Knowledge-Based Systems 100 (2016) 160–174

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Recompiling learning processes from event logs

Juan C. Vidal*, Borja Vázquez-Barreiros, Manuel Lama, Manuel Mucientes

Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), Universidade de Santiago de Compostela, Rúa de Jenaro de la Fuente Domínguez, 15782 Santiago de Compostela, Spain

ARTICLE INFO

Article history:
Received 3 September 2015
Revised 22 January 2016
Accepted 4 March 2016
Available online 10 March 2016

Keywords:
Learning flows discovery
Process mining
Adaptive rules mining
IMS Learning Design

ABSTRACT

In this paper a novel approach to reuse units of learning (UoLs) – such as courses, seminars, workshops, and so on – is presented. Virtual learning environments (VLEs) do not usually provide the tools to export in a standardized format the designed UoLs, making thus more challenging their reuse in a different platform. Taking into account that many of these VLEs are legacy or proprietary systems, the implementation of a specific software is usually out of place. However, these systems have in common that they record the events of students and teachers during the learning process. The approach presented in this paper makes use of these logs *(i)* to extract the learning flow structure using process mining, and *(ii)* to obtain the underlying rules that control the adaptive learning of students by means of decision tree learning. Finally, *(iii)* the process structure and the adaptive rules are recompiled in IMS Learning Design (IMS LD) – the *de facto* educational modeling language standard. The three steps of our approach have been validated with UoLs from different domains.

1. Introduction

While designing a course, there are two main concerns that worsen the realization of an educational scenario: *(i)* how to model a practical pedagogical scenario to achieve the educational objectives, and *(ii)* how to reuse this scenario in another context than the original. Teachers do not only define the learning content to be consumed by the learners, but they also include the different educational objectives, the order in which the learning activities must be undertaken to achieve these objectives, the evaluation methods, etc. Hence, to reuse and better validate an educational scenario, it should be explicitly written. Although these learning designs, i.e., the descriptions of the educational process, are usually portrayed with documents that use natural language, they can be formally described through Educational Modeling Languages (EMLs). Moreover, when interacting with a virtual learning environment (VLE), learners also perform additional activities than the specifically defined by the teachers, such as interacting in the forum, checking the bibliography, etc. This information should be also highlighted to enable teachers to improve the *learning flow*, i.e., the *real* workflow of learning activities, as well as the *evaluation process* [33]. Therefore, the defined educational scenario is more complex than the learning design explicitly documented by teachers.

In the last decade a great effort has been made for developing EMLs. The main idea underlying these languages is to describe, from a pedagogical point of view, the learning process of the course, i.e., the *sequence of steps* the learners should undertake to achieve the educational objectives of the course, by using the available educational resources and services. Regarding the wide variety of specifications for representing learning designs, one standard *de facto* has jumped into e-learning panorama: the IMS Learning Design (IMS LD) specification [8]. IMS LD enables the formal description of learning processes for a wide range of pedagogical contexts in a VLE. Although there is some controversy about whether IMS LD is too complex to be understood by teachers from a practical point of view [11] – especially with the levels B and C – most of authors highlight this complexity as a barrier for adopting IMS LD [25].

To deal with this issue, a number of user-friendly authoring tools have appeared [5,9,14,16,19,24], but even with these tools, authoring process of IMS LD units of learning[1] (UoLs) is not easy for teachers when these UoLs are complex or require to use advanced features of this standard. The *automatic* reconstruction of UoLs could relieve this issue [25], promoting the use of IMS LD by teachers and instructors. Taking as starting point the event log files, which stores all the events generated by the learners, it

* Corresponding author. Tel.: +34 881816388.
E-mail addresses: juan.vidal@usc.es (J.C. Vidal), borja.vazquez@usc.es (B. Vázquez-Barreiros), manuel.lama@usc.es (M. Lama), manuel.mucientes@usc.es (M. Mucientes).

[1] An UoL represents a variety of prescribed activities, assessments and services provided by teachers, in a course or lesson, which is the result of the learning design.

http://dx.doi.org/10.1016/j.knosys.2016.03.003
0950-7051/

is possible to mine the *real behavior* undertaken by the students during the UoL, i.e., what the learners really did, and the rules that constraint the behavior of the model. Then, by combining these two models, it is possible to reconstruct the UoL to a specific target language. Therefore, this process facilitates the reuse of defined UoLs no matter the VLE that has been used, as the techniques used for both mining the variable values and the formal model are totally independent of the domain. Therefore, teachers can design their courses within their VLE, avoiding the need to use an authoring tool with a specific EML notation, and still be possible to reconstruct the UoLs from the scratch to a target language – such as IMS LD.

In this paper, we present an approach to *automatically* reconstruct the IMS LD representation of an UoL from the events generated by the learners in the VLE. This objective is achieved in three different steps. Firstly, the learning flow of the UoL is automatically extracted from the logged sequences through a *process discovery algorithm*. Then an algorithm based on the knowledge about the IMS LD control structure is applied to determine which IMS LD components should be created. Finally the adaptive rules of the UoL are automatically extracted from the event logs – more specifically, from the variable values of the logs – by a decision tree learning algorithm, and integrated into the IMS LD structure. The contributions of this proposal are: *(i)* a new framework to facilitate the reuse of UoLs between different VLEs; *(ii)* the automatic discovery of learning processes from event logs and its recompilation to IMS LD; and *(iii)* the automatic identification of the adaptive rules from event logs.

Notice that IMS LD has a high expressiveness to allow the definition and orchestration of complex activity flows in a multi-role setting, but at the expense of complexity. In fact, current IMS LD research seems to accept the assumption that specification's conceptual complexity hiders the authoring process [11]. Taking this into account, another objective of this paper is to reduce this barrier and facilitate the adoption of UoLs specified in IMS LD by instructors. Specifically, the proposed semi-automatic approach hides the complexity of the EML language, so instructors only have to decide which one of the recompiled processes fits better with the learning objectives of the UoL, in terms of structure and adaptive criteria. Henceforth, the main research question addressed in this paper is the automatic reconstruction of UoLs from scratch, as the state of the art heavily relies in the participation and feedback from all appropriate personnel and users during the whole process, hindering the reuse of UoLs in different platforms.

The remainder of this paper is structured as follows. Section 2 briefly introduces the main features of the IMS LD specification. Section 3 describes the different approaches that have already been proposed and that motivated our approach. Then, Section 4 presents the framework that supports the mining of log files and the reconstruction of IMS LD. Sections 5 and 6 detail, respectively, how the learning flow – through a process discovery algorithm – and the adaptive rules – through a decision tree algorithm – are mined from the logs. Then, Section 7 details the transformation from these two models to the actual target language (IMS LD). Section 8 shows the results and, finally Section 9 points out the conclusions and future work.

2. IMS Learning Design

IMS LD specification is a meta-data standard that describes all the elements of the design of a teaching–learning process [8]. This specification is based on: *(i)* a well-founded *conceptual model* that describes the vocabulary and the functional relations between the concepts of the learning design; *(ii)* an *information model* that details in natural language the semantics of every concept and relation introduced in the conceptual model; and *(iii)* a *behavioral model* that specifies the constraints imposed to the software system when a given learning design is executed in run-time. In other words, the behavioral model defines the semantics during the execution phase. Furthermore, IMS LD defines three levels of implementation depending on whether the learning design is adaptive or not:

- *Level A*. This first level contains the main components of a UoL: participants (roles), pedagogical objectives, resources (services and contents), and learning design. This last component is understood as the coordination of the learning activities to be performed by the participants to achieve the pedagogical objectives, i.e., the learning design describes the learning flow – or learning path – to be followed by learners in a UoL. To describe this learning design, the IMS LD specification follows a theater metaphor where there are a number of plays, that can be interpreted as the runscripts for the execution of the UoL and that are *concurrently* executed, being independent of each other. Each one of these plays is composed by a set of acts, which can be understood as a module or chapter in a course. Acts are performed in *sequence* and define the activities that participants must do. This model also allows the assignation of roles to the participants and partitioning the activities of an act according to those roles. In this case, each one of the partitions can run *in parallel*. Finally, activities can be simple or complex, the latter may consist of a *sequence* or *selection* of activities (simple or complex).
- *Level B*. This level adds *properties* and *conditions* to level A. It also adds monitoring services and global elements which allow users to create more complex structures. The properties store information about people (preferences, outcomes, roles, etc.), personal information, or even about the learning design itself. Level B also establishes *(i)* the visibility of the elements of the learning flow; *(ii)* if properties are transient or should persist across multiple sessions; and *(iii)* the set of operators and expressions that may transform the value of properties and the visibility of elements. For instance, adaptation is usually based on the visibility of the activities of the learning flow, since IMS LD does not have control structures such an *if-then-else*. Therefore, the adaptation rules use properties, such as a test score, an answer to a specific exercise, and so on, to decide the learning path of the student through the visibility of the activities.
- *Level C*. The last level incorporates notifications to level B. Notifications fire automatically in response to events triggered in the learning process. For example, if a student submits a job, an email to report the event could be automatically sent to the teacher.

Taking this into account, the objective of this paper can be defined more precisely as recompiling the structure of the learning process defined at level *A* and the properties and adaptation rules at level *B* from event log files.

3. State of the art

We have focused our analysis of the state of the art in the topics and fields that motivated our approach: *(i)* IMS LD authoring tools; *(ii)* the reconstruction of IMS LD; and *(iii)* the applications of process mining in education.

For the last years, a number of IMS LD authoring tools have been developed. These solutions allow a better analysis of the related educational design approaches by trying to relieve the complexity of this standard to teachers and instructors. ASK-LDT [19] provides a graphical interface that allows to hide the complexity of the IMS LD control structure and adaptive components. The authors define an abstract high-level architecture for designing pedagogical scenarios that can be reused in different virtual learning