Adaptive Content Presentation in Asynchronous Learning Environments

http://dx.doi.org/10.3991/ijet.v7i2.1911

Georgios Tsoulouhas, Dimitrios Georgiou and Alexandros Karakos
Democritus University of Thrace, Xanthi, Greece

***Abstract*—In Adaptive Educational Hypermedia Systems (AEHS), we expect that the learning content presentation should be appropriately retrieved from learning object repositories, and dynamically tailored to each learner's needs. Each learner has a profile, subject to continuous change. The basic components of the learner's profile include his/her cognitive characteristics, background of knowledge, previous experience, and current emotional situation. This paper proposes the architecture of a Petri net-based workflow engine – scheduled to be implemented in a AEHS - aiming to provide a reliable and efficient platform for the execution of learning course flows in a grid environment. Dealing with the question of adaptive management of learning content, the proposed p-timed Petri net is capable of presenting learning content adapted to the learner's Learning Style and knowledge background. The proposed schema is accurately tested using a p-timed Petri net simulator. The schema may now be extended to include other components of the learner's profile.**

***Index Terms*—Adaptive Educational Hypermedia Systems, Expert Systems, P-timed Petri Nets.**

I. INTRODUCTION

Aiming to maximize the impact of teaching, tutors are called to reschedule the teaching strategy during the delivery of a lesson. The tutor continuously monitors the learner's behavior [1] and evaluates any feedback received during teaching. Using the collected information, tutors can adjust teaching to the learner's changing needs. As regards asynchronous e-learning, one expects that its governing systems should stand on a par with the best human tutor's teaching. It is expected that a well-designed asynchronous online educational system should "understand" each learner's needs, and reply serving learning material "tailored to those learner needs". Therefore, the asynchronous educational system must be adaptive to the learner's needs.

Traditional Technology-Enhanced Learning systems offer very few strategies for the personalization of educational offerings. For example, the widely used Intelligent Tutoring Systems do not offer personalized learning. Adaptive Educational Hypermedia Systems (AEHS) have been developed to address learner dissatisfaction by attempting to personalize the learning experience. The adaptivity dimension that inspires today's web technology in searching and retrieving information from globally distributed databases and repositories also influences research on educational technologies. Adaptivity in AEHS may be viewed as the system's ability for the adaptive retrieval of learning objects, and (learner) adaptive course presentation. Aiming to contribute to the learner-adaptive presentation of learning material, this paper presents a system that takes into account the learner's cognitive characteristics and the assessment of his/her progress, in order to adjust the teaching flow to their needs. Therefore, the purpose of implementing such a system in an AEHS is to dynamically organize the teaching flow as a function of two variables: the learners' Learning Style (LS) and learning progress. At this point we would like to underline that the proposed P-TPN is one out of several AEHS operating components, such as the cognitive characteristics detector, the monitoring and assessment device, or the LOs retrieval unit. Namely, our P-TPN serves as the dynamic learning content presenter. This part of AEHS uses outcomes from other system devices, such as the LS detector and the assessment of learner subsystems. Furthermore, the system makes use of the LOs retrieval unit, in order to present the "best fitted to learner" lesson.

The educational purpose of such adaptive educational offerings is to maximize learner satisfaction, learning speed (efficiency) and educational effectiveness. LSs with other means of adaptivity (e.g. user goals, prior knowledge) provide some improvements in learner satisfaction and knowledge gain. However there are very few studies of adaptive e-Learning which limit the adaptivity to just LS adaptivity. AEHS methods have been known to change the deployment of the most important resource in the education system: teacher and the learner time. Finally, referring to its efficiency, AEHS is expected to make a radical difference to education, specifically, the quality and effectiveness of the learning experience with one of its key contributions being "personalized learning" [2]. The ability of the proposed schema to dynamically manage the teaching flow contributes to the educational impact of AEHS.

Importantly, in order to design an AEHS component able to provide the best possible course presentation, we should take a number of factors into consideration. These primarily include the learner's cognitive characteristics, knowledge background, current emotional state, etc. The proposed system may be extended to take more than two factors into account.

LS detection is of great importance in AEHSs as adaptivity is strongly linked to the learner's characteristics. For example, in ACE (Adaptive Courseware Environment), a WWW-based tutoring framework, which combines methods of knowledge representation, instructional planning and adaptive media generation to deliver individualized courseware over the WWW, LSs play an important role. Experimental studies within ACE showed that the successful application of incremental linking of hypertext is dependent on students' LS and their prior knowledge [3].

Reference [4], shows how cognitive traits and LSs can be incorporated in web-based learning systems by providing adaptive courses. The adaptation process includes two steps: the individual needs of learners must be detected and then the courses must be adapted according to the identified needs. The LS estimation in their work is made by a 44-item questionnaire based on Felder-Silverman LS model [4]. This leads to the question of finding methods for user LS detection, as it became an AEHS's important issue as adaptivity is strongly linked to learners' personal characteristics. According to the authors' best knowledge, limited efforts on LS formal detection evaluation. Considerably more efforts have been made so far regarding the online LS detection [5]. To this direction, empirical studies were conducted on two educational systems (Flexi-OLM & INSPIRE) to investigate learners' learning and cognitive style, and preferences during interaction [6]. The Index of LSs questionnaire was used to assess the style of each participant according to the four dimensions of the Felder-Silverman LS model. It was found that learners have a preference regarding their interaction, but no obvious link between style and approaches offered was detected, to investigate methods for online detection of LSs.

Recently, results regarding online LS estimation in asynchronous e-learning systems appeared either based on a Bayesian network application [7] or using a formal Fuzzy Cognitive Maps (FCM) schema [8]. Both of these works are also based on Kolb's LSI [9]. Instead of using a static questionnaire to estimate the learner's LS, authors in the first work implemented the Fault Implication Avoidance Algorithm (FIAA) and a Probabilistic Expert System. Taking into account the structure of Kolb's LSI, the FIAA dynamically creates a descending shorting of learner's answers per question, decreases the amount of necessary input for the diagnosis, which in turn can lead to a limitation of possible controversial answers. The applied Probabilistic Expert System analyzes information from responses supplied by the system's antecedent users (users that complete the questionnaire before the present user) to conclude to a LS diagnosis of the present user. Evidence is provided that the effect of some factors, such as cultural environment and lucky guesses or slippery answers, which hinder an accurate estimation, is diminished. Their system gives a "clear" LS estimation avoiding "grey" estimation areas, of two equally important LSs. In this paper, we take advantage of the FIAA application to avoid the impacts of wrong answers. Another approach tends to pursue adaptation according to the generated user profile and its features that are relevant to the adaptation, e.g. user preferences, knowledge, goals, navigation history and possibly other relevant aspects that are used to provide personalized adaptations [10]. Researchers discuss the lesson content design tailored to individual users by taking into consideration user's LS and learning motivation. They relied on the Kolb's LS model and suggest that every LS class should get a different course material sequencing.

Other examples which implement different aspects of the Felder-Silverman Index of LSs are WHURLE, [11],[12] and ILASH [13]. The development of an adaptive hypermedia interface, which provided dynamic tailoring of the presentation of course material based on the individual student's LS, was part of the research work in [14]. By tailoring the presentation of material to the student's LS, authors believe students learned more efficiently and effectively. Students determine their LS by answering a series of 28 questions. These forms were based on an assessment tool developed at North Carolina State University based on B.S. Solomon's and Felder's Inventory of LSs. In iWeaver the Dunn & Dunn model is applied [15].

Petri Nets (PN) is a formal and graphical appealing language, which is appropriate for modeling systems with concurrency and resource sharing. PNs has been under development since the beginning of the 60s, where Carl Adam Petri defined the language. It was the first time a general theory for discrete parallel systems was formulated. Historically speaking the PN has its origin in Carl Adam Petri's dissertation [16] submitted in 1962 to the faculty of Mathematics and Physics at Technical University of Darmstadt, Germany. Later, the concept was refined and formalized by Holt [17]. The language is a generalization of automata theory such that the concept of concurrently occurring events can be expressed. Since those days, PNs became a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic and/or stochastic. Furthermore, as T. Murata in [18] explains, as a graphical tool PNs can be used as a visual-communication aid similar to flow charts, block diagrams and networks. PNs have been proposed for a variety of applications. They can be applied to any area or system that can be described graphically like flowcharts that represent parallel or concurrent activities.

Although we easily find a wide range of PN applications, very limited efforts have been made so far in Web-based adaptive learning applications. In Ref. [19] Gao and Dew performed a pioneering work, in which they proposed a high level colored timed PN based approach to providing some level of adaptation for different users and learning activities. Similarly, in [20] we can see a high level timed PN based approach to provide some kinds of adaptation for learning activities. Examples were given while explaining ways to realize adaptive instructions. To the best of authors' knowledge, PNs have not been applied so far to AEHS for adaptive course content presentation.

The main advantages of the method are both technical and educational. The technical refer to the study and organization of the dynamic concurrent behavior of the parallel and distributed system of a discrete course flow. The educational concern the representation of a flexible system that is able to adjust the teaching strategy at every step of the course. The aim of the proposed system is to form an optimal learning experience. To this end, the proposed PN has components assigned to: (1) various LO metadata, (2) a four-level model that groups students' learning preferences into learning processes based on experiential learning, and (3) assessment of learner performance at each level of the learning process. The system navigates the learner through a wide range of learning material. One may consider as the main educational advantage, not only its ability to present learning material tailored to learner needs, but its ability to rearrange the teaching strategy in case of failure in learner performance.

II. KOLB'S LEARNING CYCLE

Learning Theories diverge with respect to the fact that students learn and acquire knowledge in many different ways, which have been classified as LSs. Students learn by observing and hearing; reflecting and acting or by reasoning logically and intuitively. Students also learn by memorizing and visualizing; drawing analogies and building mathematical models [21]. Learning behavior has been also extensively examined in cognitive psychology. There is a great variety of models and theories in the literature regarding learning behavior and cognitive characteristics.

The issue of a learner's LS estimation in the scope of providing learning tailored to his / her educational needs has been extensively addressed in the literature. Besides exploring foundations posed by Dewey, Lewin and Piaget for experiential learning, Kolb presented a model of 4 particular elements, which together constitute an optimal learning process [9]. These are: active experimentation, concrete experience, reflective observation, and abstract conceptualization. The model is widely known (and depicted) as a learning cycle and Kolb also used its elements to identify 4 LSs, each corresponding to the spectrum between 2 elements - e.g. The Diverger, who supposedly prefers to learn through concrete experience and reflective observation. Let us focus on the four core elements and use them to illustrate and discuss activities in different teaching and learning environment. The model is represented in a two-dimensional graph, as shown in Fig. 1.

The preference is found by analyzing subject responses to a number of appropriate questions. A wide range of LS Inventories (LSI) and related questionnaires have been proposed to serve as LS detection tools. The LSI has been the subject of analysis in [22], [23],[24] and [25]. Their findings lent some support to the LSIs' two-dimensional structure; however they did not consider LS in relation to other constructs. Kolb's learning theory sets out four distinct LSs (or preferences), based on a four-stage cycle, which might also be interpreted as a "learning cycle". In this respect, Kolb's model is particularly elegant, since it offers both a way to understand individual people's different LSs, and also an explanation of a cycle of experiential learning that applies to the vast majority of humans. Kolb made a self-test LSI, which can reveal the weak and strong points of learning. As noted in [26], several systems that attempt to adapt to LS had been developed; however it was still not clear which aspects of LS are worth modeling, and what can be done differently for users with different styles. Since then efforts have been made and many surveys have been published stating the benefits of adaptation to LS.

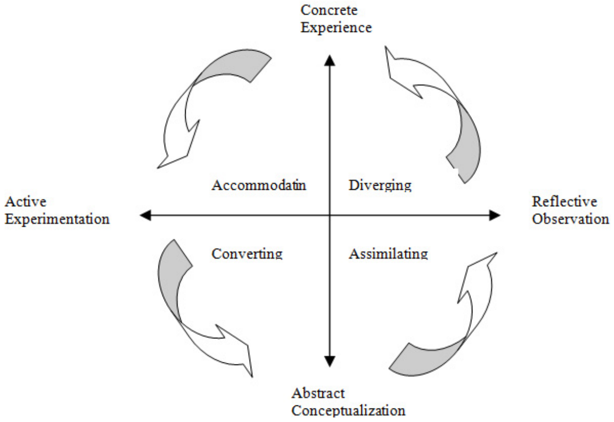

Figure 1. D. Kolb's learning cycle

To estimate a user's LS, Kolb in [27] introduced an LS inventory. Learners respond to 8 items, each of which contains four statements. The four statements appear in every possible combination of two, and the student has to choose one out of the pair, which means every item has 6 pairs. After the 48 answers have been given, the educator uses a two-dimensional schema to point out the leading LS that better expresses the learner's cognitive preferences. In some cases the leading LS may not be clear, as the final score shows a style between two adjacent styles in the LS cycle. If this is the case, educators are suggested to adapt a combined teaching style that fits to both LSs of this specific learner.

Based on Kolb's learning cycle, which is a schema representing the four factors with a decisive role in learning, certain probabilistic or fuzzy techniques have been introduced to render computers capable of recognizing users' LS [7],[8],[28]. Authors put efforts in the direction of diminishing the influence of factors that hinder an accurate estimation and providing service to an AEHS. Also, in cases were methodologies as those of David Kolb's concludes with two LSs of equal weight, our method provides a dominant LS as it makes use of the system's knowledge, i.e. the LS estimation of previous users. Kolb and many other cognitive scientists agree that in cases of learners between two LSs, it would be actually better to reflect this in some way rather than deciding on one. In AEHS this point of view should also be applied, and research should deal with this, designing more integrated architectures.

Practically, such systems can be applied, with minor modifications, to inventories of any kind, making them capable of taking under consideration both the examined user's responses and past users' classifications. In this paper, we consider that the late methodologies can provide accurate LS estimations to work with.

III. PETRI NETS

A. General

A PN is a particular kind of directed graph, together with an initial state called the initial marking M0, in which an information flow is depicted by a flow of tokens or markers, which are simply a conceptual depictions of a condition in the graph. The underlying graph N of a PN is a directed, weighted, bipartite graph consisting of two kinds of vertices, called conditions and transitions, where edges are either from a condition to a transition or from a transition to a condition. A marking (state) assigns to condition p a nonnegative integer k that is interpreted as the condition p is marked with k tokens. Pictorially, we condition k black dots in condition p. A vertex representing a condition in the graph is known also as a place and is represented as a circle. Vertices that represent transitions are represented as parallelograms. Pairs of two consecutive adjacent vertices allow tokens to pass from condition to condition through the interfering transition. Therefore, any condition of the net must be separated from the next by an event. The movement of tokens along the edges is controlled by a transition that is called an event (Fig. 2). A condition is said to be incident on an event if there is a directed edge from the condition to the event. If there is a directed edge from an event to a condition, the condition is a successor of the event. It is thus possible to define for all events an input set consisting of all conditions incident on the event and an output set containing all conditions that are successors of the event. Conditions within the net are capable of containing tokens such that a condition is said to hold or be true if a token is present in it. If all members of the input set of an event hold, the event is "enabled" and sometime later will "fire," removing the tokens from its input set and placing tokens in all members of its output set (Fig. 2).

Multiple edges directed away from a condition indicate that a token present in the condition may travel over either edge, but not both. Multiple edges directed to a condition indicate that a token may enter the condition through one of several paths.

The state of a PN is the set of conditions that hold at an instant in time. A net is live if, for any event, it is impossible for the net to reach a state from which that event cannot be enabled. A net is safe if there can never be more than one token in a condition at one time.

"Two events which share a common input condition can be in conflict if both events are enabled at the same time. If there are conflicting events, it is indeterminate which will occur. However, the occurrence of one will remove a token from the input set of the other and disable it. A net that is not safe or has conflicting events can often have these situations resolved by properly constraining the inputs to the net. Thus, we can define for a net a constraint set that contains illegal combinations of inputs.

We see that in order to effectively utilize a PN for speed independent design, we must guarantee that it is safe and conflict-free. Allowing tokens to collect in a condition would involve excessive complexity in a hardware simulation of the net, for it must keep track of the number of tokens present. Conflicting events in a net would appear as race conditions in a circuit, possibly causing a nondeterminate output. PNs will be further restricted by requiring them to be live, thus assuring that all portions of a circuit are utilized. A P-net is defined as a PN, limited by a set of constraint conditions, C~, which force the net to be safe and conflict-free. It is possible for C to be the null set if the original net already contains the qualities of safeness and freedom from conflict. It is also possible that there might not exist a set of constraint conditions which could force a given net to be safe and conflict-free. In such a case, the net cannot be converted into a P-net.

B. P-Timed Petri Nets

P-timed Petri Net (P-TPN) is an extension of traditional PNs [16] when used to describe the temporal behavior of a target system. For instance, in a P-TPN, if one time attribute is associated with condition, the firing rules are that a transition is enabled after tokens deposited in its input conditions take a fixed, finite amount of time. During that time, the tokens are not available. A transition e is enabled only after all of its input conditions p have at least one token and are still within (t_{min}, t_{max}) delay interval.

After the time delay, the transition becomes enabled. If fired, tokens are moved into the output conditions of that transition. If two time attributes are adopted, one is defined as the minimum delay t_{min} and the other as maximum delay t_{max}, the firing rules are that a transition is enabled after t_{min}; it remains enabled in (t_{min}, t_{max}) interval; if after t_{max}, the enabled transition has not been fired, it is forced to do so, moving tokens from its input conditions to output conditions. If the transition cannot fire, the token becomes unavailable. This "dead end" should be avoided by setting appropriate (t_{min}, t_{max}) and adjusting them dynamically.

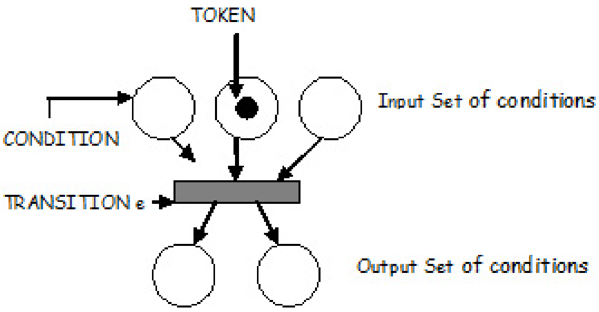

Figure 2. A typical element of PN. Adjacent vertices of the graph are of different kind: a condition and a transition are connected by an oriented edge.

By building a P-TPN based learning model to present learning material tailored to each student's needs we can benefit from the dynamic executive semantics of the PN and consequently obtain a powerful adaptation model.

IV. P-TPN BASED ADAPTIVE LEARNING MODEL

A course distributed by a web-based AEHS has a predefined LOs space. Each LO accompanied by a set of metadata expressing its semantics and other attributes such as hypermedia contents and numbers of knowledge nodes, recommended learning time, and linkage relation with other LOs, etc. For AEHS a leading role in the set of metadata is played by those expressing the learning profile of the user to whom the LO is addressed [29]. The learning profile has a number of components, such as LS, learning behavior, or current emotional state. As far as students differ in LS and learning abilities, they need to follow different learning paths, and to consume learning time in their own way.

Aiming to build an AEHS capable of managing learning content in the most appropriate way for each learner we use the P-TPN model. The P-TPN map in Fig. 3 shows the logical structure for adaptive presentation of learning material in terms of LS.

The upper level of the P-TPN at the first condition P_{10}, the user is called to respond to the Kolb's LS Inventory. Based on techniques described in [6] the system estimates the best possible LS that fit the user's cognitive characteristics. As far as the available tools for such estimation rank the LS from the most to the less alike, the proposed P-TPN may use the second best fitted to the user LS.

The transitions of the proposed P-TPN are supplied with a number of logical instructions that either energize societies of agents to retrieve LOs from LOs repositories, or direct the token whenever fired from an adjacent condition. For example, using sophisticated techniques, the system presents LOs tailored to the user's LS at one of the conditions P_{11}, P_{12}, P_{13}, or P_{14}. Then, transitions Tr_5, Tr_6, Tr_7 or Tr_8 , send the fired token to either P_{21}, P_{22}, P_{23}, or P_{24} where the gained knowledge is assessed.

At this point we recognize two alternative paths the token may follow.

a. Successful assessment of the first learning content presentation, and
b. Unsuccessful assessment of the first learning content presentation.

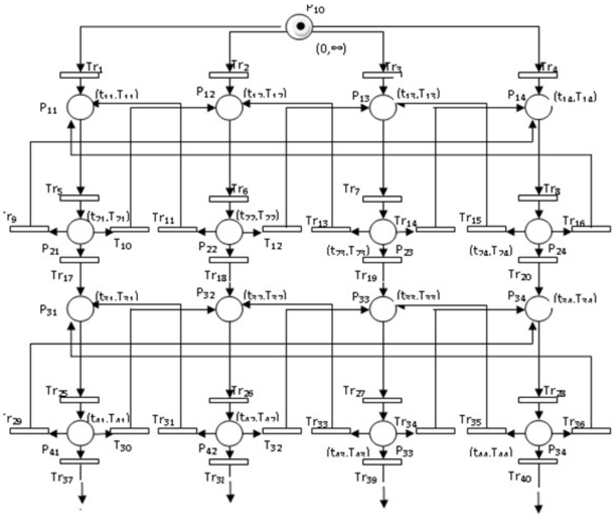

Figure 3. The token is fired to one of the Conditions Pi, i =1,2,3 or 4 that represent four different ways to present the same concept (one for every LS). Assessment of the knowledge gained by the learner takes condition at one of the conditions Pi, i =5,6,7 or 8 that lays on the corresponding to the previously chosen condition Pi, i=1,2,3 or 4 . In case of failure the token is redirected to the condition that corresponds to the second best LS estimation. After that the token is forced to be fired to the next LO (shadowed area).

In the first case, the token is fired from the condition P_{2j} to the condition P_{3j} of the next level, which is associated with the next learning content of the course sequence.

In the second case, the token will be fired from the condition P_{2j} to the condition P_{1k} for k≠j. In fact, k is the index of the LS that is the adjacent LS in the resulted rank of the LS estimation in condition P_{10}. In this case, even if at the condition P_{2k} results inadequate assessment of knowledge, the token will be fired to the condition P_{3k}. This was decided because there is no common border for three LS, and thus there is no reason for the same learning content to be taught using a third teaching style.

A. Firing Rules of P-TPN Based Adaptive Learning Model

In the proposed P-TPN model, the presence of the token in any condition but the P_1, activates the presentation of its corresponding learning content. In condition P_1, the LS detection algorithm is applied. The allowable time interval for the learning content's presentation is decided according to two factors. The statistics resulted from the time consumed by all the learners who made use of this learning content before, and the degree of present user concentration on the learning procedure.

Let us now denote by $t_{ij,\ min}$ the minimum time needed for a specific learner to work on the learning content presentation at the i level addressed to the j LS. In a similar way, we denote the maximum time $T_{ij,\ max}$ for the learner to spend time working on the same learning content. Thus, as i assumes as many values as the number of content presentations in the course sequence. Index j assumes up to four values (as many as the Kolb's LS are).

As learner begins to focus his/her attention on a specific learning content hypermedia presentation, time related information stored in the database. The token at the condition $P_{i,j}$ stays there for at least time units. The learner must wait at least $t_{i,j,min}$ before he exits the presentation. As soon as $T_{ij,\ max}$ expires the adjacent enabled transition will be chosen automatically. That means that no learner can stay more than $T_{ij,\ max}$ time units. It appears that $T_{ij,\ max}$ is subject to reduction but not to increase. The reduction of maximum time is a statistical result of the time spent by predecessor learners on the same learning content presentation. The reduction -when it occurs-, does not influences dramatically the maximum time $T_{ij,\ max}$.

The predefined transition priorities usually reflect users' learning profiles. For instance, in Fig. 3, condition P_i, for i≠1, stimulates a learning content presentation tailored to the needs of a learner as the token is conditioned on it. Conditions P_{11}, P_{12}, P_{13}, or P_{14} are candidate to be reached by the token at any time due to the minimum time that is 0. According to the LS detection of the user estimated at P_{10}, a unique condition can be reached at the level 1 of the PN.

Suppose that P_{13} is selected, P_{10} remains visible on the screen as far as the user works on it, and the learning content that corresponds to P_{13} appears on the screen as soon as the token is fired from P_{10}.to P_{13}. The learner is allowed to work on the learning content for at most T_{13}, max time units. He/she is not allowed to exit from this interface that corresponds to condition P_{13} in an interval shorter than $t_{13,\ min}$.

B. PN simulation

To the purpose of study the behavior of the proposed schema, we made use of the "S/T Petri-Net Simulation System" applet [30]. The applet was written as part of a software project with the purpose of implementing a simple simulation system (PNS) for extended condition/transition PNs. An example of such simulation appears in Fig. 4. Our PN schema implied in the simulator that has been set to run sequentially, and the choice of transition enabled randomly.

The purpose of this simulation is to justify the validity of the proposed P-TPN. Instead of using random numbers generator to fire the token from a condition to another condition, and to define the time needed to stay in a condition, the proposed P-TPN supplies the simulator with the outputs of the LS detection engine, and time restrictions. Moreover, transitions are supplied with logical statements (using Boolean operators). The statements at the first four transitions refer to learner LS. In Fig. 4, the token has been fired to condition in the lower-left side of the frame.

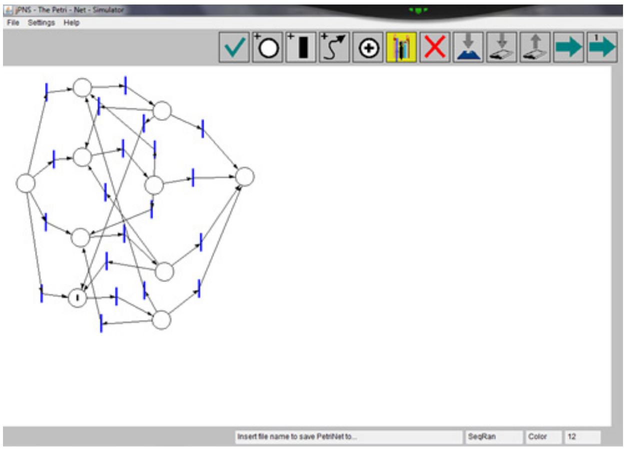
JPNS - The Petri - Net - Simulator
File
Settings
Help
Insert file name to save PetriNet to...
SeqRan
Color
12

Figure 4. The Braunl's PN's simulator supplied with appropriate time and learner's LS rules gave crucial information that improves the system's efficiency.

This represents a learning content presentation tailored to the needs of this particular learner LS. At the same condition the assessment of gained knowledge results an output. The transitions connected to this condition are supplied with logical statements that direct the token either to the exit of this course unit, or to a second presentation of the same learning content (this time using tailored to the needs of the second best detected LS for this specific learner. In case of second learning content presentation the transition restrictively fire the token to the exit of this learning stage.

V. RESULTS DISCUSSION AND FUTURE WORK

An issue (among others) in AEHS is the system's ability to manage dynamically the course of study. The current results point out to the suitability of application of PNs to describe the management of learning content of an entire course of study. In order to describe management of learning content it is appropriate to employ such graphic tools to describe and formulate the basic rules that govern complex management situations. We found that P-TPN represents a course presentation flow that dynamically adapts time-dependent changes of learner's profile.

Moreover, the proposed schema simulates the logical process that a human tutor follows in order to plan a teaching strategy. The tutor decides about the LO presentation sequence, on a number of inputs provided by the learner. Similarly, an AEHS equipped with an application that simulates such human actions has the ability to navigate the learner through a large number of available LOs .We found that the use of P-TPN is a powerful tool that allows the AEHS to present learning content tailored to continuously changing learning profile of a learner. This should be considered as the major contribution of the proposed schema to the integration of an AEHS. The authors look forward to implementing the proposed PN in the AEHS under construction in Democritus University. Looking forward to future work, the authors intend to study the impact of P-TPN application on the assessment of a test group. The results will be compared to the outcomes of teaching without P-TPN. It is also intention of the authors to extend the use of P-TPN to more complex schemata, including more components of the learner's cognitive characteristics. Moreover, it is the authors' intention to extend the system's operability using information referring to integrated learner profiles and implement such schemata in parallel operation, for handling huge numbers of users at the same time.

ACKNOWLEDGMENT

Most of this work was done while the second author was with Department of Computer Science at University of Cyprus. We thank both institutions for their support and hospitality.

REFERENCES

[1] G. Tsoulouhas, D. Georgiou and A. Karakos, "Detection of Learner's Affective State Based on Mouse Movements," *Journal of Computing*, vol. 3, issue 11, pp 9-18, November 2011.

[2] C. Mulwa, S. Lawless, M. Sharp, I. Arnedillo-Sanchez, V. Wade "Adaptive educational hypermedia systems in technology enhanced learning: a literature review," *Proceedings of the 2010 ACM Conference on Information Technology Education, 7–9 October, Midland, USA.*

[3] M. Specht, R. Oppermann, "ACE - Adaptive Courseware Environment," *The New Review of Hypermedia and Multimedia*, vol 4 (1), 1998, pp. 141 -161. http://dx.doi.org/10.1080/13614569808914699

[4] S. Graf, T. Liu, K. N. Chen, S. J.H. Yang, "Learning styles and cognitive traits – Their relationship and its benefits in web-based educational systems," *Computers in Human Behavior*, vol. 25, Issue 6, pp. 1280-1289, November 2009.

[5] S. Graf, Kinshuk, T. C. Liu, "Supporting Teachers in Identifying Students' Learning Styles in Learning Management Systems: An Automatic Student Modelling Approach," *Educational Technology & Society*, vol. 12(3), 2009, pp. 3–14.

[6] K.A. Papanikolaou, A. Mabbott, S. Bull, M. Grigoriadou, "Designing learner-controlled educational interactions based on learning/cognitive style and learner behaviour," *Interacting with Computers*, vol. 18, 2006, pp. 356-384. http://dx.doi.org/10.1016/j.intcom.2005.11.003

[7] S. Botsios, D. A. Georgiou, N. Safouris, "Contributions to AEHS via on-line Learning Style Estimation," *Journal of Educational Technology and Society*, vol. 11 (2), 2008.

[8] D. A. Georgiou, D. Makry, "A learner's style and profile recognition via fuzzy cognitive map," *Proceedings of IEEE International Conference on Advanced Learning Technologies*, Joensu Finland, 2004. http://dx.doi.org/10.1109/ICALT.2004.1357370

[9] D. A. Kolb, "Experimental learning: Experience as the source of learning and development," *Jersey: Prentice Hall*, 1984.

[10] D. Milosevic, M. Brkovic, M. Debevc, R. Krneta, "Adaptive Learning by Using SCOs Metadata," *Interdisciplinary Journal of Knowledge and Learning Objects*, vol. 3, 2007, pp. 163-174.

[11] Moore, Brailsford, Stewart, "Personally tailored teaching in WHURLE using conditional transclusion," Short Paper, *12th ACM Conference on Hypertext and Hypermedia*, 2001 Århus, Denmark, pp. 14-18, August 2001.

[12] Brown and Brailsford, "Reappraising cognitive styles in adaptive web applications," *Proceedings of the 15th international conference on World Wide Web Edinburgh*, Scotland, 2006, pp. 327 – 335.

[13] N. Bajraktarevic, W. Hall and P. Fullick, "Incorporating learning styles in hypermedia Environ-ment: Empirical evaluation," viewed 4 Jul 2008 from http://www.is.win.tue.nl/ah2003/proceedings/www-4/

[14] C.A. Carver Jr., R. A. Howard and W. D. Lane, "Enhancing student learning through hypermedia courseware and incorporation of student learning styles," *IEEE transactions on Education*, vol. 42 (1), 1999, pp. 33-38. http://dx.doi.org/10.1109/13.746332

[15] C. Wolf, "iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education," *Proceedings of the Fifth Australasian Computing Education Conference on Computing Education*, vol. 20, 2003, pp. 273-279.

[16] C. A. Petri, "Communication with Automata," *New York: Griffins Air Force Base*, Tech. Rep., vol. 1, Suppl. 1, 1966, pp. 65-377.

[17] A. Holt, "Information system theory project," *Dept.of Commerce*, Clearinghouse, Springfield, VA 22151, 1968.

[18] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, Issue 4, 1989, pp. 541-580. http://dx.doi.org/10.1109/5.24143

[19] S. Gao and R. Dew, "Enhancing Web-based Adaptive Learning with Colored Timed Petri Net," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4798, 2007, pp. 177-185.

[20] S. Gao, Z. Zhang, J. Wells and I. T. Hawryszkiewycz, "Supporting adaptive learning with high level timed Petri nets," *International Journal of Intelligent Systems Technologies and Applications*, vol.4 (3-4), 2008, pp. 341 – 354. http://dx.doi.org/10.1504/IJISTA.2008.017277

[21] R. M. Felder and L. K. Silverman, "Learning and teaching styles in engineering education," *Engineering Education*, vol. 78 (7), 1988, pp. 674-681.

[22] L. Willcoxson and M. Prosser, "Kolb's learning style inventory (1985): Review and further validity and reliability," *British Journal of Educational Psychology*, vol. 66, 1996, pp. 247–257. http://dx.doi.org/10.1111/j.2044-8279.1996.tb01193.x

[23] I. Yahya, “Willcoxson and Prosser’s factor analyses on Kolb’s (1985) LSI data: reflections and re-analyses,” *British Journal of Educational Psychology*, vol. 68, 1998, pp. 281–286. http://dx.doi.org/10.1111/j.2044-8279.1998.tb01290.x

[24] R. Loo, “The Distribution of Learning Styles and Types for Hard and Soft Business Majors,” *Educational Psychology*, vol. 22(3), 2002, pp. 349-360. http://dx.doi.org/10.1080/01443410220138566

[25] R. Loo, K. Thorpe, “Confirmatory factor analyses of the full and short versions of the Marlowe-Crowne social desirability scale,” *Journal of Social Psychology*, vol. 140, Issue 5, 2006, pp. 28-635.

[26] P. Brusilovsky, “Adaptive hypermedia. User Modeling and User Adapted Interaction,” *Ten Year Anniversary Issue (Alfred Kobsa, ed.)*, vol. 11 (1/2), 2001, pp. 87-110.

[27] D. A. Kolb, “Learning Style Inventory - version 3: Technical Specifications,” *TRG Hay/McBer, Training Resources Group*, 1999.

[28] D. A. Georgiou and S. D. Botsios, “Learning style recognition: A three layers fuzzy cognitive map schema,” *IEEE International World Congress on Computational Intelligence FUZZ-IEEE*, 2008, Hong Kong.

[29] S. Botsios, D. A. Georgiou, “Recent contributions to a Generic Architecture Design that supports Learning Objects Interoperability,” *International Journal of Distance Education Technology*, vol. 7 (4), 2009, pp. 17-37. http://dx.doi.org/10.4018/jats.2009071002

[30] T. Bräunl, “Bräunl's Petri Net Java Applet” from http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/java/Braunl , 2005

AUTHORS

G. S. Tsoulouhas completed his B.Sc. degree on Electrical Engineering at the Democritus University of Thrace and he is working for his PhD on Data Mining and Adaptive Educational Hypermedia Systems. He deals with programming in Windows and Unix environments. (e-mail: gtsoulou@ee.duth.gr)

D. A. Georgiou Born in 1948. BSc. AUTh Mathematics Department , PhD. DUTh Postdoc, UCBerceley. Visiting Scholar at UC, Davis (1980-1982). Visiting Professor URI (1989-92). Associate Professor, School of Engineering DUTh since 1991. Instructor of several undergraduate and postgraduate courses he also teaches vocational courses to Hellenic Power Corporation, Center for Productivity and Development, Training School for High School Teachers, Hellenic Air Force e.a. He published six text books. Research interests: 1) Qualitative behavior of solutions of ODE, Difference Equations, and PDEs, 4) Numerical Methods for Boundary Value Problems, 5) Intelligent Tutoring Systems. 6) Computer Networks. Research Papers published in Several Scientific Journals and conferences. Member of IEEE Computer Society, AACE, AMS, ECMI e.a. Referee for the Journal of Mathematical Analysis and its Applications and other journals in mathematics and educational technology. (e-mail: dgeorg@ee.duth.gr)

A. S. Karakos received the Degree of Mathematician from the Department of Mathematics from Aristotle University of Thessaloniki, Greece and the Maitrise d' Informatique from the university PIERRE ET MARIE CURIE, Paris. He completed his PhD studies at university PIERRE ET MARIE. He is Assistant Professor at the Dept. of Electrical and Computer Eng., Democritus University of Thrace, Greece. His research interests are in the areas of learning systems, data analysis and programming languages. (e-mail: karakos@ee.duth.gr).

Manuscript received 21 December 2011. Published as resubmitted by the authors 27 May 2012.