Adaptive vs. Non-adaptive e-Learning Systems – a Petri Net-based Evaluation Approach

Emilija Spasova Kamceva[1] and Pece Mitrevski[2]

Abstract – **This paper deals with the principles of specification, modeling and implementation of an educational information system. The behavior in a sample e-learning application is described using a formal model, based on Petri Nets. An adaptive model of student's transitions during the course of an e-learning cycle is proposed and a comparison with non-adaptive systems is made. Depending on the type of student, learning style, level of knowledge and dynamic behavior, a general conclusion is drawn that the students spend more time in non-adaptive vs. adaptive e-learning systems.**

Keywords – **Adaptive E-learning system, Petri Nets, modeling, simulation.**

I. INTRODUCTION

Education theories argue that different students use different strategies in the learning flow and demonstrate different adjustments when using the learning materials. Similarly, results indicate that learning styles can be identified on an individual basis, and the adaptation to build a personal style increases the efficiency of learning in some students [1].

Very often, web based e-learning systems manifest some technical problems, e.g. in case if the corpus of learning materials is too burdened, or some content data are lost, etc. Adaptive e-learning has the purpose to solve the problems of understanding the learning content and disorientation of the students, i.e. to change with some user adaptive methods, which optimize the learning material and decrease the time spent for the learning process [2]. These systems generate intern representation for every student. For example, personal characteristics, purposes and knowledge are taken into account. In the past, with decades, different strategies of how students can adjust to learning were developed, like learning materials modification, adaptation of the learning content, etc.

Chapter 2 introduces math formulas which help to determine the efficient learning time which the students spend in a course, the time passed in a given state, the time that a student spends while searching in some unit, and the time when the student is in answering or testing phase. Later on, in Chapter 3, a Petri Net model of a sample course is presented, which provides a student to decide the style of learning of the course and, based on his/her previous knowledge, to decrease/increase the required time to pass the course.

[1]Emilija Spasova Kamceva is with the Faculty of Informatics, FON University, Bul. Vojvodina bb, 1000 Skopje, Republic of Macedonia, E-mail: emilija.kamceva@fon.edu.mk

[2] Pece Mitrevski is with the Department of Computer Science and Engineering, Faculty of Technical Sciences, St. Clement Ohridski University, Ivo Lola Ribar bb, 7000 Bitola, Republic of Macedonia, E-mail: pece.mitrevski@uklo.edu.mk

II. THE FOUNDATIONS OF AN ADAPTIVE METHOD

One of the aims of this paper is to identify the states in which an e-student can be found and to estimate the average time that a student spends in a particular state. Initially, we use graph representation to describe the method, following by a Colored Petri Net-based model. The places in the Petri Net are titles, subtitles, exams and examples, while the colored tokens represent students.

In an adaptive method, learning style is checked in every node, and the path is built for each student. The next node is chosen according to the previous level of knowledge and the points obtained (scores). For example, in the Petri Net, tokens with time stamp which is equal to time of response are used. It is necessary to calculate the time of searching which student spends in a unit, the time needed the student to make estimation and the time of remaining in the queue.

The time LT is the time of arrival of the next student and it is calculated by Poisson distribution (Eq. 1):

$$LT = \frac{e^{\lambda t}(\lambda t)^n}{n!} \tag{1}$$

where t is time of waiting of the student in the queue before the start to use the system, λ is number of arrivals and LT is used to calculate the time when the student will arrive, when n=1.

The time of searching through a given learning unit is calculated by Normal distribution (Eq. 2):

$$BT = \frac{e^{\frac{-1}{2}\left[\frac{AVG_B - \alpha}{\delta}\right]^2}}{\sqrt{2\pi}\delta} \tag{2}$$

where AVG_B refers to the average length of the time in which the student remains in the learning unit (lesson), α is variation of the spent time among the students and δ is the standard deviation of the spent time of learning among the students.

The time when the student is into a state of answering or testing is also calculated by Normal distribution (Eq. 3):

$$BT = \frac{e^{\frac{-1}{2}\left[\frac{AVG_A - \beta}{\rho}\right]^2}}{\sqrt{2\pi}\rho} \tag{3}$$

where AVG_A is the average time of testing, β is the variation of the spent time in the node of testing among the students, ρ is the standard deviation of the spent time in node of testing among the students.

The score which the student is obtaining is again calculated by Normal distribution (Eq. 4):

$$Score = \frac{e^{\frac{-1}{2}\left[\frac{AVG_S - \mu}{\gamma}\right]^2}}{\sqrt{2\pi}\gamma} \quad (4)$$

where AVG_S is the average score, γ is the standard deviation, μ is the variation of the score as a result of learning and testing among the students.

In terms of colors used for representing the user characteristics, we need some additional colors for calculating the time of response [3]. These colors are the following: the time of arrival of the student, the total sojourn time of the student into the system (initialized to zero at the start), the path of the student, and the time which elapses on every unit by the student.

III. A SAMPLE COURSE AND THE PROPOSED MODEL

The sample course of the subject Calculus 1 will be displayed as a whole (Fig. 1). In this particular course, there are two chapters (content 1, content 2), two sub-contents (content 1.1 and content 1.2), seven examples, introduction and conclusion. The graph of the course of the subject Calculus 1 is represented on Fig. 2.

This graph also can be represented using a Petri Net (Fig. 3). It is necessary to declare three base colors for the CPN tool: 1) for the style of learning, 2) for the level of learning, 3) for the score. Additionally, we need four new colors: 1) for LT – the time of arrival, 2) for the path of traversing of the students, 3) the time of learning of a single unit, 4) the time of the learning process which represents the total time of learning that the student spends in the system. Once we have defined all the characteristics of the student (i.e. "the colors"), we define a student as a mix of colors from all the abovementioned characteristics [4,5]. Also, we are going to define the set of colors called "students" which we will use for managing the FIFO queue when they arrive in the system.

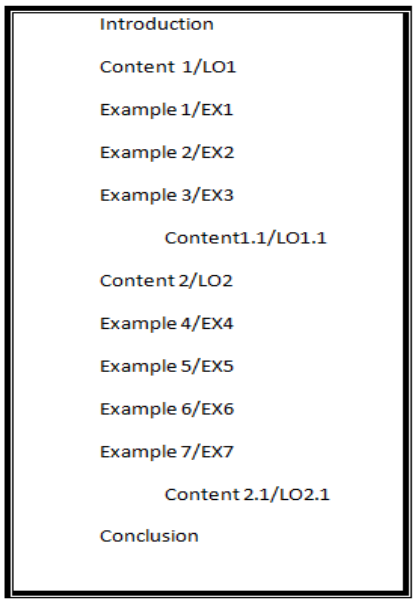

Fig. 1 The structure of a sample course (Calculus 1)

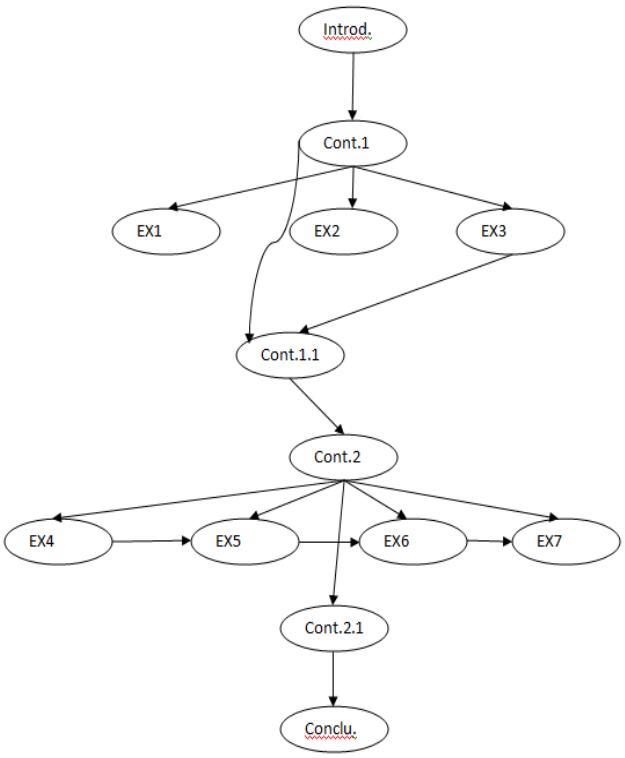

Fig. 2 Graph of the sample course structure (Calculus 1)

As the new students arrive, the *newLearner()* function is used for creating a token representing the new student. When it is called, it returns the set of colors *STUDENT* (Fig. 4). The first component represents the type of the learning style and uses the *Learning_Style.ran()* function, which is randomly generated. The second component is the level of knowledge and it uses the function *Knowledge_Level.ran()*.The third and the fourth component are the scores from *content 1* and *content 2* of the book. The fifth component is the time of arrival and the sixth component is the sum of the total time which the student spends in the system. The seventh component is the path of the student, while the eight component is the time which the student passes in each learning unit (Fig. 5).

All the students should pass the introduction and *content 1*, as a suggestion of the course teacher. Because *content 1* is a chapter with a test, according to the score and the level of knowledge the student can decide from 4 different paths to continue with learning (because in the first process the decision is based solely by the style of learning, now we have two factors which impact the decision, so we have 2^2 number of paths). After that, the student is returned to the page with *RWStyle*. In this part (Fig. 6), if the level of knowledge is low and the score is less than 10, the student should pass through *example 1*, *example 2*, *content 1.1 (LO1_1)*, respectively. As soon as they pass *LO1_1*, all the students (four in total) should pass through *content 2*. For that purpose, the students go on the next chapter *LO2*. In the subpart of *LO2*, firstly the decision for the next path is made based onthe level of knowledge and the score from *LO1* [6].

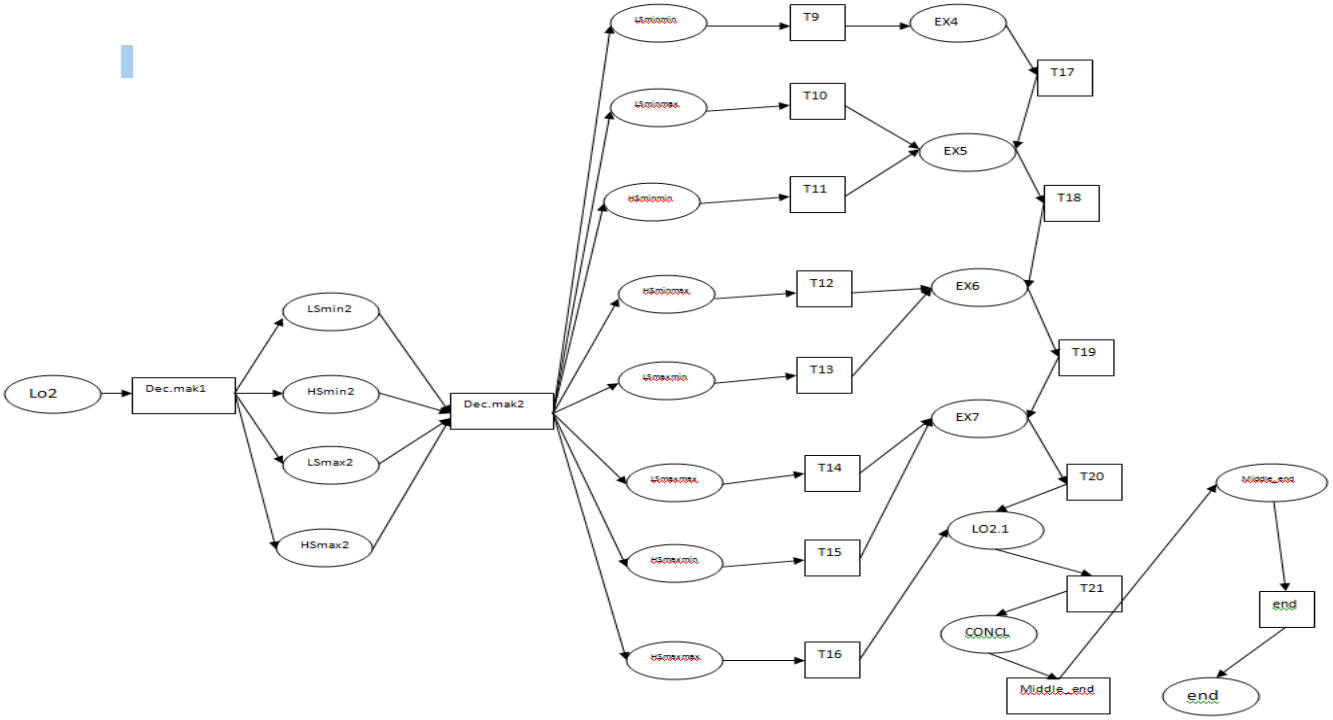

Fig. 3. Petri Net presentation of the sample course structure (Calculus 1)

Fig. 4. Declaration of the set of colors

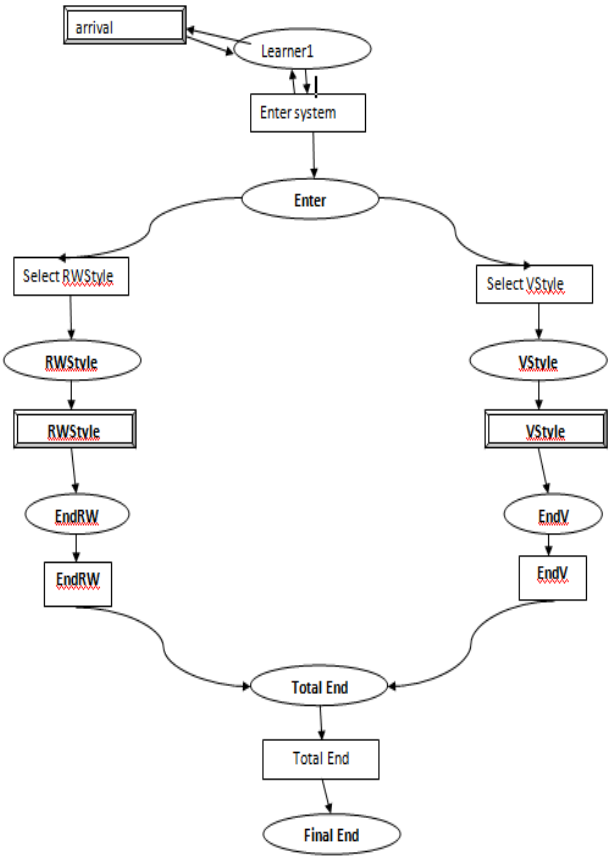

Fig. 5. The graph of the main page of the sample course

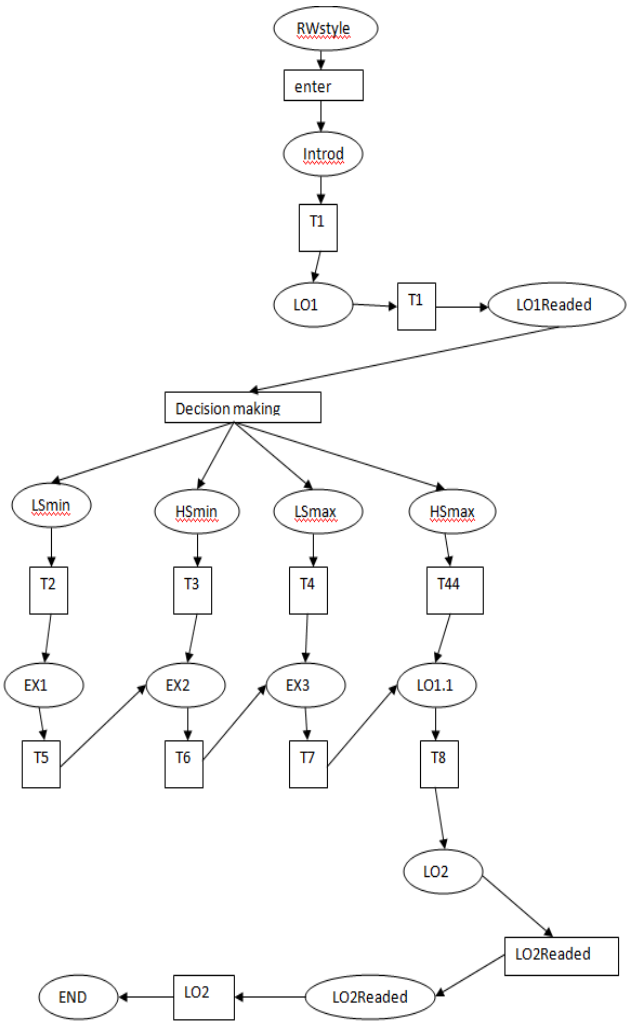

Fig. 6. RWstyle subpages

For the evaluation of the efficiency of the proposed model, we should calculate the time of response for the two types of users and to compare it with the time of response of the students which do not use an adaptive system. We should underline that, in the adaptive system, the students choose the learning units, which is not the case in a non-adaptive system.

We want to illustrate that if the user has a higher level of knowledge, he/she does not need to pass some of the units of the course, and will arrive at the end in shorter time. Otherwise, the user should pass through additional contents, which directly translates in longer sojourn time.

If the user has lower level of knowledge and did not pass the test successfully, then he should pass the course for shorter time, but he/she can also skip some units. For calculating the time of response, the conditions for intercrossing are removed and they allow the student to choose arbitrary paths without restrictions.

If we summarize the results of the two types and the time of the response of both the adaptive and non-adaptive systems, a conclusion could be drawn that the time of response of adaptive systems is shorter than the time of response of non-adaptive systems.

IV. Conclusion

Nowadays, with the fast development of the technology and the Internet, the traditional way of education is more or less abandoned, giving its way to the e-learning paradigm. An e-learning system is a web based environment where all the individuals have the access, willing to learn and expand their knowledge. In an e-learning system, all the coursework can be accessed in some of the available formats: text, audio, video, photography, presentations, tests, etc. A plenty of e-learning systems exist, based on different technologies, and they use different algorithms for evaluating the efficiency of the system. Among all of them, adaptive and dynamic systems appear to be the best choice. In our current research efforts, we tried to investigate and prove that these systems exhibit the best time of response.

The main characteristics, which were included in the research, are: the learning style, the level of knowledge and the score. These are the main characteristics, based on which the efficiency of a given system is evaluated. As a next research step, we will focus on calculating the transition probabilities, in order to compare the theoretically obtained results and the simulation results obtained by employing the class of Deterministic and Stochastic Petri nets (DSPNs) [7].

References

[1] K.L. Rasmussen and G.V. Davidson-Shivers, Hypermedia and learning styles: can performance be influenced?, Journal of Educational Multimedia and Hypermedia 7 (4), pp. 291 308, 1998.

[2] Kamceva, E., Mitrevski, P., “On the General Paradigms for Implementing Adaptive e-Learning Systems”, ICT Innovations 2012, Web Proceedings, pp. 281-289, Ohrid, Macedonia, 2012.

[3] Y. C. Chang, Y. C. Huang and C. P. Chu, “B2 Model: A Browsing Behavior Model Based on High-Level Petri Nets to Generate Behavioral Patterns for E-Learning”, Expert Systems with Applications, Vol. 36, No. 10, pp. 12423-12440, doi:10.1016/j.eswa.2009.04.044, 2009.

[4] K. Jensen, L. M. Kristensen and L. Wells, "Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems," International Journal on Software Tools for Technology Transfer, Vol. 9, No. 3-4, pp. 213-254, doi:10.1007/s10009-007-0038-x, 2007.

[5] A. Ratzer, L. Wells, H. Lassen, M. Laursen, J. Qvortrup, M. Stissing, M. Westergaard, S. Christensen and K. Jensen, "CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets," Proceedings of the 24th International Conference on Applications and Theory of Petri Nets, Eindhoven, pp. 450-462, 2003.

[6] F. Omrani, A. Harounabadi, V. Rafe , ”An Adaptive Method Based on High-Level Petri Nets for E-Learning” –Journal of Software Enqineerin and Application (JSEA), No. 4, pp. 559-570, 2011.

[7] H. Choi, V. G. Kulkarni, K. S. Trivedi, “Transient analysis of deterministic and stochastic Petri nets”, Application and Theory of Petri Nets, Lecture Notes in Computer Science 691, Springer, Berlin, pp. 166–185, 1993.