

WEB APPLICATION FOR AN ADAPTIVE MULTI-AGENT E-LEARNING SYSTEM: A CONTINUOUS IMPROVEMENT OF E-CONTENT

SALAH HAMMAMI HASSAN MATHKOUR

College of Computer and Information Sciences, King Saud University, Arabia Saudi

shammami@ksu.edu.sa mathkour@ksu.edu.sa

Received April 15, 2013
Revised November 18, 2013

This paper presents a methodology for the continuous improvement of e-content in an e-learning system based on agent technology. Multi-agent systems play an important role in today's software development of web application. For this purpose, we introduce here a novel concept involving the development of an adaptive e-Learning web application that explores several recent technologies and including web design. The Adaptive E-Learning system based on agent technology is useful in developing continuous improvement strategies. The goal this work is to propose a generic model to assess and evaluate the students learning outcomes related to e-content. Based on the analysis and evaluation of the students learning outcomes results, the instructor identifies the low achievers of the e-content and makes a correction plan to improve them. An exploratory implementation has been developed and used in practice. database.

Key words: e-assessment, continuous improvement, multi-agent system, web application.

Communicated by: M.D. Lytras & P.O. de Pablos

1 Introduction

To develop a continuous improvement process of an e-learning e-content, a need for a genuine move towards adopting and utilizing the current technology evolution in acquiring knowledge has become a necessity. Today, the learning practice has turned out to be heavily technology and Internet dependent shaping a new term known as electronic learning or e-Learning. E-learning must be enhanced towards a more user-centred, interactive and collaborative model of learning using web application. This requires interactions among different people or organizations with different goals, intentions and the potential for the emergence of conflicts [1]. The complexity of designing such e-Learning systems has been addressed in many studies. Marie et al. [2] have shown that only putting course contents on the e-learning systems, without (i) the use of modern information technologies such as the web technology to present the learning content, (ii) using appropriate pedagogical models and principles, and (iii) taking into account the communication between participants and instructors, is not enough to accomplish educational goals. Following this claim, new methods for ensuring the reliability and the quality of e-content in such systems are necessary. We introduce here a new approach to address these

issues. This approach is an adaptive e-learning system based on multi-agent technology which provides an easy way to communicate, collaborate, and coordinate actions and resolve conflicts and raises modelling the interactions among the agents and their behaviours as the main challenge facing designing of adaptive multi-agent architectures [4].

Multi-agent systems provide powerful resources to develop educational systems and as a support of lifelong learning [1]. We adopt such technologies to enhance e-learning systems in an intelligent manner that is robust and adaptive. The multi-agent methodology using web-based technologies is a new trend in the modeling and development of learning environments [4, 6, 7, 23]. Web-based learning architecture facilitates interactions between learners and authors [34]. Intelligent and reactive agents in an e-learning architecture enable researchers to obtain a personalized e-learning system that adapts to the goals and characteristics of each learner [8]. Modeling the interactions between the agents and their behaviours is a major challenge and is receiving more attention nowadays when designing multi-agent architectures. Interactions must be clearly identified, validated and correctly implemented to enable a reliable description of agent interactions [5, 9].

This paper shows how to build an adaptive e-learning web application that explores several recent technologies, including agent technology and web design which facilitate quick and easy access to information. The Adaptive E-Learning system based on agent technology helps us to develop and implement continuous improvement strategies to assess and evaluate the students learning outcomes related to every e-content, identify their low achievers and make a correction plan to improve them.

The remainder of this work is organized as follows: In Section 2, previous and related studies are presented. In Section 3, we present an overview of the adaptive e-learning system based on multi-agent system supporting the development of e-content continuous improvement process. In Section 4, we describe the design and the implementation of the E-Assessment-Multi-Agent System for e-content continuous improvement process. In Section 5, the technical and implementation issues will be described. Finally, in Section 7, we present our conclusion, discussion and provide recommendations for ongoing work.

2 Previous and Related Studies

The literature review for the paper was conducted for two main objectives. The first was to establish an overview of adaptive e-learning system based on multi-agent technology, while the second aim was to identify approaches for continuous improvement process of e-content.

For the first aim, the idea of learning systems that make use of multi-agents architectures has been introduced in previous research work. In the Baghera learning environment [27] a2-level architecture is implemented. The first level is composed of number of agents that provides the main functions of the educational system. The second level is composed of reactive agents that are in charge of analyzing students' knowledge. Educational decisions taken on the first level are based on input from the second level. Students interact with three artificial agents: Student's Personal Interface Agent, Tutor Agents, and Mediator Agent. Teacher interacts with two artificial agents: Teacher's Personal Interface Agent that provides the teacher with information about the whole learning environment, and Assistant Agent for the creation and distribution of new activities.

Brusilovsky [16] introduces a distributed architecture that makes use of reusable intelligent learning activities. The architecture includes two kinds of agents: a knowledge-inferring agent and an activity-inferring agent. The architecture assumes the presence of at least four kinds of servers: activity servers, value-adding services, learning portals, and student model servers. Gascueña [8] proposes 4-component architecture for an Intelligent Tutoring System (ITS). The components are: The Student Model, the Domain Model, the Pedagogical Model, and the Educational Model. The pedagogical model uses four agents: The Preferences Agent to manage the user-preferred style of presentation, the Accounting Agent to monitor the student interaction with the interface, the Exercises Agent to select exercises that will be presented to the student, and the Tests Agent to form the test questions.

Nabeth [17] presents the InCA system with an open agent architecture that integrates a set of interactive features on an attempt to allow personalized and adaptive course generation. It is composed of three components: The domain model, that contains a set of structured knowledge elements to be delivered to the user, the user model that covers elements such as level of knowledge sharing, learning goals, domains of interests, etc., and a set of expert agents that access the domain model and propose intervention strategies to be offered to the user. Maia [6] compares to architectural approaches for distributed learning management systems that make use of agents to represent various learning functions. This comparison takes into account the complexity and the notion of single-point failure. A virtual educational environment model that uses collaboration as a form of social interaction is discussed in [8]. The general architecture is composed of three levels (user, mediators, and provider) and for each level there are heterogenic families of human and software agents corresponding to it. Salvatore [18] proposes Profile-based recommender system that is an enhanced extension of the MASHA system (Multi-Agent System Handling Adaptivity). Agents in this system include: Device agent, Student agent, Tutor agents, and Teacher agent.

Safiye [1] presents an agent system whose components are teachers, students, and resources. The Petri net method has been used to model and review the Inter component relations. The principle of the architecture proposed for an adapted training service is to allow the adaptation of knowledge transmission from a teaching function managing a virtual group of learners by a communication system. The agents are: Student Agent, Teacher Agent, (iii) Course Agent, and Resource Agent. Existing courses can be reused and integrated with existing web courseware or educational materials. The processes and activities of the system are illustrated through a case study. Kannan [39] presents a framework that consists of two main parts that is multi agent based e-Learning System and an Ontology Model. There are six types of agents: (i) main agent that start up the ontology by invoking the Ontology Manager; (ii) ontology agent that activates the User Interface Agent and displays the interface to users; (iii) dispatcher agent, that pass the parameter or search criteria entered by the user to be sent over Searcher Manager at various platforms (iv) searcher agent, which searches the external sources such as files and, parse the retrieved result into XML, and notify the Main agent and send the result XML to it; (v) saver agent, that saves the XML into the internal database, for record keeping and future use and (vi) presenter agent, that presents the results.

The architecture discussed in [40] has four layers: (i) a database layer to store, share and reuse courses and teaching materials, (ii) an adaption layer which allow personalized courses generation, (iii) a presentation layer that arrange personalized courses into learning paths, and (iv) an interface layer to develop several learning interfaces (e.g. for use via web or mobile devices). The four layers are

detailed in [40], but the interactions among the layers are not fully exploited. MATHEMA [41] is a learning environment that is based on a multi-agent architecture. The architecture of MATHEMA consists of six components: (i) a motivator that represents human external entity for motivation purposes; (ii) a human learner; (iii) a Micro-society of Artificial Tutoring Agents (MARTA) for problem solving purposes; (iv) a Human Experts Society (HES) that acts as a knowledge used by MARTA; (v) an Interface Agent that is between the human learner and MARTA; and (vi) a Communication Agent that provides the interaction between MARTA and HES. In this research work, the communication and cooperation among tutoring agents are stressed. It is modelled using a Petri net based object oriented approach named G-Nets.

For the second aim, a comprehensive search for continuous improvement of e-learning e-content was conducted. Most of these articles presents assessment methods, but few of them who analyze these results in order to improve the e-learning e-content.

Hossein [41] describes an assessment process that measures student-learning outcomes and develop a model for a continuous improvement process of student skills and performance in courses and programs. The approach of continuous improvement is based on outcomes of both course and program level assessment. This allows instructor to evaluate curriculum and syllabi based on the results of the course VGLU rubric and to help him to improve their course outcomes performance and take adequate actions to improve the student learning outcomes in future tasks. Heinrich [42] presents survey to identify a range of assessment approaches using e-tools for specific steps in the assignment process. Crespo [43] develops framework for the integration of learning outcomes, assessment and units of learning as key concepts. It is an outcome based assessment where the assessment process must be aligned with the learning outcomes. The author proposes model which captures the influence of learning outcomes in the learning assessment process, and determines appropriate assessment methods and resources to be used. Raghu [44] presents an e-learning system uses an adaptive assessment algorithm followed by an evaluation and expert module to evaluate student performance and attitudes and enhancing the content and format to their special needs. Gloria [45] describes the use of an assessment approach based on fuzzy logic and computational theory of perceptions of learning process to generate an automation assessment report on the achieved skill level. This technique implements assessment criterion using inference systems based on linguistic rules. Nangkula [46] defines an approach to align the effectiveness of formative assessment with learning outcomes through Rasch measurement model in order to develop and build student's skills cumulatively during his undergraduate course. The Rasch measurement uses student's data assessment, for a given task and transformed them into logic scale which have equal interval, and analyzes it to measure variables such as abilities, attitudes, and personality traits. Kamsuriah [47] develops methodology based on the same Rasch model. It consists of three phases: planning, classification and results analysis. In the planning phase, the course learning outcome was identified and evaluated. The classification phase focuses on the pre-processing of the students' assessment results on each course learning outcome. Then data were transformed on grade rating of mark cluster and treated. The results from this study used to monitor the performance for each learning outcome and to improve its related course.

3 Web Application Architecture for Adaptive E-learning System

Web applications of multi-agent systems require a formal exchange of messages between agents and their synchronous parallel functionality. In the following sections, we propose architecture for e-learning system with adaptability and intelligence features and we define a structure for a blackboard agent to ensure communication between the agents [5]. The blackboards provide an easy way for agents to communicate, collaborate, coordinate their actions and resolve conflicts. There are three different design approaches for blackboards [11, 24], but in the present context of e-learning systems, several arguments motivate the use of the distributed blackboard approach. Therefore, the blackboard is used not only to ensure communication among agents but also to help the agents accomplish their tasks, share knowledge, request services, present information about their plans or decisions, and satisfy their goals. Based on the above considerations, the proposed architecture is shown in figure 1.

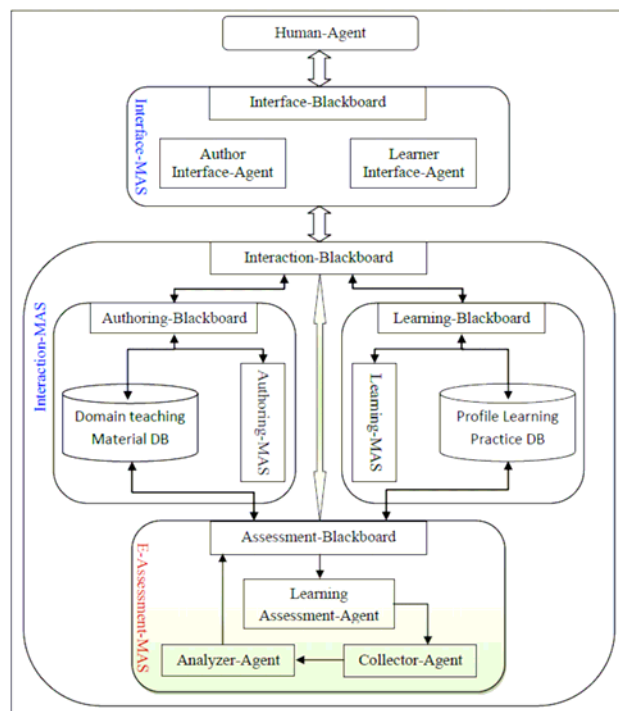


Figure 1 Architecture of Adaptive E-learning System.

This architecture uses distributed intelligent blackboards [4, 5], which facilitate the implementation of a shared address space on a distributed system. Our idea is to distribute the shared data (profile, e-content, ...) over the distributed blackboard and use a message passing sub-system, totally transparent to the different multi-agent sub-systems.

The proposed multi-level architecture (Figure 1) is made up by three main levels:

- *Human Agents* (Learner, Authors (instructor, tutor)) ;

- *Interface Multi-Agents System (MAS)* contains agents that acts with the human agent of the system;
- *Interaction-MAS*:
 - Interaction Blackboard Agent contains the agents that supervise interaction between different agents and how they manipulate the data;
 - Learning-MAS contains agents that responsible of learning process;
 - Authoring-MAS contains agents that responsible of the learning content;
 - E-Assessment-MAS: It is a MAS that manage the assessment process to measure and evaluate the students' learning outcomes (SLO) related to every e-content in order to determine how well the e-content objectives and SLO are being attained. It is a continuous quality improvement of the e-content.

All of the layers are organized hierarchically and are able to run concurrently and communicate through the distributed intelligent blackboard agent. We analyze the agents' behaviors using Object Petri Net [3, 33]. The proposed Object Petri net models make evident certain problems that can appear when agents have to simultaneously treat more goals [4, 7]. Object Petri Net enables the verification and validation of the specification and facilitates the transition to the implementation.

3.1 Global view of the Adaptive E-learning System behaviour

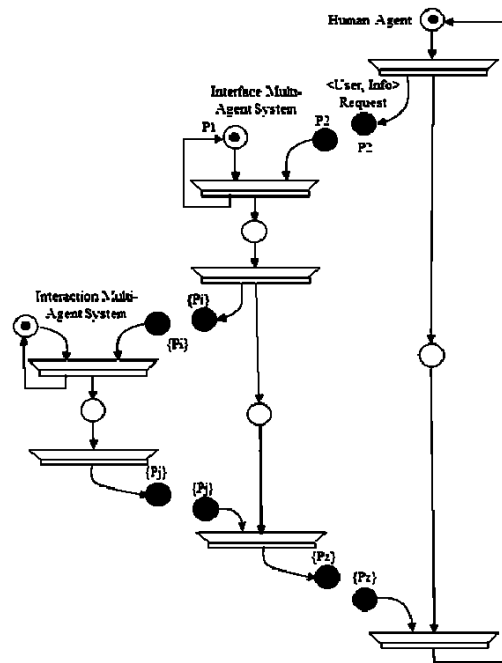


Figure 2 OPN model representing the global view of the e-learning system behavior.

Figure 2 provides a global view of the e-learning system behavior described by Object Petri Nets. It is composed of three sub-Object Petri Nets: Human Agent, Interface Multi-Agent System and Interaction

Multi-Agent System. Communications between these multi-agent systems are represented through black places as illustrated in the figure.

3.2 The Interface Multi-Agent System behaviour

The Interface MAS contains agents that provide functionality needed by the human agents to interact with the system. Figure 3 gives an idea about the behavior of the Interface MAS. The Interface MAS is composed of three types of agents: the Interface Blackboard Agent, the Learner Interface Agent, and the Author Interface Agent:

- The Interface Blackboard Agent starts with the agent receiving a request from the Human Agent (the request from the Human Agent is represented by the black place P2) and returns the result of the request to it (the response to the Human Agent is represented by the black place Pz). This agent is responsible for controlling the Human Agent requests and distributing them among the intended interface agents (the Learner Interface Agent or the Author Interface Agent) by notifying these agents about the arrival of new requests. The response originates from the interaction will be forwarded to the Human agent.

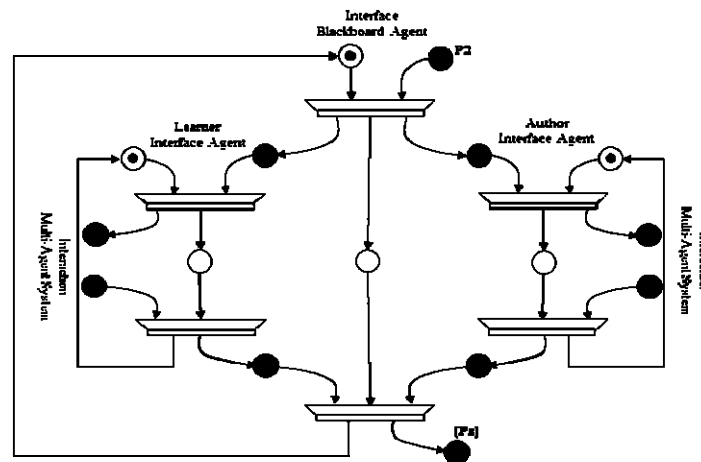


Figure 3 OPN of the Interface Blackboard Agent

- The Learner Interface Agent monitors the Learner's actions and gives access to system resources through the Interaction Multi-Agent System.
- The Author Interface Agent makes available to the author information about the whole learning process and mediates communication with the Interaction Multi-Agent System.

3.3 The Interface Multi-Agent System behaviour

The Interaction Blackboard Agent controls and mediates communication between the Interface-BA and Interaction-MAS. The main tasks of this agent are receiving requests, identifying and notifying the intended agents about requests, and notifying them about received results.

The Interaction-BA obtains the requests from the agents in the Interface-MAS and submits them to the suitable agents: Learning-MAS, Authoring-MAS, E-Assessment-MAS. Additionally, the Interaction Blackboard Agent receives the requests' responses from the Learning-MAS/Authoring-MAS/ E-Assessment-MAS and submits them to the various agents in the Interface-MAS. Figure 4 details the Object Petri Net, which represents the communication and the transfer of data between the Interface Blackboard Agent and the Interaction-MAS through the black places.

The details of the Interaction Blackboard Agent, Learning-MAS, and Authoring-MAS are developed in [5]. In the following section we present the design and the implementation of the E-Assessment-MAS.

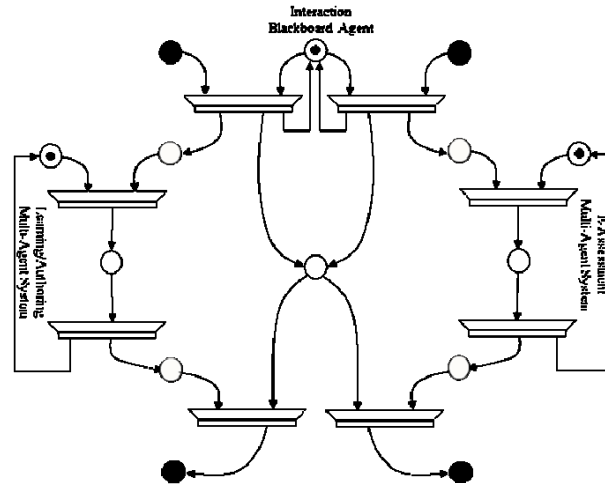


Figure 4 OPN of the Blackboard Agent.

4 E-Assessment-MAS for Continuous Improvement Process of e-content

The aim of this section is to show how the Adaptive E-Learning system based on Agent technology and OPN helps us to implement an E-Assessment Multi-Agent System to measure and evaluate the students' learning outcomes (SLO) related to every e-content in order to determine how well the e-content objectives and SLO are being attained: it is a continuous quality improvement e-content. Based on the analysis and evaluation of the SLO results, the instructor identify the low achievers of the e-content and make a correction plan to improve them.

4.1 Description of the SLO assessment process

The assessment of SLO is performed every semester based on different type of exams (assignments, midterms, final exam, project) taking different formats. They are posted through the AEL-AOPN at a given time and their responses must be posted before a predefined due date.

Data of the SLO assessment are collected in automated templates to facilitate the assessment process and makes it uniform for all e-content. These data provide an opportunity for instructor to effectively validate and align the e-content by identifying and building the course strengths, and

developing recommendations and strategies that can enhance SLO of low achievers the course activities. Figure 5 presents the SLO assessment process based on the following agents: *Learning Assessment Agent*, *Collector Agent* and *Analyzer Agent* which are described subsequently:

- In every e-content, the *Learning Assessment Agent* specifies which *Assessment Resources* (e.g. tests, test items, peer assessment forms) and *Assessment Methods* (e.g., multiple-choice test, peer assessment, learning contract, and oral examination) are to be used during the learning provision process to assess the achievement of intended student learning outcomes. The learning assessment process is expected to help the instructor to assess the achievements of the relevant SLO in his/her course. The assessment is preferably done close to the end of the semester.

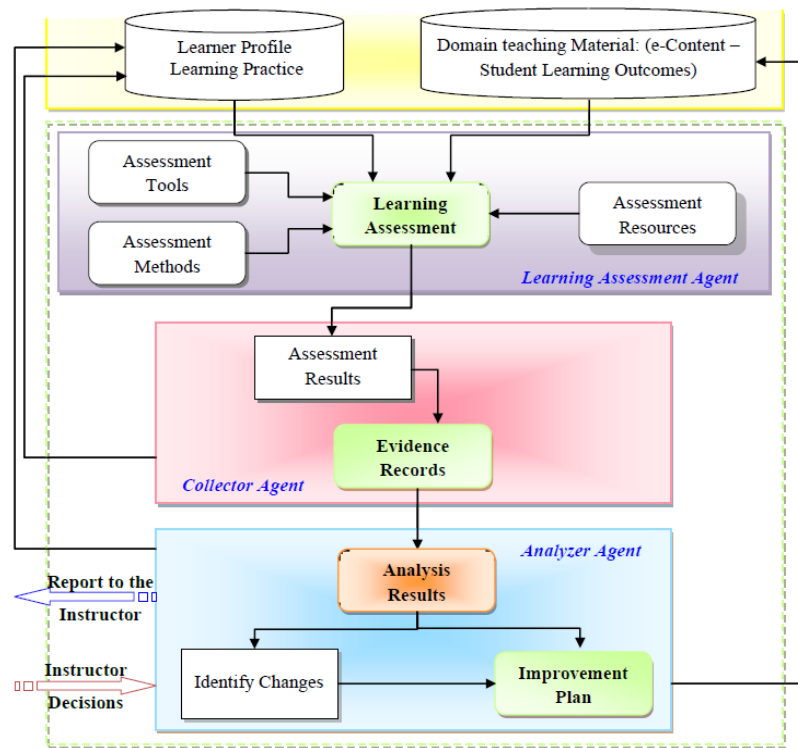


Figure 5 Student's Learning Outcomes Assessment Process.

- The *Collector Agent*, collects, checks, transforms, and validates data/grade to make sure it is consistent and usable. All these data transformed into *Evidence Records* to elaborate a report related to the SLO achievement in every e-content. A set of metrics are calculated (depending on the data) such as average score achieved in a specific SLO, as a percentage of students achieving the satisfactory-exemplary levels in a specific outcome, etc.
- The *Analyzer Agent* analyses and evaluates the assessments results. The attainment of objectives and outcomes is measured by comparing the real achievements with the targets for each objective and outcome. If achievements are higher or equal to targets, the objective or

outcome is considered attained. If the assessment reveals any weaknesses in a specific SLO, the *Analyzer Agent* elaborates a report to the instructor to identify the cause of that weakness. If achievements are lower than the targets, the objective or outcome is considered to be not attained. The instructor will identify the reasons behind the non-attainment of that objective or outcome and introduces adequate action plans that can improve learning outcomes of low achievers through the e-content tasks. Based on the analysis of the reasons behind the non-attainment of an objective or an outcome, an Improvement Plan is implemented in order to overcome the identified issues. The proposed corrective actions are implemented the following semester and their impact on the outcome achievement shall be assessed. Then, a new assessment cycle starts.

4.2 Methodology for the e-content continuous improvement

The methodology for the continuous improvement e-content is developed to assess student learning outcomes through the e-content and program based on the ABET/CAC program requirement [48]. For Computer Science program, the program enables students to achieve, by the time of graduation, *a* to *k* outcomes [48].

Each e-content target one or more student learning outcome. The e-content descriptions describe how it supports a subset of the student learning outcomes. When several student learning outcomes are targeted in one e-content, they can be targeted at different percentages. In every e-content description, a mapping table between the e-content learning outcomes and the student learning outcomes is included. This mapping table contains also the percentage of coverage of every student learning outcome in the e-content.

Therefore, for each e-content, we found in the courseware repository:

- learning objects and teaching methods,
- e-content learning outcomes definition,
- The ABET-related standards student learning outcomes (from *a* to *k*) [48] that are measured in the e-content,
- mapping of the e-content learning outcomes to student learning outcomes,
- assessment resources,
- assessment methods.

For example, upon the successful completion of the e-content of Software Engineering course (CSC342), a student should be able to achieve the following learning outcomes:

1. Apply key elements and common methods for elicitation and analysis to produce a set of software requirements;
2. Create and specify the software design using a software requirement specification, an accepted design methodology (e.g., structured or object-oriented), and appropriate design notation;
3. Use tools necessary for analyses and design activities;
4. Use a software testing strategy;

5. Work in team;
6. Make ethical professional decisions and practice ethical professional behavior.

The student learning outcomes (a through k) of the e-content of CSC342 are:

- b. an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution .
- c. an ability to design, implement and evaluate a computer-based system, process, component or program to meet desired needs.
- d. an ability to function effectively on teams to accomplish a common goal Students work on projects for both analysis and design.
- e. an understanding of professional, ethical, legal and social issues and responsibilities.
- i. an ability to use the current techniques, skills, and tools necessary for computing practice.
- k. an ability to apply design and development principles in the construction of software systems of varying complexity.

This mapping table 1 contains also the percentage of coverage of every student learning outcome in the e-content.

Table 1 Mapping of Course learning outcomes to student learning outcomes in CSC342.

Course Learning Outcomes	Student outcomes					
	b	c	d	e	i	k
	30%	10%	10%	10%	10%	30%
1	√					
2						√
3					√	
4		√				
5			√			
6				√		

The mapping between the course learning outcomes and the student learning outcomes with the percentage of coverage of every student learning outcome in the e-content helps the instructor to identify the assessment resources and assessment methods that should be used to assess a specific student learning outcome. The achievement of different student learning outcomes was measured by final exam, midterms, and project. For example, the evaluation, of the student learning outcomes of CSC342 Spring 2013, is made according the following table 2.

The evaluation of student learning outcome is considered to be attained; if the student's score (SS) demonstrates a very good or good understanding and performance through important concepts of the course materials. We consider the SLO attained if the SS is 70% or above in the corresponding SLO. There are four levels of satisfaction.

- **E:** Excellent- Demonstrates a very good understanding and performance through important concepts of the course materials. $SS \geq 90.00$; SS can be used for a grade of B+ and A.

- **G**: Good: Demonstrates a good understanding and performance through important concepts of the course materials: $70.0\% \leq SS < 90.0\%$; SS can be used for a grade of C+ and B.
- **W**: Week - Demonstrates that the student has mastered this SLO is provided, but it is weak or incomplete: $60.0\% \leq SS < 70.0\%$; X can be used for a grade of D+ and C.
- **P**: Poor - Demonstrates unsatisfactory: $SS < 60.0\%$; SS can be used for a grade of F and D.

Table 2 Mapping of Course learning outcomes to student learning outcomes in CSC342.

		b	c	d	e	i	k
Final Exam	Ex 1	√					√
	Ex 2				√		
	Ex 3					√	
	Ex 4						√
	Ex 5		√				
Midterm 2	Ex 1	√					
	Ex 2						√
	Ex 3					√	
Midterm 1	Ex 1						
	Ex 3	√					
	Ex 4				√		
Project	P1-Analysis	√					
	P2- Design						√
	P3-Team work			√			

For every course, the system provides course assessment reports where they report the assessment results. This report comprises two tables and a course assessment chart and includes all data relevant to the course. Table 3, includes grades related to each student learning outcome and the levels of satisfaction per student per specified student learning outcome in the CSC342 Spring 2013. At course level, the achievements in each of the specified student learning outcomes in the course is compiled and calculated as the percentage of student achieving a satisfactory level in the concerned outcome:

$$\text{Outcome Achievement} = \frac{(\text{number of E} + \text{number of G})}{(\text{number of E} + \text{number of G} + \text{number of F} + \text{number of P})}$$

Table 4 summarizes the result shown in table 3 and provides of the details of course's outcome achievement and provides an opportunity for instructor to effectively review, collect the necessary data and develop an adequate corrective action plan to improve student learning outcome achievement in the following semester.

Table 3 Students' performances in achieving a CSC342 course's outcomes - Spring 2013.

KSU- CCIS- Spring 2013 - Course Title: CSC 342 Software Engineering																																			
				SO (b)						SO (c)				SO (d)				SO (e)				SO (i)				SO (k)									
				25%	25%	25%	25%			100%			100%			55%	45%			55%	45%			25%	25%	25%	25%								
				M1	M2	P	F	%	Satisfaction	F	%	Satisfaction	Team Work	%	Satisfaction	M1	Q1	%	Satisfaction	M2	Q3	%	Satisfaction	M2	P	F	F	%	Satisfaction						
				3.75	1.5	8	10			10.0			4			3.75	2			5	4			8.5	8	10	10								
No	ID	Name	Status	1	1	1	1			1			1			1	1			1	1			1	1	1	1								
1	430104651			3.75	1.0	8.0	6.0	82	G	10.0	100	E	4.0	100	E	2.25	2.0	78	G	4.0	4.0	89	G	8.5	7.0	8.0	10.0	92	E						
2	430105018			1.25	1.0	8.0	6.0	65	F	7.0	70	G	4.0	100	E	0.50	0.0	7	P	4.0	3.0	78	G	6.0	7.0	8.0	8.0	80	G						
3	430106435			1.50	1.0	7.0	9.0	71	G	2.0	20	P	4.0	100	E	2.25	2.0	78	G	4.0	1.0	55	P	4.0	8.0	5.0	6.0	64	F						
4	430106627			0.00	0.0	0.0	0.0	0	P	0.0	0	P	0.0	0	P	0.00	0.0	0	P	0.0	0.0	0	P	0.0	0.0	0.0	0.0	0	P						
5	430106682			0.00	0.0	0.0	0.0	0	P	0.0	0	P	0.0	0	P	0.00	0.0	0	P	0.0	0.0	0	P	0.0	0.0	0.0	0.0	0	P						
6	430107030			3.75	1.0	7.0	4.0	74	G	5.0	50	P	4.0	100	E	3.50	2.0	96	E	4.5	4.0	95	E	6.0	8.0	6.0	10.0	83	G						
7	431100232			3.75	1.0	7.0	7.0	81	G	9.0	90	E	2.0	50	P	0.00	2.0	45	P	4.0	3.0	78	G	6.0	7.0	8.0	10.0	85	G						
8	431100655			2.75	0.5	7.0	6.0	64	F	8.0	80	G	3.0	75	G	2.00	2.0	74	G	5.0	4.0	100	E	5.0	7.0	6.0	6.0	67	F						
9	431100713			3.75	1.0	8.0	7.0	84	G	8.0	80	G	4.0	100	E	2.00	2.0	74	G	5.0	4.0	100	E	8.5	7.0	9.0	8.0	89	G						
10	431100775			1.50	1.0	7.0	8.0	69	F	9.0	90	E	3.0	75	G	0.00	1.0	23	P	4.0	3.0	78	G	5.5	7.0	5.0	5.0	63	F						
11	431100800			3.75	1.0	7.0	7.0	81	G	8.0	80	G	4.0	100	E	3.75	2.0	100	E	3.5	4.0	84	G	6.0	8.0	7.0	9.0	83	G						
12	431101087			3.75	1.5	8.0	9.0	98	E	10.0	100	E	4.0	100	E	3.75	2.0	100	E	5.0	4.0	100	E	8.0	8.0	9.0	8.0	91	E						
13	431101112			2.75	1.5	8.0	9.0	91	E	5.0	50	P	4.0	100	E	0.75	2.0	56	P	5.0	4.0	100	E	5.0	7.0	8.0	9.0	79	G						
14	431101514			3.75	1.0	8.0	9.0	89	G	10.0	100	E	4.0	100	E	3.75	2.0	100	E	5.0	4.0	100	E	8.5	8.0	9.0	8.0	93	E						
15	431101537			2.00	1.0	7.0	7.0	69	F	9.0	90	E	3.0	75	G	0.75	3.0	79	G	5.0	1.0	66	F	7.0	7.0	6.0	4.0	67	F						
16	431101553			3.75	1.0	8.0	8.0	87	G	10.0	100	E	4.0	100	E	3.50	2.0	96	E	5.0	4.0	100	E	8.5	8.0	7.0	10.0	93	E						
17	431101634			3.75	1.0	9.0	7.0	87	G	8.0	80	G	2.0	50	P	2.00	2.0	74	G	5.0	4.0	100	E	5.0	8.0	9.0	10.0	87	G						
18	431101755			3.75	1.0	8.0	5.0	79	G	10.0	100	E	3.0	75	G	2.25	0.0	33	P	5.0	1.0	66	F	6.5	7.0	9.0	8.0	83	G						
19	431102056			3.75	1.0	7.0	8.0	84	G	9.0	90	E	4.0	100	E	3.50	2.0	96	E	5.0	4.0	100	E	8.0	8.0	8.0	10.0	94	E						
20	431102339			3.75	1.0	7.0	6.0	79	G	10.0	100	E	4.0	100	E	3.00	2.0	89	G	4.0	4.0	89	G	8.0	8.0	8.0	7.0	86	G						
21	431102444			2.75	1.0	8.0	2.0	65	F	2.0	20	P	3.0	75	G	2.25	2.0	78	G	5.0	4.0	100	E	4.5	7.0	7.0	5.0	65	F						
22	431102509			3.75	1.5	8.0	7.0	93	E	8.0	80	G	4.0	100	E	3.75	1.0	78	G	3.0	2.0	56	P	8.0	7.0	6.0	9.0	83	G						
23	431103612			2.00	1.0	8.0	7.0	73	G	8.0	80	G	4.0	100	E	2.25	2.0	78	G	4.0	5.0	100	E	7.0	7.0	8.0	9.0	85	G						
24	431103943			3.75	1.0	8.0	7.0	84	G	10.0	100	E	4.0	100	E	2.25	2.0	78	G	4.5	4.0	95	E	7.5	7.0	9.0	8.0	86	G						
25	431104585			3.75	1.0	8.0	3.0	74	G	10.0	100	E	4.0	100	E	3.75	2.0	100	E	4.5	4.0	95	E	7.0	7.0	7.0	9.0	82	G						
26	431106226			3.75	1.0	8.0	7.0	84	G	10.0	100	E	4.0	100	E	3.00	2.0	89	G	5.0	4.0	100	E	8.0	8.0	6.0	9.0	86	G						
27	431100789			3.75	1.0	8.0	8.0	87	G	10.0	100	E	4.0	100	E	3.75	2.0	100	E	5.0	4.0	100	E	8.0	8.0	8.0	10.0	94	E						
28	431101660			2.50	0.8	8.0	8.0	74	G	10.0	100	E	3.0	75	G	2.25	2.0	78	G	2.5	2.0	50	P	3.5	7.0	7.0	8.0	70	F						
29	431102896			3.75	1.0	7.0	8.0	84	G	9.0	90	E	4.0	100	E	3.75	2.0	100	E	4.5	4.0	95	E	7.0	8.0	8.0	6.0	81	G						

Table 4 Course satisfactions related to each student learning outcome shown in Table 3.

	Satisfaction (b)		Satisfaction (c)		Satisfaction (d)		Satisfaction (e)		Satisfaction (i)		Satisfaction (k)	
SLOs' Statistics	E	3	E	16	E	19	E	5	E	16	E	6
	G	19	G	7	G	6	G	12	G	6	G	15
	F	5	F	0	F	0	F	0	F	2	F	6
	P	2	P	6	P	4	P	12	P	5	P	2
Outcome Achievement (E + G)		75.9%		79.3%		86.2%		58.6%		75.9%		72.43%
Course's Outcome Achievement	G		G		G		P		G		G	
Action Plan	None		None		None		Yes		None		None	

The Analyzer Agent generates a Course Assessment Chart (figure 6) based on the course assessment template to show the continuous improvement of the student learning outcome achievement related to one course during different semesters.

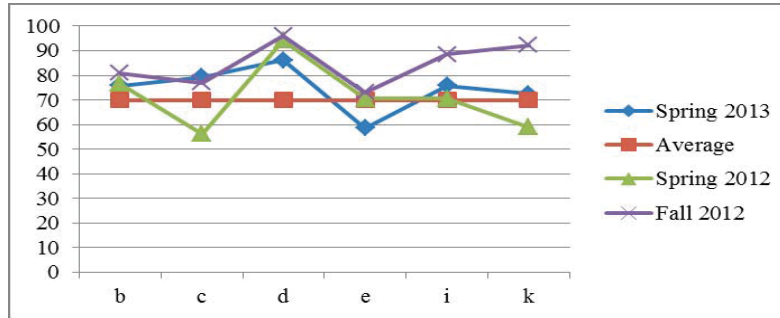


Figure 6 Course Assessment Chart.

5 Technical and Implementation Issues

An Environment for the E-Learning System has been developed at the Research Center of the College of Computer and Information Sciences of the King Saud University in conformity with our architecture. For the multi-agent system (MAS) implementation, we used JADE Intelligent Agents [28], [29], [30], which provide and offers the primitives for constructing and programming multi-agent systems. We chose JADE because it provides the primitives to program the MAS in Java [31]. To guarantee interoperability and ensure communication among agents, JADE is compliant with the specifications of the Foundation for Intelligent Physical Agents (FIPA) [32]. In such a direction, FIPA has developed specifications for interaction protocols, communicative acts, and content messages for agent communication [32].

To implement the Object Petri Nets, which describe the agent interaction and the communication among the agents, we used the Agent Communication Language (ACL) [10]. In addition, JADE has support for creating new ontology to produce domain-specific applications; this ontology allows agents to communicate easily and effectively.

Our adaptive e-learning management system has been developed using Web 3.0 technologies [49]. It represents a potential technology to improve communication, collaboration and sharing of resources in an e-learning system. Using Web 3.0, we include various services and tools to:

- enhance, facilitate and encourage students to access web based material anytime from anywhere, and to be lifelong learners,
- enable teachers to plan suitable online delivery structure, share goals of learning, easily manage the e-contents, upload student assignments, and improve e-content based on e-assessment.

We will attempt to provide a visualization of our system by providing screenshots of the different pages that appear to the users of the system to illustrate the main functionalities and features of the system. Figure 7 shows the interface of the system. As noted above, the user of the system may be a learner or an author. We will present one scenario for each of these users. Figure 8 shows the page that

is displayed to the author after he has logged in to the system. The author has two main responsibilities, which are to generate either adaptive course structures or adaptive course units.

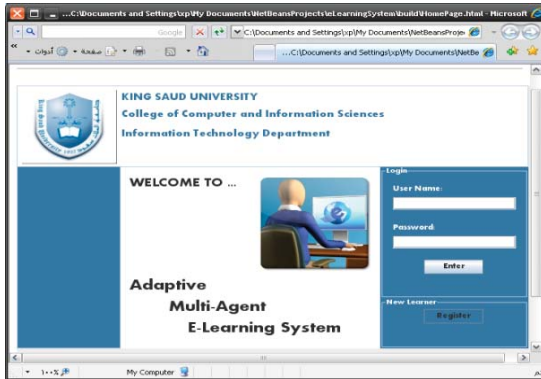


Figure 7 Home Page of the system

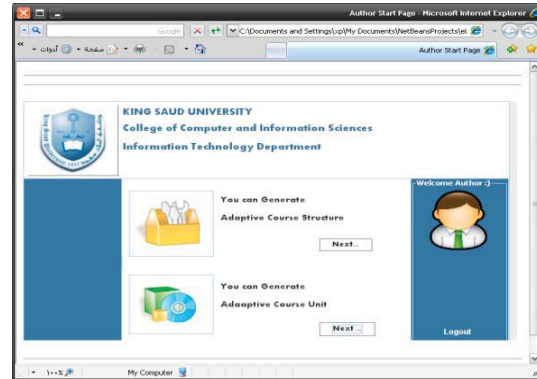


Figure 8 Main functions provided for Author.

Figure 9 shows the page that is displayed to the author if he/she elects to generate an adaptive course structure. The author should specify the course name and a brief description of the intended structure as well as the learning style and the target learner. Then, the responsible agent will retrieve the complete structure of the specified course to generate from it a course structure that is adaptive to the target learner. In Figure 10, we present the generated course structure from the previous step. As observed in the Figure 10, the author is able to alter the generated structure by adding, deleting or changing the arrangements of the nodes.

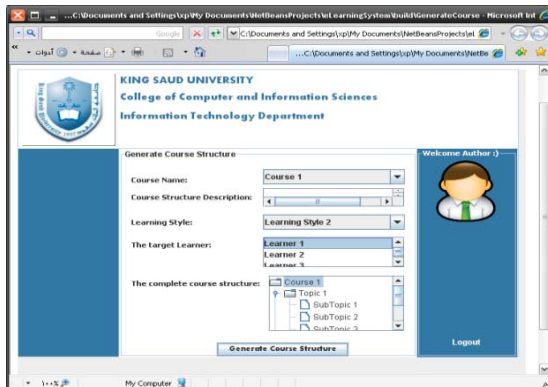


Figure 9 Generated Course Structure Screen

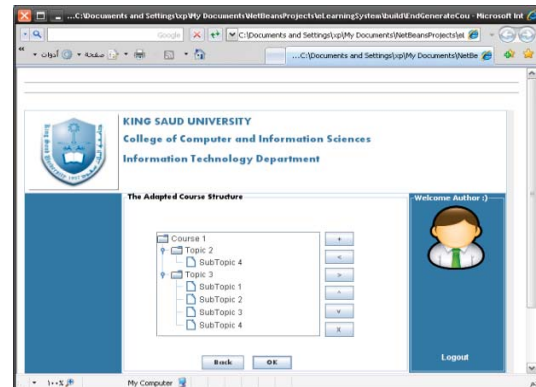


Figure 10 The Generated Adapted Course Structure

In contrast, Figure 11 shows the page that is displayed to the learner when he/she has logged in to the system. The learner can request to generate an adaptive course presentation or to generate an adaptive practice (Figure 12).

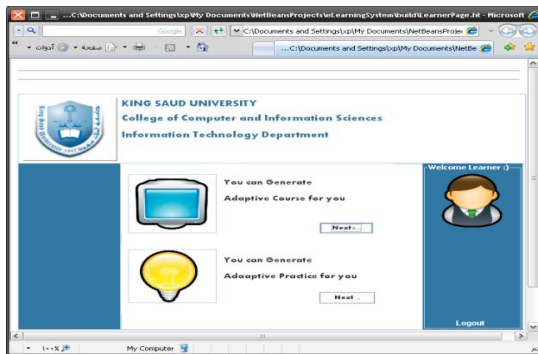


Figure 11 Main functions provided for the Learner

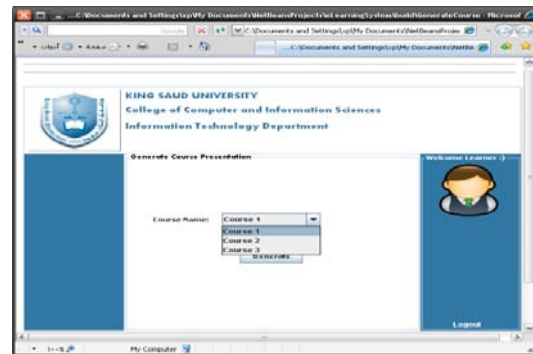


Figure 12 Generated Adaptive Course Presentation

Then, the responsible agent will generate the adaptive course presentation based on many factors, such as the learning process of the learner and the adaptive course structure, as well as the course units that have been generated before (Figure 13). If the learner elects to generate an adaptive practice for himself/herself, he/she should first specify the course to which this should be applied (Figure 14).

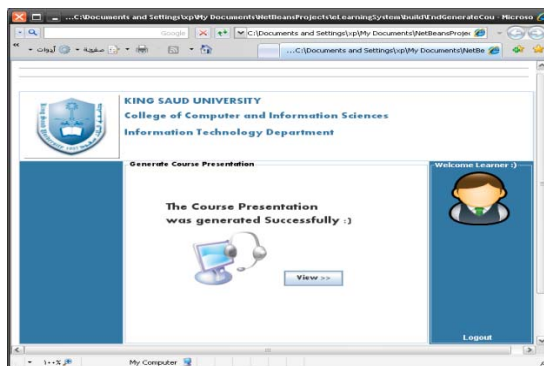


Figure 13 End of the Course Generation

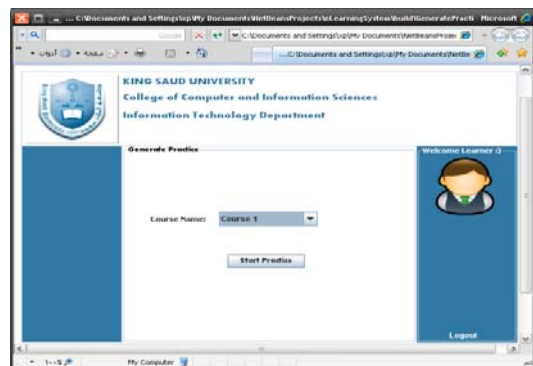


Figure 14 Generated Adaptive Practice Page

Then, the system will present questions that the learner should answer (Figure 15). When the learner has answered all the questions in the practice set, the responsible agent will evaluate the answers and display the result (Figure 16).

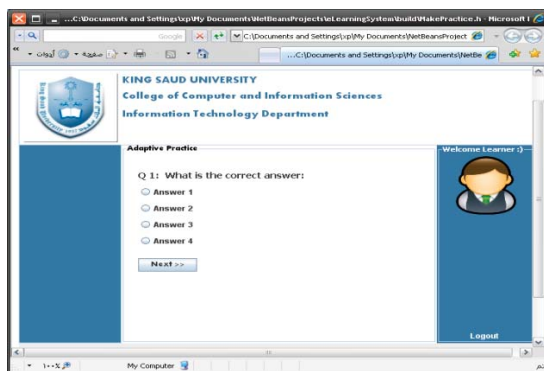


Figure 15 One Question of the Generated Practice Set

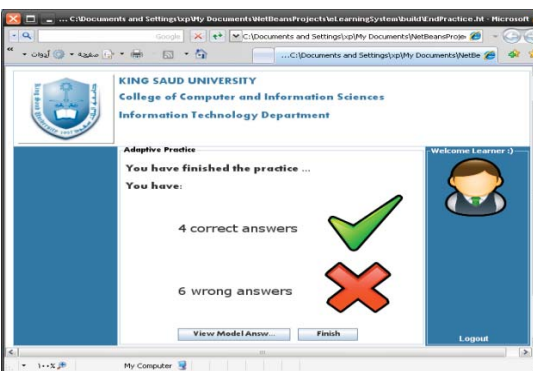


Figure 16 Evaluation of the Learner's Answers

6 Discussion and Conclusion

This paper presents a cyclical methodology for the continuous improvement of e-learning e-content based on ABET-related standards for student learning outcomes. We have presented an overview of an adaptive e-learning web application based on agent technology and object Petri nets. The proposed application has been demonstrated to better provide students with educational material that is best suited to their individual learning profiles. In addition we have shown that this application enables developing and implementation of continuous improvement strategies aiming at measuring student learning outcomes related to e-content.

In a previous work [5], we have proposed an adaptive e-learning system based on agents and Object Petri Nets and compared it with the state of the arts. Here, we have proposed an upgrade to this approach by adding new capabilities not previously reported in similar works such as the ABET based e-content continuous improvement standards.

In future work, we plan to develop an Academic Advisor Agent whose goal is to monitor and to track the level of achievement of individual student in all student learning outcomes during his/her study period.

Acknowledgements

The authors wish to acknowledge the collaborative funding support from the Research Center in the College of Computer and Information Sciences, King Saud University.

We also thank A.M. Artoli for fruitful discussion.

References

1. Safiye, T.: A Multi-Agent System Approach for Distance Learning Architecture. *Turkish Online Journal of Educational Technology*, Vol.4, Issue 4, pp. 21-26 (2005)
2. Grazia Violante, M., Vezzetti, E.: Implementing a new approach for the design of an e-learning platform in engineering education. *COMPUTER APPLICATIONS IN ENGINEERING EDUCATION*, DOI: 10.1002/cae.21564 (2012)
3. Miyamoto, T.: A Survey of Object-Oriented Petri Nets and Analysis Methods. *IEICE Trans. Fundamentals*, Vol. E88–A, No.11 (2005)
4. Hammami, S., Mathkour, H., Entesar, A.: A Multi-Agent Architecture for Adaptive E-Learning Systems Using a Blackboard Agent, 2nd IEEE International Conference on Computer Science and Information Technology Beijing, China, (2009)
5. Hammami, S., Mathkour, H.: Adaptive E-Learning System based on Agents and Object Petri Nets (AELS-A/OPN). Wiley Periodicals, Inc. It will be published on the *Computer Applications In Engineering Education* (Received 12 September 2012; accepted 21 September 2013), (2013)
6. Lin, F., Holt, P., Leung, S., Hogeboom, M., Cao, Y.: A multi-agent and service-oriented architecture for developing integrated and intelligent web-based education systems. *Proc. International Workshop on Semantic Web Technologies for E-learning*, (2005)
7. Gascueña, J., Fernández, A.: An Agent-based Intelligent Tutoring System for Enhancing E-learning/ E-teaching. *International Journal of Instructional Technology and Distance Learning*, Vol.2, No.11 (2005)

8. Paurobally, S., Cunningham, R., Jennings, N.: Developing agent interaction protocols using graphical and logical methodologies. In Workshop on Programming MAS, AAMAS (2003)
9. Mencke, S., Dumke, R.: Agent-Supported e-Learning. Technical Report, Otto-von-Guericke University of Magdeburg (2007)
10. Tnazefti, I., Arantes, L., Marin, O.: A Multi-Blackboard Approach to the Control/Monitoring of APS. WSEAS International Conference on Applied Mathematics and Computer Science (AMCOS'0Z), Brazil (2002)
11. Maia, R.F. Netto, M.L., "Work in Progress - A Distributed Approach to a Learning Management System using Multi-Agent Technology", Frontiers in Education, Proceedings 35th Annual Conference, pp.21-22, (2005).
12. Tveit, A.: A Survey of Agent-Oriented Software Engineering. Proceedings of the First NTNU Computer Science Graduate Student Conference, Norwegian University of Science and technology (2001)
13. Emil M. Popa, Ioana Moisil, Iulian Pah, Daniel Hunyadi , "A multi-agent architecture for e-learning" (2006).
14. Carolis, D., Pizzutilo, S., Cozzolongo, G., Drozda, P., Muci, F.: Supporting Students with a Personal Advisor. Educational Technology & Society, 9(4), pp. 27-41 (2006)
15. Brusilovsky, P.: KnowledgeTree: A distributed architecture for adaptive e-learning. Proceedings of the Thirteenth International World Wide Web Conference, New York, ACM Press, pp. 104-113 (2004)
16. Nabeth, T., Razmerita, L.: A. Angehrn and C. Roda, INCA: A cognitive multi-agents architecture for designing intelligent & adaptive learning systems. Computer Science and Information Systems, Vol.2(2), pp.99-114 (2005)
17. Salvatore, G., Domenico, R., Giuseppe, M.: MASHA-EL: A Multi-Agent System for Supporting Adaptive E-Learning. 19th IEEE International Conference on Tools with Artificial Intelligence, pp. 103-110 (2007)
18. Silveira, R., Vicari, R.: Improving interactivity in e-learning systems with multi-agent architecture. Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Springer, Malaga, Spain, pp. 466-471 (2002)
19. Elisa, B., Eliseo, B., Rosa, M.: An affective agent-based virtual character for learning environments. Workshop on Motivation and Affect in Educational Software, Amsterdam, 18-22 July (2005)
20. Shanghua, S., Mike, J.: Learning objects & learning styles as a foundation for a multi-agent web-based education system. Proceedings of the IASTED International Conference Web-Based Education, Feb. 21-23, Grindelwald, Switzerland (2005)
21. Jin, Q.: Design of a virtual community based interactive learning environment. Information Sciences, 140, pp.171-191 (2002)
22. Dragoslav, P., Milan, V., Mirjana, I., Zoran, B., Jovana, V.: Usage of Agents in Document Management. Computer Science and Information Systems. Special Issue on Advances in Computer Languages, Modeling and Agents. ComSIS Vol. 8, No. 1, pp. 193-210, DOI:10.2298/CSIS090608019P, Sep. (2011)
23. Gonzaga, T., Bentes, C., Farias, R., de Castro, M., Garcia, A.: Using Distributed-Shared Memory Mechanisms for Agents Communication in a Distributed System, Seventh International Conference on Intelligent Systems Design and Applications, ISDA (2007)
24. Hammami, S.: Evaluating E-Learning Systems Using E-Traceability Systems. Journal of Computer Science, Science Publications, Vol.6, No.2, pp.210-216 (2010)
25. Hammami, S., Ameer, T., Mathkour, H.: Course E-Generation using Atomic Learning Object. International Journal of Instructional Technology & Distance Learning, Vol.7, No.4 (2010)

26. Webber, C., Bergia, L., Pesty, S., Balacheff, N.: The Baghera project: a multi-agent architecture for human learning. Proceedings of the Workshop Multi-Agent Architectures for Distributed Learning Environments, AIED 2001, San Antonio, TX, USA. pp. 12-17 (2001)
27. Bala, M.B.: Developing Intelligent Agent Applications with JADE and JESS. Knowledge Based Intelligent Information and Engineering Systems, Vol.5179, Publisher: Springer Berlin / Heidelberg, pp. 236-244 (2008)
28. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE: A software framework for developing multi-agent applications. Lessons learned. Information and Software Technology, Vol. 50, Issues 1–2, January, pp. 10-21 (2008)
29. Salvatore, V., Vincenzo, C., Carmelo, M., Filippo, S.: An extended jade-s based framework for developing secure multi-agent systems. Computer Standards Interfaces, Vol.31, Issue 5, Publisher: Elsevier B.V., pp. 913-930 (2009)
30. Matteo, B., et al: Adding Organizations and Roles as Primitives to the JADE Framework. Dagstuhl Seminar Proceedings 08361, Programming Multi-Agent Systems, <http://drops.dagstuhl.de/opus/volltexte/2008/1639>, (2008)
31. James, O.: Foundation for Intelligent Physical Agents, 2012. <http://www.fipa.org>.
32. Carlos, P., Doru, P.: Modeling a Holonic Agent based Solution by Petri Nets. Computer Science and Information Systems. Special Issue on Advances in Computer Languages, Modeling and Agents. ComSIS Vol. 9, No. 3, pp. 1287-1305, 10.2298/CSIS111223031P, Sep. (2012)
33. I-Ching, C., Dong-Her, S., Shuen-Cheng H.: Augmenting a Web-Based Learning Environment through Blending Formative Assessment Services. Journal of Web Engineering, Vol.11 No.1, pp. 048-065. (2012)
34. <http://adaptweb.sourceforge.net> (20/01/2012)
35. Musa, D., Palazzo, J.: Sharing Learner Information through a Web Services-based Learning Architecture. Journal of Web Engineering, Vol.4 No.3, pp. 263-278 (2005)
36. <http://dbgroupp.comopolimi.it/brambilla/webml> (24/01/2012)
37. Ceri, S., Dolog, P., Matera, M., Nejdl, W.: Adding Client-Side Adaptation to the Conceptual Design of e-Learning Web Applications. Journal of Web Engineering, Vol.4 No.1, pp. 021-037, (2005)
38. Kannan, S. and Kasmuri, E.: Agent Based Content Management for E-Learning, 2005.
39. Vincenza, C., Alessandro, L., Michele, M., Giuseppe, M.: An Architecture to Support Adaptive E-Learning, IJCSNS International Journal of Computer Science and Network Security, Vol. 7 No. 1 pp. 166-178. (2007)
40. Costa E.B., Perkusich A., Jatoba A.A.: Petri net based modeling of the cooperative interaction in a multi-agent based intelligent learning environment. CESA'96 IMACS, (1996)
41. Hossein, R., Naveen S.: Student Learning Outcomes: An Integrated Continuous Improvement Process for Course and Program Assessment. Latin American and Caribbean Journal of Engineering Education, Vol. 2, No. 2, pp. 54-62, (2008)
42. Heinrich, E., Milne, J., & Moore, M.: An Investigation into E-Tool Use for Formative Assignment Assessment – Status and Recommendations. Educational Technology & Society, 12 (4), pp. 176–192. (2009)
43. Crespo, R. M., Najjar, J., Derntl, M., Leony, D., Neumann, S., Oberhuemer, P., Totschnig, M., Simon, B., Gutierrez, I., & Delgado Kloos, C.: Aligning Assessment with Learning Outcomes in Outcome-based Education. IEEE Education Engineering 2010, pp. 1239-1246. (2010)
44. Raghu, R., Prema, N.: Performance Improvements in Schools with Adaptive Learning and Assessment. 4th International Conference on Distance Learning and Education (ICDLE), pp. 10-14. (2010)

45. Gloria, S.T., Carmen, T.B., Gracian, T.: An approach to automatic learning assessment based on the computational theory of perceptions. *Expert Systems with Applications* 39, pp. 12177–12191. (2012)
46. Nangkula, U., Badiossadat, H.: Aligning assessment with learning outcomes. Published by Elsevier Ltd. Selection and/or peer reviewed under responsibility of the UKM Teaching and Learning Congress 2011, doi: 10.1016/j.sbspro.2012.09.372, *Procedia - Social and Behavioral Sciences* 60, pp. 228 – 235. (2012)
47. Kamsuriah, A., Nazlena M.A., Suhaila, Z.: An Improved Course Assessment Measurement for Analyzing Learning Outcomes Performance Using Rasch Model. *Kongres Pengajaran dan Pembelajaran UKM*, 2010, Published by Elsevier Ltd. doi:10.1016/j.sbspro.2011.05.065, *Procedia Social and Behavioral Sciences* 18, pp. 442–449. (2011)
48. ABET Board of Directors: Criteria for Accrediting Computer Computing Programs. http://www.abet.org/uploadedFiles/Accreditation/Accreditation_Process/Accreditation_Documents/Current/cac-criteria-2012-2013.pdf
49. Rajiv1 and Manohar Lal. Web 3.0 in Education & Research BIJIT - BVICAM's International Journal of Information Technology Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi Copy Right © BIJIT – 2011; July – December, 2011; Vol. 3 No. 2; ISSN 0973 – 5658 335