR,  BEFORE  ?P1
```

THEN

```
RELEVANT   HIST-SOURCE ?S3
```

Figure 8: Generation of the initial plausible version space rule.

Rule refinement

The representation of the PVS rule in the right hand side of Figure 6 shows the most likely relation between the plausible lower bound, the plausible upper bound and the hypothetical exact condition of the rule. Notice that there are instances of the plausible upper bound that are not instances of the hypothetical exact condition of the rule. This means that the learned rule in Figure 8 covers also some negative examples. Also, there are instances of the hypothetical exact condition that are not instances of the plausible upper bound. This means that the plausible upper bound does not cover all the positive examples of the rule. Both of these situations are a consequence of the fact that the explanation of the initial example might be incomplete, and are consistent with what one would expect from an agent performing analogical reasoning. To improve this rule, the educator will invoke the rule refinement process represented schematically in Figure 9. The educator will ask the agent to use the learned rule to generate examples similar with the one in Figure 4. 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 natural language equivalent of this example (which looks like the one in Figure 4) is shown to the educator 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 through empirical induction. A negative example is used to elicit additional explanations from the educator and to specialize both bounds, or only the plausible upper bound. Figure 10 shows an example generated by Disciple, by analogy with the initial example from Figure 4. The agent's analogical reasoning is schematically represented in Figure 11. The explanation from the left-hand side of Figure 11 indicates why the initial example is correct. The expression from its right hand side is similar with this explanation because both of them are less general than the analogy criterion from the top of Figure 11. Therefore, one may infer by analogy that the expression from the right hand side of Figure 11 explains an example that is similar to the initial example (that is, the generated example from the right hand side of Figure 11 and from Figure 10).

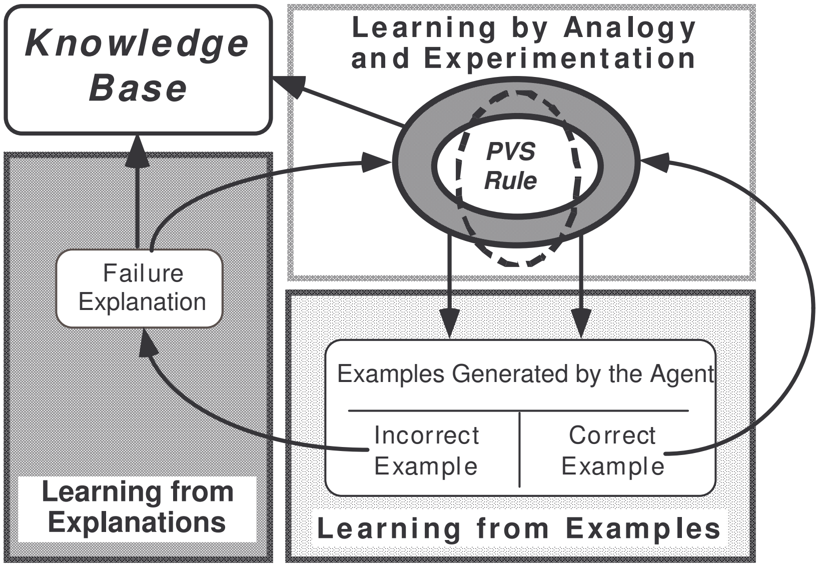
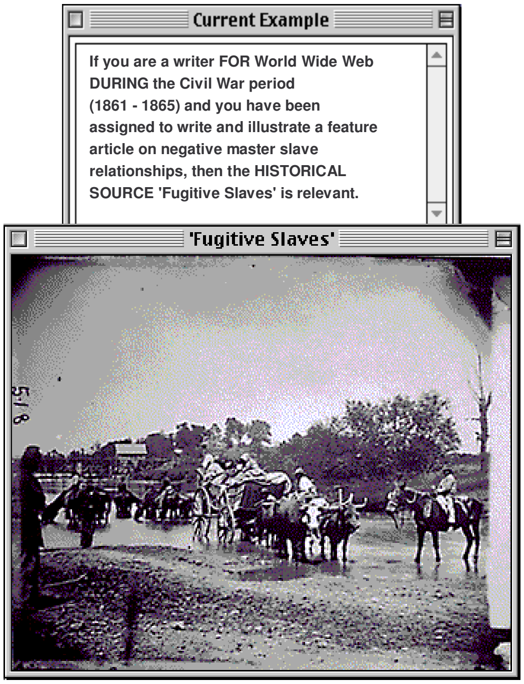

Knowledge Base
Learning by Analogy and Experimentation
PVS Rule
Failure Explanation
Examples Generated by the Agent
Incorrect Example
Correct Example
Learning from Explanations
Learning from Examples

Figure 9: The rule refinement method of Disciple.

The example in Figure 10 is incorrect and was rejected by the educator. In such a case the agent needed to understand why this example, which was generated by analogy with a correct example, is wrong. By comparing the two examples, the educator and the agent were able to find out that the generated example is wrong because the WORLD-WIDE-WEB was not issued during the CIVIL-WAR period. On the contrary, the initial example was correct because SOUTHERN-ILLUSTRATED-NEWS was issued during the CIVIL-WAR period. This explanation is used to specialize both bounds of the version space. This process will continue until either the two bounds of the rule become identical or until no further examples can be generated that are not already covered by the plausible lower bound. The final learned rule is the one from Figure 5. This training phase of the agent continued until 54 relevancy rules were learned.



If you are a writer FOR World Wide Web DURING the Civil War period (1861 - 1865) and you have been assigned to write and illustrate a feature article on negative master slave relationships, then the HISTORICAL SOURCE 'Fugitive Slaves' is relevant.

Figure 10: An example generated by the agent by analogy with the example in Figure 4*

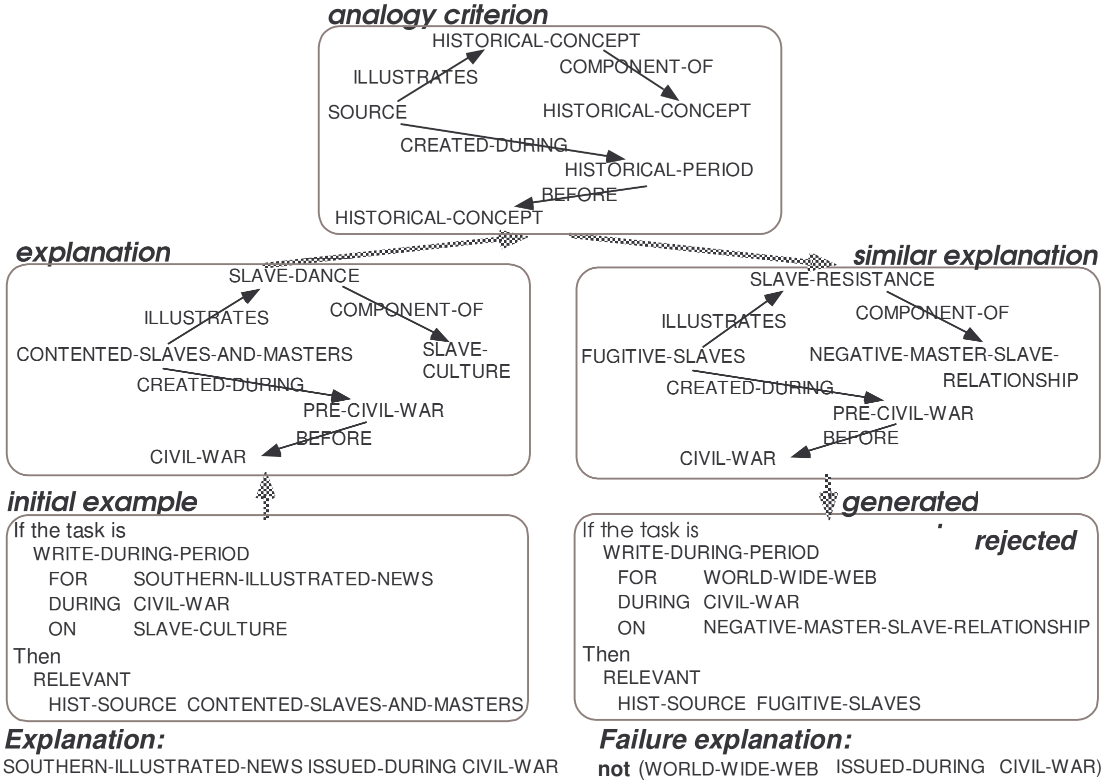


Figure 11: Analogical reasoning in Disciple.

* Picture reproduced from LC-USZ622-14828, Library of Congress, Prints & Photographs Division, Civil War Photographs

Developing the test generation engine

One of the agent's requirements was that it generates not only test questions, but also feedback for right and wrong answers, hints to help the student in solving the tests, as well as explanations of the solutions. Moreover, the agent's messages needed to be expressed in a natural language form. Although the rules learned by the agent contain almost all the necessary information to achieve these goals, some small adjustments were necessary. In the case of the rule in Figure 5, the educator and the agent needed to define the templates for the Hint, Right Answer and Wrong Answer, shown in Figure 12.

Hint

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

Right Answer

> The source ?S3 is relevant to your task because it illustrates ?S4 which was a component of ?S2, ?S3 was created during ?P2 which was before ?P1 and ?S1 was issued during ?P1.

Wrong Answer

> 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.

Figure 12: Additional templates associated with the rule in Figure 5.

The Hint in Figure 12 is the part of the Explanation in Figure 5 that refers only to the variables used in the formulation of the test question. The Right Answer in Figure 12 is generated from the Operation Description and the Explanation in Figure 5, and the Wrong Answer is a fixed text.

It should also be noticed that in the cases where the refined rules still contained an upper bound condition and a lower bound condition, the upper bound condition was dropped and the lower bound condition became the only condition of the rule.

The learned rules can be used to generate different types of test questions. In the current version of the agent we have chosen to develop a test generation engine that can generate the following four classes of test questions listed in ascending order of the level of difficulty:

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

Similar test questions could be generated for each evaluation skill such as, IF CREDIBLE test question or WHY CREDIBLE test question.

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 one hint and one explanation that will indicate one reason why S is relevant to T. 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.

The generation of an IF RELEVANT test question where the source is relevant is a very efficient process. The case where the source is not relevant is more computationally expensive. In such a case, the agent has to 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 an 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. For instance, to generate an WHICH RELEVANT test question the agent first has to generate an IF RELEVANT test question with a task T and relevant source S. Then two additional sources that are not relevant to T are looked for, as described above.

The method for generating WHY RELEVANT test questions is the following one. First an example E_1 of a relevancy rule R_1 is generated. 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. This rule could be from another family of relevancy rules, or could be a rule corresponding to another evaluation skill, for instance credibility or accuracy. 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. For instance, it could explain why S is relevant to another task T_1.

It should also be noticed that the agent is always choosing at random an element from a set. Therefore, its behavior is different from one execution to another.

EXPERIMENTAL RESULTS

The ontology of the test generation agent includes the description of 252 historical concepts, 80 historical sources, and 6 publications. The knowledge base also contains 54 relevancy rules grouped in four families, each corresponding to one type of reporter task. Two of the families contain 18 relevancy rules and the other two contain 9 relevancy rules.

There are 40,930 instances of the 54 relevancy rules in the knowledge base. Each such instance corresponds to an IF-RELEVANT test question where the source is relevant. In principle, for each such test question the agent can generate several IF-RELEVANT test questions where the source is not relevant, as well as several WHY-RELEVANT, WHICH-RELEVANT and WHICH-IRRELEVANT test questions. Therefore, the agent can generate more than 10^5 different test questions.

The 54 relevancy 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 four types of experiments with the test generation agent. The first experiment tested the correctness of the 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 teaching expert. 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, given that assessing relevance is, to a certain extent, a subjective judgement. The third and the fourth experiments tested the quality of the test generation agent, as judged by students and by teachers.

The results of the first two experiments are summarized in Table 1. To test the predictive accuracy of the knowledge base, 406 IF RELEVANT test questions were randomly generated by the agent and answered by the developing expert, each time recording the agreement or the disagreement between the expert and the agent. They agreed in 89.16% of the cases.

Reviewer	Number of reviewed questions	IF questions with relevant sources	IF questions with irrelevant sources	Time spent to review questions	Accuracy on relevant sources	Accuracy on 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%	–	–

Table 1: Evaluation results.

We have performed a similar experiment with a domain expert who was not involved in the development of the agent. This independent expert has answered another 401 randomly generated IF RELEVANT test questions. This time, the expert and the agent agreed in 85.76% of the cases and disagreed in 14.24% of the cases. These disagreements were analyzed by the developing expert and by the independent expert. There were cases where the two experts disagreed themselves, mainly because the independent expert had a broader interpretation of some general terms (such as slave culture, activities related to slavery, cruelty of slavery, and master slave relationships) than the developer of the knowledge base. However, the independent expert agreed that someone else could have a more restricted interpretation of those terms, and, in such a case, the answers of the agent could be considered correct. There were also 5 cases where the independent expert disagreed with the agent and then, upon further analysis of the test questions, agreed that the agent was right.

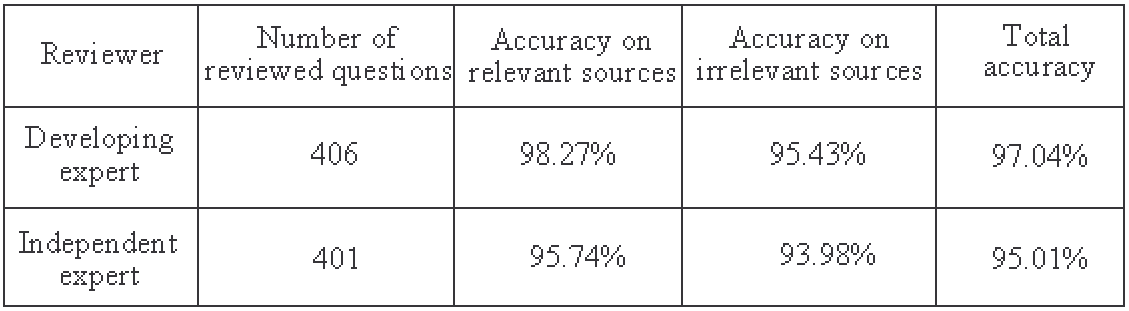
These two experiments have also revealed a much higher predictive accuracy in the case of IF RELEVANT test questions where the source was relevant. This was 96.53% in the case of the developing expert and 95.53% in the case of the independent expert. The predictive accuracy in the case of irrelevant sources was only 81.86% in the case of the developing expert and 76.35% in the case of the independent expert. To confirm these results we have conducted an additional experiment with the independent expert, who was shown other 1,326 IF RELEVANT test questions where all the sources were relevant (for a total of 1,524 such questions). In this case the predictive accuracy of the agent was 96.19%.

We have also analyzed in detail each case where both the developing expert and the independent expert agreed that the agent failed to recognize that a source was relevant to a certain task. In most cases it was concluded that the representation of the source was incomplete. This analysis suggested the following "Projection" principle for the representation of the historical sources in the knowledge base of the agent:

Any historical source must be completely represented with respect to the concepts from the KB.

This means that if the knowledge base contains a certain historical concept, then the representation of any historical source referring to that concept should contain that concept. This does not mean, however, that the contents of the historical sources have to be completely represented (a task that would be very hard, especially for pictures). That is, if a certain concept C is not part of the agent's ontology, then the representation of a source need not contain C even if, in general, one would use C to describe the content of S. Operationally, the projection principle states that if the expert decides to describe a new source in terms of some new concept C, then the expert has to review again the descriptions of each source S from the knowledge base. If the expert decides that S refers to C, then she or he has to include C in the representation of S.

Following the Projection principle, we have compared the description of each of the 80 historical sources from the agent's knowledge base with the 252 historical concepts from the knowledge base and we have extended some of the sources' descriptions to be complete with respect to these 252 concepts known to the agent. Then we have rerun the agent to answer again the test questions from Table 1 and compared the agent's answers with those of the experts. These new results are shown in Table 2. As one could notice, there is a significant improvement in the agent's performance. This time, the experts and the agents agreed, on average, in 96% of the cases.



Reviewer	Number of reviewed questions	Accuracy on relevant sources	Accuracy on irrelevant sources	Total accuracy
Developing expert	406	98.27%	95.43%	97.04%
Independent expert	401	95.74%	93.98%	95.01%

Table 2: Evaluation results after the application of the Projection principle.

Table 1 also indicates the evaluation time because, unlike automatic learning systems, interactive learning systems require significant time from domain experts, and this factor should be taken into consideration when developing such systems. First of all, one could notice that it took twice as long to the independent expert to analyze 401 test questions than it took to the developing expert. This is because the independent expert was not familiar with any of the 80 historical sources used in the questions, and he had to analyze each of them in detail in order to answer the questions. However, once the independent expert became familiar with the historical sources, he answered the new 1,326 test questions much faster.

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 d isagree. Figure 13 presents the results from 7 of the most informative survey questions.

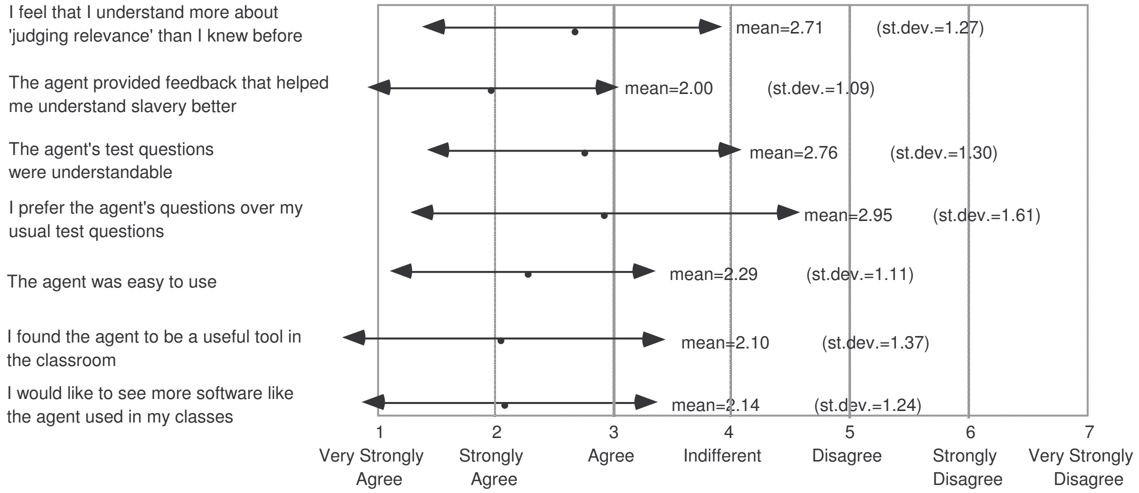


Figure 13: Student survey results.

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 teacher’s responses are presented in Figure 14.

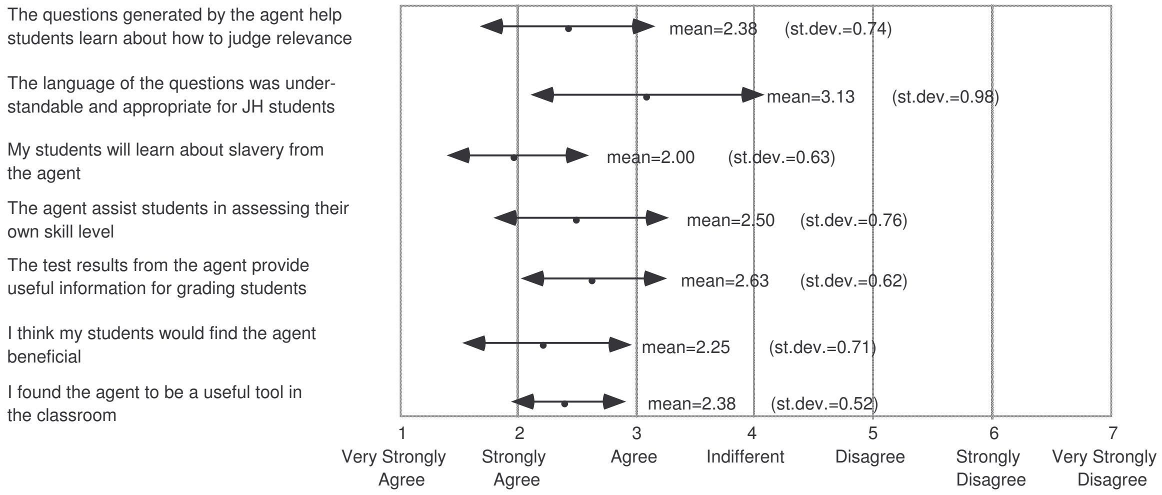


Figure 14: Teacher survey results.

CONCLUSIONS

In this paper we have presented the Disciple approach and its application to developing an educational agent that generates test questions to assess student's understanding and use of higher-order thinking skills in the domain of history. 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 developed test generation agent provides the educator with a flexible tool that lifts the burden of generating tests for large classes. The agent is also a useful tool for the students that can use it in the self-assessment mode, and can even receive limited tutoring from it, in the form of hints to answer the test questions, answers and explanations of the answers. Since 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 (Hamburger and Tecuci, 1998). From the point of view of the artificial intelligence research, this work shows an integration of machine learning and knowledge acquisition with problem solving and intelligent-tutoring systems. From the point of view of the education research, it shows an automated computer-based approach to the assessment of higher-order thinking skills, as well as an assessment that involves multimedia documents. Our experience with the development of this and other agents has shown that Disciple can easily be applied to a wide range of domains (Tecuci, 1998; Tecuci et al., 1999). Future work includes further development of the Disciple approach and its application to other challenging problems, including that of building a statistical analysis assessment and support agent.

ACKNOWLEDGMENTS

Tomasz Dybala and Kathryn Wright contributed to the development of the Disciple Learning Agent Shell. Lynn Fontana, Rich Rosser, and David Webster assisted in the development of the knowledge base of the agent. Harry Keeling was the developing expert. Lawrence Young, who holds an MS in History Education, acted as the independent expert. Ward Cates and M.J. Bishop provided useful suggestions for the test questions to be generated by the agent. We would also like to thank the reviewers who provided useful comments on this paper. This research was done in the Learning Agents Laboratory at George Mason University in Fairfax, Virginia. The research of the LA Lab 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, as part of the Computer-Aided Education and Training Initiative, and by the NSF grant No. CDA-9616478, as part of the program Collaborative Research on Learning Technologies.

REFERENCES

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

Beyer, B. (1987). *Practical Strategies for the Teaching of Thinking.* Boston, MA: Allyn and Bacon, Inc..

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

Bloom, B. (1956). *Taxonomy of Educational Objectives*. New York: David McKay Co., Inc.

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,* San Mateo, CA: Morgan Kaufmann.

Chaudhri, V. K., Farquhar, A., Fikes, R., Park, P. D., and Rice, J. P. (1998). OKBC: A Programmatic Foundation for Knowledge Base Interoperability. In *Proc. AAAI-98*, pp. 600 – 607, Menlo Park, CA: AAAI Press.

DeJong, G. and Mooney, R. (1986). Explanation-Based Learning: An Alternative View, *Machine Learning*, 1, 145-176.

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*.

Forbus, K. and Gentner, D. (1989). Structural evaluation of analogies: what counts? *Cognitive Science*.

Frasson and Gouarderes, (1998). *Proceedings of the ITS'98 Workshop on Pedagogical Agents*, San Antonio, Texas.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R. (editors), *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic.

Hamburger H. and Tecuci G. (1998). Toward a Unification of Human-Computer Learning and Tutoring, In Goettl, B.P., Halff, H.M., Redfield, C.L., and Shute, V.J. (editors), *Intelligent Tutoring Systems*, 444-453, Berlin: Springer-Verlag.

Mengelle, T., De Lean, C., and Frasson, C. (1998). Teaching and Learning with Intelligent Agents: Actors. In Goettl, B.P., Halff, H.M., Redfield, C.L., and Shute, V.J. (editors), *Intelligent Tutoring Systems*, 284-293, Berlin: Springer-Verlag.

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

Mitchell, T.M., Keller, T., and Kedar-Cabelli, S. (1986). Explanation-Based Generalization: A Unifying View, *Machine Learning*, 1, 47-80.

Mitchell T.M., Mahadevan S., and Steinberg L.I. (1990). LEAP: A Learning Apprentice System for VLSI Design. In Kodratoff Y. and Michalski R.S. (Eds.), *Machine Learning*, vol III, San Mateo, CA: Morgan Kaufmann.

Munro, A., Johnson, M., Pizzini, Q., Surmon, D., Towne, D., and Wogulis, J. (1997). Authoring Simulation-Centered Tutors with RIDES. *International Journal of Artificial Intelligence in Education,* 8, 285-316.

Quillian, M. R. (1968). Semantic Memory, In Minsky, M. (editor), *Semantic Information Processing*, 227-270, Cambridge, Mass: MIT Press.

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

Tecuci G. (1998). *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*, Academic Press.

Tecuci, G., Boicu, M., Wright, K., Lee, S.W., Marcu, D., and Bowman, M. (1999). An Integrated Shell and Methodology for Rapid Development of Knowledge-Based Agents. In *Proceedings for the Sixteenth National Conference on Artificial Intelligence,* Menlo Park, CA:AAAI Press.

Veermans K. and Van Joolingen, W.R. (1998). Using Induction to Generate Feedback in Simulation Based Discovery Learning Environments. In Goettl, B.P., Halff, H.M., Redfield, C.L., and Shute, V.J. (editors), *Intelligent Tutoring Systems*, 196-205, Berlin: Springer-Verlag.