Modelling Adaptive Distance Learning Course Using Petri Nets

Pavlina Vladimirova[1] and Daniela Ilieva[2]

Abstract – **This paper considers the modelling of interactive education and adaptation process. A model of interactive adaptive tutoring course, using coloured Petri nets is offered. The transitions are connected with predicate functions, which give the rules for firing the transitions.**

Keywords – **modeling, Petri nets, learning course, adaptive**

I. Introduction

Distance education has opened the gates of "Learning Anytime and Anywhere". The constantly increasing requirements to these courses and cost of the development and deployment is a fact, therefore it is important their preliminary design and testing. Some of the important requirements for an e-learning course are:

• Adaptability - the ability to provide educational material for each student individually;

• Possibility of restructuring - the updating of the product to be convenient and law costing. It is often necessary because of change and development of the topics or improving the quality of the course;

• Ability to research - it is very important to examine the training system in the development, implementation and operation in order to improve its quality and learning outcomes;

The mathematical model description of the learning course is essential to meet these requirements.

II. Colored Petri Nets

CPN (Colored Petri Net) have much more modelling power and have better structuring facilities as types and modules. CPN are hierarchical because they contain facilities for representing a model as a hierarchical structure. CPN use basic data types that can be used for building composite data types.

Colored Petri net is defined as a tuple [1]

CPN = (Σ, P, T, A, N, C, G, E, I) satisfying the requirements:

Σ is a finite set of non-empty types, called color sets;
P is a finite set of places;
T is a finite set of transitions;
A is a finite set of arcs such that

$A \subseteq (P \times T) \cup (T \times P),\ P \cap T = P \cap A = T \cap A = \varnothing;$

N is a node function, where N: $A \rightarrow (P \times T) \cup (T \times P)$;
C is a color function. It is defined as C: $P \rightarrow \Sigma$;
G is a guard function. It is defined from T into expressions such as:

$$\forall t \in T : [Type(G(t)) = Bool \wedge Type(\ Var(G(t))\ ) \subseteq \Sigma]$$

E is an arc expression function. It is defined from A into expression such as:

$$\forall a \in A:[Type(E(a)) = C(\ p(a))_{MS} \wedge Type(Var(G(t)) \subseteq \Sigma],$$

where p(a) is a place of N(a);

I is an initialisation function. It is defined from P into closed expression such as:

$$\forall p \in P:[Type(I(p)) = C(p)_{MS}\ ] .$$

CPN provides typing data (color sets) and sets of values from specified types for each place. The expression E (p, t) is the variable name associated with an incoming arc from place p to transition t, and the expression E (t, p) is associated with transformation (action), which transition t performed on incoming to it data to obtain outcome data for the place p. The so-called guard is a logical expression associated with transition t, which must be true to enable transition t.

Consideration and modeling of the system in terms "condition-event", allows visualization of the passage of the training course, possibility to study the dynamics and the different properties of the modeling system and establishing the simulation algorithms. Colored Petri nets allow introducing additional conditions for activation of the transitions, to link places with certain data types and tokens to be associated with some of these tips.

III. Formulation of the Problem

Let us consider a course that consists of a number of modules (m), corresponding to the topics of the course. Each of these modules covers a certain amount of theoretical and reference material on the subject and problems for independent work. To design adaptive learning course on the volume and complexity of the material is required for each of the modules to develop several levels of complexity. Experience shows that it is appropriate to develop three levels of complexity:

1st level - covering the minimum required to pass the course;

[1]Pavlina St. Vladimirova is from Technical University – Varna, Faculty of Computer Sciences and Technology, 1 Studentska str., 9010 Varna, Bulgaria, E-mail: pav_varna@yahoo.com

[2]Daniela D. Ilieva is from Technical University – Varna, Faculty of Computer Sciences and Technology, 1 Studentska str., 9010 Varna, Bulgaria, E-mail: ilievadaniela@abv.bg

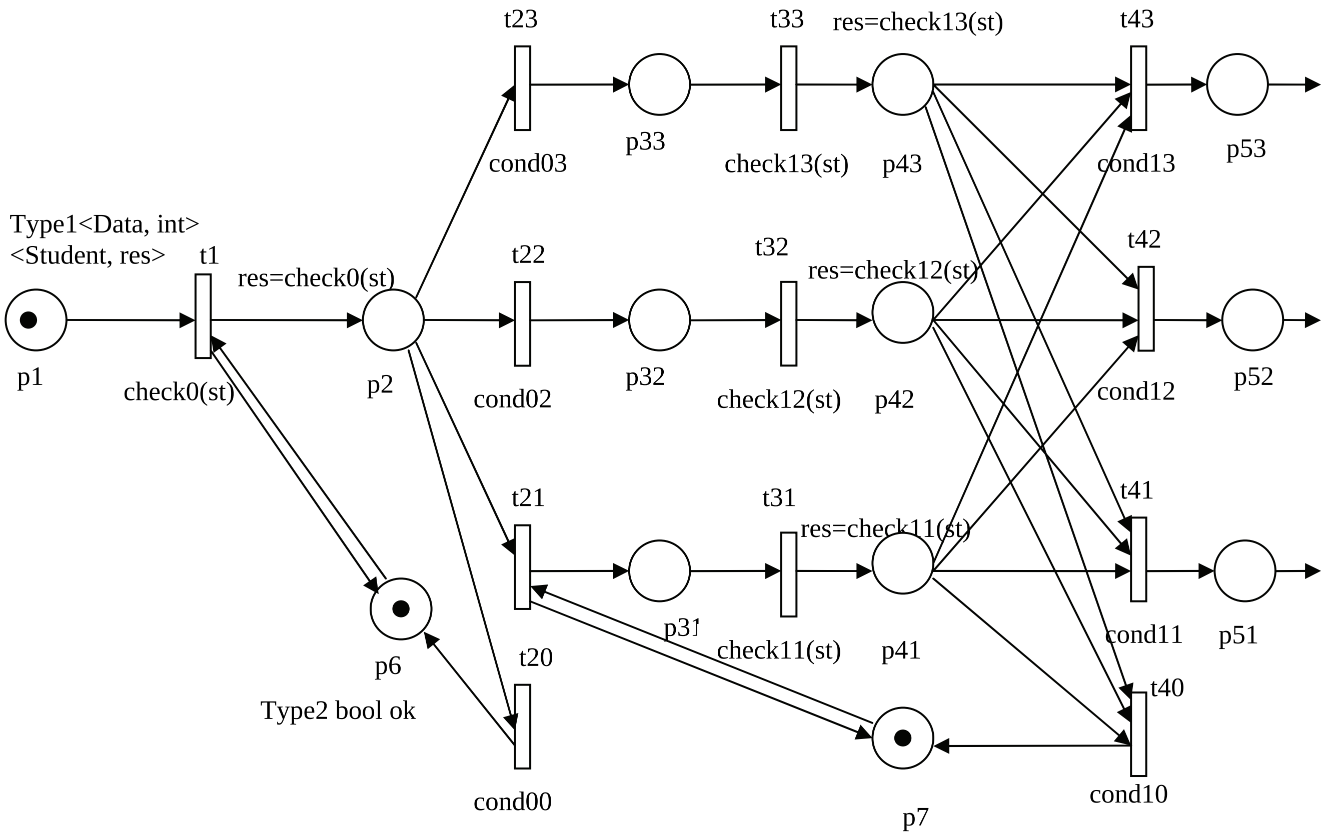

Fig.1 Model of interactive adaptive learning system

2nd level - covering 80% of material aimed at good or very good students;

3rd level - for excellent students, where the basic material is added with additional practical problems, requiring in-depth and analytical thinking.

For each of these components and their levels is necessary to develop tests (questions, problems), checking the degree of understanding the material. Regardless of the initial level of complexity, the student is able to change it depending on the results of tests on the previous modules. Adaptability of the course lies on the fact that every student can choose a suitable path of training, i.e. individual sequence of modules, depending on the recorded knowledge and skills.

IV. INTERACTIVE MODEL OF ADAPTIVE LEARNING SYSTEM WITH COLORED PETRI NETS

Model of interactive adaptive learning system is given on Fig.1.

Places p_i correspond to the different states of the student during his passage through the training course, which most often are: ready to start a new training module, completion of a module and a readiness to verify his knowledge on this module, completing the verification of knowledge and readiness to continue training the same or another level.

Transitions t_i correspond to the student activities such as: training on specified module, testing of knowledge, conducting additional training. Some transitions are associated with predicate functions and activation of the transition is executed only if the predicate function is true and if tokens are available in all input places.

Description of the developed model training system:

All the places p_i can store tokens (data) of type ordered pair (Student, res), where Student is a class/data structure for student and res is a variable of integer type holding current result of the reported knowledge of the student. Student class stores the following data:

```
class Student
{string name; / / name of student
string facnumber; / / faculty / ID number
string speciality; / / speciality
string path; / / path of the course
int totalresult; / / total score for knowledge of  the course
};
```

Places p6, p7 can store and data logical type {1,0}.

p1 - ready for starting the course and passing the entrance test reviewing the necessary basic knowledge for this course;

p2 - ending of the entrance test and moving to training on the material of module 1;

p3i - completion of the i-th level of the module 1 and readiness to check knowledge on it (i = 1, 2, 3);

p4i - completion of test 1 and ready to start module 2 (i = 1, 2, 3);

p5i - completion of the i-th level of module 2 and readiness to check knowledge on it (i = 1, 2, 3);

p6, p7 - completion of additional training to the last module and a readiness to re-pass the last test. In these places there is always a token which is constant readiness for further training;

t1 - execution of entrance tests, verifying basic knowledge required;

t2i - learning the material of module 1, level i (i = 1, 2, 3);

t20 - additional training due to insufficient knowledge for starting the course;

t3i – executing the test 1 checks knowledge on module 1, level i (i = 1, 2, 3);

t4i - learning the material of module 2, level i (i = 1, 2, 3);

check0 (st) - function related to transition t1, forming the value of variable res;

check1i (st) - function related to transition t3i, forming the value of variable res;

cond01 - logical condition (guard) associated with the transition t21 to switch to study the material from module 1 level 1 (res $\in$ [40, 60));

cond02 - logical condition (guard) associated with the transition t22 to switch to study the material from module 1, level 2 (res $\in$[60, 80));

cond03 - logical condition (guard) associated with the transition t23 to switch to study the material from module 1, level 3 (res $\in$[80, 100));

cond00 - logical condition (guard) associated with the transition t20 to return to study additional material (res <40);

cond1i - logical condition (guard) associated with the transition t4i to switch to study the material of module 2, level i (i = 1, 2, 3 and corresponding logical conditions res $\in$[40, 60), res $\in$[60, 80), res $\in$ [80, 100));

cond10 - logical condition (guard) associated with the transition t40 to return to study additional material for module 1 (res <40);

The model extends beyond the places p5i by analogy to the network from t2i to p5i, i.e. test on another module follows, with subsequent adaptive guidance to the appropriate level to the next module and these iterations continue according to the number of the modules in the course.

The model allows adding the additional variable of time for the tokens and duration to transitions, which would increase the criteria for analysis and evaluation of the system. Also, the network can be considered as hierarchical, i.e. some of the transitions are represented as macros that the next level can be presented in a more detailed look.

V. Conclusion

1. The developed model of adaptive learning system allows testing, evaluation and quality management of the system during the development, deployment and its operation.
2. Using Colored Petri nets as a mathematical model allows using variables and data accumulation about statistical information of system performance which will be used for analysis and subsequent modifications of the learning course in order to enhance the effectiveness of training.
3. The model allows easy and comfortable transition to program implementation.

References

[1] K. Jensen, “Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use” – New York, NY: Springer-Verlag, Vol.1 – 1996, Vol.2 – 1997, Vol.3 – 1997.

[2] P. Vladimirova, D. Ilieva,” Fuzzy based approach for decision support system”, ICEST 2010, Conference Proceedings, pp. 743-746, Ohrid, Macedonia, 2010.

[3] M. Mitev, P. Vladimirova,” Graph ‑ Based Analytical Approach to Testing Programs”, ICEST 2010, Conference Proceedings, pp. 719-722, Ohrid, Macedonia, 2010.