

Finding Learning Paths Using Petri Nets Modeling Applicable to E-Learning Platforms

Rogério Campos-Rebelo^{1,2}, Anikó Costa^{1,2}, and Luís Gomes^{1,2}

¹ Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal

² UNINOVA-CTS, Portugal

{rcr, akc, lugo}@uninova.pt

Abstract. This work proposes an approach for course modeling using Petri nets. The proposed modeling method can be applied to support development of e-learning platforms (namely learning management systems - LMS) allowing student guidance when considering reaching a specific goal. This goal could be as simple as getting a set of sequential courses (or a degree), or as complex as combining different modules from different courses having different types of dependencies in order to obtain a qualification. Each course is characterized by a set of modules and their relations. Each module is represented by a Petri net model and the module structure representing the course's dependency relations is translated into another Petri net model. Additional courses or modules can be included into the offer as their associated Petri net models can be easily composed using net addition operation. The contribution of this paper foresees the usage of common Petri nets analysis techniques (such as state space analysis, invariants, trace finding) to constraint student's options in order to optimize his/her path to reach a degree or a qualification. A simple example considering a scenario with a few courses and modules is used to illustrate the approach.

Keywords: Learning, Petri Nets, Course, Study Plan.

1 Introduction

With the growth and evolution of the internet, online systems have increased. The education was not left behind and e-learning tools and platforms started to appear allowing managing educational process or even getting qualifications online.

Some examples can be found in the literature, targeted to optimize the e-learning process. As an example, an integrated and unified e-learning model describing e-Learning by using knowledge characterization from five views (course, learner, task, learning process, and learning resource) is presented in [1]. To set up the dynamic mapping between these views (sub-models) the paper presents a set of rules and supporting theorems.

In [2] a new centralized learner model technique applicable for e-learning systems allowing the optimization of the learning process for the individual learning requirements is proposed.

Several examples can be found in the literature where Petri nets were used in the modeling of learning systems. Petri nets are used in systems where the teacher becomes the main element acting at the core of the educational process [3], model a study course [4] or use high level Petri nets to reuse and aggregate learning resources in e-learning [5]. However, no work modeling an entire open system of education was found.

This paper aims to model with Petri nets a complete e-learning system offering several courses and modules (provided by one or several institutions) and supporting guidance to students to find specific paths to accomplish their goals according with their own requirements.

This paper starts with the presentation of their contribution to value creation, followed by a short Petri Net overview. Next, in section 4, the course and their model characterization are presented, including the description of modules and their association. In this section, modeling qualification and their characteristics are also presented. In section 5, the proposed method to create the study plan is presented. Finally, last section concludes and presents future works.

2 Contribution to Value Creation

This work allows a progress in the development of the open educational systems, proposing an online system where the educational offers (courses and modules) are modeled and all different types of users can interact and obtain important information.

An institution can offer their modules, courses, and obtain information about performance of their students, thus being able to reformulate their offers in order to make them more competitive. A student can obtain the information about all available courses and modules, in order to choose his study plan or create it automatically, introducing in the system the features (such as cost, duration, etc.) that suit him. In this sense, the proposed modeling approach can bring added value to all users of the e-learning platform.

3 Petri Nets Overview

Petri nets are a modeling formalism with a graphical representation and a precise syntax and semantics [6][7]. They allow the modeling of characteristics like parallelism, and synchronization.

A Petri net can be defined by a tuple $C = (\mathbf{P}, \mathbf{T}, \mathbf{I}, \mathbf{O})$ [8] where \mathbf{P} is a set of places, \mathbf{T} is a set of transitions and disjoint of \mathbf{P} , \mathbf{I} is a set of input arcs in transitions \mathbf{O} is a set of output arcs on transitions, and a marking set, $\mathbf{M} = \{m_1, m_2, \dots, m_n\}$.

The graphical representation of a Petri net is done through a graph with two types of nodes: places, represented with a circle, and transitions, represented with a bar. These are connected by arcs. These are the three constituents of a Petri net [9]. Some authors still consider the tokens (associated with place marking) as the fourth constituent element of Petri nets.

The state of the system, modeling the static part of the system, is associated with the marking of the places. The transition is responsible for changing the state of the system through the creation and destruction of tokens (modeling the dynamics of the system).

4 Course Characterization and Modeling

This section focuses on key modeling issues, starting with the representation of the module-based course structure and its modeling through Petri nets, and ending with the characterization of qualifications/degrees and associated Petri nets models.

4.1 Module-Based Course Structure

In the proposed approach, courses are composed by several modules and are represented with diagrams that specify the precedence between them. These relationships can be characterized from two points of view: from the inputs of the module, and from the outputs of the module.

The precedence relationship looking at the inputs can be specified in three different types:

- 1 to 1, when the only requirement of the module is the accomplishment of one previous module.
- 1 of N to 1, when a module has as requirement the accomplishment of a module within a set of N modules (OR precedence).
- N to 1, when a module has as requirement the accomplishment of all modules in a set of N (AND precedence).

Looking from the outputs side, it is possible to specify two different types of precedence relationships:

- 1 to 1, when the module is requirement to only one other module.
- 1 to N, when the module is precedent to a set of N modules.

To better explain the method, and due to the constrained space available, a simple problem is presented as an example. Two courses are considered: a course A (C_A) and a course B (C_B) both with five modules, as presented in Fig. 1.

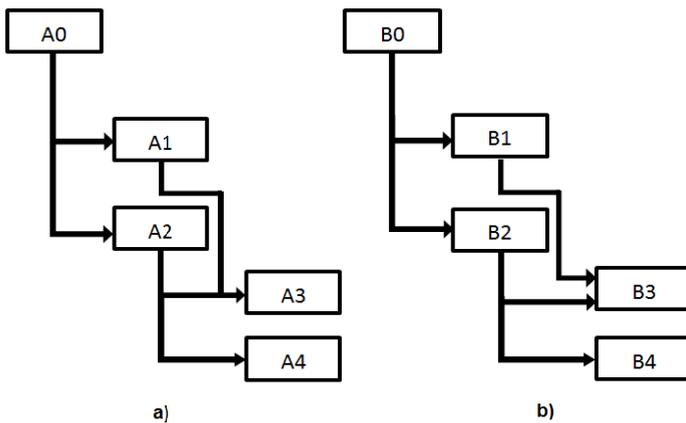


Fig. 1. Two examples of diagrams that represent precedence between the various modules in a course; a) Course A; b) Course B

Course A starts with module A0. When module A0 is accomplished Modules A1 and A2 become ready to be done. To do A4 is required the module A2 accomplishment. However to be able to do A3, modules A1 and A2 must to be previously accomplished. This models a AND dependency in the module requirements.

Course B starts with module B0. When module B0 is accomplished Modules B1 and B2 become ready to be done. In this case, to be able to do B3, one of the modules, B1 or B2 must to be accomplished. This models a OR dependency in the module requirements.

These modules are presented in Fig. 1.

Each module has a set of characteristics that define it (which are not relevant for the contents of this work). As examples of those characteristics, one can mention “the number of hours that a student will have to spend to do the module”, or “the cost of the module”.

4.2 Petri Net Modeling of the Course Structure

In the proposed approach, the module diagrams are translated into Petri nets. Each module has a common Petri net model, which will be used as the basic building block to construct the course model.

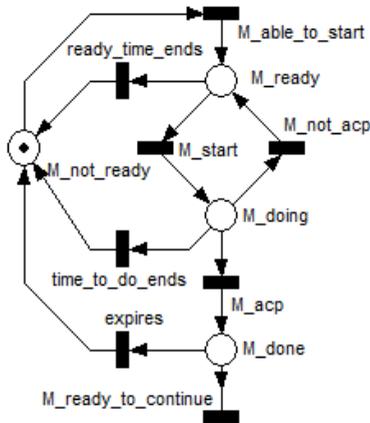


Fig. 2. Example of a generic module represented in Petri nets

Each module is represented by four places, and eight transitions. In Fig. 2 is presented a generic module M where this can be seen.

The places are: The place *M_ready*, which represents the student being able to do the module. The place *M_doing*, that represents the time while the student is doing the module. The place *M_done* represents the accomplishment of the module. Finally the place *M_not_ready* represents the student not being able to start the module.

There is always one and only one token on the module, which means that only one of these places can be marked at a time.

The transitions are: The transition *M_able_to_start* fires when the student accomplishes all the requirements to do the module. The transition *M_start* fires when the student starts the module. The transition *M_acp* fires when the student accomplishes the module. The transition *M_not_acp* fires if the student does not accomplish the module with success. The transition *M_ready_time_ends* fires if the student does not start the module before the specified time. The transition *M_time_to_do_ends* fires if student does not accomplish the module in the time that he has to do it. The transition *M_expires* fires if the time that the module was valid ends. Finally the transition *M_ready_to_continue* fires when the student is able to continue to the next step.

These modules' models are used to create the course model. It is composed by a set of modules' models connected depending on their precedence links.

This net addition is done by merging transitions *M_able_to_start* of the dependent module and *M_ready_to_continue* of the previous module model.

Fig. 3 illustrates three situations how to add the modules to create a course.

In Fig. 3 a) a 1 to 1 precedence is presented where the fusion of the transition *M_ready_to_continue* of the previous module with the transition *M_able_to_start* of the next module is foreseen.

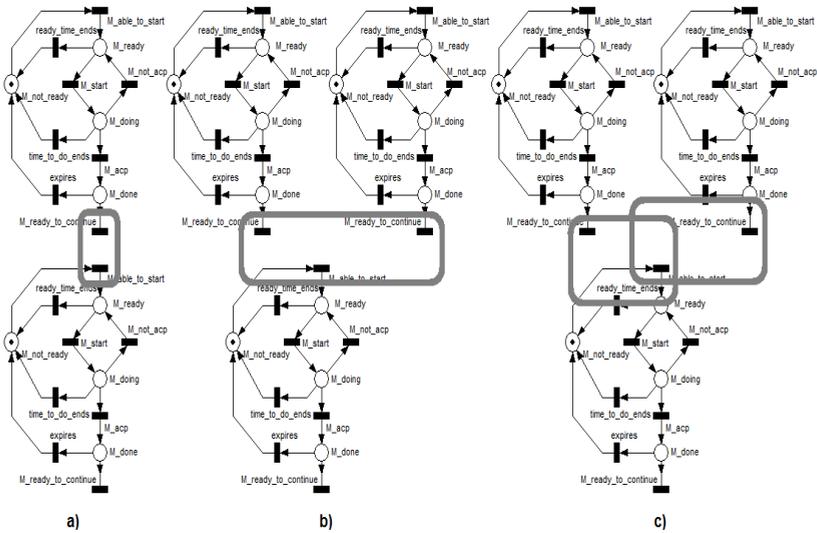


Fig. 3. Modules addition. a) 1 to 1 precedence b) N to 1 precedence (AND) c) 1 of N to 1 Precedence (OR)

In Fig. 3 b) a N to 1 precedence (AND precedence) is presented, where all the transitions *M_ready_to_continue* of all previous modules and the transition *M_able_to_start* of the next module will be merged.

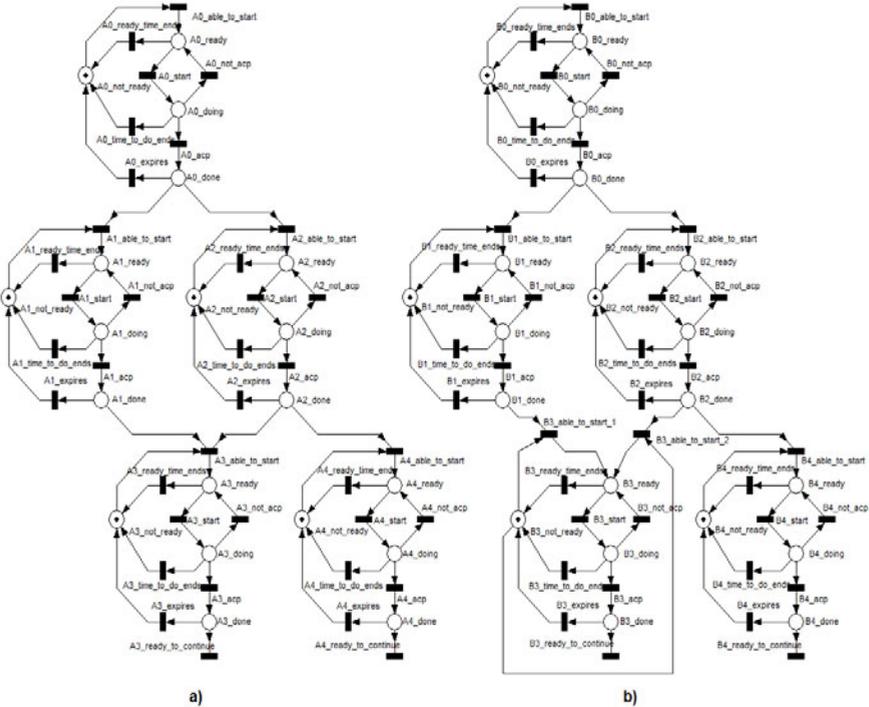


Fig. 4. Two examples of Petri nets that represent different types of precedence between the various modules in a course; a) Course A; b) Course B

In Fig. 3 c) a 1 of N to 1 precedence (OR precedence) is presented, where each transition *M_ready_to_continue* of each previous module will be merged with the transition *M_able_to_start* of the next module (using different merging operations).

The addition of the modules' models can be accomplished using the net addition operation, as defined in [10].

Fig. 4 presents the Petri net representations of the two modules presented in Fig. 1, after applying the rules presented before using the net addition operation, considering AND precedence for Fig. 4 a) and OR precedence for Fig. 4 b).

These Petri nets represent all the evolution of the student in the presented courses. It is important to note that using test arcs at the outgoing transition of each module model guarantees that Petri net conflicts are solved and that the token in each module is never removed by a subsequent module.

4.3 Qualifications Characterization

A qualification is the final goal that the student is seeking. This is obtained finishing a set of modules that are specified in the Qualification specification. The qualification is

represented by a diagram, as the one presented in Fig. 5. a), where Qualification Q1 is composed by modules A3 and B3.

This diagram does not present all modules that a student has to do, but only the (final) modules that are required for the qualification (if he/she needs to do other modules to be able to do these modules, those modules are not explicitly represented).

4.4 Qualifications Petri Net Modeling

The qualification is converted into Petri nets models too. The proposed net has only two transitions, as presented in Fig. 5. b). The first transition (Q1_start) fires when the students start their qualification and the transition Q1_acp fires when the student accomplishes all modules of the qualification.

The Fig. 5.b) presents the Petri net that represents the Qualification 1 presented in Fig. 5. a).

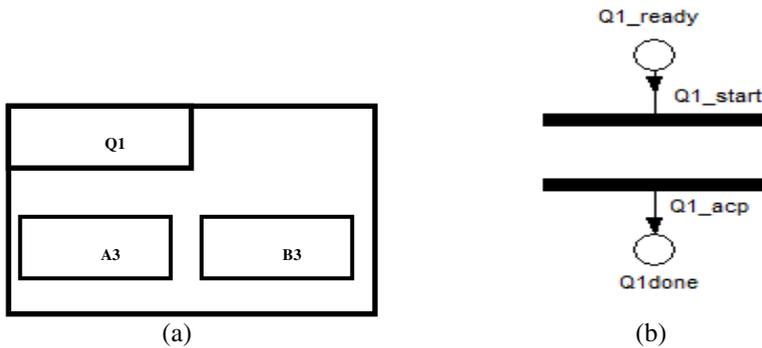


Fig. 5. Qualification Representation; **a)** Diagram that represents a qualification and the modules that are needed to obtain the qualification. **b)** Petri net model that represents a qualification.

This net is composed basically by a start transition and an end transition that represents respectively the start and the accomplishment of the qualification. Once again, the structural conflict between the several modules composing the qualifications is solved by using test arc as outgoing arc from the place Q1_ready.

5 Study Plan Elaboration

When a student wants to obtain a certain qualification is necessary to elaborate his/her study plan. As far as the qualification is defined by the deepest modules of the courses that compose it, it is possible to obtain the study plan dependencies through net addition of the qualification's model and the models of the modules.

Using the courses and qualifications Petri nets models, it is possible to automatically generate the study plan, which represents the modules the student needs to attend accordingly to the defined sequence.

The study plan model is created through the use of the net addition operator [10][11]. To create the study plan (S) for a Qualification (Q) requiring completion of n Modules (Mn) the net addition operation is used as follows:

$$S = Q + \sum Cn (...)$$

where:

- S is the study plan model,
- Q is the qualification model,
- Cn is the model associated with required modules Mn, and
- (...) represents the fusion sets involving transitions and allowing gluing together the different models (as in Fig. 3).

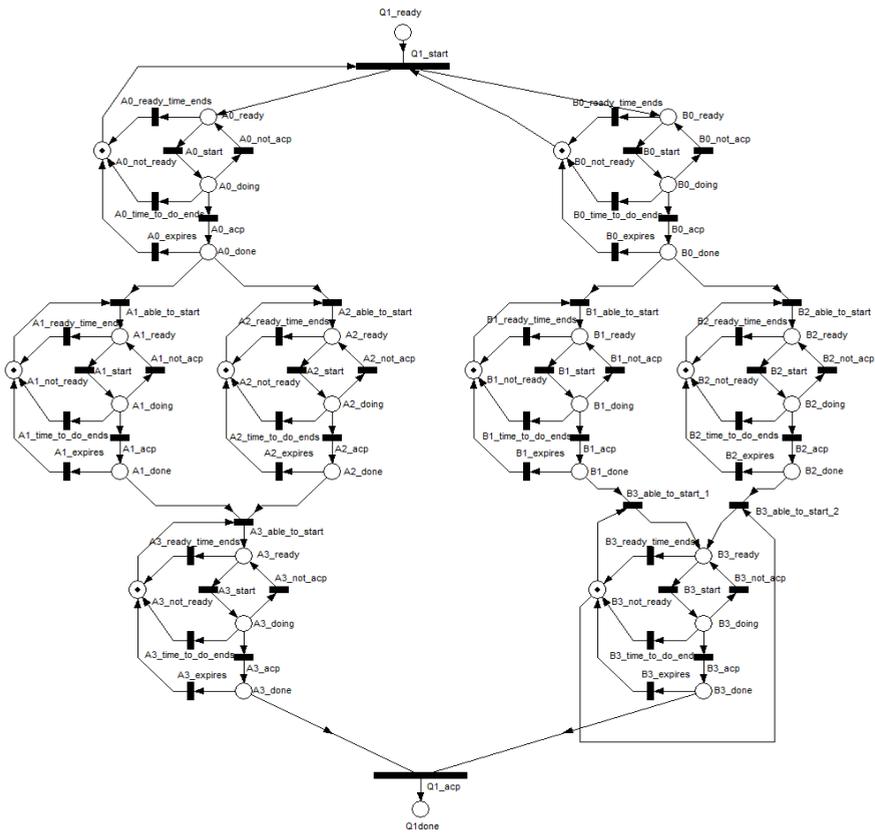


Fig. 6. Petri net that represents the Study plan for the qualification Q1, generated automatically by the proposed method

For the qualification Q1 and courses A (C_A) and B (C_B) presented before, the study plan can be generated using the following fusion sets (where before \rightarrow all transitions to be merged are identified, and after the \rightarrow the transition obtained by fusion is referred):

$$\begin{aligned} & Q1.Q1_start / C_A.A0_able_to_start / C_B.B0_able_to_start \rightarrow S.Q1_start \\ & Q1.Q1_acp / C_A.A3_ready_to_continue / C_B.B3_ready_to_continue \rightarrow S.Q1_acp \end{aligned}$$

Fig. 6 represents the study plan modeled with Petri nets. It is automatically generated applying the previous method (where only the nodes that can affect the referred A3 and B3 modules outgoing transition are represented; this strategy leads to removal of models associated with modules A4 and B4 in initial courses).

In this net we have all information about what the student needs to do to obtain the Qualification.

It is important to note that the resulting Petri net model can be integrated in “any” learning management systems and constraints the “navigation” of the student through the available modules.

On the other hand, it is possible to analyze the Petri net model in order to conclude about specific characteristics. Building associated state space allows to conclude about reachability of specific states (which means accomplishment of specific modules or qualifications), as well as determine costs associated with a specific path. By costs one can associate different types of metrics, depending on the type of data that can be associated with a student and with the modules (namely duration, fees, and so on). As an example, a student may consider to be enrolled in a sequence of modules leading to the shortest duration till accomplishment of one qualification.

6 Conclusions and Further Work

A methodology based on Petri Nets was proposed to model a complete open learning system and create aimed study plans. The technique allows modeling several study plans automatically allowing the user to choose the best way to achieve his/her goals. A complete e-learning system of a university or open system with many institutions may be modeled by the proposed strategy, where the student has the possibility to choose the modules that better fit his/her purposes independently of the institutions. A tool is foreseen to be available to support the proposed modeling strategy and its integration in a learning management system (as Moodle, for example). It is important to remark that the proposed tool can be employed for other purposes as well, for example, to model the tasks sequence of a project. In this sense, an interesting future work would be to explore the use of tool for other case studies.

References

1. Sun, X., Feng, Z.: Unified and integrated e-learning modeling supporting dynamic learning process optimization. In: 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), July 26-28, vol. 4, pp. 2137–2141 (2011), doi:10.1109/FSKD.2011.6020015

2. Gu, Q., Sumner, T.: Support Personalization in Distributed E-Learning Systems through Learner Modeling. In: 2nd Information and Communication Technologies, ICTTA 2006, vol. 1, pp. 610–615., doi:10.1109/ICTTA.2006.1684441
3. Borges, D.C., Neto, H.B., de Souza, J.N.: Work in progress - Petri Nets as applied to the modeling of E-learning cooperative systems. In: 2010 IEEE Frontiers in Education Conference (FIE), October 27-30, pp. F1D-1–F1D-3 (2010)
4. Baohua, Z.: Research on Evaluation of E-Learning Modelling Based on Petri Nets. In: International Conference on Advanced Computer Theory and Engineering, ICACTE 2008, December 20-22, pp. 699–703 (2008)
5. Liu, X.-Q., Wu, M., Chen, J.-X.: Knowledge aggregation and navigation high-level Petri nets-based in e-learning. In: Proceedings. 2002 International Conference on Machine Learning and Cybernetics, vol. 1, pp. 420–425 (2002)
6. Silva, M.: Las Redes de Petri: en la Automática y la Informática. Editorial AC, Madrid (1985)
7. Girault, C., Valk, R.: Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications. Springer, Heidelberg (2002)
8. Reisig, W.: Distributed algorithms: modeling and analysis with Petri nets. In: 1998 IEEE International Conference on Systems, Man, and Cybernetics, October 11-14, vol. 1, pp. 38–43 (1998), doi:10.1109/ICSMC.1998.725380
9. Murata, T.: Petri Nets: Properties, Analysis and Applications. In: Proceedings of IEEE, vol. 77-4 (1989)
10. Barros, J.P., Gomes, L.: Net model composition and modification by net operations: a pragmatic approach. In: Proceedings of the 2nd IEEE International Conference on Industrial Informatics, INDIN 2004 (June 2004)
11. Gomes, L., Barros, J.P.: Structuring and Composability Issues in Petri Nets Modeling. IEEE Transactions on Industrial Informatics 1(2), 112–123 (2005)