An Educational Component Based Framework for Web ITS Development

Mónica Trella, Ricardo Conejo, Eduardo Guzmán, and David Bueno

Departamento de Lenguajes y Ciencias de la Computación.
E. T. S. I. Informática. Universidad de Málaga. Apdo. 4114, 29080 Málaga. SPAIN
{trella, conejo, guzman, bueno}@lcc.uma.es

Abstract. This paper presents a framework for the integration of web-based educational systems. It is part of a research project, MEDEA[1], whose final goal is to develop a general framework to build open Intelligent Tutoring Systems (ITS). We understand "open system" as a set of autonomous educational modules which communicate between themselves, following high-level pre-established protocols. Each module can be an intelligent component with its own instruction strategies, or other components like support tools, web pages, etc. In the second case, the adaptive capabilities are left to the ITS instructor core. The architecture opens up the possibility to include new web-based components.

1 Introduction

The structure of the Internet is distributed, and in most cases it is used just as a huge repository of unorganized information. When teachers use the Internet for tutorial purposes, they can construct courses either collecting different URLs to give their students links where they can find lecture notes, exercises, etc. about the subject that they are studying; or using web-based systems to hold up their tutorial contents.

In the first approach, if not fixed courses are wanted, an intelligent behavior is required. In order to do that, teacher must redirect students to those systems that are more appropriate according to the student profile, and supervise their progress. For instance, he should suggest students to skip some parts, conduct them through pages with exercises, simulators, etc., *i.e.* he should adapt the contents to each student. Although generally the web systems used are non-adaptive, if they have some kind of intelligent behavior, it should be advisable to take advantage of these features [1].

Web Intelligent Tutoring Systems (ITS) might be constructed by gathering pages or systems, like *building blocks* in the sense of Chandrasekaran [2]. Some of them might be new, and some others reused from existing material. Most technologies used for Web ITS development are designed for small systems, and they do not have the necessary resources to guarantee a high-level conceptual organization. This kind of problems has been traditionally treated with software engineering techniques, and more recently with knowledge engineering techniques, following the Newell's *knowledge level* paradigm [3]. Using this idea, high-level methodologies and tools for the intelligent systems development have been proposed as KADS [4], PROTÉGÉ [5], KSM [6], etc.

[1] This project has been partially financed by Spanish CICYT/FEDER, with number IFD97-1286.

J.M. Cueva Lovelle et al. (Eds.): ICWE 2003, LNCS 2722, pp. 134–143, 2003.

In this regard, MEDEA system (a Spanish acronym of *Methodology and Tools for the Development of Intelligent Environments of Teaching and Learning*) is not a new ITS authoring tool, but a general framework for the development and integration of open intelligent tutoring systems [7] [8]. We understand "open system" as a set of autonomous educational components that communicate between themselves following high-level pre-established protocols. In the case of web-based open tutorial systems, this protocol is the well known HTTP. MEDEA also adds a new higher level layer to communicate intelligent tutoring components.

The components used in MEDEA can be intelligent, with its own instruction strategies. In other case, the adaptive capabilities are left to the ITS instructor core. The architecture opens up the possibility to include general-purpose components which are able to interact with the general framework. MEDEA offers to educators a generic environment to develop Web ITS, without the limitations of a close set of utilities, like in the most of the authoring systems [9].

2 MEDEA Architecture

The elements that compose MEDEA architecture can be classified in three main groups: those that contain knowledge (*knowledge modules*), those that use this knowledge for making adequate decisions along the instruction (*functional modules*), and those that are used to access and configure the system (*tools*). The base of MEDEA architecture is a core that plans the instruction, based on a set of external tutorial components.

It could be said, comparing the MEDEA architecture and the traditional ITS architecture, that *what* (knowledge about the learning subject) is explicitly represented by the domain model, *who* (knowledge about the student) is given by the student models (knowledge and attitudes) and *how* (teaching knowledge) is divided between the instructional planner, tutorial components and domain model. All knowledge models are represented in XML for computational use. Fig. 1 shows the structure of MEDEA modules:

Knowledge modules:

- **Conceptual Knowledge Domain Model**. It contains the knowledge about the subject to be taught. Both domain concepts and relationships among them are represented. It will be approached in section 4.
- **Student model**. It is decomposed into the *Student Knowledge Model* and the *Student Attitude Model*. It will be described in section 5.

Functional modules:

- **Instructional planner**. This module provides to the students the necessary guidance during the learning process. It decides at each moment the most adequate task to be performed by the student. It uses the conceptual domain knowledge, the student's state of knowledge, the student's profile and the tutorial components definitions.

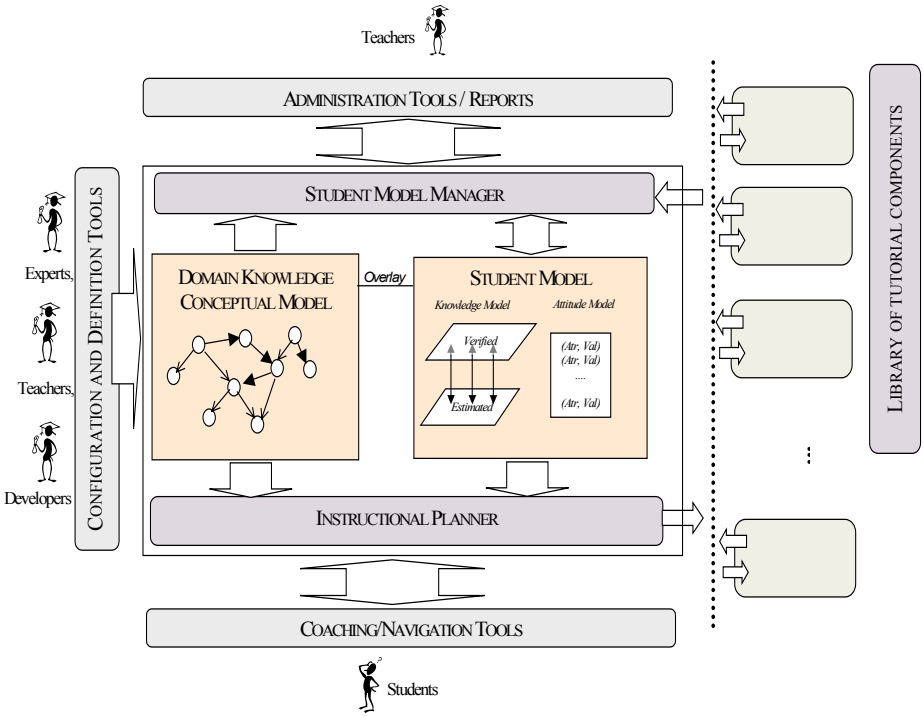

Fig. 1. MEDEA architecture

The tutoring systems generated by this tool, allow students to freely navigate. The system only recommends but does not impose the next student's action. The planner will also have the capability of justifying the recommended actions, so the student has the necessary information before deciding to accept the suggestions made by the system.

- **Student Model Manager**. It creates and updates the student model. It is updated after the execution of a tutorial component. Some components can evaluate the student knowledge about some domain concepts. The information provided by them, goes to the *Student Verified Knowledge Model.* MEDEA assumes the existence of other components which are not able to determine exactly the variations of the student knowledge level. For instance, a component which displays a text to be read. In these cases, MEDEA is just informed that the student has visited these components. This information is used to update the *Estimated Student Knowledge Model.*

- **The library of tutorial components**. They are external educational tools which make a concrete task (electronic books, simulation tools, exercises, making tests, etc.). From MEDEA point of view, the architecture of a component (Fig.2) is composed by a *partial domain model*; a *development interface* to introduce contents; a *student interface*; a *student temporal model*; a *component functional description*, where the functions of the teaching component and the way of

communicating with MEDEA planner are defined; and a *control*, that is the execution engine of the component.

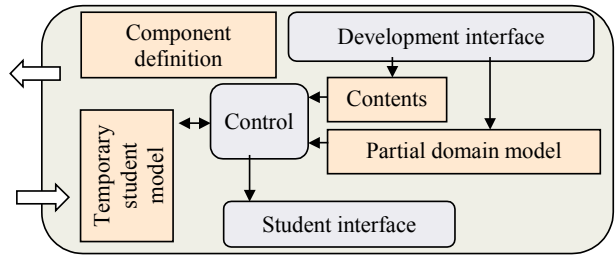

Fig. 2. Tutorial component architecture

Tools (Interfaces):

- **Configuration and definition tools**. They are used by domain experts, teachers and designers. They allow to introduce the contents, define and configure the data and knowledge modules using specific interfaces.
- **Administration tools**. They are used by teachers to monitor the evolution of their students. They show the progress of each student, statistics about the use, the average of the student's performances, and other management and administrative tasks.
- **Navigation tools**. They are used by students to support their navigation and the interaction with the whole system. It can be conceptualized as an advisor during the learning. It is currently implemented as an additional frame of the web browser.

3 The Behavior of MEDEA

To describe the behavior of MEDEA the simile of a school personal tutor can be used. Suppose that a student asks for help to his personal tutor (*instructional planner*), to study a subject (*domain model*). The tutor examines his academic expedient (*student model*) and takes into account other student's features (*student attitude model*). He selects the best topic and style of teaching for the student. After that, he consults the school staff (*tutorial components library*) and chooses the best teacher for that specific topic, according to the student's current profile. The tutor must know the teaching methods of all the available teachers (*i.e.* this usually does a lot of exercises, that prefers to explain theory, this wants the students to make exercises on the blackboard, etc.). The tutor sends the student to the selected teacher (*tutorial component*) with a message (*communication protocol*) indicating the concept which the student must study, and maybe, other information (*e.g.* this is the third time this student tries this concept, he has low learning rate,…). This teacher will apply his effort and expertise to improve the student knowledge in the concept proposed (*intelligent component*); or for instance, he may just give the student a proper text to

read (*non-intelligent component*), without controlling what he is doing. When the teacher ends the class, he will send back a message to the tutor explaining the student's behavior (*evaluation feedback*), or simply he will give back the control to the tutor without any message. With this new information, the tutor (now *the student model manager*) actualizes the student expedient. This process will be repeated until the student completes the subject.

4 The Domain Model

The domain model represents the knowledge about the subject to be learned. There are different approaches [10] depending on the nature of the represented domain. In MEDEA, declarative domains representation are used. The most extended models for representing this kind of domains are the semantic networks of knowledge units. They have been used in systems like DGC [11], Eon [12] and IRIS [13].

From a conceptual point of view, the domain is defined by *a)* a semantic network of concepts and relations between them, and *b)* pedagogical knowledge required for the instruction.

From the implementation point of view, the basic elements of MEDEA domain model are:

a) *Concepts*. They are the basic pieces in which the subject is divided. During the task of modeling a knowledge area for an ITS, the pedagogical purpose of the system should be taken into account. First of all, the net granularity is chosen, *i.e.* the decomposition of the units in other simpler ones. As Anderson says [10], sometimes to model accurately a domain, a computational charge is required. From the tutorial point of view, that is not necessary. Second, it is necessary to include pedagogical knowledge in the domain model, to guide the student through the knowledge units [8]. This pedagogical information is included as attributes associated to the concepts.

b) A set of binary *relations* between the concepts. They are used by MEDEA to describe the domain. These relations are: *prerequisite*, *part of*, *is a*, *belongs to*, *is useful to understand*, *is similar to* and *is opposite to*. Each relation should define an acyclic graph of concepts which is used by the functional modules to guide the instruction (*the instructional planner)*, and to make inferences about the student's knowledge (*the student model manager*). The current semantic of the relation is very fixed in the functional modules. However, MEDEA includes an informal description about the semantic of the relations to guide authors in the creation process of these graphs. Currently, the domain model supports all these relations, but only *prerequisite*, and *part of*, are used by the *instructional planer*. Other relations can be defined by the authors course and stored for future use. The *student model manager* is still under development.

c) *Evaluation types*. The student knowledge model is based on the overlay technique. Each concept has a magnitude associated which represents the student's degree of knowledge. The *evaluation types* in MEDEA are the types of those magnitudes. They are not fixed, but they are defined, when the course is created between a set of *internal types* supported by the architecture. These types are: *enumerated* (*i.e.* the knowledge level of a concept can be A, B, C or D), *real* (a real number) and *distribution* (*i.e.* the knowledge level can be {A/0.2, B/0.3, C/0.4,

D/0.1}, indicating the probability of that the student's knowledge level in concept A is 0.2 and so on). MEDEA is a general framework which can use different components for its instructional purposes. Each of these components can have its own internal representation of the student's knowledge. Implicit conversion between internal types has been defined to support compatibility between different components.

```
<!DOCTYPE DOMAIN_MODEL SYSTEM "http://sirius.lcc.uma.es/medea/dtd/DOMAIN_MODEL.dtd">
<DOMAIN_MODEL id="domain_model01" name="Logic of proposals">
   <EVALUATION_TYPES>
        <EVALENUM id="EvalEnum" default_minimum_mark="Passed">
            <ENUMERATED id="Passed"/>
            <ENUMERATED id="Failed"/>
        </EVALENUM>
        <EVALREAL id="EvalReal" lower_boundary="0" upper_boundary="10"default_minimum_mark="5"/>
   </EVALUATION_TYPES>

   <CONCEPTS>
        <CONCEPT id="t1" idref_evaluation="EvalEnum" name="Introduction"  difficulty="Low"/>
        <CONCEPT id="t2" idref_evaluation="EvalEnum" name="Formal sintax" difficulty="Low"/>
        <CONCEPT id="t3" idref_evaluation="EvalEnum" name="Semantic" difficulty="Low"/>
        [...]
        <CONCEPT id="c32" idref_evaluation="EvalReal" name="CDN" difficulty="High"/>
   </CONCEPTS>

   <RELATIONS>
        <RELATION id="r1" id_origin_concept="c21" id_destiny_concept="c1" type="prerequisite"/>
        <RELATION id="r4" id_origin_concept="c4" id_destiny_concept="c2" type="is_a "/>
        [...]
        <RELATION id="r73" id_origin_concept="c31" id_destiny_concept="t6" type="belongs_to"/>
        <RELATION id="r74" id_origin_concept="c32" id_destiny_concept="t6" type="belongs_to "/>
   </RELATIONS>
</DOMAIN_MODEL>
```

Fig. 3. Representation of the domain model

Fig. 3 shows a fragment of an XML file, which contains the domain model of a course of Logic. At the beginning, the types used to evaluate each domain concept are defined. The second part is a list of concepts which includes some pedagogical information required for the instruction, like the difficulty level of each concept. At the end, there is a list with the relations between concepts.

5 The Student Model

The *Student Model* in MEDEA is divided into two main subcomponent. The *Student Knowledge Model* and the *Student Attitude Model*. The first represents what the student knows about the subject. The second represents other features of the student.

The *Student Knowledge Model* is an overlay model divided into two levels: *the estimated model* and *the verified model*. Each level is a list of *concept/mark* pairs. This multilayer approach has already been used by other authors like Brusilovsky, in the last versions of ELM-ART-II [14]. He uses different layer for concepts visited, evaluated, inferred, etc. At this moment, we have only proposed two layer, grouping in the estimated layer, all the uncertain information and inferences.

Fig. 4 shows an example of the student knowledge model expressed in XML. Each concept has associated a value according to the evaluation type defined in the domain model. The last concept that has been taught is stored too.

```
<!DOCTYPE STUDENT_MODEL SYSTEM "http://sirius.lcc.uma.es/medea/dtd/STUDENT_MODEL.dtd">
<STUDENT_MODEL courseid="d03" lastconcept="c5" studentid="s432">
 <ESTIMATED_STUDENT_MODEL>
      <CONCEPT id="c1" value="VERY WELL"/>
      <CONCEPT id="c2" value="VERY BAD"/>
      [...]
      <CONCEPT id="c5" value="WELL"/>
      <CONCEPT id="c6" value="VERY BAD"/>
 </ESTIMATED_STUDENT_MODEL>
 <CHECKED_STUDENT_MODEL>
      <CONCEPT id="c1" value="VERY BAD"/>
      <CONCEPT id="c3" value="REGULAR"/>
      <CONCEPT id="c5" value="REGULAR"/>
      <CONCEPT id="c6" value="VERY BAD"/>
 </CHECKED_STUDENT_MODEL>
</STUDENT_MODEL>
```

Fig. 4. Representation of the student knowledge model

There are also some relevant student's features which are important for the learning process. In MEDEA, some of them has been included in the *Student Attitude Model*: cognitive development (formalization and abstract concepts understanding skills), motivation, learning style, time dedicated to the subject study (*i.e.* the student effort degree to pass the subject), progress (the student's learning speed), experience with computers, Internet connection speed, etc.

This model is used by teachers or course designers to establish relations between a concrete student profile and some instruction parameters. For example, a teacher can specify in the course definition that, when a student with *low motivation* level makes a test, he should see the right response after he answers to each question, instead of at the end of the test.

6 The Instructional Planner

This is the core of MEDEA architecture. It is the module which puts in sequence the domain and adapts the instruction process to each student. Usually a teacher takes decisions in several levels. First, he decides the instruction goals (concept to be taught) and then, he takes other decisions like the topics content, material to be used, the pedagogical strategy to be used, etc.

There are several examples in the ITS literature in which the planner task is divided into subtasks: some oriented to concept selection, and others to decide how to teach the selected concept [13] [15] [16].

The MEDEA planner takes three main decisions: 1) Does the student need to be evaluated?, 2) if YES: about which topic will he be evaluated?, if NO: which concept should the student learn now?, and 3) How will he be evaluated or taught better? The knowledge needed to answer these questions is in the domain model, the student

models and the pedagogical elements of the system (domain model pedagogical contents and tutorial components).

The modular structure of the MEDEA architecture and the separation between knowledge representation and knowledge use, allow that this module could be easily changed. The final goal of this project is to have a planner library in the system. For the first prototype, a heuristic planner has been implemented. It carried out the task of selecting a concept to be evaluated or learned in two phases: first, it selects an ordered set of candidate concepts. This selection takes into account the relations *prerequisite* and *part of*, and the current student model state. Second, the planner selects the first concept from the candidates set. The criteria to decide if a concept is candidate are: a) The student has not reached at the minimum required by the teacher to pass the concept. b) The concept is *prerequisite* of any other concept that cannot be learned because the student has not passed this one. c) The concept is *part of* any other concept that cannot be learned because the student has not passed this one.

A weight is assigned to each candidate. The most weighted are those that are selected by the criterion 2.

The implemented planner takes this decision using the teacher's criterion. That is, when a teacher designs a course, he links each tutorial component registered in the system, to a student's profile. The planner only has to consult the *student attitude model* and to assign him the most adequate tutorial component according to the teacher.

7 Tutorial Components

The pedagogical knowledge in MEDEA system is distributed between several modules: the domain model, the instructional planner and the tutorial components. Anyway, it is not strictly necessary for the instructional process that a component provides any knowledge. Most of the instruction tasks fall on the pedagogical system core composed by the domain model and the planner. A tutorial component can complement this task providing its tutorial strategies and a more exhaustive control over the student's actions.

MEDEA classifies the educational components as *evaluation components* or *information components*. The difference is that components of the former type are able to evaluate the student's knowledge level about a concept.

The problem of the components integration can be approached as the communication between two web-based systems. This communication can be established through URLs. Therefore, the planner needs from the component, some low-level knowledge about its performance (call format, parameters, etc.), and some high-level knowledge about its pedagogical offer (tutorial strategies, options, user options, etc.). Also, this communication can be established through two different interfaces: from teacher's interface for the creation of courses, and from student's interface for the execution of instruction session.

A web-based tutoring system can be constructed by developing specific components, which are also called *general-purpose components*. Currently, MEDEA uses the following components: HERMES, which is an authoring tool to create web-

based electronic books; SIGUE [17], which is a system that allows the construction of web courses by collecting the references to existing web pages; and SIETTE [18], which is an adaptive test-based assessment system.

These components work in MEDEA as plug-ins. They are invoked from the MEDEA core. The authoring tools of these components are linked to the development tools of MEDEA; the administration tools are linked to the administration tools of MEDEA, and the course presentation is linked to the student interface.

8 Conclusions

As a consequence of the increasing importance of the distance education, and the advance of this field due to the new information technologies, many researchers have realized the necessity of applying intelligent techniques of existing educational systems to the Web.

MEDEA is a proposal of an open architecture for Web ITS development. It is an open system that contains the traditional modules of an ITS architecture. It has been designed to allow the integration and reutilization of educational tools, and teaching material already developed. The main idea is that any teacher could develop a course, reusing other material and software. After a tutorial session in a component, if this returns any kind of feedback information, it will have to implement the protocol defined. The difficulty degree of this integration depends on the information required and provided by the component. Some other attempts have been made to integrate preexisting web-based adaptive systems (see for instance [19]).

Other advantage of MEDEA is that this is a web based system, and therefore the requirements to be accessed are minimum: only a web navigator tool is required. Also, even though this system can recommend the student which is the best step to accomplish, the final decision is taken by him. At last, its inference machine is upgradable, since planners with new instructional strategies can be added.

MEDEA is still under development. Although some courses have been added, the current instructional planner is only a first attempt, and must be tested. Besides, although a first version of the communication protocol between the core and the library of components is being used, a new enhanced version is being implemented.

References

1. Eklund, J. & Brusilovsky, P. The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence. In *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia at Ninth ACM International Hypertext Conference, Hypertext'98*. (1998).
2. Chandrasekaran, B. Generic Task in Knowledge Based Reasoning: High level building blocks for expert systems design. *IEEE Expert*, 1, (1986) 23–29. Re-printed in Buchanan, B.G. y Wilkins, D. C. (eds.) *Reading in Knowledge acquisition and learning*. Morgan Kaufmann. San Mateo. CA. (1993) 170–177
3. Newell, A. The Knowledge level. *Artificial Intelligence*. (1982); 18.

4. Wielinga, B.J., Schreiber, A.T. & Breuker, J.A. KADS: a modeling approach to knowledge engineering. *Knowledge Acquisition* 4, (1992).
5. Musen, M. A. Automated Support for Building and Extending Expert Models. *Machine Learning*. (1989); 4:347–375.
6. Cuena, J. & Molina, M. KSM: An Environment for Knowledge Oriented Design of Applications Using Structured Knowledge Architectures. *Applications and impacts. IFIP'94*; (1994).
7. Self, J. Open Sesame?: Fifteen Variations on the Theme of Openness in Learning Environments. Keynote speaker at AI-ED'99. Abstract published in Lajoie, S. & Vivet, M. (eds.) *Artificial Intelligent in Education*: IOS Press (1999).
8. Murray, T. A Model for Distributes Curriculum on the World Wide Web. In *Journal of Interactive Media in Education*. (1998),5
9. Murray, T. Authoring Intelligent Tutoring Systems: an analysis of the state of the art. In *International Journal of Artificial Intelligence in Education*. (1999), 10, 98–129
10. Anderson, J. R. The Expert Module. In Polson, M.C. & Richardson, J.J. (eds.) *Foundations of Intelligent Tutoring Sytems*. Lawrence Elrbaum; (1988).
11. Vassileva, J. Dynamic Course Generation on the WWW. In Brusilovsky, P., Nakabayashi, K. & Ritter, S. (eds.) *Proceedings of the Workshop 'Intelligent Educational Systems on the World Wide Web' at AI-ED'97*. (1997).
12. Murray, T. Authoring Knowledge-Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. In *Journal of Learning Sciences*. V. 7, N.1, pp. 5–64
13. Arruarte, A., Fernández-de-Castro, I., Ferrero, B. & Greer, J. (1997), The IRIS shell: How to build ITSs from pedagogical and design requisites. In *International Journal of Artificial Intelligence in Education*. V. 8, N.3–4, pp. 341–381
14. Weber, G. & Specht, M. User modeling and adaptive navigation support in WWW-based tutoring systems. In Jameson, A., Paris, C. & Tasso, C. (eds.) *User Modeling*, Springer-Verlag, Wien (1997) 289–300
15. Woo, C. W. *Instructional Planning in an Intelligent Tutoring System: combining global lesson plans with local discourse control*. Illinois Institute of Technology, PhD. Thesis. (1991).
16. Woolf, B. & McDonald, D. Context-Dependent Transition in Tutoring Discourse. In *Proceedings of AAAI*. (1987).
17. Carmona, C., Bueno, D., Guzmán, E. & Conejo, R.: SIGUE: Making Web Courses Adaptive In De Bra, P., Brusilovsky P. & Conejo, R. (eds.) *Proceedings of the AH2002*, Springer-Verlag LNCS 2347 (2002).
18. Conejo, R., Guzmán, E., Millán, E., Pérez-de-la-Cruz, J. L. & Trella, M. SIETTE: A web-based tool for adaptive testing. In *International Journal of Artificial Intelligence in Education*. (to appear).
19. Brusilovsky, P., Ritter, S. & Schwarz, E. Distributed intelligent tutoring on the Web. In Brusilovsky, P., Nakabayashi, K. & Ritter, S. (eds.) *Proceedings of the Workshop 'Intelligent Educational Systems on the World Wide Web' at AI-ED'97*, (1997).