AGENT BASED INTELLIGENT COURSEWARE GENERATION IN E-LEARNING SYSTEMS

Ani Grubišić[1], Slavomir Stankov[1], Vlado Glavinić[2]
[1] Faculty of Science, Teslina 12, Split Croatia, {ani.grubisic, slavomir.stankov}@pmfst.hr
[2] Faculty of Electrical Engineering and Computing, Unska 3, Zagreb Croatia, vlado.glavinic@fer.hr

ABSTRACT
Since involvement of computers in educational, academic and business communities has increased, e-learning systems present involvement of information and communication technology as constituent part of education that supports a teacher in traditional learning and teaching process. In this paper, our focus is placed on learning materials in e-learning systems, how they are selected, sequenced and presented in a form of courseware. The main idea of this preliminary research related to intelligent courseware generation is in dynamic planning of courseware elements based on a student's knowledge. A term "intelligent courseware generation" has not been used so far in this form and with the semantics described later on. Here, we define a multi-agent system model for intelligent courseware generation that would enable collaboration of different intelligent agents that would, eventually, enhance student's learning and teaching process.

KEY WORDS
e-learning, courseware generation, intelligent agent, curriculum design and development, system architecture

1. Introduction

Nowadays, almost every household owns personal computer, computer usage increases in educational and academic institutions, and working without computers has become almost impossible. Involvement of computers in educational, academic and business communities has raised a question related to computerized literacy. Namely, use of computers in business sphere demands occupational retraining for new professions, or additional training for old professions, or permanent education and improving. Therefore, it is not surprising that emersion and development of new technologies has led to changes in education. Information and communication technology (ICT) has become a constituent part of education and supports a teacher in traditional learning and teaching process.

An intersection of a world of education and a world of ICT is called an e-learning [1]. E-learning systems provide access to electronically based learning resources anywhere at anytime for anyone [2]. Intelligent e-learning systems have capability to act appropriately in uncertain situations that appear in learning and teaching process. Their main goal is to be as good as highly successful human tutor. A special class of intelligent e-learning systems are intelligent tutoring systems (ITS).

The intelligent tutoring systems are computer systems that support and improve learning and teaching process in certain domain knowledge, respecting the individuality of learner as in traditional "one-to-one" tutoring ([3], [4], [5]). In order to reduce cost and time needed for ITS development another approach has been chosen, namely to create particular ITSs from flexible shells acting as program generators [6]. Each ITS consists of four components: domain knowledge, teacher module, student module [7] and communication module [8]. In this paper, we focus mainly on the teacher module and, especially, how learning materials are selected, sequenced and presented in a form of courseware.

Many authors define courseware as a "type of instructional or educational software program" (www.cybermediacreations.com) or "educational software that delivers course material and instruction via computer" (www.worldwidelearn.com), but we disagree with those definitions, because courseware is only a part of e-learning system, and, therefore, cannot be attributed as a program itself. We observe courseware as learning materials designed and presented with computer support, that are divided into learning themes, units and lessons. In order to ease courseware exchange with other e-learning systems, all courseware elements should be designed according to one of the standards, like SCORM (Sharable Content Object Reference Model, www.adlnet.gov) [9].

Usually, e-learning systems provide learning environment that is based on "not guided" learning approach, which means that student can freely choose their learning path in a courseware. In that way, it is easy to omit learning some courseware elements or learn something in completely wrong order or learn something that has inappropriate complexity level. That is the reason why intelligent e-learning systems, and especially intelligent tutoring systems, should present the student a courseware that has appropriate extent and difficulty level. Therefore, student's knowledge and courseware elements complexities are simultaneously taken into account in a process of selecting, sequencing and presenting learning materials to a student. This will eventually reduce cognitive loading and provide individualized learning guidance.

The main idea of this preliminary research related to intelligent courseware generation (ICG) is in dynamic planning of courseware elements based on a student's knowledge. A term "intelligent courseware" has first been coined in 1988 by Cunningham, Newens and Mollender [10], where word intelligent was put in quotes. In 1989, Chapelle and Mizuno [11] use this term without quotes and considers it to be "courseware which is sensitive to students' needs". Later on, it has been mentioned by Holt and Wood [12], on AIED1995 conference by Mayor [13], as well as Sarti and Marcke [14]. A term "intelligent courseware generation" has not been used so far in this form and with the semantics described later on in this paper.

A reference model for courseware generation has not been developed so far [15]. In order to optimize learning and teaching process, it is necessary to take account student's individuality, and, therefore, intelligent agents can be used to take care of courseware elements generation and their sequencing adapted to the needs of each student individually. The intelligent agents make a good choice since their abilities to learn, personalize and adapt, enable them to manage new situations and to provide pedagogically appropriate courseware presentation. Artificial intelligence (AI) planning helps intelligent agents in achieving their goals, as it provides them with the means of generating a sequence of actions [16]. Since intelligent agents can collaborate, in this paper we define a multi-agent system model for intelligent courseware generation that would enable collaboration of different intelligent agents in order to accomplish their goal – enhance student's learning and teaching process.

In this paper we present a preliminary research related to using agents for intelligent courseware generation in e-learning systems. In the second chapter we will present several approaches in the field of courseware generation that inspired us to define grounds for our novel approach. In the third chapter we describe our intelligent courseware generation model that can be used within any e-learning system. A platform that will be used to explain that model is an intelligent authoring shell eXtended Tutor-Expert System (xTEx-Sys) [1].

2. Related work

Courseware generation has been in a focus of scientists a long tome. There are many different names for courseware generation such as: curriculum sequencing, trail generation, course planning, instructional modeling. Also, terms like dynamic or adaptive or automatic or personalized or intelligent courseware generation, are used in the same contest, already defined by Chapelle and Mizuno [11]: generating "courseware which is sensitive to students' needs". Before we can present our work on intelligent courseware generation, it is necessary to elaborate some key approaches in this research field. Those different approaches are presented chronologically.

Dynamic Courseware Generation: The Dynamic Courseware Generator (DCG) [17] makes distinction between domain knowledge concepts (content planning) and educational resources (presentation planning). The domain knowledge concepts are presented in a graph. The educational resources are HTML pages linked to domain concepts. When student model and goal concepts are known, a content planner generates paths through the domain knowledge concept graph (using AI planner) that connects concepts known to the learner with the learning goal. That plan makes an outline for presentation planning. Presentation planner selects which of the educational resources linked to a concept will be presented to a student, as well as their order.

Generic Teaching Expertise: The Generic Tutoring Environment (GTE) [18] is a learning environment based on explicitly modeled knowledge. Instructional knowledge was divided into instructional tasks, methods and objects. Task is a set of activities that have to be accomplished during the teaching process. Methods perform tasks by dividing them into subtasks and then further on to primitives. A final result of task decomposition is a tree called task structure.

Adaptive Courseware Environment: The Adaptive Courseware Environment (ACE) [19] combines adaptive hypermedia techniques [20] with DCG's presentation planning [17]. Each concept from domain knowledge concept graph is linked with different types of learning materials that explain the different features of the concept. The teacher has to explicitly provide the path through the domain structure (for presentation planning) by specifying the sequence of educational resource. Therefore, learning paths is not generated automatically taking into account the learning goals. An adaptive sequencing component tries to keep student on optimal path and allows skipping of the mastered sections. Specht and others [21] developed the Web-based Intelligent Design and Tutoring System (WINDS) which is closely related to ACE.

Adaptive Personalized e-Learning Service: The Adaptive Personalized e-Learning Service (APeLS) [22] represents courses using narratives and candidate groups. A narrative is a sequence of steps through content where each step consists of educational resources sets that share the same learning goal (candidate groups). Presentation planning is restricted to selecting the particular candidate group. The candidate groups are pre-defined and have different structure of the educational resources as well as their output format. Conlan and Wade [23] continued their research in project called Intelligent Distributed Cognitive-based Open Learning System for Schools (iClass). In iClass course generation adapts intelligently and cognitively to student. Unlike APeLS, the iClass separates pedagogical information and the domain structure.

Self e-Learning Networks: The Self e-Learning Networks (SeLeNe) project investigated adaptive sequencing in a form of trail generation [24]. Trail is a linear sequence of educational resources. In SeLeNe, a student queries for educational resources. A SeLeNe component adds conditions of the queries, ranks the results by their relevance to the student and generates a trail. The SeLeNe's pedagogical knowledge is embedded in the algorithm that ranks the search results. The ranking (or sequencing) is done using relationships given in the metadata of the resources.

Personalized curriculum sequencing using modified item response theory: A personalized curriculum sequencing (PCS) approach based on the modified item response theory [25] includes an off-line courseware modeling process, four intelligent agents (learning interface agent, feedback agent, courseware recommendation agent and courseware management agent). The feedback agent collects learner explicit feedback information from the learning interface agent. The courseware recommendation agent recommends a personalized learning pathway to learner according to learner feedback response and concept relation degrees of courseware. The courseware management agent aids teachers while creating new course units, uploading courseware and deleting or modifying courseware. When the course recommendation agent gets the abilities of new students, then it ranks a series of appropriate course materials according to the new ability. The PCS model recommends appropriate courseware to the learner and provides learning paths that can be adapted to various levels of difficulty of course materials and various abilities of students.

Course generation based on statistical methods: Karampiperis and Sampson [26] use statistical methods for finding the best learning path. They first calculate all probable sequences of concepts (each concept is replaced by the related set of educational resources) and then select the appropriate one, according to a suitability function and a shortest plan algorithm. The selected sequence is called Learning Paths Graph and inherits the relations from domain structure and among resources. Suitability function, through statistical methods, determines how much particular educational resource is suited for a particular student is. It compares learning object characteristics with the ones of a student, and inversely. The authors argue that their approach is suitable for small-sized educational resources with metadata. The main downside is that the rating of educational resources is done by instructional designer with respect to a particular student.

Course generation based on hierarchical task network planning: Course generation based on hierarchical task network planning (HTN planning) was first implemented in e-learning environment FOrmacion HUMana (FORHUM) [27] and later in PAIGOS system [15]. In HTN planning, the goal is to achieve a partially or fully ordered list of activities called tasks. Those tasks are solved by decomposing into subtasks until primitive tasks are reached and carried out directly. In FORHUM, educational objectives (similar to domain structure concepts) are achieved by educational resources called educational units. Each educational unit has an associated learning style. For each concept, a HTN method is generated. The generation of the pedagogical knowledge is done automatically after the definition of the domain structure. The FORHUM teaches the prerequisites first and selects educational resources according to a learning style. In PAIGOS, generated courses are structured and adapted to different learning goals and to student's abilities. PAIGOS has an explicit representation of pedagogical knowledge derived from pedagogical experts and pedagogical theory. It allows generating courses for the same concepts but with different learning goals.

3. Intelligent Courseware Generation model

The main idea of this preliminary research related to intelligent courseware generation (ICG) is in dynamic planning of courseware elements based on a student's knowledge. This section describes the intelligent courseware generation (ICG) model architecture based on intelligent agents approach. Since intelligent agents can collaborate, this multi-agent system model for intelligent courseware generation enables collaboration of different intelligent agents in order enhance student's learning and teaching process.

First, we describe the xTEx-Sys, a platform that will be used to apply our ICG model. Then, an overview of model is presented, followed by its specifications and, finally, we compare it's compliance with SCORM Sequencing and Navigation model [9].

3.1. Intelligent authoring shell eXtended Tutor-Expert System

The xTEx-Sys is intelligent authoring shell for building intelligent tutoring systems in freely chosen domain knowledge [1]. The xTEx-Sys's most important feature is rigorous separation of domain knowledge design from courseware design.

The formalism for knowledge presentation in the xTEx-Sys is based on the semantic networks with frames and production rules. The top level of domain knowledge hierarchy is called the area, every area has sub-areas, and finally all subareas contain elementary knowledge objects with knowledge nodes and links from the semantic network that represents particular domain knowledge.

The teacher designs courseware in a specialized environment where the top level concept is a course. A courseware has a multilayered structure and it is divided into four levels: (i) unit, (ii) lesson, (iii) topic and (iv) instructional item. These courseware elements have been identified according to the pedagogical tradition in many European countries, but extended by one new expression, an instructional item which is considered to be an undividable element of the subject matter. A knowledge test can be assigned any courseware element beside instructional item.

The xTEx-Sys is SCORM compliant as its courseware structure and courseware elements are defined according to it. Each courseware element (beside test) contains one or more SCO's (Sharable Content Object) (instructional item can contain only one SCO). SCO can be an object for learning or an object for testing knowledge. In the xTEx-Sys, if SCO is used for learning, then its assets are nodes from domain knowledge. The semantic network nodes are undividable domain knowledge elements, just like assets in the SCORM. If SCO is used for testing knowledge, then its assets (questions) are randomly and dynamically assigned during realization time.

Therefore, knowledge presentation, as well as, the design and the sequence of educational contents, is done according to the SCORM model. In this way, it is possible to exchange courseware elements with other e-learning systems.

Since we wanted to adapt courseware generation in the xTEx-Sys to individual student's needs and characteristics, we have decided to make courseware a dynamic structure that modifies itself dynamically while student is interacting with content objects associated with its activities. That approach is proposed and described in the following sections.

3.2. Intelligent Courseware Generation model overview

Here, a multi-agent intelligent courseware generation that includes three intelligent agents and two databases is presented. A proposed model for intelligent courseware generation (ICG) deals with the following terms: domain knowledge, nodes, links, student model (overlay [28] and stereotype model [29] combination), courseware structure (units, lessons, topics, instructional items), course, content and presentation planning, SCOs and tests.

Due to enhancing quality and efficiency of student learning process, courseware is selected in such a manner that is suites student's individual needs. That selection depends on student's characteristics and his/her interactions with courseware, that is, courseware generation mainly depends on information stored in student model. Unsuitable courseware will only slow down student's learning process, so it is very important that an e-learning system itself takes into consideration student model and courseware elements' difficulty levels.

Therefore, in our ICG model one of the intelligent agents always checks suitability of courseware element difficulty level with the level of student knowledge. The suitability checking is done dynamically after evaluating student's knowledge based on feedback information gained from the student. It is important to emphasize that in our approach courseware is adapted to student only relating his/her knowledge, not by his/her interaction style (like in adaptive hypermedia systems).

The proposed ICG model architecture is composed of the following components (Figure 1.): (i) student stereotype database, (ii) stereotyped courseware plans database, (iii) courseware generation agent, (iv) courseware adaptation agent and (v) agent for managing student's feedback information. Prerequisites for this ICG model are formalized domain knowledge with clearly indicated concepts and relations among them or learning object (LO) repository based on one of the standards for designing learning contents (SCORM or other). Since, the xTEx-Sys enables creation of formalized domain

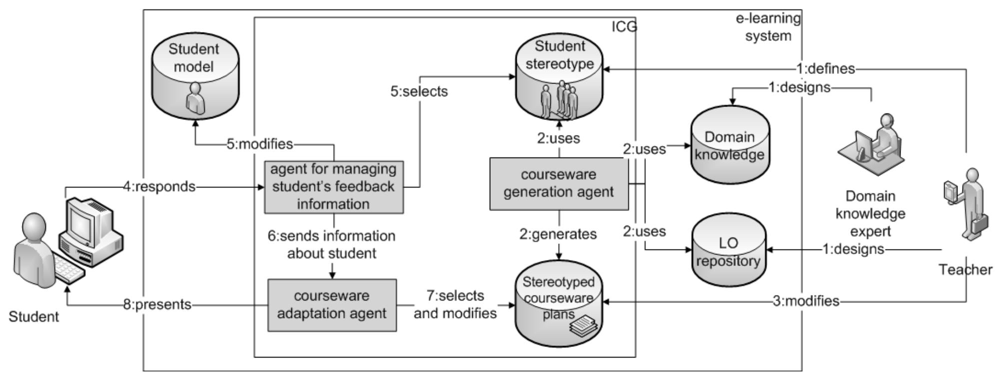

Figure 1. Intelligent Courseware Generation model

knowledge, as well as, creation of SCORM compliant courseware structure, it is be possible to implement and deploy ICG model into xTEx- Sys. Consequently, ICG model can be implemented and deployed in any e-learning system that fulfills ICG model's prerequisites.

Courseware generation agent is responsible for developing as much as courseware plans as there are stereotypes in student stereotype database. A final result of courseware generation is a tree called courseware plan. Therefore, this agent uses some of the state-based planning [16]. Those "stereotyped" courseware plans are adapted to a representative student of certain stereotype.

The purpose of the agent for managing student's feedback information is to collect explicit feedback information about student knowledge and to store them in student model, in order to ease mechanisms for adapting courseware. It uses a set of rules, from its internal mechanism, for assessing student's needs, evaluating student knowledge and deciding on difficulty level suitable for student. It then forwards collected information to courseware adaptation agent that uses that information as a criterion for filtering and adapting courseware elements. Those rules direct the behavior of the whole student and ensure personalization that will not disturb student in his/her learning and teaching process. This agent also allocates each student with one of the student stereotypes, as well as, maintains the student model.

Courseware adaptation agent is responsible for modifying generated courseware and adapting courseware to individual student based on student model, student's allocated stereotype, student's feedback information and courseware difficulty level. It compares difficulty level of each courseware element with student knowledge, selects the appropriate ones and presents them to student.

Experts create formalized domain knowledge used for developing learning objects. Teacher defines student stereotypes by defining their characteristics and can change only those stereotypes that he/she has created. The Courseware generation agent generates different courseware plan for each stereotype defined and domain knowledge that exists in e-learning system. "Stereotyped" courseware plans contain different sorts of courseware elements such as units, lessons, topics, instructional items and tests. This agent creates SCOs using domain knowledge elements as assets and stores them into LO repository. Teachers can modify "stereotyped" courseware plans in a way they believe they would fit better to a certain student stereotype. They can change the order of the courseware elements and add or delete tests.

Students can only give responses while solving tests (if presented courseware element's SCO is not a test, then response is the default one – I'm learning). Agent for managing student's feedback information uses data from knowledge testing to select student's stereotype and modifies student model. Then it sends information about student to courseware adaptation agent. Based on received information about student model and stereotype, courseware adaptation agent selects and modifies, if necessary, courseware plan so it is more suitable for that student. It presents the elements from the selected courseware plan to the student, one by one. Learning and teaching process finishes when student logs out the system.

3.3. ICG model specification

Use case diagram of ICG model is presented in Figure 2. Real life actors in this model are student and teacher. Main functionalities of actor student are: log in/out, learning, selecting domain knowledge and giving feedback through surveys or knowledge test. Main functionalities of actor teacher are: log in/out of system, defining student stereotypes and modifying generated courseware plans.

Other actors of the ICG model are three agents: courseware generation agent, agent for managing student's feedback information, courseware adaptation agent. The main functionality of the courseware generation agent is developing courseware plans for each student stereotype and domain knowledge available. Main functionalities of agent for managing student's feedback information are analyzing student feedback and modifying student model. Main functionalities of courseware adaptation agent are adapting courseware plan and presenting courseware element to individual student. Class diagram of ICG model is presented in Figure 3.

Figure 2. Intelligent Courseware Generation use case model

One of the possible scenarios that can be realized for a student that logs into the e-learning system for the first time is the following one:

1. The student logs into the e-learning system.
2. The student fills in a survey related to his/her personal characteristics.
3. The e-learning system sends gathered information about student to the agent for managing student's feedback information that selects the stereotype that best fits the student and creates new student model based on that stereotype.
4. The student selects the domain knowledge for his/her learning and teaching process.
5. The courseware adaptation agent gets stereotyped courseware plan for selected domain knowledge that suites the student stereotype best. Then it modifies it to better fit the student model and starts presenting it to the student. Selected and adapted courseware elements, and its SCOs, are presented to the student one by one.
6. The courseware adaptation agent selects the first free SCO. If there are no more SCOs in selected courseware plan, the learning and teaching process finishes.
7. If the selected SCO is not a test, then it is presented to the student for learning. The flow returns to Step 6.
8. If the selected SCO is a test, then after student solves it, agent for managing student's feedback information evaluates student's knowledge, modifies student model and sends new information to courseware adaptation agent. If the student's predefined stereotype has changed, then the flow returns to Step 5.

The three agents in the ICG model courseware, generation agent, agent for managing student's feedback information, courseware adaptation agent, collaborate together in order to accomplish their task which is presenting appropriate courseware to the student and evaluate his/her knowledge.

3.4. ICG model and SCORM

Our courseware plan corresponds to SCORM activity tree, or content organization, that enables sequencing and navigation according to activities and conditions defined. An activity is meaningful unit of instruction and it provides a learning resource (SCO) to the student or it contains sub-activities [9]. In the xTEx-Sys activity tree has a static structure. Courseware generation agent, in the ICG model, also creates a stereotyped courseware plans with static structure, but courseware adaptation agents makes their structure dynamic.

Courseware elements are organized into clusters that include parent activity and its immediate children, but not the descendants of its children. The children of a cluster are either leaf activities or other clusters. A leaf activity is not a cluster. The parent activity in a cluster contains the information about the sequencing strategy. The leaf activities of a cluster have content objects associated with them. Learning objectives are separate from learning activities. Attempts on activities occur within the context of attempts on their parent activity [9].

Each activity in the courseware plan has a set of sequencing control modes [9] that define navigation: Sequencing Control Choice (false), Sequencing Control Choice Exit (false), Sequencing Control Flow (true), Sequencing Control Forward Only (false), Use Current Attempt Objective Information (true), Use Current Attempt Progress Information (true). In our approach their default value is set as written in brackets.

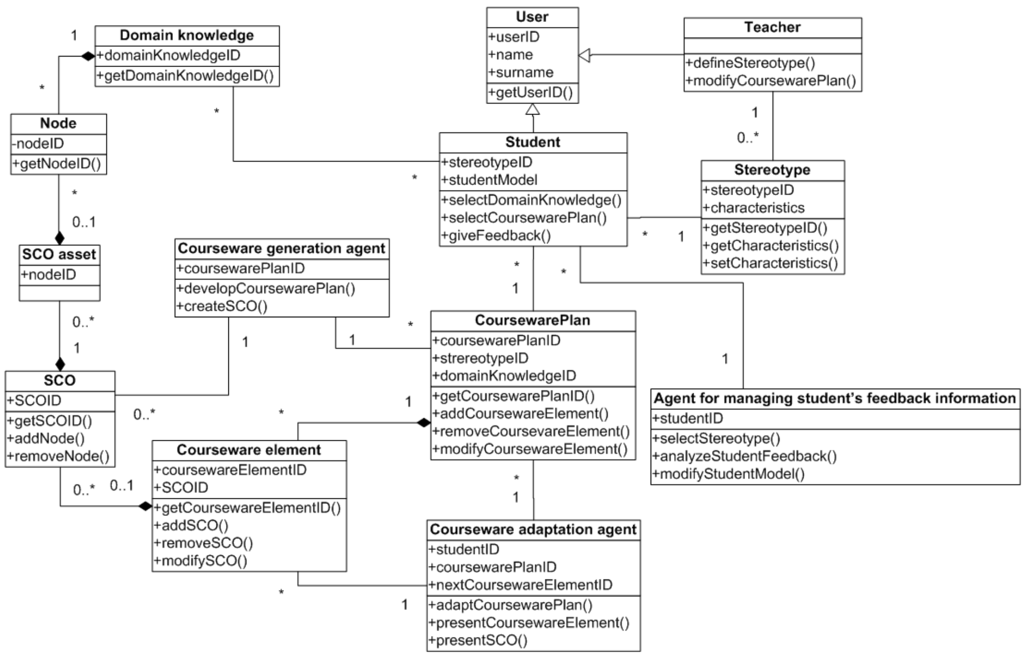

Figure 3. Intelligent Courseware Generation class model

In our approach we are limiting the number of attempts that a student is permitted on a learning activity that involves testing knowledge (Limit Condition Attempt Control = true, Limit Condition Attempt Limit = 3). As we do not want to limit the duration that can be spent in a single attempt of a learning activity, we do not define an Attempt Absolute Duration Limit for any learning activity.

SCORM employs a rule-based sequencing model. Each sequencing rule consists of a set of conditions and corresponding action (*if [condition_set] then [action]*) [9]. We use the following rule conditions: Objective Measure Greater Than, Objective Measure Less Than, Completed, Attempted, Attempt Limit Exceeded. The rule actions we use are: precondition (Skip, Disabled), post condition (Exit Parent, Exit All, Retry, Retry All, Continue, Previous), Exit.

An activity may have one or many learning objectives associated with it. Each learning objective has a set of tracking status information that allows learner progress toward the learning objective to be tracked, thus enabling conditional sequencing decisions. The process of determining a cluster activity's status information based on its children's status information is called rollup. Rollup rules may be applied to a cluster activity in order to associate cluster activities with content objects. Each rollup rule consists of a set of child activities, a set of conditions and a corresponding action (*if [condition_set] True for [child activity set] then [action]*) [9]. The rollup conditions we use are: Completed, Attempted, Attempt Limit Exceeded. The rollup child activity set consists of: All, Any, None. The rollup actions are: Completed, Incomplete.

4. Conclusion

One of the most important educational goals is to transfer knowledge to student. Traditional learning and teaching process is not directed to individual student, yet to a whole class. We know that students accept and process information in different ways and in different pace, that is, each student has his/her own learning style. That is the reason why teaching process does not always result with learning. Teaching method optimization is of great importance, but it is hard to select and apply one appropriate teaching method to a whole class.

As a result of a preliminary research related to mentioned problematic, this paper presents intelligent courseware generation model based on intelligent agent interaction, which enables automatic generation of stereotyped courseware plans, their modification and adaptation, in order to make them more suitable to individual student's needs and knowledge. Courseware adaptation is done according to student feedback information and, therefore, fully optimized and personalized. This model presents a multi-agent system, where intelligent agents communicate in order to adapt learning and teaching process to student's knowledge.

There are some key issues related to our approach to courseware generation. First, this ICG model presents an innovative approach because it can be implemented and deployed in any e-learning system that fulfills its prerequisites: formalized domain knowledge with clearly indicated concepts and relations among them or learning object repository based on one of the standards for designing learning contents (SCORM or other). Therefore, we provide domain-knowledge-free approach to courseware generation. Second, courseware generation over formalized domain knowledge is a novel approach, because it produces courseware that is knowledge-based, not document-based like in other e-learning systems. Third, student stereotypes used for courseware generation can be defined by teacher. This feature enables teachers to define as many student stereotypes as they think are necessary. Fourth, courseware generation is done prior to student's learning process, so student gets an overview of the course structure through his/her stereotyped courseware plan, but adaptation and modification is done dynamically in small steps immediately after student feedback and according to that feedback.

If we observe again the key approaches in this research field, elaborated in section 2, we can state some main differences and benefits of our approach over the mentioned ones. DCG, ACE and iClass make distinction between domain knowledge concepts and educational resources, but they mainly consider domain knowledge concepts to be domain structure, what makes our approach knowledge-based since we differentiate domain knowledge from educational resources and courseware structure. In our approach educational resources are much more than HTML pages linked to domain concepts: they are standardized learning objects that relate to domain knowledge concepts. ICG model enables working with any domain knowledge, as long it is formalized. GTE and PAIGOS structure courseware into smaller elements of different categories, but in our approach that structurization is done according to the pedagogical tradition in many European countries. iClass does content planning separately from presentation planning. In our approach it is also possible, since courseware element and SCOs make content, but their assets make presentation. Generation of the best courseware plan, for the defined stereotype and selected domain knowledge, uses some statistical methods and some of the state-based planning for defining suitability function and a shortest plan algorithm. Among all, only PCS approach based on the modified IRT enables adaptation to student knowledge, as well as presents multi-agent approach. Our courseware adaptation agent and agent for analyzing student feedback collaborate in order to evaluate student knowledge and automatically adapt courseware plan by modifying next courseware element to his/her needs. PAIGOS allows generating courses for the same concepts but with different learning goals. In our approach, different learning goals correspond to different stereotypes defined by teacher.

Our next step is to verify the effectiveness of proposed ICG model when compared with a traditional tutoring using one of the experimental designs explained in [30].

Future research and development will focus on implementing tutorial dialogue [31] as a technique that will enable testing student knowledge using controlled natural language communication. That communication will be controlled because it will use only words and phrases defined by the expert in the domain knowledge ontology.

Acknowledgements

This paper describes the results of research being carried out within projects 177-0361994-1996 *Design and evaluation of intelligent e-learning systems* and 036-0361994-1995 *Universal Middleware Platform for e-Learning Systems*, both within the program 036-1994 *Intelligent Support to Omnipresence of e-Learning Systems*, and funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

[1] S. Stankov, M. Rosić, B. Žitko, A. Grubišić, TEx-Sys model for building intelligent tutoring systems, *Computers & Education, 51* (3), 2008, 1017-1036

[2] D. Albert, E-learning Future – The Contribution of Psychology, *Proceedings of the 59th Annual Convention, International Council of Psychologists*, Winchester, England, 2001, 30-53

[3] E. Wenger, *Artificial Intelligence and Tutoring Systems* (Morgan Kaufmann Publishers, Los Altos, California, 1987)

[4] S. Ohlsson, Some Principles of Intelligent Tutoring, *Artificial Intelligence and Education, 1*, 1987, 203-238

[5] D. Sleeman, J. S. Brown, Introduction: Intelligent Tutoring Systems, 1-11. In: D. Sleeman, J. S. Brown (eds.) *Intelligent Tutoring Systems* (New York: Academic Press, 1982)

[6] T. Murray, Having It All, Maybe: Design Tradeoffs in ITS Authoring Tools, *Proceedings of the Third International Conference on Intelligent Tutoring Systems*, Montreal, 1996

[7] J. Self, Theoretical Foundations for Intelligent Tutoring Systems, *Journal of Artificial Intelligence in Education, 1*(4), 1990, 3-14

[8] B. Woolf, AI in Education, pp. 434-444, In: S. Shapiro (ed.) *Encyclopedia of Artificial Intelligence* (John Wiley & Sons, Inc., New York, 1992)

[9] SCORM 2004 3rd Edition Sequencing and Navigation (SN) Version 1.0

[10] J. Cunningham, M. Newens, I. Hollender, Practical experience in implementing "intelligent" courseware using expert systems tools, *IEEE Colloquium on Intelligent Tutorial Systems*, 1988, 711–713

[11] C. Chapelle, S. Mizuno, Student's Strategies with Learner-Controlled CALL, *CALICO Journal 7*, 1989, 25-47

[12] P. Holt, P. Wood, Intelligent Tutoring Systems: A Review for Beginners, *Canadian Journal of Educational Communications, 19*(2), 1990, 107-23

[13] N. Major, REDEEM: Creating Reusable Intelligent Courseware, *Proceedings of the International Conference on AIED*, AACE, Charlottesville, VA, 1995, 75-82

[14] L. Sarti, K.V. Marcke, Reuse in intelligent courseware authoring, *AI-ED 95 Workshop on Authoring Shells for Intelligent Tutoring Systems – 7th World Conference on AIED,* Washington, 1995, 82- 87

[15] C. Ullrich, Pedagogically Founded Courseware Generation for Web-Based Learning – An HTN-Planning-Based Approach Implemented in PAIGOS, *Lecture Notes in Artificial Intelligence*, 5260, 2008, Springer

[16] J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, 2003)

[17] J. Vassileva, Dynamic Courseware Generation: at the Cross Point of CAL, ITS and Authoring, *Proceedings International Conference on Computers in Education, ICCE'95*, Singapore, 1995, 290-297

[18] K. Van Marcke, GTE: An epistemological approach to instructional modeling, *Instructional Science 26*, 1998, 147-191

[19] M. Specht, R. Oppermann, ACE – adaptive courseware environment, *The New Review of Hypermedia and Multimedia 4*, 1998, 141-162

[20] P. Brusilovsky, E. Schwarz, G. Weber, ELM-ART: An Intelligent Tutoring System on the World Wide Web, *ITS 1996. LNCS, vol. 1086*, Springer, 1996

[21] M. Specht, M. Kravick, L. Pesin, R. Klemke, R. Authoring adaptive educational hypermedia in WINDS, *Proceedings of the ABIS 2001 Workshop,* 2001

[22] O. Conlan, V. Wade, C. Bruen, M. Gargan, Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning, *Proceedings of Second Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer, LNCS 2347, 2002, 100-111

[23] I. O'Keeffe, A. Brady, O. Conlan, V. Wade, Just-in-time Generation of Pedagogically Sound, Context Sensitive Personalized Learning Experiences, *International Journal on ELearning (IJeL), Special Issue: Learning Objects in Context, 5*,(1), 2006, 113-127

[24] K. Keenoy, M. Levene, D. Peterson, Personalization and Trails in Self e-Learning Networks, SeLeNe Deliverable 4.2., 2003

[25] C.-M. Chen, C.-Y. Liu, M.-H. Chang, Personalized curriculum sequencing utilizing modified item response theory for web-based instruction, *Expert Systems with Applications, 30*, 2006, 378–396

[26] P. Karampiperis, D. Sampson, Adaptive Learning Resources Sequencing in Educational Hypermedia Systems, *Educational Technology & Society, 8*(4), 2005, 128-147

[27] N. Mendez, C. Ramirez, J. Luna, IA Planning for Automatic Generation of Customized Virtual Courses, *Proceedings of ECAI 2004* Vol. 117, IOS Press, Valencia, Spain, 2004, 138–147

[28] I.P. Goldstein, Overlays: A Theory of Modeling for Computer Aided Instruction, AI Memo 406, MIT, Cambridge, MA, 1977

[29] E. Rich, User modeling via stereotypes, *Cognitive Science 3*, 1979, 329-354

[30] A. Grubišić, S. Stankov, M. Rosić, B. Žitko, Controlled experiment replication in evaluation of e-learning system's educational influence, *Computers & Education, 53(3)*, 2009, 591-602

[31] M. T. H. Chi, S. A. Siler, H. Jeong, T. Yamauchi, R. G. Hausmann, Learning from human tutoring, *Cognitive Science, 25*, 2001, 471-533