From Local Patterns to Global Models: Towards Domain Driven Educational Process Mining

Nikola Trčka, Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{n.trcka, m.pechenizkiy}@tue.nl

Abstract

Educational process mining (EPM) aims at (i) constructing complete and compact educational process models that are able to reproduce all observed behavior (process model discovery), (ii) checking whether the modeled behavior (either pre-authored or discovered from data) matches the observed behavior (conformance checking), and (iii) projecting information extracted from the logs onto the model, to make the tacit knowledge explicit and facilitate better understanding of the process (process model extension). In this paper we propose a new domain-driven framework for EPM which assumes that a set of pattern templates can be predefined to focus the mining in a desired way and make it more effective and efficient. We illustrate the ideas behind our approach with examples of academic curricular modeling, mining, and conformance checking, using the student database of our department.

1 Introduction

In modern education various information systems are used to support educational processes. In the majority of cases these systems have logging capabilities to audit and monitor the processes they support. At the level of a university, administrative information systems collect information about students, their enrolment in particular programs and courses, and performance-like examination grades. These data can be analyzed from various levels and perspectives, showing different aspects of the educational process.

Traditional data mining techniques have been extensively applied to find interesting patterns, build descriptive and predictive models from large volumes of data accumulated through the use of different information systems. The results of data mining can also be used for getting a better understanding of educational processes, for generating recommendations and advices to students, for improving management of learning objects, etc. [9]. However, most of the traditional data mining techniques do not focus on the process as a whole. They do not aim at discovering, analyzing, nor providing a visual representation of the complete educational process. Process mining [12] tools and techniques, on the contrary, are aimed at extracting process-related knowledge from event logs recorded by an information system.

In [8] and [11] we already illustrated the potential of process mining technology for the educational domain. We showed that this technology is not only helpful in the analysis of historical data, but also in facilitating decision support for directors of education, study advisors, and students (e.g. through the use of various *what if* scenarios that can be simulated and checked against the existing constraints). However, we also noticed that the existing process mining approaches sometimes have serious trouble mining comprehensive and easy-to-understand graphical representations of educational data. After conducting many experiments we realized that these problems should not only be attributed to the inherited difficulty of the natural phenomena under study. The large variation in the data, long-term dependencies, and explosion of strongly dominating yet meaningless patterns covering organizational issues, are all also the reasons why mining educational process models is a rather non-trivial task. Consider, e.g., the situation when scheduling the exams in certain way every year hints a strong pattern suggesting the order in which the students take the exams, while from the domain perspective all exams within an individual session are considered simultaneous.

In this paper we present a new domain-driven Educational Process Mining (EPM) framework that facilitates more focused search for local patterns and their further assembling into a global model. Our EPM framework assumes that, based on domain knowledge, we can predefine a set of pattern templates that are crucial for the problem at consideration. This pattern set can always be modified

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.159

and extended, but we assume that the templates (and possibly concrete patterns too) are pre-authored at the moment of process mining. Thus, we can reduce the search space and direct the process mining in the desired way. Conformance checking and model extension can also be focused on the patterns of our prior interest. Moreover, many other activities can be performed in exactly the same way as it is currently done in the existing process mining framework.

The rest of the paper is organized as follows. In Section 2 we introduce our framework for domain-driven EPM. Section 3 introduces and formalizes some typical patterns in academic curriculums. Section 4 gives some intuitive examples of academic curricular mining and conformance checking. Related work is briefly summarized in Section 5. We conclude with the discussion of further work in Section 6.

2 Framework for domain-driven EPM

The general ideas behind our framework can be illustrated with Figure 1. An information system that supports educational institution generates event logs that are stored in a database and represent e.g. student performance and enrolment into the courses and corresponding exams. Educators, responsible e.g. for curriculum development and monitoring its effectiveness, can help to identify typical constraints that should be enforced in the study process. Given the event log reflecting historical data and pattern templates we can mine *all* the patterns present in the database satisfying the templates. The resulting pattern set can be post-processed in a semi-automatic way, and then a unique process model (represented as a colored Petri net graph in our case) can be assembled into a graph structure (or first abstracted to a more comprehensive representation, so that domain experts are not puzzled with the lower-level primitives). On the process model we can next perform standard task of process mining, like e.g.: (i) determining popular paths or "narrow" places in the curriculum, (ii) extending the model with additional information or modifying it, (iii) executing various *what if* scenarios to facilitate real time decision making, and (iv) real-time monitoring.

Note that we do not exclude the possibility of a *manual* design of the complete process model by the domain experts. From the point of view of the analysis, the same tasks can be performed on the manually designed model as on the mined one. However, a benefit of this is that we can, using the standard conformance checking techniques, automatically check whether any of the required constraints (now captured in the designed model) have been violated in the past.

3 Academic Curriculum Patterns

Academic curriculum is a (legal) document defining course-related rules that must be respected by students through their study period. These rules typically describe a set of courses and a set of relationships between these courses. The rule "*logic1 must be passed with a grade bigger than 6 before logic2 can be taken*" (or a similar one) is, for example, commonly seen in computer science curricula.

The rules in a curriculum are usually stated informally, in a natural language, and are thus subject to multiple interpretations. It is not uncommon that students have to approach their study advisors to ask whether they are allowed to enroll certain course or what would the impact be if they do. The advisors, on the other hand, are themselves often confused and must ask the board of education for clarification.

To alleviate the above problem we propose a method for mathematical (and thus formal) modeling of a generic academic curriculum. Our main idea is to identify some typical constraints defined in the existing curricula, define them in form of patterns and use the formal (and graphical) language of *Colored Petri nets* (CPNs) [7] to express (i.e. encode) these patterns. In this way we give a precise and unambiguous semantics to the study rules.

CPNs are based on Petri nets and can thus easily model concurrency, synchronization, alternative and sequential behavior. The extension with colors adds several standard programming language-like elements to the language and allows for the modeling of timing constrains. Moreover, the CPN formalism comes equipped with a powerful toolset called CPN Tools [13], supporting simulation and various forms of formal verification. We show that CPNs are a natural choice for the modeling of a curriculum, and that all identified patterns can easily be encoded into CPNs.

The advantages of having a formal and executable model of a curriculum are not only in its elimination of ambiguity, but in the fact that through the use of CPN Tools and ProM [3], such a model almost directly offers a wide range of possibilities. For example,

- Students can automatically check, by themselves, whether they are allowed to do something or not;
- Historic data stored in the log of the educational information system can be compared against the model. In this way we could see whether the curriculum was always respected in the past;
- The same historic data can be used to equip the model with quantitative information (probabilities, delays, etc.). This immediately enables us to perform all kinds of performance analysis (finding, e.g., the average time to graduation, the most common paths, etc.) and generate recommendations; and

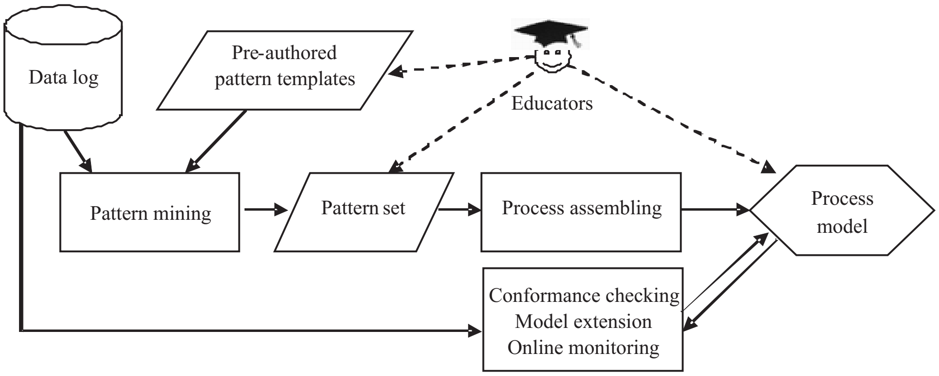

Figure 1. Our framework for domain-driven educational process mining

- Online simulation of the model can facilitate real-time detection of curriculum violations.

3.1 Basic curricular constraint patterns

In this section we show how some typical constraint-type patterns of a curriculum can be effectively modeled by CPNs. Note that it is not our intention to cover every possible constraint, but rather focus on some commonly seen requirements. We hope that the reader is convinced that the same method can be extended to cover more general and more complex cases. It is assumed that the reader is familiar with the CPN formalism; if not, [7] provides a good introduction.

The ***course/exam pattern*** is described in Figure 2. Every course is represented by 1) a place (`grades_of_C` in the figure) containing the ordered list of grades obtained for this course (per student), and 2) a transition (`C` in the figure), of which every firing corresponds to taking this course in an exam. The place `grades` is needed as the result of an exam can be any grade from 0 to 10. This place is global, i.e. visible to every element of the CPN model. When `C` fires the list of grades is appended with a new (non-deterministically chosen) grade. We also model the maximum number of times that this exam is allowed to be taken (variable `MAX_NUM_ATTEMPTS`).

Here it is important to note that the ***course/exam pattern*** needs to be modeled for every course and that it is the starting point of the whole model. The rest of the patterns use the existing places and transitions and just add more arcs (sometimes more places and transitions too) to restrict behavior according to the constraints they represent.

We next consider the ***starting pattern***. It simply defines which courses are allowed to be taken at start. The pattern is shown in Figure 3 (on the left) for two courses `C1` and `C2`. The place start contains student ids. It is this place that initially holds tokens and from which the model actually starts executing. Similarly, we model the fact that graduation is the last exam to be taken and call this the ***ending pattern*** (see the right part of Figure 3).

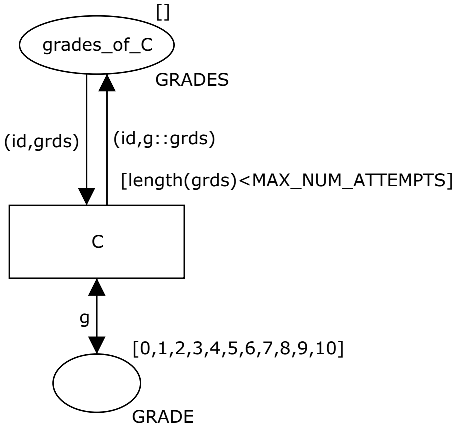

Figure 2. Course exam pattern

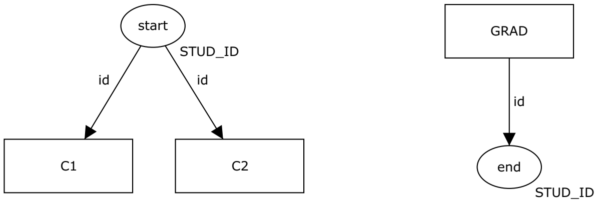

Figure 3. Starting and ending patterns.

The next pattern is more involved. It represents a realistic complex constraint saying that ***m-out-of-n*** courses must be passed before some other course can be taken. Figure 4 shows the ***2-out-of-3*** variant of this pattern where `D` is the constrained course. The condition under `D` is allowed to fire, i.e. the guard of D, is hidden in the function check2out3(grds1,grds2,grds3). This function would typically check whether the head of two lists from {grds1,grds2,grds3} contains a passing grade. It is, of course, possible to have more complicated functions and thus express more complex constrains.

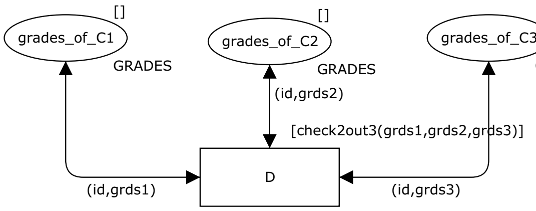

Figure 4. *m-out-of-n* pattern

Some courses are considered stronger than other courses in the sense that the result of a course also counts as the result for some other (weaker) course. We capture this by the ***dependency pattern***. Figure 5 show the instance of this pattern when the grade of D automatically becomes the new grade of C. Note that for two equivalent courses we would have this pattern in both ways.

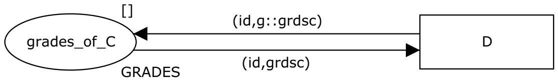

Figure 5. Dependency pattern

The list of grades for a course can only be valid for some time. The final pattern considered in this paper is the expiration pattern, depicted in Figure 6. The idea is to introduce a special transitions that takes the list of grades and returns the empty list. This transition has a guard expireCond(grds) (which can be any function returning a boolean value as a result), implementing the actual condition when expiration must take place.

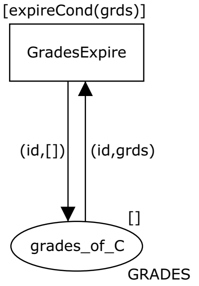

Figure 6. Expiration pattern

Any realistic curriculum would have a large list of the above (and similar) patterns. This, of course, drastically reduces the readability of the full CPN model. However, note that the model is not meant to be read by humans but rather as an input to various forms of analysis methods or as a template for process mining.

4 Mining and conformance checking

The previous section provided us with a list of patterns that appear in a typical educational curriculum. In this section we explain how these patterns can be used for the discovery of the complete educational process model. We also show, assuming that the whole model is manually constructed, how we can check whether the observer behavior confirms with the patterns.

4.1 Pattern mining

Given a set of pattern templates the process of mining (or querying) the patterns from the database is rather straightforward, but may become computationally prohibitive. For example, for the m***-out-of-***n courses constraint pattern template we would need to find patterns for all the combinations of m and n. But in practice, a user can define the upper bound for n and simplify this task.

Typically, in most cases we only need to identify whether events follow sequential execution (like course A must be taken before B), parallel execution (course A and B are usually taken together), or choice (only one of the courses A or B can be taken). There are different approaches to decide what kind of pattern we observe. A simple and intuitive way to infer whether we deal with events occurring in parallel (or there is a choice) is to check from the process instances all the occupancies of A and B: if A and B appear in the traces in any order we likely deal with a parallelism; if they never appear one after the other these two actions likely form a choice.

The set of patterns extracted from an educational database can be rather large. Therefore domain experts may benefit from additional tools allowing them to navigate through this set and remove meaningless and redundant patterns. Different ideas of mining non-derivable patterns, dominating patterns, pruning mined patters, using condensed representation of patterns and defining pattern interest, confidence- and coverage-based measured (that have been already developed in data mining research area) can be reused in our framework.

Assembling of local patterns into a global model. In order to provide the users with a comprehensive overview of the complete process a graph-based representation can be generated. This procedure is rather straightforward from

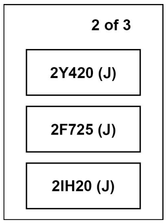

Figure 7. *2-out-of-3* pattern abstracted from CPN presentation

the computational point of view. Visualizing a possibly large spaghetti-like graph in a comprehensive way may be a challenging task. Anyhow, we should emphasize again that domain experts do not need to see an overcomplicated low-level representation with a CPN. Instead, complex CPN patterns can be represented in an abstract way. Consider, e.g., the constraint where each student has to take at least 2 courses from 2Y420, 2F725 and 2IH20; this ***2-out-of-3*** pattern would be presented as in Figure 7.

4.2 Conformance checking

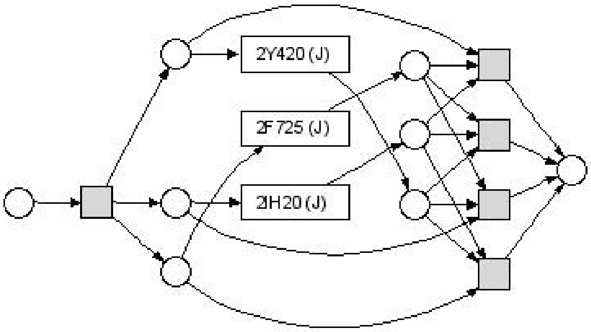

Figure 8. 2-*out*-*of*-3 **constraint modeled as a classical Petri net in ProM**

The curriculum patterns describe the possibilities the students have, and the constrains they must obey. Conformance checking is a technique that can check, against the event log, whether these constraints were indeed always respected. In this section we show how the *conformance checker plugin* [10] of ProM [3] can be used to check one 2-out-of-3 constraint from the curriculum of our department.

Figure 8 shows the already mentioned constraint where each student has to take at least 2 courses from 2Y420, 2F725 and 2IH20 (the pattern is now drawn now as a classical, not colored, Petri net as this is the input of the plugin, but the essential information is the same). Figure 9 gives the output of the conformance checker (the *Model view*), annotating the original Petri net with many details. We can now see the places in which problems occurred during the log replay and many other interesting characteristics, including path coverage, passed edges, failed and remaining tasks and token counters. Note that, in this case, there were no problems in the log and the considered constraint has always been satisfied.

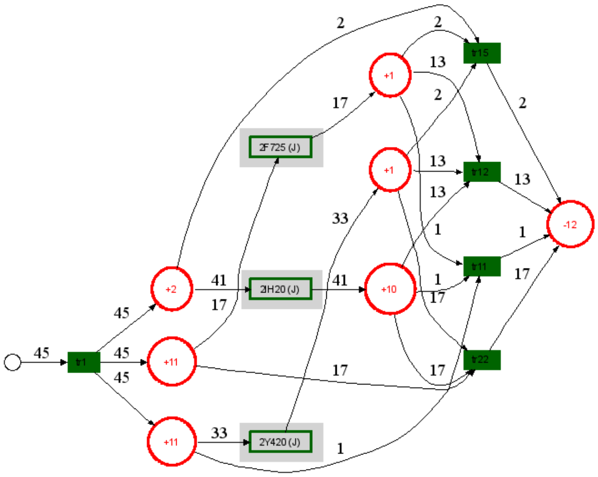

Figure 9. Conformance analysis of the Petri net representing *2-out-of-3* constraint

5 Related work

Workflow mining. In the past few years there has been an increasing interest in the analysis of process logs, particularly in the emerging area of process mining. Most of the work in process mining was concentrated on (business) workflow systems and discovery of Petri nets representations of workflows. A number of algorithms have been proposed and are supported by ProM [3].

In data mining, the focus with regard to process discovery was on mining sequential patterns (or graphs) and their assembling into the global models. The problems with mining unconnected patterns (sets of activities that are frequently executed together but do not exhibit explicit dependent relationships) in workflows were presented in [5]. In both communities, however, most of the works do not consider duration of (and between) the activities into account. A relatively new mining paradigm, temporally-annotated sequences, that is highly related to the problem of process mining and analysis was introduced in [4]. Another important step in the recent development of this area is the framework for user-interactive exploration of a condensed representation of groups of process executions [2].

Modeling curricular. The problem of curricular modeling has been widely recognized as an important an non-trivial task. However, to the best of our knowledge very few papers address the problem of the *formal modeling and analysis* of academic curricular. An interesting attempt was given in [6]. Modeling an academic curriculum plan as a mixed-initiative constraint satisfaction problem was proposed in [14]. The use of constrained sequential pattern mining and and constraint relaxations (i.e. an approximation to the constraint) was considered in [1].

6 Discussion and Further Work

Process mining is a relatively new technology emerged from business community. Recently, we showed some of the potential of this technology to the educational domain. Educational process mining (EPM) allows for different kinds of decision support, it also allows to get a better understanding of the underlying educational processes.

In this paper, we introduced a new framework that allows to integrate domain knowledge into the core of EPM and to facilitate interactive process mining. The framework is aimed at helping educators analyse educational process in a principled way based on formal modeling. We illustrated our techniques giving with basic examples of pattern template authoring, pattern mining and conformance checking.

Our future work will pursue in several directions. On the research side, we are primarily interested in developing efficient pattern mining techniques and designing an intuitive graphical language for domain experts to generate typical constrains and patterns. On the practical side, we are aimed at developing EPM plugins for the ProM framework. In the first place, we think that authoring tools and two-way translations from abstract primitives (defined in the domain experts-tailored graphical language) to CPNs need to be developed. On the experimental side we plan to conduct a case study that would illustrate the feasibility of our approach in a real educational setting.

References

[1] C. Antunes. Acquiring background knowledge for intelligent tutoring systems. In *EDM'08: 1st Int. Conf. on Educational Data Mining*, pages 11–27, 2008.

[2] M. Berlingerio, F. Pinelli, M. Nanni, and F. Giannotti. Temporal mining for interactive workflow data analysis. In *15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'09)*, pages 109–118, 2009.

[3] B. Dongen, A. Medeiros, H. Verbeek, A. Weijters, and W. Aalst. The ProM framework: A New Era in Process Mining Tool Support. In *ICATPN'05*, volume 3536, pages 444–454. Springer-Verlag, Berlin.

[4] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Mining sequences with temporal annotations. In *ACM Symp. on Applied Comp. (SAC'06)*, pages 593–597.

[5] G. Greco, A. Guzzo, G. Manco, and D. Sacca. Mining unconnected patterns in workflows. *Inf. Syst.*, 32(5):685–712, 2007.

[6] B. Hnich, Z. Kzfitan, and W. T. Modelling a balanced academic curriculum problem. In *Proc. of CP-AI-OR-2002*, 2002.

[7] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Springer-Verlag, 1997.

[8] M. Pechenizkiy, N. Trčka, E. Vasilyeva, W. van der Aalst, and P. D. Bra. Process mining online assessment data. In *EDM'09: 2nd Int. Conf. on Educational Data Mining*, 2009.

[9] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146, 2007.

[10] M. Rosemann and W. Aalst. A Configurable Reference Modelling Language. *Information Systems*, 32(1):1–23, 2007.

[11] N. Trčka, M. Pechenizkiy, and W. van der Aalst. *(to appear) Handbook on Educational Data Mining*, chapter Process Mining from Educational Data. Taylor & Francis, 2010.

[12] W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[13] A. Vinter Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen, and K. Jensen. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In *ICATPN 2003*, volume 2679 of *LNCS*, pages 450–462. Springer Verlag, 2003.

[14] K. Wu and W. Havens. Modelling an academic curriculum plan as a mixed-initiative constraint satisfaction problem. In *Proc. of Canadian Conference on AI'05*, pages 79–90. Springer-Verlag, 2005.