ABSTRACT

Educational Modelling Languages such as IMS LD, LAMS and LPCEL can be used to describe a learning process, but they depict different degrees of expressiveness to represent learning flows. This paper presents an evaluation of the expressiveness of these languages based on the analysis of different Workflow Control Patterns including Basic Control Flow, Advanced Branching and Synchronization, and Iteration; outlining the learning process design issues.

1 Introduction

Computer-aided design and execution of learning activities try to describe a learning experience from the point of view of the tasks that participants have to realize and the learning resources involved. That sort of design activities are generally known as learning design (LD), which aims at taking prescriptions from instructional design theory plus examples of best practices and applying them for defining learning courses ready to be run on the computer [1].

Educational Modelling Languages (EML) [1, 2] represent an important approximation to integrate diverse educational aspects, allowing the design and implementation of spaces and activities supported by Information and Communication Technologies (ICT). With EML it is possible to integrate, under a learning process, personalized educational material for each student —i.e. activities, resources, objectives, evaluations, profiles—, promoting the active participation of the students with their learning process. Also, these languages are based in the creation of learning scenarios inside a Unit of Learning (UoL). The different elements implicated in the learning experience —i.e. roles, objectives, results, activities, resources and services—, are modeled in these units along with the organization mechanisms and coordination of the learning experience —i.e. activity structures, assignment of the activity structures to roles and rules which determine the material display—. To represent a learning environment under different complexity levels and mixing didactic techniques, it is necessary for the EML to be expressive enough to specify complex activity structures, dependences, rules, contents, roles, and participants, which can interact to achieve the learning objectives. These structures allow the building of learning flows.

The objective of this paper is to evaluate the expressiveness of EMLs to represent learning structures (i.e. learning scenarios, UoL or learning flows), in particular IMS LD, LAMS and LPCEL. Such evaluation is realized over the analysis of different Workflow Control Patterns (WCP) [3] and the expressiveness of EMLs to represent those structures. The rest of this paper is structured as follows: section 2 presents the WCP; after that, section 3 presents EML characteristics and section 4 relates the WCP with EMLs to end with some conclusions.

2 Workflow Control Patterns

Workflows are defined on the ICT field by the Workflow Management Coalition as “The computerised facilitation or automation of a business process” [4]. Workflow Management Systems (WMS) have become gradually a key technology for business process automation, providing great support for organizational aspects, user interface, monitoring, accounting, simulation, distribution and heterogeneity [5]. When designing a WMS, there are two core components needed to think about: The Workflow Engine and the Workflow Language.

The Workflow Engine [4] is the basic workflow management control software responsible for process creation and deletion, control of the activity scheduling within an operational process and interaction with application tools or human resources. It also provides the run time execution environment for a workflow instance and is responsible for part or all of the runtime control environment within an enactment service. The workflow enactment service provides the run-time environment in which process instantiation and activation occurs, utilising one or more workflow management engines, responsible for interpreting and activating part, or all, of the process definition and interacting with the external resources necessary to process the various activities. Typically a Workflow Engine provides facilities to handle: (1) interpretation of the process definition; (2) control of process instances as creation, activation, suspension, termination; (3) navigation between process activities as sequential or parallel operations, deadline scheduling, interpretation of workflow relevant data and others; (4) sign-on and sign-off of specific participants; (5) identification of workitems for user
attention and an interface to support user interactions; (6) maintenance of workflow control data and workflow relevant data, passing workflow relevant data to/from applications or users; (7) an interface to invoke external applications and link any workflow relevant data; and (8) supervisory actions for control, administration and audit purposes.

A workflow engine can control the execution of a set of instances of a process or sub-process within a defined scope, determined by the range of object types and their attributes specified in a Workflow Language process definition(s). WCP [3] have been identified with the aim of delineating the fundamental requirements that arise during business process modelling on a recurring basis and to describe them in an imperative way. So far, 43 WCP have been identified classified in Basic Control Flow, Advanced Branching and Synchronization, Multiple Instance, State Based, Cancellation and Force Completion, Iteration, Termination, Trigger and Disclaimer.

Learning flows range from basic activity structures such as sequence, parallel, loops, split and union; up to more complex structures such as multiple choice and multiple union. The EML evaluation in this paper focuses on 10 WCP including all the Basic Control Flow Patterns and some of the Advanced Branching and Synchronization and the Iteration Patterns, because of their capability to represent the most common learning scenarios. Further work can be done to analyze the rest of the patterns.

Basic Control Flow Patterns capture elementary aspects of process control, namely: sequence, parallel split, synchronization, exclusive choice, and simple merge. Advanced Branching and Synchronization Patterns characterise more complex branching and merging concepts. Although relatively commonplace in practice, these patterns are often not directly supported or even able to be represented in many commercial offerings. The patterns of this category included in the evaluation are: multi-choice, structured synchronizing merge, multi-merge and structured discriminator.

Iteration patterns deal with capturing repetitive behaviour in a workflow. Structured Loop is the Iteration pattern included in the evaluation, and it is described as the ability to execute an activity or sub-process repeatedly. The loop has either a pre-test or post-test condition associated with it that is either evaluated at the beginning or end of the loop to determine whether it should continue. The looping structure has a single entry and a single exit point.

3 Characteristics of Design, Execution and Control of EML

An EML is a semantic information model and binding, describing the content and process within a UoL from a pedagogical perspective in order to support reuse and interoperability [2]. The temporal order in which the various learning activities unfold in a UoL is called learning flow [6]. Koper defines a number of requirements for EMLs (i.e. formalization, pedagogical flexibility, reproducibility, personalization, interoperability and sustainability, reusability, among other [1]. And due to the complexity levels and the vast didactic techniques, EMLs need to support Complex Learning Process (CLP) learning flows. A CLP [7, 8] is the result of the dynamic and unanticipated integration of mixed pedagogies and resources, and based upon the collaboration of instructors and learners, within a learning process. From a constructivist viewpoint, CLP recognize that learning is an active process, which is built upon experiences that are achieved by entwining contents, context, and educational objectives into a learning process. The learning flow of a CLP consists in the composition of complex structures, which are formed by different building blocks (e.g. activities, resources, roles, dependences, restrictions, etc.). These elements interact in a complex and collaborative manner to achieve the learning objectives. Moreover, the composition of such structures cannot be fixed or programmed in design time, but on a dynamic and unanticipated basis. In some cases, the initial form of a CLP must be adapted and redefined as the collaboration of learners and instructors changes. This is usually done in the run time, without missing the achieved state of the learning process. In order to provide CLP support, new characteristics have been defined by [7]. These include: learning flow description: EMLs need to be expressive enough to specify complex and dynamic structures (e.g. activities, dependences, rules, contents, roles, scenarios and participants); dynamic and unanticipated composition; separation of learning process and service, i.e. semantic interoperability; (5) learning service availability and containment and transaction support.

The IMS LD specification includes a language to describe learning experiences as UoLs, and provides a conceptual framework for the specification of learning process designs, which have to be afterwards deployed onto a computer-based execution engine. The LAMS approach has taken a step further by integrating design and execution of learning activities into the same LD language, design environment and execution engine. LPCFL (Learning Process Execution and Composition Language), provides a framework including the appropriate languages primitives to describe execution-aware learning designs, but still lacks of an implementation.

There is a big necessity of EMLs to represent workflow control structures where students and faculty staff may realize learning activities in a pedagogical sense. The WCPs [3] are a good starting point to apply the identified patterns into the learning field to achieve this objective.

4 Workflow Control Patterns and EML

Although WCP provide a good point of reference to design learning flows and evaluate the expressiveness of EMLs, it is important to differentiate between learning and workflows. In a learning environment the execution of a process activity does not guarantee that the objectives have been achieved, meanwhile in workflows, the execution of a process activity assures a state change. In other words, the completion of a learning activity does not assure new knowledge for the learner.

The EMLs learning flow expressiveness evaluation
Table 1: EMLs learning flow expressiveness evaluation

<table>
<thead>
<tr>
<th>Workflow Control Pattern</th>
<th>IMS-LD</th>
<th>LAMS</th>
<th>LPCEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Control Flow</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Sequence</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2. Parallel split</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Synchronization</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4. Exclusive Choice</td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>5. Simple Merge</td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td><strong>Advanced Branching and Synchronization</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Multi-Choice</td>
<td></td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>7. Structured Synchronizing Merge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Multi-Merge</td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>9. Structured Discriminator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Iteration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Structured Loop</td>
<td></td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

Based on the WCP previously described is presented in Table 1. IMS LD supports learning flows through the script theater metaphor represented by <play>, <act> and activity constructs, which control the synchronization of a scenario. In every act, the <role-part> associates a role with an activity (i.e., learning activity, support activity or activity structure). Acts may contain diverse role-parts, which are concurrent and independent, but synchronous. The learning activities of any activity structure under a role-part must be executed sequentially. As shown in Table 1, IMS-LD only has full support for the sequence pattern. A parallel split can be achieved by defining more than one play due to the concurrency and independence of them. Also, it will lead to a loss of sense of the script metaphor, because a play means a teaching-learning process and there will be many learning processes that were pedagogically only one is supposed to be. Also, there will be no synchronization for the plays later on.

In the LPCEL language, the <LPCEL-Definition> is composed by a <Complex-Learning-Process> element to represent a CLP, allowing the definition of diverse <Component-CLP>. To define learning flow, LPCEL makes reference to a collection of <Component-CLP> elements such as <Sequence>, <Parallel>, <Choice>, <Switch>, <While>, <doWhile>, <Join>, <Split> and <CompensateCA>; and also to a collection of <Basic-Component> elements such as <Action>, <Flow>, <Delay>, <Invoke>, <Gateway>, <Terminate>, <Assign>, <Create> and <CompensateBA>; including their relationships and a number of criteria to indicate the start and termination conditions of the <Component-CLP>, providing support for a vast number of WCP including all the basic ones, as seen in Table 1 but not limited only to them. These primitives allow LPCEL to support WCP, providing the necessary expressiveness to describe and guide the execution of a CLP.

5 Conclusions

It is important to remark that in a CLP, the learning flow concept is not just a matter of the workflow concept applied to a learning environment. As mentioned before, in a learning environment the execution of a process activity does not guarantee the learning objectives have been achieved. In WAMS, the execution of a process activity assures a state change. However, it is of high relevance the need of EMLs to specify complex learning flows, in order to have flexible Learning Management Systems able to support CLP. IMS LD needs to implement basic flow structures, such as parallel flow, and synchronization, and as well as LAMS, they both need to extend learning structures with more complex ones like multiple choice and merge, discrimination, and state-based flow, in order to provide full support for CLP learning flows. As well, it is important for new EMLs versions to take into account the characteristics shown in this paper. LPCEL provides the broadest support for learning flows. Future work relies upon the construction of the LPCEL engine and LPCEL editor.

6 Acknowledgements

The work is partly funded by the Avanza subprogramme (project TSI-020501-2008-53), the PCI subprogramme (project A/018126/08) of the Spanish Government and the Distributed and Adaptive Systems Lab for Learning Technologies Development, DASL4LTD (C-QRO-17/07) from the Tecnológico de Monterrey, México.

References