A multi-agent system for advising and monitoring students navigating instructional Web sites

Claudia Roda
Computer Science Department, American University of Paris
Paris 75007, France

ABSTRACT

A growing community of teachers, at all levels of the educational system, provides course material in the form of hypertext/multimedia documents. In most cases this is done by creating a "course Web site". This paper explores the issues related to the design of software systems that aid teachers in monitoring how students use their sites and proactively advise students navigating the sites. In connection to these functions two important topics in current applications of technology to education are discussed. Firstly the definition of a set of criteria allowing the evaluation of the appropriateness of multi-media and hypertext technologies vis à vis to classic course support material and in particular textbooks. Secondly the issue of the utility and acceptability of proactive user interfaces such as interface agents or personal assistant agents. A multi agent system capable of advising and monitoring students navigating instructional Web sites is introduced and it is used as a basis for discussion of the above two topics. The system generates and uses a set of indicators evaluating how much use is made of hypertext and multimedia tools as well as indicators of usefulness and cognitive support of the proactive user interface.

Keywords: Tutoring systems, Multi agent systems, World Wide Web, Autonomous Interface Agents, Digital Course Material, XML.

1. INTRODUCTION

The need for the research described in this paper arises from several years of experience teaching introductory level classroom courses providing the majority of lecture notes and student exercises in hypertext format. This type of material has the advantage of allowing the students to review the subjects covered in class at their own pace, conforming with their own distinctive reasoning style (they may follow links as they please).

The use of hypertext has appeared to be more effective, compared to a classic sequential textbook methodology, especially for teaching students with heterogeneous backgrounds, which is the case for the American University of Paris. However, monitoring the students' coverage of the material and advising them on the pathways to follow through the hypertext documents becomes very time consuming.

To address this problem, we have started research into the design of a multi-agent system capable of monitoring and advising students navigating hypertext course material. This system is intended to take full advantage of the high level of interaction that is made possible by the use of electronic media as opposed to classic textbooks. A CONcept BAsed system (ConBa) has been devised which, on the basis of the relations existing amongst the concepts introduced in each page of the course Web site, will advise students whilst keeping record of the activity performed. The information about these concepts and their relations may be incrementally supplied by the authors of the Web site.

This paper describes how the ConBa system's design responds to the requirements for efficient advising and monitoring of instructional Web sites.

2. SYSTEM OVERVIEW

A ConBa system includes one Professor agent and a Tutor agent for each one of the students following the course (see Figure). The Professor agent provides the interface to the human instructor for performing three main activities:

- Managing the course material.
- Interrogating the Tutor agents to find out how students are proceeding in the course, e.g. which subjects have been covered by the students, which performance benchmarks have been achieved, etc.
- Updating the student's state; e.g. inform the Tutor agent that a certain skill can be considered as acquired by the student.

The Tutor agent maintains a student description organised in two parts. One part is static and it is supplied by the student her/himself. The second part is dynamically generated by the Tutor agent and summarises the level and quality of course coverage in terms of:

- The topics that have been examined by the student
- The exercises that have been successfully completed
- The exercises that have been attempted but failed, etc.

The Tutor agent uses the student's description for two main purposes: to respond to the Professor agent's requests about the student's status (monitoring function) and to determine an appropriate suggested path through the course material (advising function). Both parts of the student description (static and dynamic) are used to tailor, to the student's specific needs and preferences, the Tutor agent's suggestions for possible future activities. An interface between the system and the student dynamically builds the hypertext pages presented to the student following the Tutor agent's indications and containing its suggestions.

Tutor agents interact through polling in order to exchange information on:

- The most fruitful pathways followed by students,
- The performance benchmarks which are either too easy or too hard to achieve,
- The topics (nodes) which are seldom visited, etc.

This information is used both to advise the students and to inform the human instructor on possible strengths and weakness of the course structure.

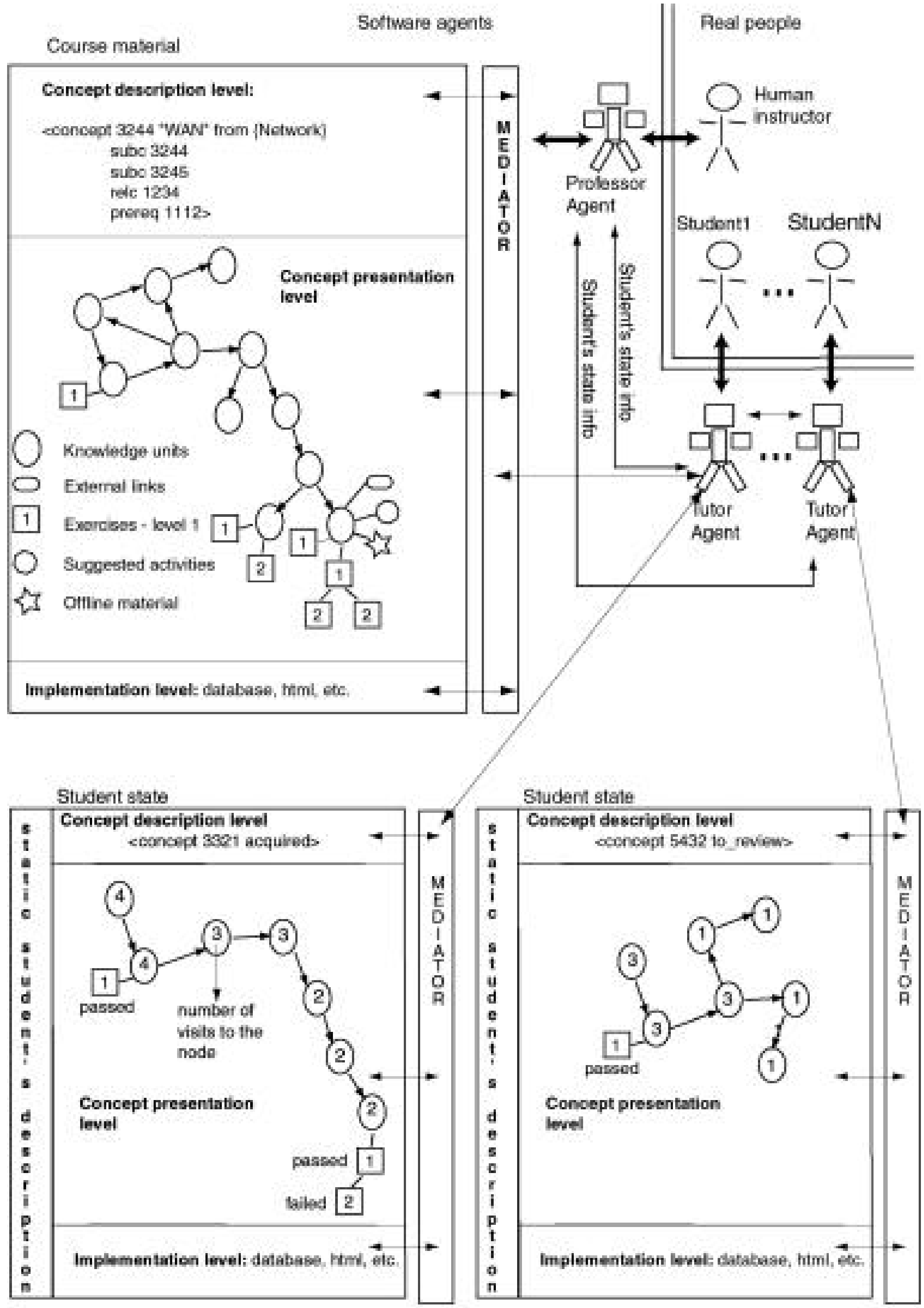

Figure 1 - Overall system structure

Focus and related works

The main focus of this research is currently on exploiting the site's structure and annotations rather than its multimedia features. Other works have concentrated more on this second issue, or on the integration of Web based presentations and other technical tools. The Internet Softbot [4], for example, provides a uniform interface to several Internet resources and facilities (ftp, telnet, gopher, etc.). The Manic system [20], concentrates on combining text, graphics and sound in order to support a multimedia distance learning environment. The Trilogy project [15] uses intelligent agents to implement a virtual research laboratory integrating physically distributed hardware and software resources.

The design of the ConBa system presents three important aspects of agenthood: decomposition, autonomy and anthropomorphism. The system described is very naturally decomposable in agent components each with a specific objective to achieve [5] and representing a user of the system. Autonomy is displayed by the system's ability to perform actions without an explicit request of the user and possibly asynchronously with respect to the user activity (e.g. a Tutor agent may be querying another one whilst the user is surfing the site). Anthropomorphism is due to the choice to present the results of the Tutor agent's reasoning as "motivated suggestions" rather than using an non-anthropomorphic presentation such as a set of dynamically modified navigation menus. This choice allows us to supply a "lightly" anthropomorphic interface that however doesn't have more complete anthropomorphic features such as the life-like characters proposed, for example in [1, 17, 21].

3. MOTIVATIONS

The ConBa system is being designed to respond to two main needs. On one hand it aims at facilitating the monitoring of the activity over instructional Web sites. On the other hand it supports this activity by proactively suggesting best paths through the site itself and related links. These goals reflect two important issues in current applications of technology to education (as well as to many other human activities). Firstly the definition of a set of criteria allowing the evaluation of the appropriateness of multi-media and hypertext technologies vis à vis to classic course support material and in particular textbooks. Secondly the issue of the utility and acceptability of proactive user interfaces such as interface agents or personal assistant agents.

Hyper/Multi-media versus classic course material

Is course material presented in the form of hyper/multi media "better" than material presented in classic book format? The fashionable answer is "yes". However some distinctions should be made. "The appearance of new means of information does not destroy earlier ones; it frees them from one kind of constraint or another" [2] In the case of hyper/multi media the new means allow for:

- The non-linear presentation of the information
- The integration of text, sound, and still and moving images

In order to evaluate the advantages of using hyper/multi-media technologies in the presentation of course material, at least one of the two characteristics above must be exploited. Some evidence of their utility can then be gained by observing how much students take advantage of these new facilities. As a consequence, the ConBa system aims at monitoring indicators such as:

- How often students visit the site in a non-linear fashion, i.e. how often surfing deviates from the main suggested path.
- How much use of non-textual information is made by the students.
- How often students access complementary material, e.g. suggested links.

These indicators should be available to the instructor along with general information about the course coverage for each student.

User interface acceptability

As briefly introduced above, the ConBa system aims at providing a *proactive* user interface capable of supporting students in the navigation of instructional Web sites. Classic user interfaces, often called *direct manipulation interfaces* [18], respond to specific requests of the user acting on explicit interface objects. The ConBa system instead, is based on a proactive interface. Such interfaces are generally based on the notion of *indirect management* [6]. The user does not have to explicitly indicate every action for the system, which is able to autonomously initiate activities and plays a proactive role. Note that ConBa's proactive interface integrates the classic direct management interface, this means that the user may initiate requests on a direct manipulation interface and the system, responding to those requests, tries to extend its performance by proposing further activities. For example, responding to a student's request for a Web page, the Tutor agent, on the basis of the information collected about the students' activities and of their profiles, may:

- Suggest possible paths through the site and
- Inform the student about relevant material of which the student may not be aware of

In order to play this active role, the Tutor agent may autonomously take actions which were not specified by the user (e.g. look for further material related to the user request, query other Tutor agents to define possible successive actions, etc.). The Tutor agent must also manage the information about the user in order to tailor its action to the specific user and his/her surfing history, i.e. the Tutor agent must learn about the user. In this sense, the Tutor agent can be seen as an interface agent as defined in [16] or as an Autonomous Interface Agent as defined in [8]. A lively debate is currently ongoing on the usefulness of interface agents; see for example [3, 7, 11, 12, 14, 19]. Some of the main questions are:

- **Usefulness** - can interface agents significantly reduce the workload of their users?
- **Cognitive support** – can interface agents present the information or tools available to the user in a fashion that allows a more efficient use of these resources and stimulates the user's interest?
- **Autonomy acceptability** - will users accept agents taking autonomous action? How can users be reassured about their level of control over the system?
- **Privacy** - up to which point an interface agent may "learn" about the user without jeopardise his/her privacy?
- **Anthropomorphism** - is the association of anthropomorphic characteristics to interface agents necessary / desirable?

We believe that, comparative studies aside, the best indicator of the agent's usefulness will be a report on how often students follow the Tutor agent's suggestions. However, the relevant suggestions for this type of evaluation should only be those which deviate from predefined paths through the Web site. For example, a suggestion to go to page X, where X is already connected by a "next" link to the current page is not relevant for the evaluation of usefulness. Note however that we consider such a suggestion absolutely legal in terms of the Tutor agent's reasoning. In fact this agent may produce such suggestions (which are essentially only based on the structure of the site) until enough information is collected on the student's surfing history and preferences so that deviations from predefined paths may be proposed. Furthermore, by gaining knowledge about the student, the Tutor agent should modify its own reasoning strategy in order to adapt to the student. Data collected about the user reactions to such strategies should be used both for the strategies evaluation and for verifying the usefulness of the interface agent. This aspect is further discussed in the next section.

Cognitive support may be seen as the level of "encouragement" that the interface agent is able to give to the user. If the resources available (whether information or tools) are presented in an uniform, accessible and stimulating way, the user will be more likely to use them efficiently. The Tutor agent tries to achieve these goals by:

- suggesting to the student relevant material that is tailored to his/her level of knowledge, specific interests, and previous surfing history
- presenting various type of material (e.g. site's pages, external links, printed material, etc.) in an uniform fashion.

The cognitive support indicator should measure how often the student will access material which is not explicitly required for the course such as suggested readings, in-depth studies, suggested exercises, etc.

As also noted in [8], the issue of autonomy acceptability is not very critical in systems such as the one described in this paper. This is because the agent does not take decisions nor acts on behalf of the user. It simply generates suggestions. The only relevant aspect is that the agent's communications to the user should be as unintrusive as possible. This is obtained in the ConBa system by reserving a small area on the bottom of the browser's window for agent's communications to the user.

The issue of privacy is still open in our study. For the time being the only feature foreseen in the system addressing this topic is a regular update to the student about the data recorded by the system. At login and logout, students will see a summary of the recorded data. However they will not be able to directly edit this information.

A "light" anthropomorphic design for the interface has been chosen. By "light" we mean that the interface may suggest human characteristics to some users but it does not explicitly presents itself as a life-like character. This is obtained simply by giving textual motivated suggestions such as: "You have failed an exercise on the topic of Boolean search, would you like to see an example for this topic?" The choice of this type of interface was made in the effort to balance two conflicting requirements. On the one hand it has been demonstrated that anthropomorphic interfaces are particularly well suited for tutoring systems and may help motivating students and making them feeling comfortable with the system [AND, LES]. On the other hand, the presence of explicit life-like characters increases the agent's "presence" and therefore the agent may become intrusive.

4. CHARACTERISTICS

Dynamic system environment – knowledge level reasoning

One of the main objectives of the design is to maintain maximum flexibility in terms of the type of material provided by the human instructor. The aim is to allow teachers, who may have their sites already designed, to use the ConBa system with minimal effort. For this reason, the system is designed to work with limited input, i.e. only the HTML documents are provided and identified as knowledge units, suggested activities or exercises. In this case the system analyses the HTML code and builds a simple graph that describes the links between the various pages. These pages are then formatted and presented to the students by the Tutor agent, which monitors progress, and gives advice using the simple graph.

However the human instructor may provide much more information on the possible pathways through the nodes. Examples are:

- Identify the concepts developed in each page (node)
- Specify precedences between concepts
- Specify relations between concepts
- Specify the number and level of exercises that need to be "passed" for the associated concept to be considered acquired.

Whenever this information is provided the Tutor agent can use it to give better advice to the students. The ideal ConBa site will be developed starting from the definition of the fundamental concepts and their relations along with the concepts presentations in Web pages. However, in practice, many sites already exists and they can be incrementally annotated adding "comments" such as the ones listed above, to pre-existing hypertext material.

The environment in which the ConBa system works (the Web site) is therefore highly dynamic. Such dynamic environments present several challenges, amongst which: the need to work with possibly inconsistent information and the impossibility to foresee all the relations amongst the objects in the environment. For example, a "student state" recorded in previous visits to the site may be inconsistent with the "course material" description which may have been updated in the meantime by the teacher. Furthermore, advising cannot be based on static relations amongst objects in the environment because these objects (pages, concepts, exercises, etc.) and their relations may not be known and may change through the life of the system.

One approach that has been successfully applied to the management of highly dynamic environments is knowledge level reasoning [13]. In this approach there exists a part of the system which is able to reason about the environment without worrying about lower level details. For example, when a Tutor agent is trying to assess how well a concept has been covered by a student, it will not be concerned about how that concept has been presented to the student and if those presentations are still available. On the other hand presentation becomes relevant when trying to assess the exposition of concepts best suited for the student. Finally, the physical display of the Web pages requires implementation level details. Creating several levels of description of the environment (see figure) allows to reason at the appropriate level and to manage dynamic changes transparently to above layers. Knowledge level reasoning is typical of deliberative agent systems.

Adapting to the user

Guiding students through instructional material should not only be a matter of specific topics but also of general student's background, preferences, experience etc. Knowledge based interface agents [10] always make use of two types of knowledge: domain knowledge (e.g. the structure of the Web site) and knowledge about the user. Moreover, human instructors normally form a "general impression" of each student, which informs their successive suggestions. This "general impression" basically allows a human tutor to tailor responses to a student's specific needs. The Tutor agents in the ConBa system mimic this mechanism by modifying the *strategies* guiding their suggestions to the user. *Strategies* are meta-level knowledge used by the Tutor agents to update their own reasoning mechanism on the basis of observed user behaviour, direct user input or on request by the human instructor. Such strategies may go from changing the "difficulty level" of the material proposed to the student, to rating certain types of concept's presentations (e.g. examples) as better suited for the student than other types (e.g. alternative explanations).

5. COURSE MATERIAL AND STUDENT'S STATE DESCRIPTIONS

The information relative to the available course material is conceptually organised in several layers.

The highest layer, called *concept* layer, is currently defined by an XML vocabulary that specifies that concepts:

- may be derived from more general concepts,
- may have more specific concepts,
- may be related to other concepts,
- may require other concepts to be understood,
- are *presented* by certain lower level resources.

Consider, for example, some course material relating to computer networks. The concept of Local Area Network is derived from the more general concept of Network, has more specific concepts such as the Token ring concept, is related to the Wide Area Network concept, etc.

The *presentation* elements in the concept description allow to create a direct connection between the concept level description and the presentation description. The presentation may be within one of the site pages or on some other resource whether online or not.

The *presentation* layer description is also currently defined by an XML vocabulary that for each presentation element indicates:

- The source type and location e.g. one of the site nodes (pages), an external link, an entry in a library database, etc.
- The presentation type: whether it is an introduction to the concept, an example, a definition, a full paper, etc.
- The presentation level: easy, average, complex.
- The target audience: professional, novice, computer science student, communication student, etc.

Obviously there may be several presentations for the same concept and a single page of the site may include several presentation elements.

At the lowest level, the site itself is represented using the Structured Graph Format (SGF) vocabulary [9]. SGF is an XML compliant markup language based on structured graphs for capturing Web site structures'.

The student's state description contains a "static" section supplying general information about the student such as name, major, email, and specific interests. This section is called "static" because it is not modified at run time by the system. It may however, be updated by the student. The dynamic student description is an annotated partial copy of the course description summarising the student's surfing history. The annotations recapitulate the course coverage (e.g. "this concept is acquired", "this exercise has been successfully completed", etc.) and contain the indicators described earlier in the paper (e.g. deviations from standard paths; access to external links, optional material, sound or video clips).

6. CONCLUSIONS AND FUTURE WORK

Thanks to the availability of tools enabling the easy and efficient creation of hypertext/multimedia documents, a growing community of teachers is experimenting with the use of such documents to support their classroom or distance-learning courses. It is time now to capitalise on the teacher's effort by developing specialised systems that allow both the easing of the burden of monitoring and guiding the students through the digital course material, and the evaluation of the effectiveness of this new type of material. This paper has analysed some of the major issues and requirements involved in the development of such specialised systems. The relevant information that should be gathered in order to evaluate the course material has been defined. Possible problems related to the acceptability of the system have been indicated. These are related both to the flexibility required in order to allow teachers to continuously update their courses, and to the design of an interface suitable to the students' need. A specific possible implementation of the system, based on a multi-agent design, has been described.

We intend to continue our project by completing the implementation and initiate trials possibly as soon as the next fall semester. Amongst the open issues the most pressing ones are related to the inclusion of students' annotations to the digital course material and the integration of sites developed, from several teachers, for different courses.

7. REFERENCES

[1] G.Ball et Al. Lifelike computer characters: the Persona project at Microsoft, *Software Agents* J.M.Bradshaw Editor, 191-222, AAAI press/ MIT press 1997

[2] U. Eco, Why new media won't kill new books, *World Press Review*. June 1996: 16-17

[3] T. Erickson, Designing agents as if people mattered, *Software Agents* J.M.Bradshaw Editor, 79-96, AAAI press/ MIT press 1997

[4] O.Etzoni, D.Weld, A softbot-based interface to the Internet, *Communications of the ACM*, 37(7):72-76 July 1994.

[5] N.R.Jennings and M.Wooldridge, Agent Oriented Software Engineering, Handbook of Agent Technology J. Bradshaw Editor AAAI/MIT Press. (to appear)

[6] A. Kay, User Interface: A personal view, *The art of Human-Computer interface design* B.Laurel Editor, 191-208, Addison-Wesley 1990

[7] B. Laurel, Interface Agents: metaphors with character, *Software Agents* J.M.Bradshaw Editor, 67-77, AAAI press/ MIT press 1997

[8] H. Lieberman, Autonomous Interface Agents, *ACM Conference on Human-Computer Interface* CHI-97, Atlanta, March 1997

[9] O. Liechti, M. Sifer, T. Ichikawa, Structured graph format: XML metadata for describing Web site structure, *Computer Networks and ISDN Systems* 30 (1998) 11–21

[10] P.Maes, Agents that reduce work and information overload, *Communications of the ACM* 37(7),

[11] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, D. Zabowski, Experience with a learning personal assistant, *Communications of the ACM* 37(7) 81-91, 1994

[12] N. Negroponte, Agents: from direct manipulation to delegation, *Being digital* by N.Negroponte, Alfred A. Knopf, 1995.

[13] A. Newell, The knowledge level, *Artificial Intelligence* 18, 87-127, 1982

[14] D.A.Norman, How might people interact with agents, *Software Agents* J.M.Bradshaw Editor, 49-55, AAAI press/ MIT press 1997

[15] T.J.Norman and N.R.Jennings "Constructing a virtual training laboratory using intelligent agents" Int. Journal of Continuous Engineering and Life-Long Learning. (to appear)

[16] H. S. Nwana, Software Agents: An Overview, *Knowledge Engineering Review* Vol.11, N.3, pp.205-244, October/November, Cambridge University Press, 1996.

[17] E. Shaw, W.L. Johnson, and R. Ganeshan, Pedagogical Agents on the Web, Proceedings of the Third International Conference On Autonomous Agents, pp. 283-290, May, 1999

[18] B. Shneiderman, Direct Manipulation: a step beyond programming languages, *IEEE Computer* 1983 16(8) 57-69

[19] B. Shneiderman, Direct manipulation versus agents: paths to predictable, controllable, and comprehensible interfaces, *Software Agents* J.M.Bradshaw Editor, 96-106, AAAI press/ MIT press 1997

[20] M.Stern, J.Steinberg, H.I.Lee, J.Padhye J.F.Kurose, Manic: Multimedia Asynchronous Networked Individualised Courseware, Proceedings of Educational Multimedia and Hypermedia, 1997

[21] B.A.Stone and J.C.Lester, Dynamically sequencing an animated pedagogical agent, *Proceedings of the National Conference on Artificial Intelligence* 424-431 AAAI 1996.