

Integrating E-learning Environments with Computational Intelligence Assessment Agents

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

Abstract—In this contribution an innovative platform is being presented that integrates intelligent agents in legacy e-learning environments. It introduces the design and development of a scalable and interoperable integration platform supporting various assessment agents for e-learning environments. The agents are implemented in order to provide intelligent assessment services to computational intelligent techniques such as Bayesian Networks and Genetic Algorithms. The utilization of new and emerging technologies like web services allows integrating the provided services to any web based legacy e-learning environment.

Keywords—Bayesian Networks, Computational Intelligence techniques, E-learning legacy systems, Service Oriented Integration, Intelligent Agents

I. INTRODUCTION

THE wide adoption of e-learning environments in all levels of human education has led scientific research in the field of adaptive and intelligent e-learning systems in order to provide higher quality services towards the end users of the e-learning systems.

Adaptive and intelligent e-learning systems is more than clear that satisfy the demanding need of the e-learning users for personalization. It is widely known that the most efficient products are tailored made to the needs of the client. The same applies in the e-learning market. People are eager to buy

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.



C. E. Alexakos is with the Pattern Recognition Laboratory, Department of Computer Engineering & Informatics, University of Patras, Rio 26500, Patras, Greece (corresponding author); phone: 0030-2610-997520; fax: 0030-2610-969001; e-mail: alexakos@ceid.upatras.gr).

K. C. Giotopoulos is with the Pattern Recognition Laboratory, Department of Computer Engineering & Informatics, University of Patras (e-mail: kgiotop@ceid.upatras.gr).

E. J. Thermogianni is with the Pattern Recognition Laboratory, Department of Computer Engineering & Informatics, University of Patras (e-mail: thermog@ceid.upatras.gr).

G. N. Beligiannis is with the Pattern Recognition Laboratory, Department of Computer Engineering & Informatics, University of Patras (e-mail: beligian@ceid.upatras.gr).

S. D. Likothanassis is with the Pattern Recognition Laboratory, Department of Computer Engineering & Informatics, University of Patras (e-mail: likothan@ceid.upatras.gr).

products or services that fit exactly to their personal needs and interests.

That is the main reason for the huge scientific effort in all market sectors to introduce features in products or services that satisfy the need for personalization. The same applies to the new and emerging sector of e-learning.

E-learning systems need to introduce the aforementioned features in terms of functionalities in order to meet the market trends and user requirements. This aspect comprises the key aspect for success in all e-learning environments.

Our proposal introduces a scalable and interoperable integration platform supporting various assessment agents for e-learning environments. These agents are implemented in order to provide intelligent assessment services based on computational intelligence techniques such as Bayesian Networks and Genetic Algorithms. The utilization of new technologies such as web services allows to integrate the provided services to any web based legacy e-learning environment. A first approach to the integration of computational intelligence techniques in order to assist e-learning process has been presented in [1].

The paper is structured as follows. Section II includes a brief presentation of the related work in the field of adoption of intelligent agents in e-learning systems. In section III the architecture and the functionalities of the proposed agent platform are described. Section IV presents an example of a query assessment agent based on Bayesian Networks, which is integrated in a legacy e-learning system. Finally, section V summarizes the conclusions and suggests future applications and extensions of the e-learning platform.

II. RELATED WORK

Utilization of agents is significant for providing intelligence in e-learning environments. Thus a lot of work has been done concerning these fields, mainly focusing on adoption of intelligent agents to integrate e-learning systems. Buraga [2] proposes an agent-oriented extensible framework based on XML family for building a hypermedia e-learning system available on the world-wide web. It deploys mobile agents that can exchange information in a flexible way via XML-based documents. The intelligent tutoring system is composed of four major components. The information processed by each component can be stored by XML documents. Some of the components have been implemented as intelligent agents.

Angehrn et al. [3] suggests the use of K-InCA to provide a personalized e-learning system. K-InCA is an agent-based system designed to assist people in adopting new behaviors. The agents within the system examine users' actions and maintain a "behavioral profile" reflecting the level of adoption of the desired behaviors. Based on the user profile, and relying on a model borrowed from change management theories, the agents provide at different stages customized guidance, including mentoring, motivation or stimulus, in order to support real learning and smooth adoption of new behaviors.

In order to recommend useful learning material to students Andronico et al. [4] proposed a multi-agent recommendation system that suggests educational resources to students into a mobile learning platform that supports mobile learning processes. They have first extended the Learning Management System (LMS) in order to incorporate mobile technologies, allowing users to interact with the systems using mobile devices like PDAs, cellular phones etc. This extension arises the problem of designing new learning models that will be able to adapt to the changes of student's performance during the learning process. Another extension of an LMS presented in that paper regards the integration of a multi-agent recommendation system, whose aim is to collect data about the users' behavior and preferences and then suggest educational resources to them according to their profile.

Zaiane [5] suggests the use of web mining techniques to build such an agent that could recommend on-line learning activities or shortcuts in a course web site based on learners' access history to improve course material navigation as well as assist the online learning process. The automatic recommendation system takes into account profiles of on-line learners, their access history and the collective navigation patterns, and uses simple data mining techniques.

According to the service-oriented integration for e-learning systems based on web services, Pankratius [6] provide a related architecture and describes the extensions to support software agents. It focuses on using intelligent software agents for the distributed retrieval of educational content. In this architecture intelligent software agents can be used to acquire user specifications of learning content and search for matching Learning Objects on behalf of the user. The work utilizes the web services as a wrapper around Learning Objects.

III. AGENT PLATFORM

The contribution presented introduces a scalable implementation architecture that is based on an agent platform. This platform is used in order to manage the execution of the various intelligent agents for supporting legacy e-learning systems. In this section, a detailed description of the functionalities and the implementation of the agent platform architecture, is presented. Furthermore, the web services technology is utilized in order to provide communication and interoperability between the proposed agent platform and e-learning legacy systems.

A. Platform Architecture

The Agent Platform Architecture that this contribution proposes is based in open and interoperable standards, since main effort has been given in the reusability perspective of the specific platform. The main idea behind the specific architecture is to use a multi-agent platform to design and develop the features (agents) that will provide the added value to legacy e-learning systems. In order to follow the international standards in the implementation of multi-agent systems, the proposed agent platform is developed utilizing Java Agent Development Framework (JADE) [7]. JADE is a java-based platform that provides the basic mechanism for the implementation of peer-to-peer agent based applications according to the FIPA [8] guidelines for the development of multi-agent systems.

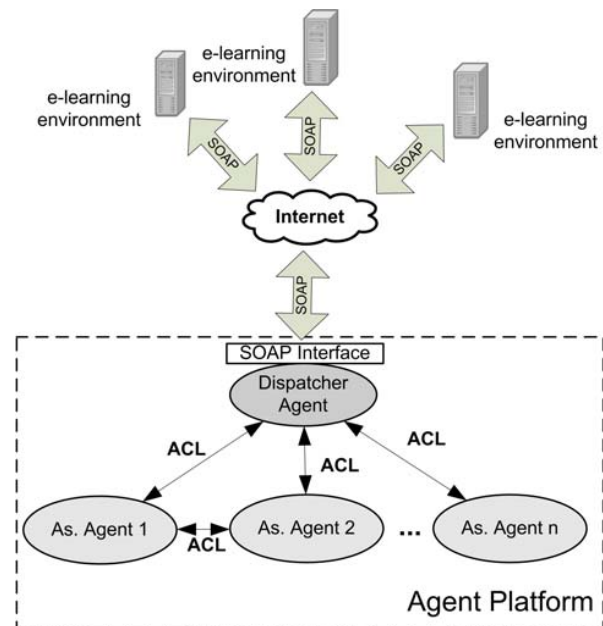


Fig. 1 Agent Platform Architecture

The main concept is to exploit the features of the proposed agent platform and develop agents that will implement the features that are needed. The basic aspect for the architecture is the fact that the agent platform architecture (Fig. 1) communicates with the legacy e-learning system through the use of web services and more specifically by using the SOAP protocol. The requests coming from the e-learning system are being processed by the Dispatcher Agent. This specific agent is the front end of the agent platform and is responsible for identifying the requests coming from the e-learning system and for allocating them to the appropriate agent in order to be processed. Each agent is designed to implement functionalities that arise from the user requirements.

The communication within the agent platform is being realized with the ACL (Agent Communication Language) [9] that is part of the FIPA template. It is important to emphasize that the Dispatcher Agent is the most important component of the agent platform. The reason for this is its multiple behavior

and usefulness. It serves as process identifier, process distributor and communication handler. Thus, it is a critical agent for the proper operation of the agent platform.

The Dispatcher Agent is "informed" by the agents with the functionalities that they provide. Its main responsibility is to recognize the incoming requests from the e-learning systems through a SOAP Interface and translate them to ACL messages in order to reroute them to the corresponding agent.

The utilization of the JADE framework allows affective communication between different agents.. It is possible that an agent will require the execution of functionalities of another agent. In that case ACL messages are exchanged between the involved agents establishing interoperability features between different agents.

B. Communication with the E-learning Environment

For the interconnection of the agent platform with the legacy e-learning systems Web Services [10] 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 artifacts 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. According to the above 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.

According to the prospects that are supported by the web services, the main agent platform is enhanced with a SOAP server in order to provide a flexible and standard based service for accessing the agents' capabilities and functions. A client script is also installed in each of the integrated systems in order to invoke the provided web services from agent platform.

The utilization of web services is based on the capability of "easy" integration that is achieved through an intermediate adapter layer that relays commands and data between the web and the system. A lot of implementation has been introduced according to the exposition of systems capabilities and services to the internet, such as .NET framework and Common Object Model (COM) for Microsoft based applications, Enterprise JavaBeans (EJB) for java based applications and PHP classes for web based applications.

IV. QUESTIONER ASSESSMENT AGENT

The use case example presented in this contribution shows the utilization of the proposed platform for assessment of a student in a questioner-based examination process. In order to manage the student's answers given to the questioner, a Questioner Assessment Agent is implemented. This agent uses Bayesian Networks in order to determine the sequence of questions that are given to the student according to his/her

possible answers.

A. Bayesian Network for Questioner Optimization

The agent is based on Bayesian Networks' techniques in order to manage the questioners of an e-learning system. Bayesian Networks [11], [12] are compact networks of probabilities that capture the probabilistic relationship between variables, as well as historical information about their relationships. They are very effective in modelling cases where some information is already known and incoming data is uncertain or partially unavailable [13]-[15]. These networks also offer consistent semantics for representing causes and effects (and likelihoods) via an intuitive graphical representation.

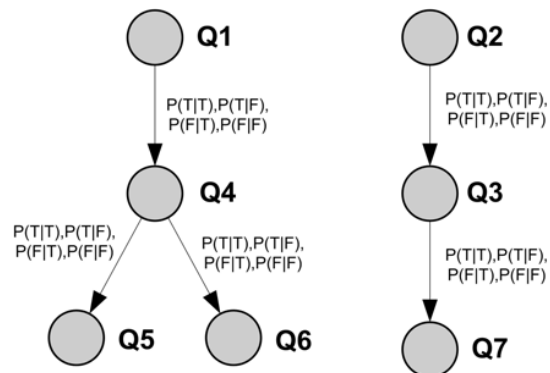


Fig. 2 Graphical Representation of a Questioner

According to the presented approach the questions are structured in a decision tree that denotes the relevance between two successive questions. A graphical representation of a questioner used to explain the functionality of our module is depicted in Fig.2.

Each of the questions in the Bayesian network is represented by a node. Each node can be in several states. Each state corresponds to a different set of probable values for each variable of the node. In our case, for example, each question is represented by a 2-state variable that can be either true (correct answer) or false (false answer). Nodes are connected to show causality with an arrow (edge) indicating the direction of influence.

The Bayesian network also contains probabilistic relationships among some of the states of the domain. These relationships are used to answer questions like the following: If the student answered correctly the fourth question, was it more likely to have answered correctly the first question or not?

The probability of any node in the Bayesian network being in one state or another without current evidence is described using a conditional probability table. Probabilities on some nodes are affected by the state of other nodes, depending on causality. Prior information about the relationships among nodes may indicate that the likelihood that a node is in a specific state is dependent on the specific state of another node. For example, prior information may show that if the

student answered correctly the first question, the likelihood of answering correctly the fourth question is higher. An example of the conditional probabilities for our Bayesian network is shown in the following table.

TABLE I
CONDITIONAL PROBABILITIES

Parent (Q1)	Child (Q4)	
	True	False
True	0.8	0.2
False	0.6	0.4

The table shows that the likelihood of giving a correct answer at the fourth question is 0.8 if a correct answer was given at the first question and 0.6 if a false answer was given at the first question respectively. Similarly the likelihood of giving a false answer at the fourth question is 0.2 if a correct answer was given at the first question and 0.4 if a false answer was given at the first question respectively. In cases where a node does not have a parent, the table has only two values that show the likelihood of giving a true or a false answer (Table II).

TABLE II
NODE WITH NO PARENT

Q1	true	False
	0.5	0.5

After storing all the essential history information stored in the conditional probability tables, Bayesian networks can be used either to help making decisions or as a way to automate a decision-making process. Someone can use Bayesian networks to perform inductive reasoning (diagnosing a cause, given an effect) and deductive reasoning (predicting an effect, given a cause). The operation of Bayesian networks is based on a well known mathematical rule, the Bayes' rule. Most simply, Bayes' rule can be expressed as follows [16]:

$$P(\omega_i / \mathbf{x}) = \frac{P(\mathbf{x} / \omega_i) \cdot P(\omega_i)}{p(\mathbf{x})}$$

where ω_i is a state of nature, \mathbf{x} is the vector of the monitored characteristics, $P(\omega_i)$ is the a priori probability, $P(\mathbf{x}/\omega_i)$ is the probability density function (likelihood), $P(\omega_i/\mathbf{x})$ is the a posteriori probability and $p(\mathbf{x})$, which is known as marginal likelihood, equals $\sum_{j=1}^n P(x/\omega_j) \cdot P(\omega_j)$ (n is the number of different states of nature).

According to our implementation, an xml file is created which contains all the information needed for handling the questioners; such as the questions, the correct answers, a decision-threshold, the initial likelihood for each node and the values of the conditional probability table (for example 0.8, 0.6, 0.2, 0.4). The system retrieves the first question from the xml file and prompts it to the student. Based on the answer of each student and according to the Bayes' rule, the system

estimates the likelihood of being correct for the next consecutive answers (based on the respective Bayesian tree structure).

Questions that are considered to be answered correctly easily, according to a decision-threshold which is initially configured by the administrator, are by-passed and never prompted to the user of the e-learning system. The decision-threshold denotes the difficulty of a particular question. This is implemented by using a Black List in which all the questions to be skipped are deposited.

The benefit of this scheme is that the student does not have to spend time by answering questions that are considered to be far easy for his/her knowledge level.

B. Implementing Questioner Assessment Agent

In order to provide the questioner assessment functionalities according to the Bayesian network - based approach, introduced in this paper, a Questioner Assessment Agent is implemented. This agent is developed using the Java programming language and the JavaBayes [17] class library. This library comprises a set of tools for the creation and manipulation of Bayesian networks. The Questioner Assessment Agent functionalities are implemented in a java class that includes the following methods:

Initialize_Questioner(xml_file): This method has as input an xml file that is structured according to the XMLBIF xml schema and is used from the JavaBayes class in order to pass to the Bayesian network the probabilities of the questions and the decisions' thresholds that are assigned from the tutor to each specific questioner. Also, there are additional information such as the questioner id and the student id that are used by the agent in order to identify the requests.

Get_Questioner_Answer(student_id,questioner_id,query_id,answer): This method is used in order to execute the Bayes inference rule and calculate the next question that will appear to the student according to the answer that he/she gave to the previous question. .

Get_Questioner_Result(student_id,questioner_id): This method is executed when the students is finished with the questioner. It processes the wrong and correct answers and, by taking account the questions that do not appear to him/her, it calculates the result of the questioner exam for the specific student.

The next step of the implementation is to publish the functionalities of the Questioner Assessment Agent to the Dispatcher Agent. This works by adding references of the corresponding methods to the initialization function of the Dispatcher Agent.

The Questioner Assessment Agent is communicating directly with the Dispatcher Agent of the proposed agent platform through ACL messages. Using the Bayesian_Logic library the Query Assessment Agent is responsible to react to the requests coming from the Dispatcher Agent.

The main processes are:

1. Initialize the questioner for the user.

2. Identify the next question that the user has to answer.
3. Calculate and provide to the e-learning legacy system whether the user has passed the questioner or not.
4. Run a Genetic Algorithm that classifies the users (students) in categories (eg. good, average, bad) and update the respective user model

C. E-learning Platform Integration

Using the Eclipse Web Tools Platform [18] a SOAP wrapper is generated on the Dispatcher Agent in order to publish the functionalities of the Questioner Assessment Agent through the Internet infrastructure. The Web Service that is generated has three operations similar to the functionalities of the Questioner Assessment Agent:

- initialize_questioner().
- get_questioner_answer().
- get_questioner_result().

Additionally, the corresponding WSDL document is created, having the invocation and grounding description of the above operation. Both the WSDL document and the SOAP server code are deployed in an application server (in our case the Apache Tomcat has been used) in order to be accessed through the Internet.

The three mentioned operations can be easily invoked from a legacy e-learning system with a SOAP client which can be implemented in any of the well known programming languages. A legacy e-learning system can use these functionalities and interoperate with the Questioner Assessment Agent through the Internet by adding some lines of code. The legacy e-learning system used in our application was implemented using the PHP language. Some blocks of code were added to the e-learning system in order to allow the tutor to assign probabilities and threshold values to the questions of a questioner. Also, a SOAP client is adjusted to the routines implementing the exam for the communication with the Dispatcher Agent. Furthermore, the e-learning system is capable of retrieving the results of the agent's execution and use it in order to assess the examination process of a particular student.

V. CONCLUSION

The proposed agent platform provides an integrated approach towards achieving the utilization of various assessment agents in legacy e-learning environments. Our proposal enhances significantly the overall system in terms of flexibility and efficiency while it introduces a high degree of agents and e-learning platforms interoperability utilizing web services technology. The presented agent platform can support various intelligent agents that provide assessment services based on computational intelligence techniques such as

Bayesian Networks and Genetic Algorithms.

The specific contribution establishes the basis for a continuous scientific work in the field of e-learning systems and more specifically in intelligent and adaptive e-learning environments. The utilization of artificial and computational intelligence techniques can be extended and provide even more sophisticated intelligent services as components of e-learning systems.

More specifically, the utilization of a multi-agent platform with more sophisticated agents that extend the intelligent and adaptive services to the learners and tutors, is in progress. The use of Genetic Algorithms in order to identify more suited threshold values to the questions, a more sophisticated Genetic Algorithm for the updating procedure of the user model and, of course, designing agents that facilitate not only the process of the exam, but also the learning content, will be the main issues of our future work.

REFERENCES

- [1] K. C. Giotopoulos, C. E. Alexakos, G. N. Beligiannis and S. D. Likothanassis, "Computational Intelligence Techniques and Agents' Technology in E-learning Environments", *International Journal of Information Technology*, Volume 2, Number 2, 2005, ISSN:1305-2403, pp. 147-156.
- [2] S. Buraga, "Developing Agent-Oriented E-Learning Systems", in *Proceedings of The 14th International Conference on Control Systems And Computer Science - vol. II*, I. Dumitrache and C. Buiu, Eds, Politehnica Press, Bucharest, 2003.
- [3] A. Angehrn, T. Nabeth, L. Razmerita, and C. Roda., "K-inca: Using artificial agents for helping people to learn new behaviors", in *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001)*, Madison USA, August 2001, pp. 225-226.
- [4] A. Andronico, A. Carbonaro, G. Casadei, L. Colazzo, A. Molinari, and M. Ronchetti, "Integrating a multi-agent recommendation system into a Mobile Learning Management System", in *Proceedings of Artificial Intelligence in Mobile System 2003 (AIMS2003)*, October 12, Seattle, USA.
- [5] O. R. Zaiane, "Building a Recommender Agent for e-Learning Systems", in *Proceedings of the International Conference on Computers in Education*, Auckland, New Zealand, December 2002, pp. 55-59.
- [6] V. Pankratius, O. Sandel, W. Stucky, "Retrieving Content with Agents in Web Service E-Learning Systems", Symposium on Professional Practice in AI, IFIP WG12.5 - in *Proceedings of the First IFIP Conference on Artificial Intelligence Applications and Innovations (IAAI)*, Toulouse, France, August 2004.
- [7] F. Bellifemine, G. Caire, A. Poggi and G. Rimassa, "JADE A White Paper", *Telecom Italia EXP Magazine*, Volume 3, Number 3 September 2003
- [8] Foundation of Intelligent Physical Agents (FIPA), <http://www.fipa.org/>
- [9] FIPA ACL Message Structure Specification, FIPA Standard, <http://www.fipa.org/repository/aclspecs.html>
- [10] W3C - Web Services Activity <http://www.w3.org/2002/ws/>
- [11] F. V. Jensen, *An Introduction to Bayesian Networks*, Springer Verlag, New York, 1996.
- [12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.
- [13] J. Martin and K. VanLehn, "Student assessment using bayesian nets", *International Journal of Human-Computer Studies*, vol. 42, pp. 575-591, 1995.
- [14] 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.
- [15] C. Conati, A. S. Gertner, K. Van Lehn, 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.

- [16] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley-Interscience, 2nd edition, October 2000.
- [17] JavaBayes, Bayesian Networks in Java, <http://www.cs.cmu.edu/~javabayes/>
- [18] Eclipse Web Tools Platform (WTP), <http://www.eclipse.org/webtools/>