Learning Content Adaptive Presentation Using a P-timed PN Simulator

Marina Vlanti, Vasia Katsageorgiou, Eleni Panagou and Dimitrios Georgiou

Department of Electrical & Computer Engineering, School of Engineering,
Democritus University of Thrace, GR 67100 Xanthi, Greece

Abstract

Research in the area of Adaptive Educational Hypermedia Systems is aiming to develop sophisticated methods for adaptive content presentation. The educational content forms a sequence of Learning material presentations that are collected from a distributed over the net Learning Object repositories and are presented adaptively to every isolated user. Here, adaptive content presentation is considered in terms of i) the user's style, ii) the user's knowledge background monitoring and iii) the user educational needs. To the purpose of establishing a flexible structure for content adaptive presentation we propose a P- timed Petri Net with inhibitor edges

A path that has been ran by a token represents the optimum flow of learning content tailored to the cognitive and learning needs of each learner.

Keywords: P-timed Petri Nets, Adaptive Educational Hypermedia Systems, Expert Systems.

Introduction

Aiming to design an AEHS component capable to take advantages from the user's learning profile recognition in order to provide the best possible course presentation, we made use of a suitable p-timed Petri net. Learning Style (LS) is one of the learning profile components.

LS recognition becomes of great importance in AEHSs as far as the dimension of adaptivity is strongly associated to the learner's personal 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 also to their prior knowledge (Specht & Oppermann [27]). In their research, Graf and Kinshuk [14] show how cognitive traits and LSs can be incorporated in web-based learning systems by providing adaptive courses. The adaptation process includes two steps. Firstly, the individual needs of learners have to be detected and secondly, the courses have to 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 (Graf & Kinshuk 2007). Such ascertainment leads to the question of finding methods for user's LS detection, as it became an AEHS's important issue as far as the dimension of adaptivity is strongly associated to learner's personal characteristics. According to authors' of this paper best knowledge, limited efforts have been put so far, on LS formal recognition. Even less efforts have been put so far regarding the online LS recognition. 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 (Papanikolaou et al. [25]). The Index of Learning Styles 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 do have a preference regarding their interaction, but no obvious link between style and approaches offered, was detected to investigate methods for online recognition of LSs.
Recently, results regarding online LS estimation in asynchronous e-learning system appeared either based on a Bayesian network application (Botsios, Georgiou, & Safouris [3]) or using a formal FCM schema (Georgiou & Makry [9]). Both of these works are also based on Kolb's LSI (Kolb, [18]). Instead of using a static questionnaire to estimate the learner's LS, authors in the first work implemented the FIAA and a Probabilistic Expert System. Taking into account the structure of Kolb's LSI, 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 result to 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, that hinder an accurate estimation, is diminished. Their system gives a "clear" LS estimation (no "grey" estimation areas), making the results of practical use in an AEHS. In this paper we take advantage of the FIAA application, to avoid the impacts of wrong answers.
Another approach tend to pursue adaptation according to generated user profile and its features which are relevant to the adaptation, e.g. the user's preferences, knowledge, goals, navigation history and possibly other relevant aspects that are used to provide personalized adaptations (Milosevic et al. [21]). Researchers discuss lesson content's design tailored to individual users by taking into consideration user's LS and learning motivation. They relied on the Kolb's learning style 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 Learning Styles are WHURLE, (Moore, Brailsford, & Stewart [23]; Brown & Brailsford [4]) and ILASH (Bajraktarevic, Hall, & Fullick [2]). 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 by Carver Jr et al [6]. By tailoring the presentation of material to the student's LS, authors believe students learned more efficiently and more 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 Learning Styles. In iWeaver the Dunn & Dunn model is applied (Wolf [29]).

Petri Nets is a formal and graphical appealing language which is appropriate for modelling systems with concurrency and resource sharing. Petri Nets has been under development since the beginning of the 60'ies, where Carl Adam Petri defined the language. It was the first time a general theory for discrete parallel systems was formulated. Historically speaking the Petri net has its origin in Carl Adam Petri's dissertation [26] 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 [15]

The language is a generalization of automata theory such that the concept of concurrently occurring events can be expressed. Since those days, Petri nets 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 explains, as a graphical tool Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams and networks. Petri nets have been proposed for a variety of applications. They can be applied to any area or system that can be described graphically like flowcharts and the needs some means of representing parallel or concurrent activities. Although one can easily find a wide portion of Petri net applications, very limited efforts have been put so far to Web-based adaptive learning applications. Pioneering work may be considered the work of Gao S. and Dew R.[11], Authors propose a high level colored timed Petri Net based approach to providing some level of adaptation for different users and learning activities. In a related way Gao, S., Zhang, Z., Wells, J. and, Hawryszkiewycz, I.[12] proposed a high level timed Petri Net based approach to provide some kinds of adaptation for learning activities. Examples were given while explaining ways to realizing adaptive instructions. To the best of authors' knowledge, Petri nets have not been applied so far to Learning Management Systems (LMS) for adaptive course content presentation.

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 Learning Styles. 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 (Felder, Silverman, [8]). Learning behaviour has been also extensively examined in cognitive psychology. There exists a great variety of models and theories in the literature regarding learning behavior and cognitive characteristics i.e.

The issue of learner's LS estimation in the scope of providing tailored to his / her educational needs, has been addressed in the literature several times. 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. The elements 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 4 core elements and use them to illustrate and discuss activities in different teaching and learning environment. The model is represented in a two dimensions graph, as shown on 0.

The preference is diagnosed by analyzing subject's responses in a number of appropriate questions. A wide range of LS Inventories (LSI) and related questionnaires have been proposed to be serving as LS recognition tools. The LSI has been the subject of analyses by (Willcoxson & Prosser, [28] Yahya [30], Loo & Thorpe [19] and Loo [20]). Their findings gave 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), which are 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, that can reveal the weak and strong points of learning. As noted by Brusilovsky in his 2001 work, 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 (Brusilovsky [5]).Since then efforts have been made and a quite large number of surveys have been published that remark the benefits of adaptation to LS.

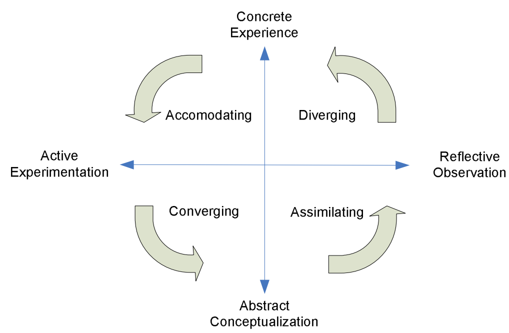

Figure 1: D. Kolb’s learning cycle

To estimate a user’s LS Kolb introduced an LS inventory. Learners respond to 8 items, each of them contains four statements. The four statements appear in every possible combination of two, and the student has to choose one out of the pair. That is in every item appear 6 pairs to work with. After the 48 answers have been given, the educator uses a two dimensional schema to point out the leading LS that expresses better the learner’s cognitive preferences. In some cases there may not be clear the leading LS, as the final score shows a style in between two adjacent styles in the learning style 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 that is a schema representing the four factors that have a decisive role to play in learning, certain probabilistic or fuzzy techniques have been introduced to the purpose of making the computer capable in recognizing user’s LS (Georgiou and Makry [9], Georgiou and Botsios [10], Botsios e.a.[3]). 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 far as it makes use of the system knowledge, i.e. the LS estimation of previous users. 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.

Petri nets

A Petri net is a particular kind of directed graph, together with an initial state called the *initial marking* M_0, in which information flow is depicted by a flow of *tokens* or *markers* which are simply a conceptual means of holding a place in the graph. The underlying graph *N* of a Petri net is a directed, weighted, bipartite graph consisting of two kinds of vertices, called *places* and *transitions*, where edges are either from a place to a transition or from a transition to a place. A *marking* (state) assigns to place *p* a nonnegtive integer *k* that is interpreted as the place *p* is marked with *k tokens*. Pictiorally, we place *k* black dots in place *p*. A vertex representing a *place* in the graph is known also as a *condition* and is represented as a circle. Vertices that represent *transitions* are represented as parallelograms. Pairs of two consequtive 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 the occurrence of a transition that is called *event* (Figure 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 exists 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 which 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 (Figure 2).

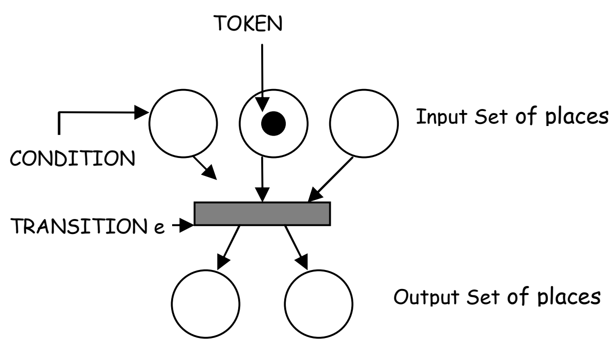

Figure 2

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 Petri net is the set of conditions which 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 place can be in *conflict* if both events are enabled at the same time. If there exist 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 which 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 which contains illegal combinations of inputs.

One can see that in order to effectively utilize a Petri net for speed independent design, one 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. Petri nets
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 Petri net, 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.

P-Timed Petri Nets

P-timed Petri Net is an extension of traditional Petri Nets (Murata T., [24]) when used to describe the temporal behaviour of a target system. For instance, in a P-timed Petri Net, if one time attribute is associated with place, the firing rules are that a transition is enabled after tokens deposited in its input places 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 places 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 places 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 places to output places. If the transition cannot fire, the token becomes unavailable. This “dead end” should be avoid by setting appropriate (t_{min}, t_{max}) and adjusting them dynamically.
By building a P-timed Petri Net based learning model to present learning material tailored to each student’s needs one can be benefiting from dymanic executive semantics of the Petri Net and consequently obtain a powerful adaptation model.

3.1 P-Timed Petri Net 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 have those expressing the learning profile of the user to whom the LO is addressed. Learning profile has a number of components such as Learning Style, Learning behaviour, or current emotional state. Students differ in Learning Style and have different learning abilities, which results in different learning paths and time consumption. To provide personalized learning instructions, the system should be able to adjust state attributes to adapt to individual's learning preferences and behavior. For example, the system using sophisticated techniques presents LOs tailored to the user's learning style or displays supplementary materials after assessment (if needed), or reduces homework if students do not need it. This results to a system capable of presenting to students personalized interfaces with LOs built in the best possible way for each learner.

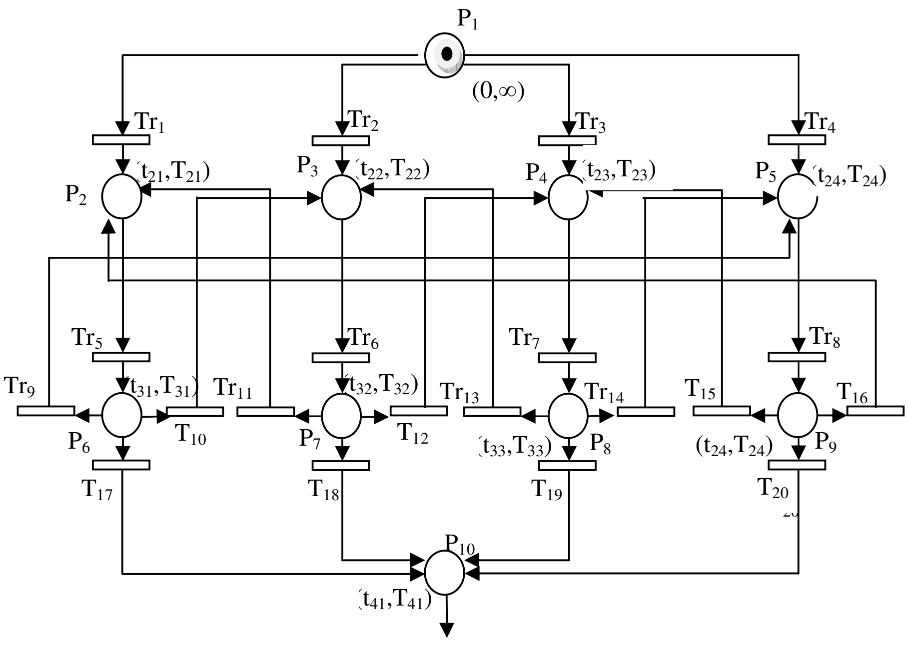

Figure 3

model (P-TPN) for adaptive presentation (in terms of Learning style) of learning material, we first map learning state to the conditions. To start with, at the first place the user is called to respond at the Kolb's Learning Style Inventory. Based on given answers the system recognizes the user's LS. For the place Figure 3

To build a P-timed Petri Net based adaptation P_1 the predefined maximum time is as long as a user can ask. The proposed PN operates sequentially: A fired token in P_1 follows a path that ends at P_{10}. Lerner responds to the Kolb's inventory and the system applies methods described in [3] and [9], recognizes the LS and the token from P_1 is fired to one of the places P_2, P_3, P_4 or P_5, according to the transition that represents estimated LS. This group of places contains introductory course presentations tailored to the learning preferences of the four different Kolb's LSs. They also contain questions on the presented concepts to be learned and related problems. To each of the places P_2, P_3, P_4 and P_5, a predefined minimum time indicates the minimum time the user may spend working one the learning material of the place. Both the minimum time t_{min}, and the maximum time t_{max} may change dynamically according to a simple probabilistic expert system's outcomes. This system, takes into account the time each new user needs to accomplish the tasks in the place.

As soon as the user finishes working in the place P_i, $i \in \{2,3,4,5\}$ where is the current location of the token, or if the maximum time expires the adjacent transition fires the token forward to the corresponding place P_6, P_7, P_8, or P_9. At this places evaluation of the knowledge gained at P_2, P_3, P_4 or P_5, is taking place. According to the results of this evaluation one of the transitions Tr_i, $i \in \{9,10,...,20\}$will be fired. At this level, transitions fire forward if the assessment of the work done by the learner is above a certain level. In case of incomplete work the token fired to the place that contains learning material adjusted to the LS estimated as being closer to the leading one. It is expected that this second, but different presentation of learning material at this stage will add value to the learning process. In such a case, there will be a forward firing to P_{10}. For example, if a learner's LS estimated as Diverger (P_2) with a score of 28 % followed by e second best estimation "Assimilator" (P_3) , and moreover this learner evaluated as "incomplete" at this stage of the course, the corresponding to Diverger's place will fire the token backward to Assimilator's LS Interface. The token now is forced to be fired to P_3 . At this place complementary pieces of information appear and moreover general information regarding the next session of the course are supplied. Here, the LOs are presented adaptively to Assimilator's learning preferences.

3.2 Firing Rules of P-Timed Petri Net Based Adaptive Learning Model

In the proposed P-TPN model, when a place contains a token, its contents are accessible to the user/student. Let us denote t_{min} by t_{ij} *,and* t_{max} *by* T_{ij} $i, j \in I$. Before t_{ij}, $i, j \in I$ no other but the *i* interface is reachable. Learners are forced to concentrate on current learning state as it appears on *Int i/j* $i \in I$, $j \in \{1,2,3,4\}$, and for a time length equal to $\left|T_{ij} - t_{ij}\right|$. As soon as t_{max} expires the adjacent enabled transition will be chosen automatically. The predefined transition priorities usually reflects users' learning profile. For instance, in Figure 3, place *INT 1* appears on the screen as far as the token placed in it. Its four links through LS_1, LS_2, LS_3, or LS_4 to are immediately reachable due to the 0 value of t_{min} but only one will be stimulated according to the LS estimation of the user taken place at Int 1.
If *LS3* is selected, *Int 1* remains visible on the screen as far as the user works on it, and *Int 2,3* becomes visible consequently. Student can stay in *Int 2,3* for at most $|T_{23} - t_{23}|$ time units before is forced to leave for the next place. At Int 3, students are required to spend's at least but only one will be stimulated according to the LS estimation of the user taken place at Int 1.$|T_{31} - t_{31}|$ time units. To activate the best fitted to the user LS *As 2/i*, $i \in \{1,2,3,4\}$ the system takes as minimum time t_{31} equal to the time the student leaves the place Int 2/i, $i \in \{1,2,3,4\}$. The maximum time T_3
Is now equal to $t_3 + T$, for a predefined T by the authors of the course.

in related delay intervals, consequently resulting in "dead end". This problem could be solved by agents which monitor students' status and adjust delay pair values dynamically.

PN simulation

Those who willing to simulate the proposed P-timed PN may find several PN simulators in the world wide web. The "S/T Petri-Net Simulation System" [1] applet below is Written by Thomas Bräunl et al. The applet was written as part of a sofware project with the purpose of implementing a simple simulation system (PNS) for extended place/transition Petri nets.

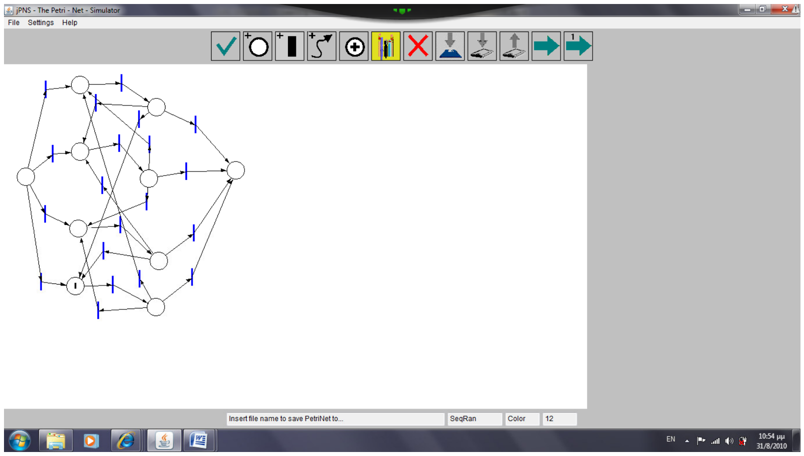

Figure 4

An example of such simulation appears in figure 4. The PN has been set to run sequentially, and in the choice of transition to be enabled is random. For the needs of our system, the choice of transitions will be based on user's assessment.

Future work

The authors of this paper look forward to implement the described P-timed PN. Furthermore, this PN will be assessed in an AEHS that is under construction.
It is also intention of the authors to generalize the system's operation in parallel operation , to the purpose of handling simultaneously huge number of users.

References

1 Bräunl T. http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/java/Braunl/

2 Bajraktarevic, Hall, & Fullick., http://www.ascilite.org.au/conferences/melbourne08/procs/willems.pdf 2003

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

4 Brown & Brailsford **,** Reappraising cognitive styles in adaptive web applications, Proceedings of the 15th international conference on World Wide Web Edinburgh, Scotland, Pages: 327 - 335 2006

5 Brusilovsky, P., Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) **11** (1/2), 87-110, 2001

6 Carver C.A. Jr., Howard R. A., & Lane W. D., Enhancing student learning through hypermedia courseware and incorporation of student learning styles, IEEE transactions on Education, Vol. 42 (1),pages 33-38, 1999

7 DeVillers I. R. and Louchard G. , Realization of Petri Nets without Conditional Statements *Information Processing Letters* **2** (1973) 105-107. North-Holland Publishing Company.

8 Felder, R. M., & L. K. Silverman. 1988. Learning and teaching styles in engineering education. *Engineering Education* 78 (7):674-681

9 Georgiou, D. A., & D. Makry. 2004. A learner's style and profile recognition via fuzzy cognitive map. Paper read at Advanced Learning Technologies, Proceedings. IEEE International Conference on Advanced Learning Technologies, 2004.

10 Georgiou, D. A., & S. D. Botsios. 2008. Learning style recognition: A three layers fuzzy cognitive map schema. Paper read at IEEE International Conference on Fuzzy System

11 Gao S. and Dew R., 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 LNAI, 2007, Pages 177-185
12 Gao, S., Zhang, Z., Wells, J. and, Hawryszkiewycz, I. T., 'Supporting adaptive learning with high level timed Petri nets', International Journal of Intelligent Systems Technologies and Applications, Volume 4, pages 341 - 354Number 3-4 / 2008
13 Georgiou, D. A., & D. Makry. 2004. A learner's style and profile recognition via fuzzy cognitive map. Paper read at Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on.
14 Graf & Kinshuk, Learning styles and cognitive traits - Their relationship and its benefits in web-based educational systems, *Computers in Human Behavior* Vol. 25 (6) , Pages 1280-1289, 2009
15 Holt A., Information system theory project, 1968. Dept.of Commerce, Clearinghouse, Springfield, VA 22151.
16 Holt, A.and Commoner, F., Events and Condition, .3 Vols. *Applied Data Research*, Wakefield, Mass., 1968.
17 Kolb, D. A. (1984). *Experimental learning: Experience as the source of learning and development*. Jersey: Prentice Hall.
18 Kolb, D. A. (1999). *Learning Style Inventory - version 3: Technical Specifications*: TRG Hay/McBer, Training Resources Group.
19 Loo R, Thorpe K., Confirmatory factor analyses of the full and short versions of the Marlowe-Crowne social desirability scale, *Journal of Social Psychology***,** Vol. 140, Issue 5, Pages 628-635, 2000
20 Loo, R., The Distribution of Learning Styles and Types for Hard and Soft Business Majors, *Educational Psychology*, **22**: 3, pages 349 — 360, 2002
21 Milosevic, D., Brkovic, M., Debevc, M. & Krneta, R., Adaptive Learning by Using SCOs Metadata. Interdisciplinary Journal of Knowledge and Learning Objects, 3**,** 163-174. 2007
22 Misunas D., Petri nets and speed independent design**,** Communications of the ACM, Volume 16 , Issue 8 (1973), Pages: 474 - 481
23 Moore, Brailsford, & Stewart Personally tailored teaching in WHURLE using conditional transclusion, Short Paper, 12th ACM Conference on Hypertext and Hypermedia. Århus, Denmark, August 14-18, 2001.
24 Murata T., Petri nets: properties, analysis and applications, *Proceedings of the IEEE*, Vol. 77, Issue 4, April 1989, Pages 541-580
25 Papanikolaou, K.A., Mabbott, A., Bull, S., Grigoriadou, M.: Designing learner-controlled educational interactions based on learning/cognitive style and learner behaviour. Interacting with Computers 18, 356-384 2006.
26 Petri C. A., Communication with Automata, New York: Griffiss Air Force Base. Tech. Rep. RADC-TR-65-377, vol. 1, Suppl. 1, 1966.
27 Specht M, Oppermann R., ACE - Adaptive Courseware Environment. The New Review of Hypermedia and Multimedia, Vol 4 (1), pages 141 -161, 1998
28 Willcoxson, L., & Prosser, M., Kolb's learning style inventory (1985): Review and further validity and reliability. *British Journal of Educational Psychology*, *66*, 247–257.1996
29 Wolf, C.: iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education. Proceedings of the Fifth Australasian Computing Education Conference on Computing Education 2003, Vol. 20 (2003) 273-279
30 Yahya, I. (1998). Willcoxson and Prosser's factor analyses on Kolb's (1985) LSI data: reflections and re-analyses. British Journal of Educational Psychology, 68, 281–286. 1998