World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:10, 2008

Computational Intelligence Techniques and Agents' Technology in E-learning Environments

Konstantinos C. Giotopoulos, Christos E. Alexakos, Grigorios N. Beligiannis and Spiridon D. Likothanassis

Abstract**—**In this contribution a newly developed e-learning environment is presented, which incorporates Intelligent Agents and Computational Intelligence Techniques. The new e-learning environment is constituted by three parts, the *E-learning platform Front-End*, the *Student Questioner Reasoning and the Student Model Agent.*

These parts are distributed geographically in dispersed computer servers, with main focus on the design and development of these subsystems through the use of new and emerging technologies. These parts are interconnected in an interoperable way, using web services for the integration of the subsystems, in order to enhance the user modelling procedure and achieve the goals of the learning process.

Keywords**—**Computational Intelligence, E-learning Environments, Intelligent Agents, User Modelling, Bayesian Networks.

I. Introduction

A. What is E-learning

E-LEARNING refers to learning that is delivered or enabled via electronic technology. It encompasses learning delivered via a range of technologies such as the internet, television, videotape, intelligent tutoring systems, and computer-based training. E-Learning is a subset of two large worlds, specifically, "information technology" and "education and training". It can be valuable when used as a part of a well-planned and properly supported education and training environment. However, e-learning does not replace or render obsolete existing pedagogical theories and approaches. Many learning and technology professionals believe that e-learning will take its place when we will stop referring to it using a separate name and regard it as an integral part of a complete learning environment.

B. Technology Based Learning

Technology-based learning is seen as an alternative to the traditional classroom model of learning. Rather than attending traditional classes, students receive their learning through a computer, at a time and place that is most convenient for their schedule. Much of the justification for implementing technology-based learning is based on reducing employee time away from the job, reducing travel expenses, and shortening the amount of time students spend in their learning experience. A major weakness of technology-based learning is that little consideration is given to delivering effective instruction with the medium. Most technology-based learning is little more than an electronic form of instruction guides. Etienne Wenger [28] describes technology based learning as "computer-based training programs that walk students through individualized sessions covering reams of information and drill practice." Another significant weakness of technology-based learning is its failure to provide effective social transactions for learners. Brown and Duguid [29] describe learning as "a remarkably social process. Social groups provide the resources for their members to learn." In a traditional classroom, students are able to interact with each other and their instructor is able to socially construct their knowledge. In technology-based learning, this social aspect of learning is significantly reduced or is completely unavailable to students. The learning interaction is a one-on-one relationship between the student and the instructional content. [3]

C. Blended Learning

Blended learning is a term that has been widely accepted as a very important key-factor in the field of learning. The basic idea of blended learning is to apply and use several learning mediums in the learning process, which is usually a combination of instructor-led learning, with tools supported by web services. Using blended learning in the e-learning process is not new on the field, especially in the higher education. There is a very good example of the Greek Open University [13] that has introduced not only an e-learning platform that enables the distant learning process, but also

We thank the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program HERAKLITOS, for funding the above work.

K. C. Giotopoulos is with the Department of Computer Engineering & Informatics, University of Patras, Rio 26500, Patras, Greece (corresponding author, phone: +306973740389; fax: +302610997706; e-mail: kgiotop@ceid.upatras.gr).

C. E. Alexakos is with the Department of Computer Engineering & Informatics, University of Patras, (email: alexakos@ceid.upatras.gr)

G. N. Beligiannis is with the Department of Computer Engineering & Informatics, University of Patras, (email: beligian@ceid.upatras.gr)

S. D. Likothanassis is with the Department of Computer Engineering & Informatics, University of Patras and with the Research Academic Computer Technology Institute (R.A.C.T.I.), Riga Feraiou 61, 26221 Patras, Greece (email: likothan@cti.gr)

traditional means such as periodical lectures, group meetings, books and exams. These means are used and considered by students as far preferable to online technology that supports their frequent and day to day activities. It is more than clear that e-learning does not exclude and isolate existing educational methods and technologies. On the contrary, it is a useful tool that complements them when it is used appropriately.

D. E-Learning Systems

The technological progress and development, recent advances and the high speed Internet access have dramatically increased the use of collaborative environments and distributed learning technologies. The result of this expansion has lead to production of a wide range of new products related to this specific subject. Also, several new companies have entered to the learning technology market. The huge expansion of this market [2] has introduced several scientific challenges in the aforementioned field and several scientific groups and leaders of technological achievements have focused on learning technologies. In today's economy status, business challenges include speed of change, quality, customer retention, revenue growth and cost reduction. Originally, marketing hype claimed that e-learning would meet these critical needs, and more. Yet a few years on, expectations have not been met. Research shows that up to 60% of e-learning implementations failed, despite improvements in technology and content development and increased knowledge and awareness. Quite rightly organisations are carefully re-considering the role of e-learning and the true value that it can deliver. [5] The results of the current technological and market trends have lead to the production of new categories of products that encompass new capabilities and other products that combine existing functionalities into new and emerging product configurations. It is a great challenge to identify the interrelation of these systems and how they can complete an e-learning environment. Thus, the expansion of e-learning does not mean that existing market products and applications are obsolete. The reuse of existing technologies, components and functionalities is a main controversial issue of great scientific research. It seems that integration of different systems is the most important issue that will lead the technological interest in the future.

E. E-learning Process

The e-learning process can be divided into two major phases: content development, and content delivery and maintenance (Fig. 1). A typical e-learning process has planning, design, development, evaluation, delivery, and maintenance stages. The e-learning process is iterative in nature. Although *evaluation* is a separate stage of the e-learning process, ongoing formative evaluation for improvement (i.e., revision) should always be embedded within each stage of the e-learning process. Individuals involved in various stages of the e-learning process should be in contact with each other on a regular basis and revise materials whenever needed.

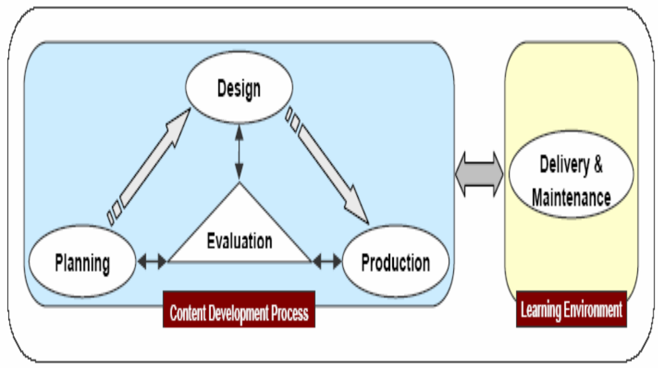

Fig. 1 The iterative process of e-learning

Based on the size and scope of the project, the number of individuals involved in various stages of an e-learning project may vary. *Some roles and responsibilities may overlap*, as many e-learning tasks are interrelated and interdependent. A large-sized e-learning project requires the involvement of various individuals. In a small or medium-sized e-learning project, some individuals will be able to perform multiple roles. When an e-learning course is completely designed, developed, taught, and managed by a single individual, it is clear that the same individual has performed the role of content expert, instructional designer, programmer, graphic artist, project manager, etc. In all cases, there are specific roles that must be included in an e-learning process in order to be a complete one. The supported roles in a typical e-learning environment are: learner, teacher, and administrator. The aforementioned types of roles cover the main necessities of the e-learning process.

F. E-learning Functional Model

In order to proceed in a better understanding of the integration of different systems that constitute an e-learning platform, it is very useful to identify a functional model of an e-learning application environment. In Fig. 2 there is a visual representation of the disperse components that provide the basis of a complete e-learning application environment and the several learning objects that are moved and communicated between the components [1]. It is important to emphasise the fact that Fig. 2 must not be treated as an architecture reference model but rather as a conceptual model that can be used to identify the e-learning systems and their functionalities.

II. System Description

The aforementioned chapters have clearly identified main issues for the concept of an e-learning system and the state-of-the-art concerning the functionalities of an e-learning environment. In the system proposed in this paper, there is an initiative to introduce AI (Artificial Intelligence) techniques and technologies in order to enhance the performance and the effectiveness of the e-learning system. AI is a rather new concept in the field of e-learning and it is widely known that constitutes the basis for the emerging and future trends of the architecture of IT applications in several fields. It is important to identify the innovative features that the proposed system is trying to introduce and how the specific features provide an added value to existing solutions. It is important to provide an introduction to the e-learning system that is under development and in the following chapters to provide the necessary details of the architecture and the development challenges that were faced. The basis of this specific work in-progress had been the idea to design and develop an autonomous e-learning application that will integrate AI techniques.

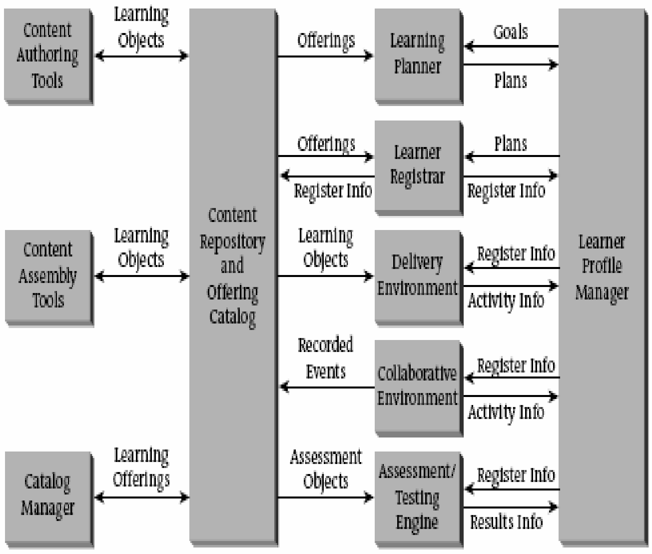

Fig. 2 E-learning functional model

The most important issue in the existing e-learning environments is to provide a customised solution to the learner's needs in order to attract his interest. The solution that we come up was to try to introduce personalisation features in order to facilitate the learner activities within the environment. The first dilemma was to decide about the technologies that will be used for the proper integration of AI techniques. This first step leads us to the decision that a web-based environment will provide the appropriate infrastructure to support the personalisation features of the e-learning platform. This enabled the development of a very sophisticated platform that introduces the AI concept without major difficulties. In order to provide a better understanding of the e-learning platform it is important to identify and clarify the exact services that are provided to the end user. This is achieved by the combination of knowledge routes concept and software agent technology leading to the establishment of a learner-centred learning system which:

- is based on open architectures and standards
- is cost-effective, flexible and adaptable
- incorporates common functional, re-usable, interoperable and platform-independent components and building blocks
- allows adaptive learning by defining the learner's prior knowledge status, and by determining the nature, the quantity and the level of lessons/training material to be imported to the system so that a certain learning goal will be achieved.

The specific approach is centred on the extension of current educational content interchange formats (e.g. the Content Packaging Specification currently under extension by international e-learning bodies such as the IEEE LTSC, CEN/ISSS LTW, IMS and ADL SCORM) to facilitate the definition of adaptive and conditional navigation rules taking into account user characteristics (user profiles, testing sessions, etc). These navigation rules will be described together with the learning assets within a single content packaging format. The aim of such a notation is to enable instructional designers to describe (in a common, reusable, interoperable and machine readable way) navigational logics which define how knowledge packages (i.e. educational courses, sets of learning assets, etc) can be disaggregated and presented, in a different way, according to different learner characteristics. As a result, e-learning applications and services can provide different knowledge routes to each individual learner, according to his/her characteristics and needs. The specific format will be used as the interchange format between the specific architectural modules developed, which address the needs of all envisaged players of the e-learning arena, namely learners and educational content and/or applications and services providers. The main objectives of the proposed system are to provide the necessary means to the e-learning users to achieve the specific learning goals and complete the e-learning process taking into account the specific and personal needs and interests of the learner.

This is the reason for the implementation of a new specific business model that is compliant to the learning needs and goals of the system. The actors of the new Business Model are:

- *front-end users*, i.e. learners accessing adaptive educational content (mainly through the internet and the world wide web) for personalised learning
- *back-end users*, including: (i) Knowledge providers, willing to publish learning material/assets in a common re-usable and interchangeable format (ii) Knowledge brokers mediator (e-learning) services, willing to provide value-added matching between learner requests and content providers, and (iii) Knowledge deliverers, e.g. educational service providers (ESPs), willing to provide end users (learners) with personalised learning services

The key elements of the system's underlying business model are depicted in Fig. 3.

The new business model highlights a process with three main roles and two main control flows, while delivering learning material to learners by means of effective and efficient learning experiences:

- *roles*: educational content publishers in the broad sense (content authors, owners, providers and publishers,

instructional designers, etc), delivering educational content to end users (learners) by means of intermediators (brokers, e-learning service providers, etc), seeking to provide value-added services, such as user tracking, monitoring and assessment, content selection, re-packaging and standards compliance testing
- *flows:* control rules and models governing the delivery of educational content from content providers to learners (content downstream) where the objective is content affordability, re-usability, interoperability and interchangeability, and, on the other hand, from learners to content providers (content upstream) usually demanding increased levels of accessibility and personalisation.

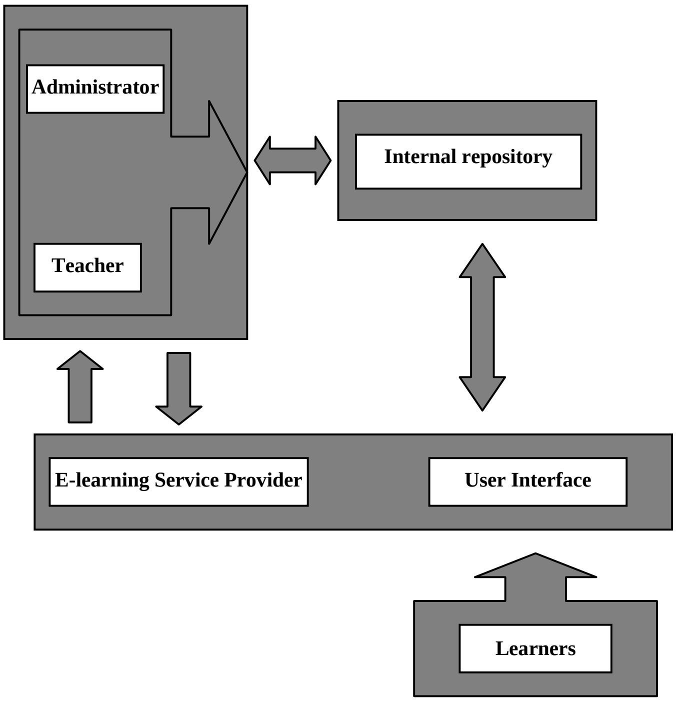

Fig. 3 E- learning business model

The most important aspects of the aforementioned business model are efficiency, personalisation features and user modelling issues. The overall issue is to use AI techniques in order to provide the necessary infrastructure to achieve the specific targets. The basic idea is to integrate intelligent agent technologies together with Bayesian networks in order to provide (to the end user) a personalised guidance within the e-learning process and a necessary tool to model the knowledge of the learner based on interactions between the learner and the system. Special focus is being given to the fact that knowledge is dynamic and continuously updated. The basis for the support of the aforementioned observation is to provide a necessary infrastructure that will initialise the model of the learner and will also map the interaction to a specific model tree based on the interactions of the learner with the system during a specific course. The teacher will support this procedure by all means. First of all, the teacher will establish the basic models for the knowledge and will also provide the initial rules for the decision making policy of the agent based on the input of the Bayesian network. The models that will be continuously communicated from the Bayesian network to the agent and vice versa will be updated and each model will be associated to a specific learner. This means that the profile of a learner will be just an XML file with a specific structure that will support all the appropriate decisions both of the system and the teacher. Thus, it will support the continuous effort of the agent to cover the interests of the learner with the search of all available contents (courses, sources, services, etc.) from the web.

It is more than clear that this specific effort and the proposed system have basically focused on two main issues:
1. integration of Intelligent Agents Technologies with Neural Networks
2. provision of appropriate infrastructure for the support of the student modelling procedure and the learning process.

III. System Architecture

The enforcement of the presented newly developed system requires a specific architecture that makes it possible to support the modelling features of the e-learning platform in a flexible and interoperable way. The system's architecture is constituted by the following three modules:
1. *E-learning platform Front-End, which is the main e-learning platform of the whole system supporting all the known training features. In addition, it provides the necessary mechanism to tutors in order to create weight-based questioners.*
2. *Student Questioner Reasoning, which is the module that elaborates the answers of each student and provides the possibility of answering correctly the following questions of the questioner.*
3. *Student Model Agent, which builds student models according to the performance and the interests of each student. Also, takes as input information that is estimated by the Student Questioner Reasoning or the E-learning platform Front-End and updates the stored model of the student providing the necessary input in order to guide the student to the next steps of the learning process.*

A. E-learning platform front end

The E-learning platform Front-End consists of four subsystems:
- e-Learning Content Presentation Subsystem
- Tutor Content Management Subsystem
- General Content Management Subsystem
- Database Subsystem

The e-Learning Content Presentation Subsystem access the Database System in order to obtain the necessary e-learning content and make it available to the users – students and teachers – of the e-learning system. The user-interface is a web-based application that provides the necessary information to users utilizing dynamically created HTML pages, which are easily accessed from a simple web browser. The e-Learning Content Presentation Subsystem supports all the basic e-learning features such as courses, lectures notes and question-based tests.

The Tutor Content Management Subsystem provides the appropriate tools to tutors in order to manage their content in the e-Learning Platform. The tutor can import, edit and delete the courses which he teaches as well as the associated learning content. A specific tool allows the tutor to create questioners for student evaluation purposes assigning weights, proportional to the importance of each question, in order to initialize the construction of the aimed student model.

The General Content Management Subsystem is used by the administrator of the e-learning platform and provides all the necessary tools for managing the content and the registered user accounts. The administrator is responsible to ensure the regular operation of the e-Learning Front-End module by checking the integrity of the e-learning content and managing the roles of the users in the system.

The Database Subsystem is the repository of all data used by the e-learning system. This data consists of the content and the associated information of the courses, the weight-based evaluation questioners and the users' profiles. The e-Learning Content Presentation Subsystem, the Tutor Content Management Subsystem and the General Content Management Subsystem access the Database Subsystem in order to retrieve or store data, as depicted in Fig. 4.

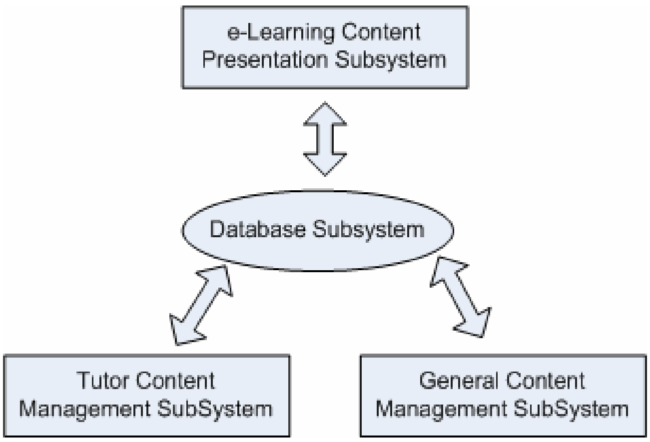

Fig. 4 E-learning platform front-end

B. *Student Questioner Reasoning*

The *Student* Questioner Reasoning aims to assist the student modelling process based on the answers in the evaluation questioners. The input to this module is an XML file containing the answers to each particular questioner given by students and the weights (probability of the student to answer correct in the given question) given by the teacher. Using these weights, a Bayesian Network is created where leafs present the questions. According to the answers of each student, the student modelling algorithm is activated in order to balance the network. The result of the algorithm's execution is the possibility of each student to answer correct or wrong the following questions of the questioner.

C. *Student Model Agent*

The *Student Model Agent* is designed in order to take as input information from other systems in order to update a particular student model and take decisions about the next step of the learning process of a student. It provides a scalable implementation that makes it possible to add different student modelling and decision makings routines in order to update the student model in different stages of the learning process. These routines can support a variety of Computational Intelligence techniques that can be applied in different cases according to the information that are sending to the agent. These routines are implemented in executable format as Java packages and can be executed dynamically from the *Student Model Agent*. The architecture of the *Student Model Agent* consists of four modules, as depicted in fig. 5: the Routine Decision Module, the Routine Execution Module, the Routine Repository and the Student Model Repository.

In the Routine Repository there are all the available routines for updating the student model and for decision making in executable format. The Routine Repository provides a storage unit for the composed student models. In the execution time of the system the Routine Decision Module is responsible to receive requests and input data from the other systems and to decide witch routine must be executed by the Student Model Agent. The Routine Execution Module is informed for the appropriate routine and retrieves it from the Routine Repository in order to initialize its execution.

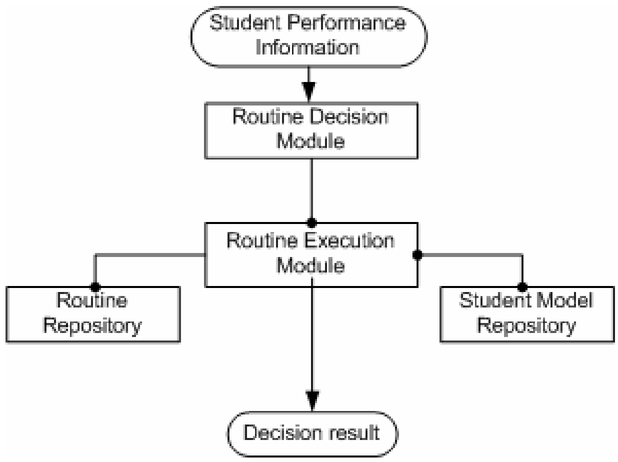

Fig. 5 Student Model Agent Implementation

In the presented system the *Student Model Agent* takes as input the probability that is estimated by the *Student Questioner Reasoning* and builds the model of each student. It also provides the necessary input to provide a possible decision about the next steps of the questionnaire. The input to the student model agent is an XML file that incorporates all necessary data in order to be able to lead the next steps of the learner. Based on the answers of the questionnaire the system leads the learner to the learning process.

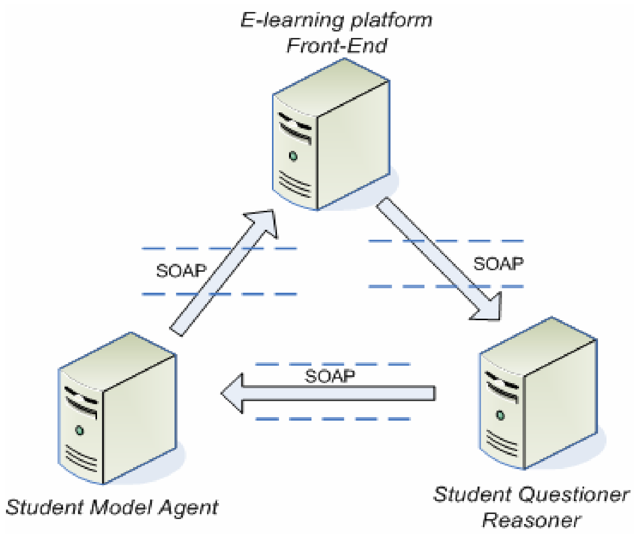

Fig. 6 E-learning system architecture

D. System modules interconnection

The system modules are three applications that are distributed in different computer systems connected over the internet network. Also, these applications are implemented using dissimilar technologies which complicate the issue of exchanging information and data between these modules. In order to accomplish the interconnection of the three applications, the system architecture is based on the Web Services technology and the protocols that implement it. The data is transferred in XML format according to the SOAP protocol [10] over HTTP compatible networks such as the WWW. When a student, who is connected to the E-learning platform Front-End, is answering a question, the corresponding data is sent to the Student Questioner Reasoning in a SOAP message. After the completion of the answers and the balance of the corresponding Bayesian Network the output is fed to the Student Model Agent through another SOAP message. The Student Model Agent , after identified the required modelling routine, reconstructs the student model and updates with a third SOAP message the E-learning platform Front-End. It is possible the routine that is executed in the Student Model Agent to require further information for the reconstruction of the student model. In that case the Student Model Agent using the Routine Execution Module requests, with a SOAP message, the necessary information from the E-learning platform Front-End. The system architecture is depicted in Fig. 6.

IV. User Modelling Issues

The problems uncovered in empirical evaluations of open learning environments indicate that additional support is needed to make the environments beneficial for all users. One way of providing this support is to supply each student with feedback on the exploration process throughout the interaction. Since the sense of control and freedom that open learning environments provide to the user has the potential to be very beneficial for learning, it is important to interrupt to provide this feedback only when warranted. Augmenting these environments with a student model is fundamental to determining when and how sensible feedback should be provided. User modelling is the process of building data structures and inference mechanisms that allow an application to assess certain properties of its user and tailor the interaction accordingly. Examples of relevant properties include i) behaviour patterns, ii) cognitive states such as knowledge, preferences and goals, iii) non-cognitive states such as emotions and personality traits. Equipped with this information, the application can adapt its behaviour to meet the needs of the user in an informed manner. Student modelling is sometimes thought of as a sub-problem of the user modelling problem [14], where the target application is an ITS. In a way, however, student modelling is a more difficult problem since no assumptions can be made concerning the user's knowledge level, which is constantly changing. Student models are used in ITSs for a number of purposes, including deciding when and how to advance the student through the curriculum, offer advice (both solicited and unsolicited), generate problems and activities, and provide tailored explanations [15].

A. Types of assessment

So far, the majority of student modelling work has focused on assessment during problem solving. One such assessment is the student's knowledge of concepts in the domain (e.g., [16]). A second common form of assessment is of the quality of a student's solution steps (e.g., [17]) which can also be used to assess the student's mastery of the rules used to generate the solutions (e.g., [18]). There are many challenges involved with assessing this type of behaviour. One challenge is to determine if a mistake was due to a slip or to incomplete knowledge. On the other hand, when the student demonstrates some correct behaviour there is the possibility that he/she was just guessing [19]. Other challenges include recognizing the student's solution path (plan recognition) and the uncertainty involved in judging the quality of the solution. Recently there have been new initiatives in student modelling that have gone beyond knowledge assessment. One of these initiatives has been to model meta-cognitive skills, such as the student model for effective example studying in [20]. Modelling meta-cognitive skills brings a new set of challenges since assessing this type of skill requires access to information that cannot be found in the student's final answer or the steps that lead to that final answer. In addition to knowledge, this type of assessment also requires information on factors such as the student's focus of attention and his/her reasoning process.

B. Student modelling using Bayesian networks

Each type of student model assessment requires different kinds of information about the student. The less of this pertinent information that the model is able to obtain directly through the student's interaction with the system, the more uncertainty there is in the modelling process. The issue of the amount and quality of information about the student that is available to the model is referred to as the bandwidth issue [15]. When trying to assess a student's knowledge during problem solving, a high bandwidth situation would mean that the model would be able to view not only the steps of the student's solution, but also the mental processes that he/she used to generate these steps. This has the lowest degree of uncertainty since the model would be able to directly use the correctness of the student's solutions along with the domain principles he/she used to generate his/her solutions to determine which set of concepts and rules he/she understands. In a low bandwidth situation, the model might only be able to view the student's final answer, requiring the model to infer the student's understanding of the concepts involved in the problem from very limited information. High bandwidth is more difficult to come by when modelling a skill such as self-explanation. Rarely would the model be provided with direct information on factors such as the amount of time the student spends reasoning about different parts of the material and the quality of this reasoning process. Thus, the bandwidth is often increased by either designing a restricted interface or by asking the student enough questions. For instance, in Conati and VanLehn's [30] assessment of self-explanation, they provided a restricted interface to obtain information on what portion of the interface the student was focusing his/her attention. This data was obtained by masking regions of interest and forcing the students to click on the masks to read the underlying material. The bandwidth issue creates special challenges for student modelling in open learning environments. As is the case with effective self-explanation, assessing effective exploratory behaviour requires modelling factors that are not easily observable. With open learning environments, however, designing a restricted interface or disrupting the user in any way would remove some of the freedom and control that has the potential to be very beneficial.

the potential to be very beneficial.

C. Student modelling using Bayesian networks

Bayesian Networks [21] are directed acyclic graphs whose nodes represent random variables and whose arcs represent direct probabilistic dependencies among variables. Each random variable has a set of values that it can take on, such as True or False for binary random variables. Each node in the network with predecessors has an associated Conditional Probability Table (CPT), while nodes without predecessors have prior probabilities. The arcs in the networks define the dependencies amongst variables in the network, rendering nodes either dependent or independent given evidence (see [22] for a good explanation of independence in Bayesian Networks). The network can be queried at any time to obtain the belief that a given node has a specified value. New evidence can be introduced into the network by setting the values of one or more of the variables in the network to a specific observed value. In any low-bandwidth situation, interpreting user characteristics such as domain knowledge and meta-cognitive skills based on limited observations of student behaviour involves a great deal of uncertainty. Bayesian networks provide a sound way of modelling and processing this uncertainty, making them very suitable for student modelling. Apart from the ability to formalize the uncertainty in the modelling task, the inferences provided by Bayesian Networks are well suited for student model assessment since the value of every node can be computed given whatever evidence is available. This includes being able to predict effects given causes (i.e., predict students' actions given information on the students' features) and vice versa.

D. Examples of Bayesian student models

A number of other ITSs have used Bayesian student models, including [23], [20], [19], [24], [25] and [26]. Much of their use so far, however, has been for knowledge assessment, such as in [24], [25] and [26]. The student model in HYDRIVE [25], an environment in which students learn how to trouble-shoot an aircraft hydraulics system, assesses the students' knowledge of aircraft components and the strategies they use to these components. Mayo and Mitrovic [24] use Bayesian Networks to assess students' knowledge of SQL, information that is used to tailor the tutor's curriculum. Finally, the student model in OLAE [19] assesses the student's knowledge of the laws and rules of physics. The student model in Andes [23], the successor to OLAE, also performs plan recognition since it can determine which solution path the student was following. Recently, there have been some interesting applications of Bayesian Networks that have extended beyond knowledge assessment. These applications include extending the student model in Andes to assess how effectively the student is self-explaining [20] and the work done in the I-Help project, which employs inspectable Bayesian Networks in a distributed setting [26]. The Student Model in ACE provides another form of innovative assessment in that it assesses the effectiveness of the student's exploration.

In this contribution we use the Bayesian network approach presented in [27] in order to model effectively each student and his/her specific characteristics. The use of this Bayesian network approach, for simulating the student model, seems to have been a complete success. The design allowed elegant, precise representation of sensible interpretation policies; it did not increase the knowledge management task, and it did not slow the system down too much. Moreover, empirical evaluations of the resulting coaches indicated that students learned more with them than with conventional instruction.

E. The approach used by the proposed system

The proposed system allows students to pursue different correct solutions during problem solving. As a result, one of the main issues that the student model of the proposed system needs to address is the assignment of credit for both plan recognition and assessment. That is, if a student action belongs to different solutions, which solution and corresponding student knowledge should be credited for it? The proposed student model handles this issue through a technique based on Bayesian networks. The technique was pioneered by the OLAE system for off-line assessment [31], and was extended to on-line assessment and plan recognition by the POLA student modeler, but only in prototype form [32], [33].

In order to illustrate the basic technique, which is fundamental to the operation of the student model of the proposed system, consider a simple case where there are just two explanations for a particular student action. Suppose each explanation involves just one rule, and the preconditions of both rules are satisfied directly by the state immediately prior to the student's action. This would be represented with the Bayesian network shown in Fig. 7. The two parent nodes represent the rules, and the third node represents the student's action. The two parent nodes represent binary variables: true means the student has mastered the rule, and false means the student has not. Each rule node has a prior probability of mastery, which is the system's best estimate of the student's competence prior to seeing this action. The action node is also binary, and its value represents whether the action either has or has not occurred. The action node has a conditional probability table that says, in essence, if either parent rule is mastered, then the action will probably be executed; if both parent rules are non-mastered, then the action will probably not be executed. When the action is observed to occur, the action node is clamped to its "occurred" value, and the network is updated. This causes the two parent nodes to acquire a marginal posterior probability, which represents the system's new best guess about the student's competence. If one node has a much higher prior probability than the other then, by the Bayesian update rule, it will get most of the credit for explaining the student's action in that its posterior probability will increase more than the other node's.

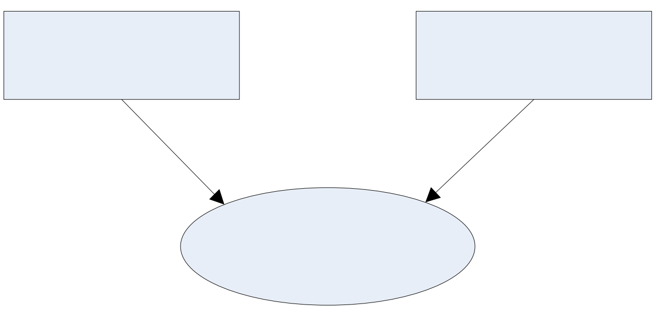

Fig. 7 A simple example of a Bayesian network for assignment of credit

This method of assigning credit is mathematically sound. Of course, whether it accurately predicts student competence depends on both the cognitive model network topology and on the prior and conditional probabilities in the nodes.

A Bayesian network can also handle the dual problem, assignment of blame. For instance, suppose that two rules must both be applied in order to produce a certain action. If the student cannot do the action, then the student is probably unable to use one or both rules. This would also be modeled by the simple network of Fig. 6, except that the conditional probability table for the action node would be different: if both parents are mastered, then the action will probably be executed; if either is non-mastered, then the action will probably not be executed. If the student is unable to perform the action, then the action node's value is clamped to "non-occurred" and the network is updated. This causes the posterior probabilities of mastery on both rules to be less than their prior probabilities. Because the student must be ignorant of at least one of the rules, the network has "blamed" both of them for the non-occurrence of the action. If one rule has a much lower prior probability of mastery, then it is more likely to be the missing rule, so it gets more of the "blame". This is a mathematically sound technique for solving the assignment of blame problem.

Although these illustrations have emphasized determining probabilities of mastery, which is the assessment part of the student modelling task, the same basic approach can be used for plan recognition. One represents possible student goals as parent nodes of observable actions, and the network will assign posterior probabilities to them that indicate the likelihood that those goals (intentions) underlie student actions.

Most importantly, such Bayesian networks cause assessment and plan recognition to interact seamlessly. Goals from explanations that involve probably mastered rules receive more credit for observed actions (and less blame for non-observed actions) than goals from explanations that involve probably unmastered rules. The adoption of this approach for the student model of the proposed system has been motivated by its mathematical soundness and elegance.

However, the proposed system is a real-world tutoring application, and thus we had to solve many technical problems to adopt the described approach for its student model. One was simply scaling the technology up. Many actions may be involved in solving a problem, and each action can require many rule applications to explain it. This makes the Bayesian network representing the problem solution quite large. Moreover, since the same rule can be used many times in solving a problem, each network can be highly interconnected. The size and topology of the networks could make them difficult to update in real time.

A second scaling problem is simply managing all the networks representing the many different exercises covered by the proposed system, and their relationships. Since the proposed system is designed in order to support over a 1000 problems in it, it is not feasible to create each problem's network by hand. Moreover, the networks are all related because they all share the same rule nodes. Yet, it is clearly infeasible to use one monolithic network that spans all actions in all problems. Moreover, if even a small change in the knowledge representation is required, it can affect the networks of many problems.

In addition to these problems of scale, realistic student modelling requires interpreting much more than just problem solving actions made by students. In particular, the Bayesian networks supported by the proposed system must deal with the following issues:

1. *Context specificity:* knowledge is sometimes acquired first in a more specific form then generalized, thus making near transfer (i.e., transfer to very similar application contexts) easier to obtain than far transfer (i.e., transfer to dissimilar application contexts) [34]. A major question is how to track the generality of competence.
2. *Mutually exclusive strategies:* some problems have multiple, mutually exclusive solution strategies. Should evidence that the student is following one strategy be interpreted as evidence that they are not following the other strategy?
3. *Guessing:* some actions are easier to guess correctly than other actions. A major question is how the assignment of credit should reflect this issue.
4. *Errors:* should errors of omission (missing actions) be interpreted the same way as errors of commission (incorrect actions)? What is really the evidence that a student does not know how to do an action?
5. *Old evidence:* how should evidence from earlier problems affect the interpretation of evidence from the current problem? If students learn a rule, then old evidence that they were ignorant of the rule should be ignored. Should forgetting also be explicitly modeled?
6. *Reading latency:* when students study examples, they pause longer as they read some lines than others, and this may be evidence of self-explanation. How can we properly interpret the latency of reading times?
7. *Hints:* when the student has received hints before entering a correct action, how much credit should the goals and rules that explain that action receive?
8. *Self-explaining ahead:* some students prefer to self-explain solution steps before reading them in the example, while others prefer to read steps then self-explain them [35]. A major question is how such preferences should affect the interpretation of reading latencies.

Although Bayesian networks were adopted in order to handle the assignment of credit and blame problems, they simplified handling the problems listed above as well. By using Bayesian networks, we were able to express sensible policies for handling those issues. We do not know if the individual policies are correct – that would require testing each one in isolation – but the policies did function well enough as a group.

V. Description of Technologies

In this section we present the main technologies that are used for the implementation of the new e-learning system. The three modules of the e-learning platforms are implemented separately as different applications. The E-learning platform Front-End is a web-based application implemented using the PHP [6]. The server side also includes technology for the computational parts and the MySQL [7] database for storing the data. The Student Questioner Reasoning and Student Model Agent are applications which demand a significant amount of computational power. In order to ensure the efficient running of these applications, they have been implemented as Java classes. Especially for the Student Questioner Reasoning the JavaBayes [8] is used for constructing and balancing the Bayesian Network.

For the interconnection of the three modules the Web Services [9] technology is utilized. Web Services are referred as "software applications identified by a Uniform Resource Identifier (URI), whose interfaces and binding are capable of being defined, described and discovered by XML artefacts and support direct interactions with other software applications using XML-based messages via Internet-based protocols". Web services are loosely coupled, communicating through XML based documents. The key enabler for Web Services technology is XML and the initiatives that form its foundations are SOAP, WSDL, and UDDI.

Simple Object Access Protocol (SOAP) [10] is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It is not tied to any particular transport protocol, although HTTP is the most popular. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.

The Web Services Description Language (WSDL) [11] provides a model and an XML format for describing Web services. WSDL enables separating the description of the abstract functionality offered by a service from the concrete details of this service description, such as "how" and "where" that functionality is offered. WSDL describes a Web service in two fundamental stages: an abstract one and a concrete one. According to the abstract description, a Web Service is given in terms of the messages it sends and receives. At the concrete level, a binding specifies transport and wire format details for one or more interfaces. An endpoint associates a network address with a binding. Finally, a service groups together endpoints that implement a common interface.

The next piece of the puzzle is UDDI (Universal Description Discovery & Integration) [12], which supports both design-time and run-time discovery of Web Services.

The last piece of the puzzle is the *JavaBayes* system that is a set of tools for the creation and manipulation of Bayesian networks. The system is composed of a graphical editor, a core inference engine and a set of parsers. The graphical editor allows you to create and modify Bayesian networks in a friendly interface. The parsers allow you to import Bayesian networks in a variety of formats. The engine is responsible for manipulating the data structures that represent Bayesian networks [8].

VI. Conclusion

In this contribution we have described a newly developed e-learning system that integrates Computational Intelligence Techniques and more specifically Bayesian Networks with the use of Intelligent Agents. The produced outcome of the specific effort is a distributed web-based e-learning platform that has being developed through the use of new and emerging technologies. Special focus was given in the integration of the three subsystems through the use of web services in an interoperable way.

Future work will be focused on the introduction of newly developed evolutionary algorithms within the e-learning environment in order to enhance the training functionalities of the system. Thus, special effort will be given in order to enhance the e-learning environment and introduce special self-adaptive features to the system.

References

[1] Eduworks Corporation, 2002. Available: http://www.eduworks.com
[2] J. S. Brown, "Growing up digital: How the web changes work, education, and the ways people learn", *Change Magazine*, pp. 11-20, March/April 2000.
[3] A. Richardson, "An ecology of learning and the role of e-learning in the learning environment, a discussion paper", Sun Microsystems 2002.
[4] B. H. Khan, "People, process and product continuum in e-learning: The e-learning P3 model", *Educational Technology*. vol. 44, no. 5. pp. 33-40, September-October 2004.
[5] L. Overton (2004), "Linking learning to business", *Bizmedia*. Available: http://www.elearningage.co.uk
[6] PHP: Hypertext Preprocessor, http://www.php.net
[7] MySQL Network, http://www.mysql.org
[8] http://www-2.cs.cmu.edu/~javabayes/Home/
[9] W3C - Web Services Activity. Available: http://www.w3.org/2002/ws/
[10] M. Gudgin, M. Hadley, N. Mendelsohn, J. J. Moreau, H. F. Nielsen, "Simple Object Access Protocol (SOAP) 1.2. W3C", 2003. Available: http://www.w3.org/tr/soap
[11] R. Chinnici, M. Gudgin, J. J. Moreau, J. Schlimmer, S. Weerawarana, "Web Services Description Language (WSDL) 2.0. W3C", 2003. Available: http://www.w3.org/tr/wsdl
[12] K. Januszewski, E. Mooney, R. Harrah, S. Lee, J. Munter, C. von Riegen. UDDI ver, 3.0. UDDI.org, 202, http://uddi.org
[13] Greek Open University. Available: http://www.eap.gr
[14] A. Jameson, "Numerical uncertainty management in user and student modelling: An overview of systems and issues", *User Modelling and User-Adapted Interaction*, vol. 5, no. 3-4, pp. 193-251, 1996.
[15] K. VanLehn, "Student modelling", in *Foundations of Intelligent Tutoring Systems*, M. C. Polson and J. J. Richardson, Eds, Erlbaum, Hillsdale, NJ, 1988, pp. 55-78.
[16] J. A. Collins, J. E. Greer, and S.X. Huang, "Adaptive assessment using granularity hierarchies and Bayesian nets", *in Proceedings of the Third International Conference on Intelligent Tutoring Systems (ITS'96)*, G. Gauthier C. Frasson and A. Lesgold, Eds, Berlin, Springer, pp. 569-577, 1996.
[17] J. R. Anderson and R. Pelletier, "A development system for model-tracing tutors", *in Proceedings of the International Conference on the Learning Sciences*, L. Birnbaum, Ed, Association for the Advancement of Computing in Education, 1991, pp. 1-8.
[18] J. R. Anderson, "The expert module", *in Foundations of Intelligent Tutoring Systems*, M. C. Polson and J. J. Richardson, Eds, Erlbaum, Hillsdale, NJ, 1988, pp. 21-53.
[19] J. Martin and K. VanLehn, "Student assessment using bayesian nets", *International Journal of Human-Computer Studies*, vol. 42, pp. 575-591, 1995.
[20] C. Conati and K. VanLehn, "Toward computer-based support of meta-cognitive skills: a computational framework to coach self-explanation", *International Journal of Artificial Intelligent in Education*, vol. 11, 2000.
[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, Los Altos, CA, 1988.
[22] E. Charniak., "Bayesian networks without tears", *AI Magazine*, vol. 12 no. 4, pp. 50-63, 1991.
[23] C. Conati, A. S. Gertner, K. VanLehn, and M. J. Druzdel, "On-line student modelling for coached problem solving using Bayesian networks", *in Proceedings of the Sixth International Conference on User Modelling*, A. Jameson, C. Paris, and C. Tasso, Eds, Vienna, New York, Springer, 1997, pp. 231-242.
[24] M. Mayo and A. Mitrovic, "Using a probabilistic student model to control problem difficulty", *in Proceedings of the Fifth International Conference on Intelligent Tutoring Systems (ITS 2000)*, G. Gauthier, C. Frasson, and K. VanLehn, Eds, Berlin, Springer, 2000, pages 524-533.
[25] R. J. Mislevy and D.H. Gitomer, "The role of probability-based inference in an intelligent tutoring system", *User Modelling and User-Adapted Interaction*, vol. 5, no. 3-4, pp. 253-282, 1996.
[26] J. D. Zapata-Rivera and J. E. Greer, "SModel server: Student modelling in distributed multi-agent tutoring systems", *in Proceedings of AIED 2001, 10th International Conference on Artificial Intelligence in Education*, J. D. Moore, C. L. Redfield, and W. L. Johnson, Eds, San Antonio, TX, 2001, pp. 446-455.
[27] Cristina Conati, Abigail Gertner and Kurt Van Lehn, "Using Bayesian Networks to Manage Uncertainty in Student Modelling", *User Modelling and User-Adapted Interaction*, vol. 12, pp. 371-417, Kluwer Academic Publishers, Printed in the Netherlands, 2002.
[28] Etienne Wenger, "Communities of practice learning as a social system", *The Systems Thinker*, vol. 9, no. 5, 1998.
[29] J. S. Brown, P. Duguid, *The social life of information*, Harvard Business School Press (Boston), 2000