Journal of Universal Computer Science, vol. 19, no. 6 (2013), 805-830
submitted: 27/7/12, accepted: 25/3/13, appeared: 28/3/13 © J.UCS

LeadFlow4LD: A Method for the Computational Representation of the Learning Flow and Data Flow in Collaborative Learning

Luis Palomino-Ramírez
(ITESM, Campus Guadalajara, Mexico
lpalomin@itesm.mx)

Miguel L. Bote-Lorenzo, Juan I. Asensio-Pérez
(University of Valladolid, Spain
migbot@tel.uva.es, juaase@tel.uva.es)

Laurence Vignollet
(University of Savoie, France
laurence.vignollet@univ-savoie.fr)

Yannis A. Dimitriadis
(University of Valladolid, Spain
yannis@tel.uva.es)

Abstract: A computer representation of teaching-learning processes in collaborative learning settings consists of modelling not only the sequence of learning activities and educational resources as existing learning design languages propose, but also modelling both the sequence of invocations of tools needed to carry out the learning activities and the flow of data among those tools. Existing data flow approaches only model data with activities but not data with tools. In this paper, we present LeadFlow4LD, a learning design and workflow-based method to achieve such a computational representation of collaborative learning processes in an interoperable and standard way. The proposed method has been assessed through the specification and enactment of a variety of non-trivial collaborative learning situations. The experimental results indicate that the level of expressiveness of the proposal is adequate in order to represent the flow of tools invocations and data which was missing in other existing research approaches.

Keywords: Data Flow, Learning Design, Learning Flow, Workflow, IMS LD, CSCL
Categories: L.3.0, L.3.6

1 Introduction

Educational Modelling Languages (EMLs) have been proposed as a means for instructional designers and educators to describe learning processes in a computer-interpretable form [Rawlings, 02]. Representing the learning process as prescribed scripts by means of an EML opens the possibility of using computer-based systems to automatically guide participants throughout the sequencing of activities, the sequencing of tools, as well as to manage the flow of information on their behalf [Koper, 08]. Moreover, by specifying the learning process using a standard EML, educators around the world could define the logic and rationale of their designs in an interoperable and standard way, thereby permitting design interchange between educators and tools, and reuse of designs in different contexts and situations [Koper, 08].

There are different EMLs that can be employed to specify a learning process [Koper, 01][Rodríguez-Artacho, 04][Martel, 06][Caeiro-Rodriguez, 06], including IMS Learning Design (IMS LD) [IMS, 02]. IMS LD is an EML that follows the so-called learning design (LD) approach [Conole, 13], which pays special attention to the learning activities –performed by participants while interacting with data produced by tools along the learning process,— instead of just focusing on the delivery of the educational content, as did previous approaches [Britain, 04]. With background in workflow domain [IMS, 02] [Marino, 07] [Vignollet, 10], the main goals of the LD approach become similar to those in workflow [van der Aalst, 02]: the process automation, the reuse of the process definition, and the interchange of the process definition among educators and tools.

IMS LD allows the specification of many aspects of a learning process, aspects such as: the roles to be played by participants, the sequence of activities to be performed by participants, and the tools and documents that can be employed to support each activity. However, there are two important aspects of learning processes that cannot be specified with IMS LD: the relationship between data and tools (IMS LD only specifies the relationship between data and activities) [Bordies, 12] [König, 10] [Caeiro-Rodríguez, 10] [Neumann, 09] [Miao, 05] [Peter, 05] [Wilson, 05], and the sequence of tools that are expected to be employed by participants in the same activity [Palomino-Ramírez, 08b]. These two aspects imply that the data flow among the tools employed to support the learning process cannot be fully automated. As a consequence, the participants in the learning process take responsibility for handling such data flow; i.e. taking the output generated by a given tool and using it as the input of another tool.

IMS LD is the de facto standard for the LD community and approach [IMS, 02] [Derntl, 11][König, 10]. IMS LD has not yet been widely adopted due to several organizational and technological reasons [Griffiths, 08]. Nevertheless, there is a renewed interest [Alario-Hoyos, 13][Katsamani, 11] in using IMS LD as an interoperability format to deploy learning designs produced by multiple existing authoring tools (e.g. OpenGLM [Derntl, 11b], Collage [Hernández-Leo, 06], and CADMOS [Katsamani, 11]) into mainstream and widely adopted Virtual Learning Environments (VLEs) (e.g. Moodle, LAMS, and Blackboard). These recent research efforts, together with the large set of available IMS LD compliant learning design authoring tools, still suggest the significant research challenge required in order to overcome the expressive pitfalls of IMS LD while still trying to maintain the validity of existing IMS LD tools as much as possible.

The aforementioned limitations are especially relevant within the context of Computer Supported Collaborative Learning (CSCL) in which different researchers propose the use of IMS LD to describe collaborative learning processes [König, 10] [Paramythis, 08] [Hernández-Leo, 07] [Dan, 07]. Such relevance is due to the fact that many collaborative learning processes include complex data flows that are difficult for participants to handle, thus hindering the realization of learning activities. In this sense, several findings reported in [Palomino-Ramírez, 08] suggest that the manual handling of data flow in collaborative learning processes unnecessarily increases the cognitive load of participants and leads to error-prone situations.

Shown in [Palomino-Ramírez, 07], the aforementioned limitations can be tackled by combining the use of IMS LD with a standard workflow language. In this approach, the workflow language is employed to specify the relationship between data and tools as well as the sequence of tools, thus enabling the automatic handling of the data flow within the context of a learning process. However, and in spite of the evidences supporting the feasibility of the approach, the proposal of a detailed method to specify learning processes following such an approach is still missing in the literature. This paper introduces a method called LeadFlow4LD that combines IMS LD with the XPDL (XML Process Definition Language) workflow language [Norin, 02] to specify learning processes in a computer interoperable way that enables the automatic handling of data flow.

The rest of the paper is structured as follows. Section 2, introduces a case study and the related work in order to show the nature of the problem. Section 3 describes the proposed method. Next, section 4 describes a prototype enactment system specified according to the proposed method and used as proof-of-concept to demonstrate that a collaborative learning process can be enacted. Section 5 presents the assessment of the expressiveness and reusability of the description of collaborative learning processes specified according to the proposed method. Finally, conclusions and future work can be found in section 6.

2 The Data Flow Problem in Learning Design

In this section a simple but significant collaborative learning (CL) situation is described in order to illustrate the lack of expressiveness of IMS LD when modelling the data flow. The section also analyses different research initiatives that have proposed ways of modelling the data flow.

2.1 An Illustrative Example of the Data Flow Problem in LD

The peer-review [Bartels, 03] is a well-known CL technique in which students first generate an artefact that is then reviewed by one or more of their peers. The CL process based on this technique typically follows the structure depicted in Fig. 1. The structure starts with an editing activity (A1) that is supported by an editing tool (T1) employed to generate an artefact (D1). Then, a reviewing activity (A2) takes place with the support of a reviewing tool (T2) that employs the artefact generated in the previous activity (D1) as input. The sequence of learning activities (A1, A2) along with the tools (T1, T2) employed to support them define what is called the learning flow structure. The sequence of tool invocations (T1, T2), as well as the data passed between them (D1) define what is called the data flow structure.

It is noteworthy that similar CL processes, also called CL situations [Osuna-Gómez, 99], can be derived from the peer-review structure depending on the grouping of participants as well as on the actual data flow among the tools used by the different groups. Fig. 2(a) illustrates the case of a peer-review situation in which participants are grouped in pairs (p1, p2) while Fig. 2(b) represents an example a peer-review situation with participants grouped in triples (p1, p2, p3). Both figures show the need of providing a different instance of each tool for each participant in a group (e.g. instance T1-1 of the editing tool for participant p1) as well as the fact the tool instances handle different instances of the artefacts (e.g. instance T1-1 generates artefact D1-1 while instance T1-2 generates artefact D1-2). Fig. 2(c) shows another example of a peer-review situation with participants grouped in triples in which the data flow is different from the one considered in the previous situation (e.g. tool instance T2-3 uses artefact instance D1-1 as an input in one case and artefact D1-2 in the other).

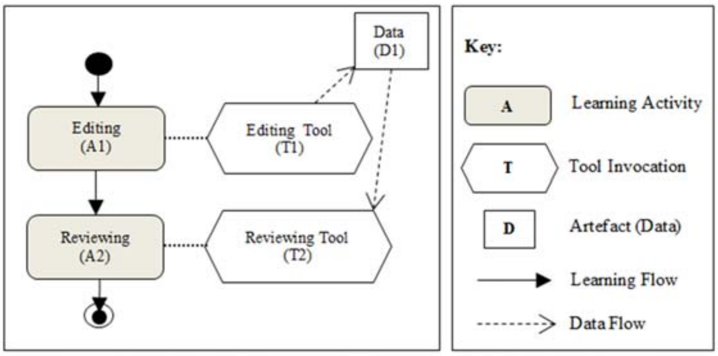


Figure 1: The CL structure of many peer-review CL situations

Unfortunately, IMS LD cannot be employed to specify any of the CL situations illustrated in Fig. 2 since it does not support the specification of the data flow between tools, as stated by several authors in LD literature [Bordies, 12] [König, 10] [Caeiro-Rodríguez, 10] [Neumann, 09] [Miao, 05] [Peter, 05] [Wilson, 05]. Moreover, it is not possible to specify the sequence in which several tools should be used by the learners in an activity in the cases where more than one tool is needed [Palomino-Ramírez, 08b]. As a consequence, it is not possible to use IMS LD to generate computer-interpretable descriptions of CL situations, descriptions that could be employed to automatically determine the sequence of tool invocations and manage the exchange of data. Both issues are called here “the data flow problem in LD”.

2.2 Related Work

Some authors [Koper, 08] [Miao, 08] have proposed ways of exploiting IMS LD expressiveness in CL so as to include information about data flow between activities in IMS LD descriptions of learning processes. In this approach, the data flow is specified between activities but not between tools. This implies that the system does not have enough information to deliver automatically an input data to the proper tool, thus the system relies on user intervention.

Other authors [König, 10] have proposed an extension of IMS LD in order to provide sufficient support for (collective) artefacts. The extension models the flow of artefacts between activities, however, the extension does not enable the provision of the information required to automatically handle the data flow between tools.

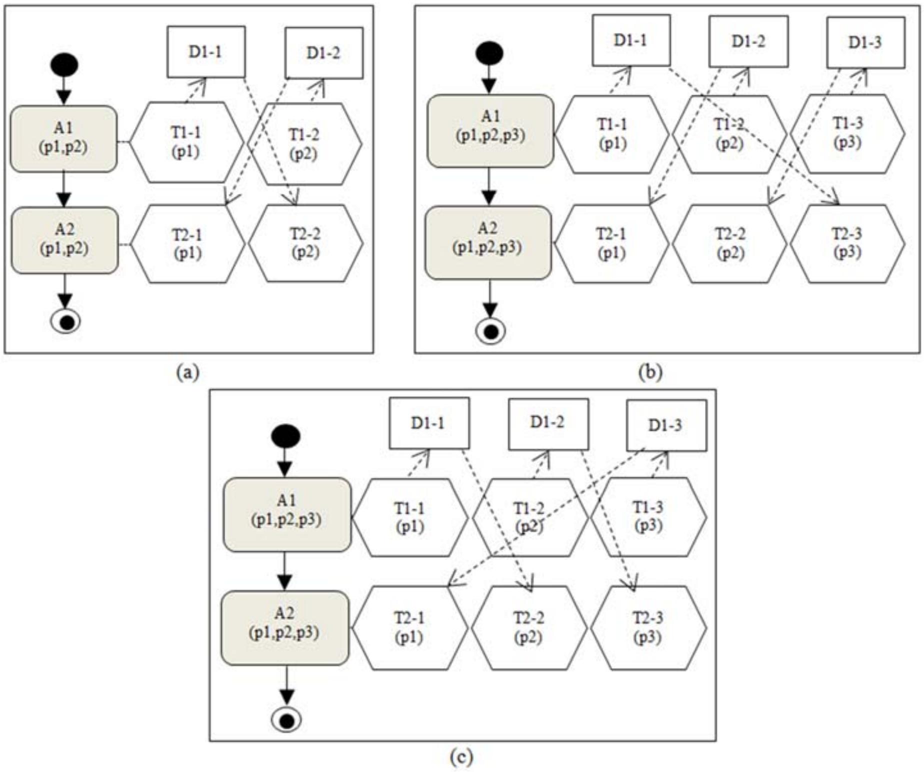


Figure 2: Three CL situations derived from the same peer-review structure: (a) participants grouped in pairs, (b) and (c) participants grouped in triples with different data flows

Another work [Vantroys, 03] studied the possibility of substituting IMS LD with the XPDL standard workflow language. The authors showed that XPDL enabled the automatic handling of the data flow in learning processes, but the distinction of the collaborative work performed by different students participating in a collaborative activity is still necessary. This is the reason why they proposed a new workflow language that extended the capabilities of XPDL. The lack of interoperability of this solution has hindered its adoption.

An alternative approach, proposed by [Miao, 05], merges the IMS LD and XPDL meta-models in order to create a new EML called CSCL scripting language. The main goal of yet another EML addresses different IMS LD issues in the CL specification including the data flow between tools. Again, the proposed CSCL scripting language is not an interoperable EML with the current LD standard (IMS LD).

There are also other works in the literature that study the data flow problem in LD but they do not propose any solution. For example, Peter and Vantroys state that IMS LD lacks data flow management features [Peter, 05], but they do not go further in defining the associated defects. Wilson criticizes that IMS LD does not consider whether the results of a tool are going to be exported to other tools [Wilson, 05]. Moreover, he does not provide any solution to this issue. Dalziel states that IMS LD requires new mechanisms in order to transfer data between tools with a possible information processing between them [Dalziel, 06], and yet he does not propose the associated mechanism. Palomino-Ramírez et al. state that IMS LD lacks system-support for addressing the automation of data flow management [Palomino-Ramírez, 07] [Palomino-Ramírez, 08b], but their actual solution is still missing. The IMS LD expert group states in [Neumann, 09] that IMS LD does not allow management of data flow between the tools since these are viewed as “black boxes” by the learning design. Caeiro-Rodríguez et al. refer to the data flow problem as a limitation of the IMS LD expressivity in CL [Caeiro-Rodríguez, 10]. Vignollet et al. describe a generic specification of the data flow problem in the LD field [Vignollet, 09]; however, they state that it is just a conceptual approach to be taken into account by future EMLs.

3 LeadFlow4LD

Overviewed in Fig. 3, LeadFlow4LD is a method to describe CL processes that combines the use of standards languages of LD and workflow in order to provide computer-interpretable representations of CL structures and situations. According to Fig. 3, LeadFlow4LD involves a series of steps resulting in the creation of five documents which specify all the information needed to completely describe a CL process. This section describes the steps that the instructional designer and educator must follow in order to achieve such descriptions.

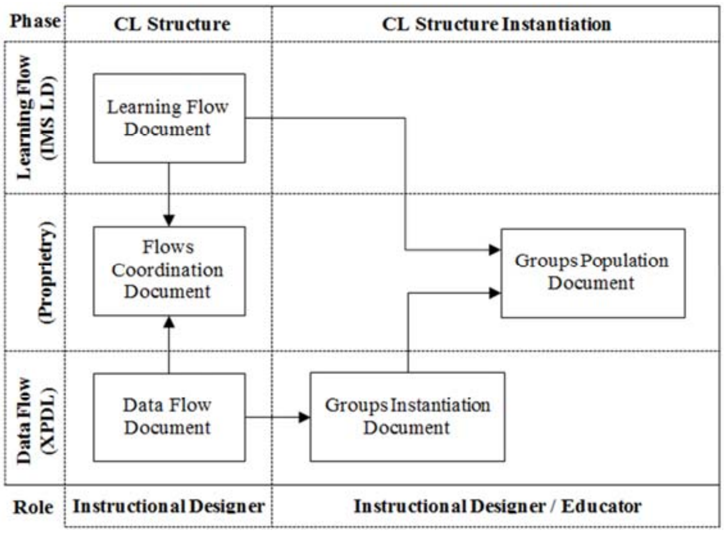


Figure 3: Overview of the LeadFlow4LD method

3.1 Collaborative Learning Structure

First of all, as illustrated in Fig. 3, the instructional designer should define a CL structure by generating the learning flow and the data flow documents. On the one hand, the **learning flow document** specifies the sequence of learning activities to be carried out in the CL scenario, and the roles to be played by participants in each activity (teachers, tutors, learners, etc.). IMS LD is the language employed for coding the learning flow document. We employ IMS LD due to its widespread adoption by the technology-enhanced learning community, as well as to its capability for expressing all the concepts involved [Hernández-Leo, 05]. On the other hand, the **data flow document** specifies the sequence of invocations of tools that will take place during the learning process, the data generated by the tools, as well as the relationship among data and tools. Additionally, this document describes the roles of participants in each tool invocation. XPDL is the language employed for coding the data flow document due to its capability for expressing all the concepts involved [Norin, 02] and because of its wide acceptance and usage as a workflow process definition interchange format [WfMC, 94].

Once the learning flow and data flow (workflow) have been specified in their corresponding documents, the information that states the relationship between learning activities and tool invocations (represented as XPDL activities) should be specified since both flows are interdependent. Furthermore, since the tools are intended to help the learner to reach the objectives of the learning activities, it seems reasonable for LeadFlow4LD to follow a master-slave coordination mechanism in which the learning flow plays the master role and the workflow plays the slave role for deciding which of the flows is actually guiding the actions of the learners. Consequently, the learning flow should identify the learning activities in which the corresponding tool invocations in the workflow should be initiated by the learners involved. Similarly, the workflow should identify workflow activities in which the workflow should be started as well as identify the workflow activities in which the control should be switched back to the learning flow. Thus, a simple way to codify this relationship between learning activities and workflow activities is by defining a list of master and slave transitions. A master transition defines a connection from a learning activity to a starting workflow activity, while a slave transition defines a connection from an ending workflow activity to a learning activity. Such coordination between flows is specified by the instructional designer in a document called the **flows coordination document**, which is codified in a proprietary specification based on the XML schema and shown in Fig. 4.

3.2 Collaborative Learning Structure Instantiation

The instantiation of the learning flow structure by the simple assignment of participants to learning activities is sufficient for the purpose of enacting specific CL situations (“create cases” in workflow terminology). Nevertheless, it is not the same regarding the instantiation of the data flow structure. In collaborative learning, the simple assignment of participants to abstract tools is not sufficient for the purpose of enacting specific CL situations. Instead, as stated during the problem review, it is necessary for the instructional designer, or even the educator, to first define the groups instantiation; that is, the number of data and tool instances (related to the number of groups) as well as the relationships among them (the interaction among groups). Following, the educator defines the groups population. That is, the assignment of participants to learning activities and tool instances.

This is what happens, for example, when the instructional designer, or even the educator, has to decide how many groups will participate in a peer-review (how many data and tool instances will be created), and decide about the interactions among groups (which group will review what artefact). But the educator also has to decide about the groups population: participants that will play a certain role; participants that will handle individual or shared tool instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/..">
   <xsd:element name="flows-coordination" type="FlowsCoordinationManifestType"/>
   <xsd:complexType name="FlowsCoordinationManifestType">
      <xsd:sequence>
         <xsd:element name="uoLeadFlow4LD-info" type="UoLeadFlow4LDInfoType"/>
          <xsd:element name="activities" type="ActivitiesType"/>
          <xsd:element name="transitions" type="TransitionsType"/>
          <xsd:element name="tools" type="ToolsType"/>
      </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string"></xsd:attribute>
        </xsd:complexType>
   <xsd:complexType name="ToolsType">
      <xsd:sequence>
         <xsd:element name="tool" type="ToolType"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="ActivitiesType">
      <xsd:sequence>
         <xsd:element name="learning-activity" type="LearningActivityType"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="MasterTransitionType">
      <xsd:sequence>
         <xsd:element name="from" type="ActivityRefType"/>
          <xsd:element name="to" type="ToolRefType"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="SlaveTransitionType">
      <xsd:sequence>
         <xsd:element name="from" type="ToolRefType"/>
```

Figure 4: Flows coordination XML schema

For this reason, LeadFlow4LD distinguishes between the CL structure and its instantiation. But differently from a typical IMS LD script where the CL structure instantiation is not specified at all, in LeadFlow4LD the instantiation has to be made in two separated documents: the groups instantiation document and the groups population one. On the one hand, the **groups instantiation document** is defined by the instructional designer, or even the educator, from the data flow structure document, but taking into account the interactions between data and tool instances (the number of groups) necessary to describe the collaborative interaction of the data flow. This document is specified in XPDL similar to the data flow document from where it derives. On the other hand, the **groups population document** is accomplished by the educator to specify the particularities of the CL situation, such as the assignment of participants to the learning activities and tool instances. This document does not follow a standard specification, since the specification of the instantiation (the assignment of participants to roles at the time of instantiation) of process modelling languages is still an open issue [Caeiro-Rodríguez, 08], even in the workflow domain [Decker, 08].

Thus, LeadFlow4LD proposes an XML representation based on a conceptual model and schema proposed by iCollage [Hernández-Gonzalo, 08], which specifies the instantiation of groups as role instances as well as the assignment of participants to groups in learning flow structures. LeadFlow4LD has extended this model to include the assignment of participants to tool instances.

The use of a non-standard specification in LeadFlow4LD might hamper its adoption since LeadFlow4LD scripts would require the use of specific enactment infrastructures (see section 4). Nevertheless, a significant part of the modelling effort could still be of use even in the absence of such specific infrastructures. For example, the learning flow document could be interpreted (and eventually reused) by existing IMS LD engines (obviously, losing the data flow modelling).

3.3 LeadFlow4LD Document Distribution

In order to get a more effective distribution of the different collaboration scripts specified according to LeadFlow4LD, the five documents summarized in Table 1 can be packaged in a unit of learning called **UoLeadFlow4LD** following the IMS Content Package Specification [IMS, 03]. This approach would encourage instructional designers and educators to improve a given CL process (a CL structure) by aggregating new groups instantiation and population documents. This separation in different documents of a CL process specified according to the proposed method is aimed at fostering different levels of reusability: the learning flow, data flow and flows coordination documents –all of which actually reflect the pedagogical strategy- can be reused in different CL situations, while only the groups instantiation and population documents would need to be changed.

Document	Description	Specification	Responsible
Learning flow	Defines the sequence of learning activities.	IMS LD	Instructional Designer
Data flow	Defines the sequence of tool invocations and the data flow between tools.	XPDL	Instructional Designer
Flows coordination	Defines the relationship between learning activities and tool invocations.	Proprietary	Instructional Designer
Groups instantiation	Defines the number of data and tool instances as well as the relation among them.	XPDL	Instructional Designer/Educator
Groups population	Defines the assignment of participants to learning activities and tool instances.	Proprietary	Educator

Table 1: Summary of documents generated by the LeadFlow4LD method

4 Enactment of a UoLeadFlow4LD

A UoLeadFlow4LD describes CL structures and situations from the points of view of the instructional designers and educators. Unfortunately, the UoLeadFlow4LD cannot be executed “as-is” by learning flow (IMS LD compliant) and workflow (XPDL compliant) engines because of a proprietary specification which describes the coordinated execution of both engines.

Therefore, three different implementation approaches have been identified in order to execute the UoLeadFlow4LD. The first approach consists in directly executing a UoLeadFlow4LD throughout a LeadFlow4LD-compliant engine. The drawbacks, however, are that the approach does not take advantage of existing learning flow and workflow standard engines and that, of course, the implementation of the LeadFlow4LD engine would be a very hard task. Rather than that approach, we propose to re-use existing engines through master-slave implementation approaches. Then, we propose to implement a master-learning flow and slave-workflow approach, because we consider that the learning activities defined in the learning flow, describe the main steps of the learning situation and should be considered as the master flow rather than the opposite approach.

The enactment of a UoLeadFlow4LD using the selected approach requires a software component called LeadFlow4LD parser. The main task of the parser is to automatically generate (from a set of defined rules) two new documents, called the **synchronized learning flow document** and the **synchronized workflow document**. While the former, is an IMS LD compliant UoL, the latter is a workflow process file written in the workflow enactment language such as XPDL.

The way in which both engines invoke each another is described in the sequence diagram shown in Fig. 5. First, participants start their learning activities using the graphical user interface that LeadFlow4LD delivers for all participants. Second, when participants decide to use tools (delivered by the workflow engine), the workflow engine launches tool clients, which in turn invoke the associated Web tool. Third, when participants finish using the last tool (from the sequence of tools), then the control flow is switched back to the learning flow engine, thereby implying that participants are able to resume the next learning activity.

As shown in Fig. 5, the engines do not communicate directly. Instead, they communicate through the so-called “engine invoking resources”. The LedFlow4LD parser aggregates the workflow engine invoking resources to the synchronized learning flow document and enables the modification of the workflow variables used as conditions to start the data flow. Similarly, the parser aggregates learning flow engine invoking resources to the synchronized workflow document and enables modification of IMS LD property values –subsequently used as conditions to terminate the learning activity.

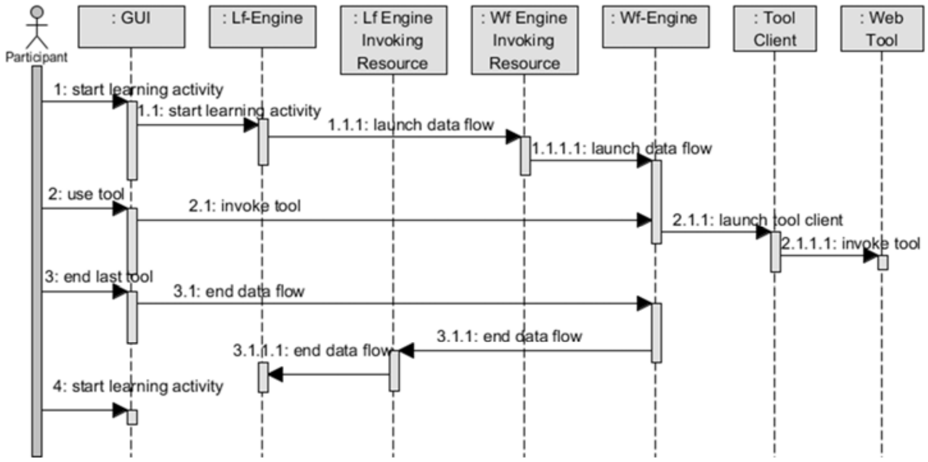


Figure 5: Sequence diagram of the coordinated execution of the learning flow and workflow engines

An existing technological infrastructure has been selected for implementation: Coppercore [Vogten, 05] as the engine able to play the synchronized learning flow document and Bonita [Surhone, 10] as the engine able to play the synchronized workflow document. The former because it is the reference engine used by the IMS LD specification, and the latter because it is open-source and because it supports XPDL. The Coppercore Web service API and the Bonita RESTful Web service API were used to communicate with the learning flow and workflow engines, respectively.

External Web tools are invoked by the Bonita workflow engine by means of the URLs that are included in the groups instantiation document of LeadFlowLD specifications. Such URLs would be invoked in the sequence prescribed by the data flow document. The current enactment infrastructure does not support an automatic way of managing those URLs (creation, modification, deletion of tool instances). That would imply a large increase in the specification burden when dealing with a significantly large number of tools instances. Those instances should be managed manually (by some kind of support staff). Nevertheless, we are currently planning to enhance the LeadFlow4LD enactment infrastructure with our proposed Glue! integration system [Alario-Hoyos, 13] so as to automate the management of third-party external Web tools instances. In the next section, different CL situations have been enacted using this infrastructure.

5 Assessment of Expressiveness and Reusability of a UoLeadFlow4LD

This section presents the assessment of the expressiveness of LeadFlow4LD to specify CL structures, structures that comprise both the learning flow and the data flow in order to be reused in different CL situations. To accomplish this dual objective of expressiveness and reusability, specific CL situations, derived from a simple well-known CL technique, were specified and assessed according to the LeadFlow4LD method. Then, LeadFlow4LD was also tested (regarding expressiveness only) with two other, but more complex, CL situations.

The assessment of expressiveness and reusability was carried out in two parts. First, both the CL structure and the selected CL situations were specified according to the proposed method, then, they were run by means of the aforementioned implementation and used as proof-of-concept to show that collaboration scripts which are specified according to LeadFlow4LD can be executed using available state-of-the-art learning flow and workflow engines.

5.1 Describing Collaborative Learning Situations for Assessment

Four situations, based on the peer-review CL technique, provide the assessment of LeadFlow4LD mainly because they deal with collaborative interactions among data and tool instances (the first assessment objective). Furthermore, the four CL situations proposed share, in pairs, the same structure to properly illustrate the reutilization of the collaboration scripts specified according to LeadFlow4LD (the second assessment objective).

Table 2 lists the selected CL situations for the assessment of LeadFlow4LD following the aforementioned criteria: expressiveness and reusability.

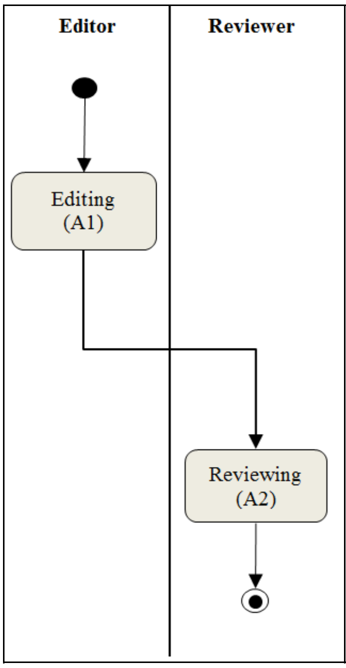
Sit.	Description
1	Two learners edit an artefact individually; then each one reviews the artefact of his/her peer.
2	Four learners edit an artefact in pairs; then each pair reviews collaboratively the artefact of their peers.
3	Three learners edit individually an artefact; then the first learner reviews the artefact of the second learner, the second learner reviews the artefact of the third learner, and the third learner reviews the artefact of the first one.
4	Six learners edit an artefact in pairs; then, working with different peer, each pair review the artefact created by another pair.

*Table 2: Selected CL situations based on the peer-review process used to assess LeadFlow4LD (*expressiveness and reusability)

5.2 The Peer-Review Collaborative Learning Structure

Since all of the selected CL situations derive from the same CL process -the peer-review-, all of them share the same structure: the learning flow, data flow and flows coordination documents.

As illustrated in Figs. 6a and 7a, the instructional designer starts with the CL structure definition comprised by the learning flow and the data flow structures, that is, the sequence of learning activities (editing then reviewing), the sequence of tool invocations (an editing tool and then a reviewing tool invocation), as well as the data flow between tools (the resultant artefact from the editing tool is an input to the reviewing tool). Then, authoring tools (Reload editor for IMS LD [Reload, 12] and Together workflow editor for XPDL [Together, 12]), specify the learning flow and data flow documents, respectively. Specified according to IMS LD and XPDL, part of these documents are shown in Figs. 6b and 7b, respectively. On the one hand, the learning flow document in Fig. 6b describes roles and learning activities as well as the dynamic behavior of the learning process in the *method* tag and which can be interpreted using the theatre play metaphor. For example, the learning process comprises two acts. In the first act, learners playing the editor role follow a script described in the learning activity A1, while in the second act, a learner playing the reviewer role follow a script described in the learning activity A2. On the other hand, the data flow document shown in Fig. 7b describes data (*DataFields*), roles (*Participants*), tools (*Applications*) and tool invocations (*Activities*). The sequence of tool invocations is described by defining transitions (not shown in Fig. 7b), while the relationship between data and tools is described through the relation between the formal parameters of tools (*FormalParameters*) and actual arguments of tool invocations (*ActualParameters*).




```
<imsld:roles>
 <imsld:learner identifier="role-1">
  <imsld:title>Editor</imsld:title>
</imsld:learner>
 <imsld:learner identifier="role-2">
  <imsld:title>Reviewer</imsld:title>
 </imsld:learner>
</imsld:roles>
<imsld:activities>
 <imsld:learning-activity identifier="la-A1">
  <imsld:title>Edition</imsld:title>
 </imsld:learning-activity>
 <imsld:learning-activity identifier="la-A2">
  <imsld:title>Reviewing</imsld:title>
 </imsld:learning-activity>
</imsld:activities>
<imsld:method>
 <imsld:play identifier="play-1">
  <imsld:title>Play</imsld:title>
  <imsld:act identifier="act-1">
   <imsld:title>Act-1</imsld:title>
    <imsld:role-part identifier="rolepart-1">
     <imsld:title>Role Part</imsld:title>
     <imsld:role-ref ref="role-1" />
     <imsld:learning-activity-ref ref="la-A1" />
    </imsld:role-part>
  </imsld:act>
  <imsld:act identifier="act-2">
   <imsld:title>Act-2</imsld:title>
    <imsld:role-part identifier="rolepart-2">
     <imsld:title>Role Part</imsld:title>
```

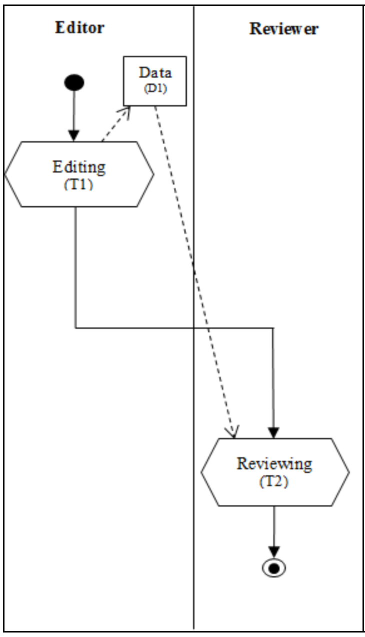
(a) (b)

Figure 6: (a) Design of the peer-review CL structure. (b) Part of the learning flow document specified conforming to IMS LD

Up to this point, the instructional designer has specified the learning flow and data flow (workflow) of the CL structure of the peer-review CL process. However, the flows coordination is still missing because the learning flow and workflow are not independent of each other. The diagram in Fig. 8a illustrates the structural relationship between the learning flow and data flow in regard to all four of the tested CL situations. Relating activities in the learning flow with activities in the workflow, this task is carried out by the instructional designer at design time. The “task” should be understood as: the moments in which the tools give the necessary support to the learning activities as well as the moments in which they are left out. Such flows coordination is specified within the flows coordination document following a proprietary specification as shown in Fig. 8b.

5.3 The Peer-Review CL Structure Instantiation

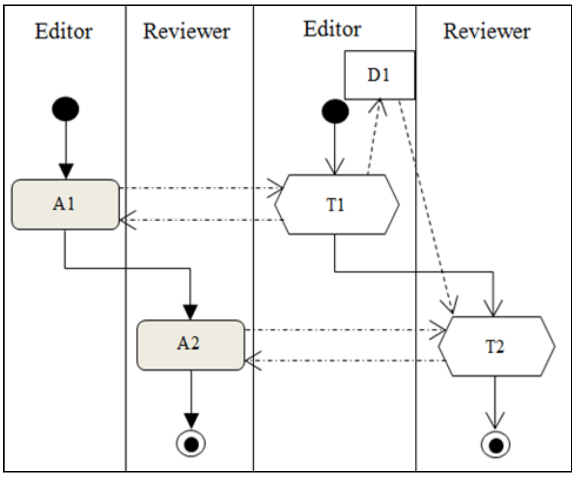
In order to define the selected CL situations of the peer-review process, first, one or more data flow structures instantiations should be defined. Thus, the goal for the instructional designer or educator is to define the necessary data flow structures instantiations to derive all four of the selected CL situations described in Table 2. As explained next, this can be reached by designing only two of these structures.




```
<xpdl:DataFields>
 <xpdl:DataField Id="D1" Name="D1"/>
</xpdl:DataFields>
<xpdl:Participants>
<xpdl:Participant Id="Editor" Name="Editor"/>
<xpdl:Participant Id="Review" Name="Reviewer"/>
</xpdl:Participants>
<xpdl:Applications>
 <xpdl:Application Id="App-T1">
  <xpdl:FormalParameters>
   <xpdl:FormalParameter Id="p-D1" Mode="OUT"/>
  </xpdl:FormalParameters>
 </xpdl:Application>
 <xpdl:Application Id="App-T2">
  <xpdl:FormalParameters>
   <xpdl:FormalParameter Id="p-D1" Mode="IN"/>
  </xpdl:FormalParameters>
 </xpdl:Application>
</xpdl:Applications>
<xpdl:Activities>
 <xpdl:Activity Id="Activity-T1" Name="T1">
  <xpdl:Implementation>
   <xpdl:Tool Id="App-T1" Type="APPLICATION">
    <xpdl:ActualParameters>
<xpdl:ActualParameter>D1</xpdl:ActualParameter>
     </xpdl:ActualParameters>
    </xpdl:Tool>
     </xpdl:Implementation>
      <xpdl:Performer>Editor</xpdl:Performer>
```

(a) *(b)*

Figure 7: (a) Design of the peer-review data flow structure (b) Part of the data flow document specified and conforming to XPDL




```
<tools>
  <tool id="H1" />
  <tool id="H2" />
</tools>
<transitions>
 <master-transition>
  <from activity-ref="A1" />
  <to tool-ref="H1" />
 </master-transition>
 <master-transition>
  <from activity-ref="A2" />
  <to tool-ref="H2" />
 </master-transition>
 <slave-transition>
  <from tool-ref="H1" />
  <to activity-ref="A1" />
    </slave-transition>
 <slave-transition>
  <from tool-ref="H2" />
  <to activity-ref="A2" />
 </slave-transition>
</transitions>
```

(a) *(b)*

Figure 8: Flows coordination (a) design and (b) document

Fig. 9 shows the design of the first data flow structure instantiation shared by the first and second selected CL situations. Both situations require the creation of two groups, and therefore two instances of each editing and reviewing tool. Meanwhile, the interaction among groups (interaction among data and tool instances) remains the same for both situations. Thus, the actual difference between these situations is the way participants share tools: while in the first situation participants manage individual tools, in the second participants share collaborative tools. This can, however, be specified later in the groups population document, while both situations share the same instance of the data flow structure. The instructional designer, or educator, produces such a design from the data flow structure shown in Fig. 7. The creation of two groups (two data and tool instances), produce the data flow interaction described in the first and second CL situations. Finally, the resultant design is specified in XPDL within the groups instantiation document.

Fig. 10 illustrates the design of the second data flow structure instantiation and shared by the third and fourth CL situations. The data flow structure, shown in Fig. 7, produces such a design. The creation of not two, but three groups (three data and tool instances), produce the data flow interaction described in the third and fourth CL situations. Similarly, such design is specified in XPDL within another groups instantiation document.

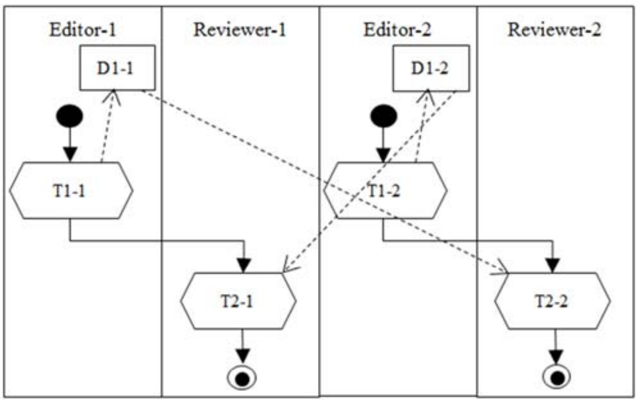


Figure 9: Instantiation of the peer-review data flow structure that is shared by the first and second of the selected CL situations

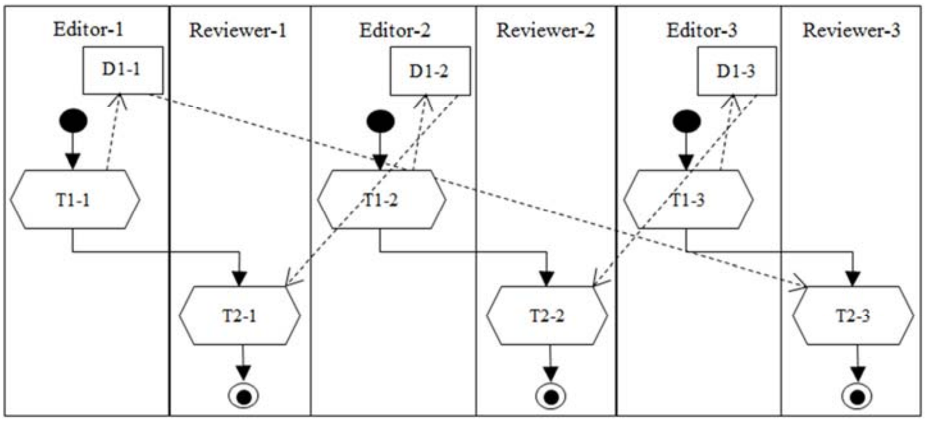


Figure 10: Instantiation of the peer-review data flow structure that is shared by the third and fourth of the selected CL situations

Once the instructional designer defines the groups instantiation documents, the educator specifies the way groups are populated for completing the specific CL situations. This assigns participants to the learning activities and tool instances. A summary of such assignments for all CL situations are shown in Fig. 11a, while part of the specification of the groups population document for the first CL situation is shown in Fig. 11b.

As seen in Fig.11a, all participants are scheduled to play both the editor and reviewer roles for all of the CL situations. Fig.11a also shows the way participants manage individual or shared tool instances.

Learning Activities Population

Role	**CL Situation**			
	1	**2**	**3**	**4**
Editor	p1,p2	p1,p2, p3,p4	p1,p2,p3	p1,p2,p3,p4 ,p5,p6
Reviewe r	p1,p2	p1,p2, p3,p4	p1,p2,p3	p1,p2,p3,p4 ,p5,p6

Tool Instances Population

Tool Instan ce	**CL Situation**			
	1	**2**	**3**	**4**
T1-1	p1	p1,p2	p1	p1,p2
T1-2	p2	p3,p4	p2	p3,p4
T1-3	-	-	p3	p5,p6
T2-1	p1	p1,p2	p1	p1,p5
T2-2	p2	p3,p4	p2	p2,p3
T2-3	-	-	p3	p4,p6

(a)

```
<role-root>
 <role id="Editor" occurrence="1"/>
 <role id="Reviewer"
occurrence="2"/>
</role-root>
<tool-instances>
 <tool-instance id="T1-1"/>
 <tool-instance id="T1-2"/>
 <tool-instance id="T2-1"/>
 <tool-instance id="T2-2"/>
</tool-instances>
<data-instances>
 <data-inst id="D1-1" url="..."/>
 <data-inst id="D1-2" url="..."/>
</data-instances>
<group-population>
 <user id="p1" name="p1"
email="...">
  <role-occu-ref ref="1"/>
  <role-occu-ref ref="2"/>
  <tool-instance-id-ref ref="T1-1"/>
  <tool-instance-id-ref ref="T2-1"/>
 </user>
 <user id="p2" name="p2"
email="...">
  <role-occu-ref ref="1"/>
  <role-occu-ref ref="2"/>
  <tool-instance-id-ref ref="T1-2"/>
  <tool-instance-id-ref ref="T2-2"/>
```

(b)

Figure 11: (a) Assignment of participants to the learning activities (roles) and tool instances for all peer-review CL situations. (b) Part of the groups population document for the first peer-review CL situation

Finally, many other specific CL situations may be derived from the same data flow structure instantiation. This reusability shows the potential of the proposed method.

5.4 Enactment

Once the synchronized learning flow and workflow documents are played by the CopperCore and the Bonita engines, respectively, these documents keep references from the engine invoking resources to the workflow and learning flow engines, thereby producing the engines-level coordination and producing the moments in which the control flow switches from one engine to the other.

Fig. 12 captures the enactment of the first CL situation at the moment when the participant *p2* starts the reviewing activity in the learning flow (see Fig. 12a). Then, when the participant *p2* is ready to give feedback to his/her peer *p1*, the workflow engine automatically launches the reviewing tool (see Fig. 12b). Moreover, the proper artefact previously created by his\her peer *p1* in the previous activity (the editing activity) is loaded automatically in the reviewing tool (see Fig. 12b).

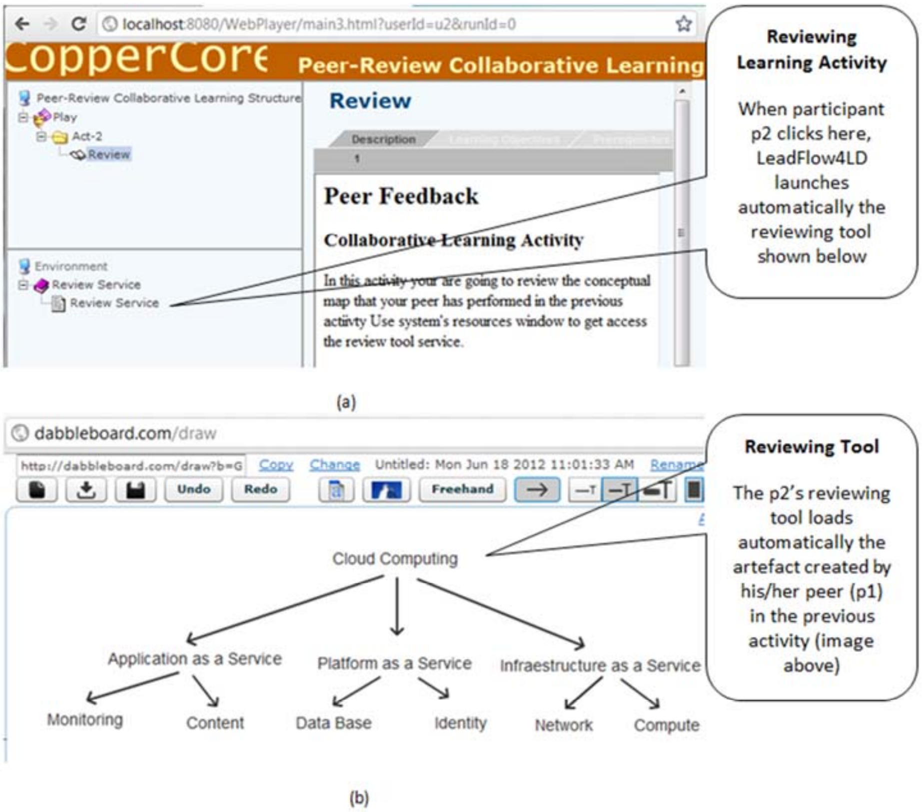


Figure 12: Enactment of the first peer-review CL situation (a) Reviewing activity presented automatically to the participant p2 by the learning flow engine (Coppercore) (b) Reviewing tool instance launched automatically by the workflow engine (Bonita) and presented automatically to the participant p2

Thus, the results obtained indicate that the data flow between tools can be specified through the proposed method. Furthermore, the system does not require user intervention for handling the data flow, thus avoiding the increase of the users' cognitive load. The system also avoids the use of situations with prompt errors for users, since the user does not have to be concerned with the data to be used in a learning activity; the system automatically delivers the necessary data to the tools. Moreover, the four selected CL situations were enacted just by changing the groups population document. This approach reused the learning flow, data flow and flows coordination documents.

5.5 Expressiveness of other more complex CL situations

Additionally to the peer-review, LeadFlow4LD has been successfully tested with other more complex CL situations. Such is the case for example of the knowledge convergence script [Miao, 08] and the so-called MOSAIC learning experience [Palomino-Ramírez, 08] which are introduced next but not discussed in detail due to space restrictions.

On the one hand, the knowledge convergence script is a generalization from the peer-review that it is well documented in the literature [Weinberger, 04]. Consider the following CL situation comprising editing, reviewing and reflection activities (the structural design is shown in Fig. 13). During the editing activity a group of three learners are required to write three reports (D1) about three different cases. Then, during the reviewing activity, each learner critiques, in rounds, the report of each of the other peers. At this stage the situation becomes interesting because each learner has to criticize two learners' reports in sequence within the same learning activity: the reviewing one. Finally, during the reflection activity each learner presents their thoughts. This CL situation was successfully specified and enacted through LeadFlow4LD, thus showing the expressiveness of the proposed method, expressiveness to specify the sequencing of tool invocations in a learning activity.

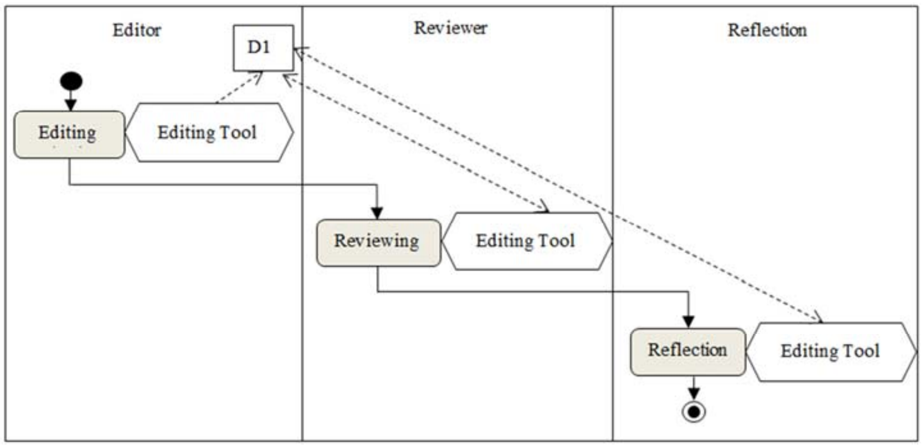


Figure 13: Representation of the learning flow and data flow structures of the knowledge convergence script

On the other hand, 12 learners participated in the MOSAIC learning experience. This experience is a CL situation derived from a combination of three well known CL structures: a 3-level pyramid, jigsaw and peer-review (the structural design is shown in Fig. 14). The main goal of this CL situation is to use the peer-review as a means to improve the outcome produced at each level or phase. The first level of the pyramid comprises 3 phases. In the first individual phase, each student reads one of three selected cases (4 students read the same one) in order to build an outcome. In the second collaborative phase all four students that read the same case work together as a group of experts. First, they make a peer-review of their previous work and then they produce a new improved outcome. In the third phase, 3 jigsaw groups, of 4 expert learners each, are formed while peer-reviewing remains again as the main activity. In the second level of the pyramid, two jigsaw groups work together in a peer-review fashion while reaching a consensus on the outcome. Finally, in the third level of the pyramid all of the 12 students present their thoughts. The successful enactment of this CL situation showed again the expressiveness of LeadFlow4LD to specify not only the data flow between tools and the sequencing of tool invocations, but also the combination of complex CL structures.

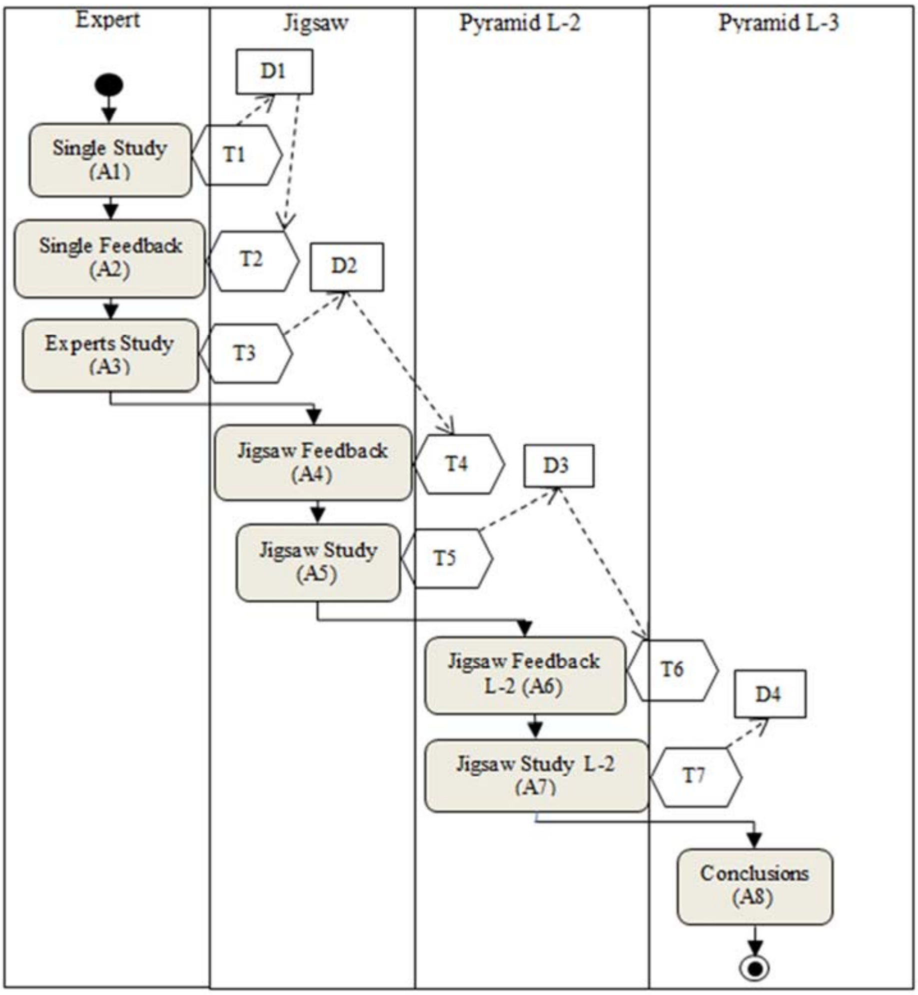


Figure 14: Representation of the learning flow and data flow structures of the MOSAIC learning experience

Due to space restrictions, all of the details of the LeadFlow4LD specification are not included in this paper. Nevertheless, Fig. 14 shows a simplification of the learning flow and data flow structures of the MOSAIC learning experience, while Table 3 shows the number of data and tool instances just to provide a glimpse of the complexity of the CL situation as well as the expressivity affordances of LeadFlow4LD.

Tool	Tool Description	Instances	Data	Data Description	Instances
T1	Conceptual map individual tool (individual study)	12	D1	Conceptual map (individual study)	12
T2	Conceptual map individual tool (individual feedback)	12	D2	Conceptual map (experts study)	3
T3	Conceptual map collaborative tool (experts study)	3	D3	Conceptual map (jigsaw study)	4
T4	Conceptual map collaborative tool (jigsaw feedback)	4	D4	Conceptual map (jigsaw study pyramid-level 2)	2
T5	Conceptual map collaborative tool (jigsaw study)	4			
T6	Conceptual map collaborative tool (jigsaw feedback pym-L 2)	2			
T7	Conceptual map collaborative tool (jigsaw study pyramid-L2)	2			

Table 3: Data and tool descriptions shown in Fig. 13 as well as the number of data and tool instances used by the MOSAIC learning experience

6 Conclusions and Future Work

The computer representation of a learning process comprising both a learning flow and data flow opens the possibility of using a computer-based system first, to guide participants through the sequence of learning activities; second to guide participants through the sequence of tools; and third, to automatically manage the data exchanged among tools.

Different existing EMLs are employed to specify a learning process, yet IMS LD has emerged as the *de facto* standard in the LD community and approach. IMS LD can specify the learning flow and the data flow between activities. IMS LD cannot, however, specify the data flow between tools. This becomes a relevant issue, especially in the CSCL field, where the data flow management is non-trivial. Complex collaborative interactions must manage the artefacts and tools that support a learning activity. When participants in the learning process, not the system, take the output generated by a given tool and use it as the input of another tool, then participants have to make the additional effort of locating and retrieving data and tools. Current solutions found in the literature focus on the proposal of other EMLs. Unfortunately, these solutions are not interoperable with current standards.

Additional to this data flow automation issue, there exists the process reuse issue. CL processes require not only the definition of the structure, but also the structure instantiation which, in turn, reflects the interaction between groups (collaborative interaction between data and tool instances), as well as the way learning activities and tool instances are populated by participants. Nevertheless, current approaches found in the literature focus on the specification of CL situations instead of the CL structure, thereby precluding the reuse of the process definition for different CL situations.

In this paper, we propose an interoperable solution to both issues, called LeadFlow4LD. This solution consists of a method for specifying both learning and data flows. It respectively uses, IMS LD and a standard workflow language such as XPDL. XPDL specifies the sequencing of tools and the data flow between tools. LeadFlow4LD proposes the specification of a CL structure and situations within a unit of learning called UoLeadFlow4LD. A UoLeadFlow4LD comprises five documents: 1. the learning flow, 2. the data flow and 3. flows coordination documents which actually describe the structure of the learning process, 4. the groups instantiation document which describes collaborative interaction between groups, and 5. the groups population document, which describes the way participants are associated to learning activities and tool instances.

The specification of a learning process in separated documents aims to foster reusability. Different generic CL situations can be derived from the same data flow structure by simply changing the groups instantiation document. Moreover, different specific CL situations can be derived from the same groups instantiation document by just changing the groups population document.

Although the flows coordination and groups population documents are both described using proprietary specifications, the fact that LeadFlow4LD is mainly based on the standards IMS LD and XPDL allows a significant reduction in authority tools development effort on the one hand (because tools can be reused) and a reduction of the learning curve while using tools based on these standard specifications, on the other hand.

Four simple but significant peer-review CL situations have been selected and specified in the proposed method and from the instructional designer and educator points of view. Furthermore, a non-trivial CL situation –based on the complex CLFP that combines the pyramid, jigsaw and peer-review patterns— has also been specified through LeadFlow4LD. In order to assess such CL situation specifications, we propose and implement a prototype enactment system used as a proof-of-concept to execute a UoLeadFlow4LD. Actual results show that artefacts and tools were delivered automatically to users when they performed learning activities (LeadFlow4LD expressiveness on data flow automation) and the proper documents of a UoLeadFlow4LD were successfully reused for different CL situations (UoLeadFlow4LD reusability).

It is noteworthy that the LeadFlow4LD method could contribute to the learning design field not only as a way of specifying CSCL scripts, but it could also be employed within the context of the proposals made by other researchers in order to enable the deployment of IMS LD based learning design in mainstream VLEs. In this way, LeadFlow4LD could improve the support provided for the realization of collaborative scenarios in both open source products such as Moodle or LAMS and commercial products such as Blackboard. These contributions should be considered for a future review of the IMS LD specification, but also by the other EMLs supporters. Furthermore, in the BPM/workflow domain, no engine supports the execution of the data-flow features added to the current version of their corresponding modelling languages. Moreover, the support of data flow between tools, at the execution time, is not even taken into account. The approach presented in this paper could be considered as a possible way to handle these features in the BPM/Workflow field.

Our future work covers four principle issues. First, the evaluation should go beyond the CL situations presented in this paper. More case studies must be conducted and evaluated from a technological and educational perspective in authentic environments in order to identify advantages and drawbacks of the proposed method. Second, a complete adoption of the proposed method requires the development of the LeadFlow4LD parser component in order to achieve an automatic generation of the synchronized learning flow and workflow documents. Third, we plan to work on the use of LeadFlow4LD to support collaborative learning in the Moodle platform. Fourth and finally, LeadFlow4LD could also be helpful for the specification of the flow of artefacts in the assessment processes, as well as the specification of flow of learning outcomes for e-Portfolio purposes. These are some of the future research lines we are currently exploring.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness (TIN2008-03023, TIN2011-28308-C03-02 and IPT-430000-2010-054) and the Autonomous Government of Castilla y León, Spain (VA293A11-2 and VA301B11-2). The authors would like to thank Tecnológico de Monterrey Campus Guadalajara, Fundación Carolina, as well as the rest of the EMIC/GSIC research group at the University of Valladolid for their support and ideas to this work.

References

[Alario-Hoyos, 13] Alario-Hoyos, C., Bote-Lorenzo, M.L., Gómez-Sánchez, E., Asensio-Pérez, J.I., Vega-Gorgojo, G., Ruiz-Calleja, A.: GLUE!: An Architecture for the Integration of External Tools in Virtual Learning Environments. Computers & Education. 60(1):122-137, January 2013.

[Bartels, 03] Bartels, N: Written Peer Response in L2 Writing. In English Teaching Forum, 2003, 41: 34-36.

[Bordies, 12] Bordies, O., Villasclaras-Fernández, E.D., Dimitriadis, Y., Ruiz-Calleja, A.: Reusability of data flow designs in complex CSCL scripts: Evaluation results from a case study Proceedings of the 18th International Conference on Collaboration and Technology, CRIWG 2012, Duisburg-Essen, Germany, September 2012.

[Britain, 04] Britain, S.A.: Review of Learning Design: Concept, Specifications and Tools. A report for the JSIC E-Learning Pedagogy Program, http://www.jisc.ac.uk/media/documents/programmes/elearningpedagogy/learningdesigntoolsfinalreport.pdf, last accessed 20th June 20011.

[Caeiro-Rodriguez,06] Caeiro-Rodríguez, M., Llamas-Nistal, M., and Anido-Rifón, L: The PoEML proposal to model services in educational modelling languages Groupware: Design, Implementation, and Use Lecture Notes in Computer Science 4154, 2006, 187-202.

[Caeiro-Rodriguez, 08] Caeiro-Rodriguez, M.: PoEML: A Separation of Concerns Proposal to Instructional Design. Handbook of Visual Languages for Instructional Design: Theories and Practices, 2008, 185-209.

[Caeiro-Rodríguez, 10] Caeiro-Rodríguez, M., Anido-Rifón, L., LLamas-Nistal, M: Challenges in Educational Modelling: Expressiveness of IMS Learning Design. In Educational Technology & Society, 2010, 13 (4): 215–226.

[Conole, 13] Conole, Grainne: Designing for learning in an open world. Vol. 4. Springer, 2013.

[Dalziel, 06] Dalziel, J.R.: Lessons from LAMS for IMS Learning Design. In Proc. of the 6th International Conference on Advanced Learning Technologies, ICALT 2006, Kerkrade, The Netherlands, 2006, 1101-1102.

[Dan, 07] Dan, Yu, and Chen XinMeng: Supporting Collaborative Learning Activities with IMS LD. Advanced Communication Technology, The 9th International Conference on. Vol. 1. IEEE, 2007.

[Decker, 08] Decker, G., Mendling, J.: Instantiation semantics for process models. In Proc. of the 6th International Conference on Business Process Management, BPM 2008, Milan, Italy, 2008.

[Derntl, 11] Derntl, M., Neumann, S., Griffiths, D., Oberhuemer, P.: The Conceptual Structure of IMS Learning Design Does Not Impede Its Use for Authoring. In IEEE Transactions on Learning Technologies, 18 Aug. 2011. IEEE computer Society Digital Library. IEEE Computer Society, http://doi.ieeecomputersociety.org/10.1109/TLT.2011.25.

[Derntl, 11b] Derntl, M., Neumann, S., Oberhuemer, P.: Community Support for Authoring, Sharing, and Reusing Instructional Models: The Open Graphical Learning Modeler (OpenGLM). Proceedings of IEEE ICALT 2011, Athens, Georgia, USA. IEEE 2011.

[Griffiths, 08] Griffiths D., Liber, O.: Opportunities, Achievements, and Prospects of IMS LD. In Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies, L. Lockyer, S. Bennett, S. Agostinho, and B. Harper, Eds. Information Science Reference, 2008, 87–112.

[Hernández-Gonzalo, 08] Hernández-Gonzalo, J. A., Villasclaras-Fernández, E. D., Hernández-Leo, D., Asensio-Pérez, J. I., Dimitriadis, Y. A.: InstanceCollage: a Graphical Tool for the Particularization of Role/group Structures in Pattern-based IMS LD Collaborative Scripts. In Proc. of the 8th International Conference on Advanced Learning Technologies, ICALT 2008, Santander, Spain.

[Hernández-Leo, 05] Hernández-Leo, D., Asensio-Pérez, J.I., Dimitriadis, Y.: Computational representation of collaborative learning flow patterns using IMS learning design. In Educational Technology & Society, 2005, 8(4): 75-89.

[Hernández-Leo, 06] Hernández-Leo, D., Villasclaras-Fernández, E.D., Jorrín-Abellán, I.M., Asensio-Pérez, J.I., Dimitriadis, Y., Ruiz-Requies, I., Rubia-Avi, B.: Collage, a Collaborative Learning Design Editor Based on Patterns Special Issue on Learning Design, Educational Technology & Society. 9(1):58-71, January 2006.

[Hernández-Leo, 07] Hernández-Leo, D.: A pattern-based design process for the creation of CSCL macro-scripts computationally represented with IMS LD. PhD Thesis, 2007. Telematic Engineering Department, University of Valladolid, Valladolid, Spain.

[IMS, 02] IMS Learning Design Specification, 2002, http://www.imsglobal.org/learningdesign/

[IMS, 03] IMS Global Learning Consortium: IMS Content Packaging Information Model v1.1.3, Technical Specification, June 2003.

[Katsamani, 11] Katsamani, Mary, and Symeon Retalis: Making Learning Designs in Layers: The CADMOS Approach. Proceedings of the IADIS Multi Conference on Computer Science and Information Systems, 2011.

[König, 10] König, F., Paramythis, A.: Towards Improved Support for Adaptive Collaboration Scripting in IMS LD. In: Wolpers, M., Kirschner, P., Scheffel, M., Lindstaedt, S., Dimitrova, V. (eds.), Sustaining TEL: From Innovation to Learning and Practice, Springer Berlin / Heidelberg , 2010.

[Koper, 01] Koper, R.: Modelling Units of Study from a Pedagogical Perspective: the pedagogical meta-model behind EML. Educational Expertise Technology Centre, Open University of the Netherlands, 2001.

[Koper, 08] Koper, R., Miao, Y.: Using the IMS LD Standard to Describe Learning Designs. In Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies, L. Lockyer, S. Bennet, S. Agostinho & B. Harper (Eds.), IDEA group, 2008, 41-86.

[Marino, 07] Marino, O., Casallas, R., Villalobos, J., Correal, D., Contamines, J. : Bridging the Gap between e-learning Modelling and Delivery through the Transformation of Learnflows into Workflows. In E-Learning Networked Environments and Architectures, Advanced Information and Knowledge Processing, Part I, Springer-Verlag (Eds.), 2007, 27-59.

[Martel, 06] Martel, C., Vignollet, L.: LDL: a Language to Model Collaborative Learning Activities. In E. Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications, Pearson & P. Bohman (Eds.), 2006, 838-844.

[Miao, 05] Miao, Y., Hoeksema, K., Hoppe, H.U., Harrer, A.: CSCL scripts: modelling features and potential use. In Proc. of the International Conference on Computer Support for Collaborative Learning, CSCL 2005: the next 10 years!, Taipei, Taiwan, 2005, 423-432.

[Miao, 08] Miao, Y., Burgos, D., Griffiths, D., Koper, R.: Representation of Coordination Mechanisms in IMS LD. In Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies, 2008, 330.

[Neumann, 09] Neumann, S., Klebl, M., Griffiths, D., Hernández-Leo, D., de la Fuente, L., Hummel, H., Brouns, F., Derntl, M., Oberhuemer, P.: Report of the Results of an IMS Learning Design Expert Workshop. In International Journal Of Emerging Technologies In Learning (IJET), 2009, 5(1): 58-72.

[Norin, 02] Norin, R.: Workflow Process Definition Interface–XML Process Definition Language. Document Number WfMC. TC-1025, Lighthouse Point (Fl): Workflow Management Coalition, 2002.

[Osuna-Gómez, 99] Osuna-Gómez, C.: Delfos: un marco telemático educativo basado en niveles orientado a situaciones de aprendizaje cooperativo: Tesis doctoral, Universidad de Valladolid, España, 1999.

[Palomino-Ramírez, 07] Palomino-Ramírez, L., Martínez-Monés, A., Bote-Lorenzo, M.L., Asensio-Pérez, J.I., Dimitriadis, Y.A.: Data Flow between Tools: Towards a Composition-Based Solution for Learning Design. In Proc. of the 7th International Conference on Advanced Learning Technologies, ICALT 2007, Niigata, Japan, 2007, 354-358.

[Palomino-Ramírez, 08] Palomino-Ramírez, L., Bote-Lorenzo, M.L., Asensio-Pérez, J.I., Dimitriadis, Y., de la Fuente-Valentín, L.: The Data Flow Problem in Learning Design: A Case Study. 6th International Conference on Networked Learning, NLC 2008, Halkidiki, Greece, 2008.

[Palomino-Ramírez, 08b] Palomino-Ramírez, L., Bote-Lorenzo, M.L., Asensio-Pérez, J.I., Dimitriadis, Y.: LeadFlow4LD: A Composition-based Solution for Learning Design in CSCL. 14th Collaboration Researchers' International Workshop on Groupware, CRIWG 2008, Omaha, Nebraska, 2008.

[Paramythis, 08] Paramythis, Alexandros: Adaptive support for collaborative learning with ims learning design: Are we there yet. Proceedings of the Workshop on Adaptive Collaboration Support, held in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Hannover, Germany. 2008.

[Peter, 05] Peter, Y., Vantroys, T.: Platform Support for Pedagogical Scenarios. Journal of Educational Technology & Society, 2005, 8:122-137.

[Rawlings, 02] Rawlings, A., van Rosmalen, P., Koper, R., Rodríguez-Atacho, M., Lefrere, P.: Survey of Educational Modelling Languages (EMLs). Technical Report, CEN/ISSS WS/LT, 2002.

[Reload, 12] Learning Design Editor Web site, http://www.reload.ac.uk/ldeditor.html

[Rodríguez-Artacho, 04] Rodríguez-Artacho, M. , Maíllo, M. F. V.: Modelling Educational Content: The Cognitive Approach of the PALO Language. Journal of Educational Technology and Society, 2004, 7: 124-137.

[Surhone, 10] Surhone, L.M., Tennoe, M.T., Henssonow, S.F.: Bonita Open Solution. VDM Verlag Dr. Mueller AG & Co. Kg, 2010.

[Together, 12] XPDL Editor Web Site, http://www.together.at/prod/workflow/twe

[van der Aalst, 02] van der Aalst, W.M.P., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT press, Cambridge, MA, 2002.

[Vantroys, 03] Vantroys, T., Peter, Y.: COW, a Flexible Platform for the Enactment of Learning Scenarios. In Proc. on the 9th International Workshop on Groupware, CRIWG 2003, 2003, 168-182.

[Vignollet, 09] Vignollet, L., Bote-Lorenzo, M. L., Asensio-Pérez, J. I., Dimitriadis, Y.: A generic specification of the data-flow issue in the learning design field, IEEE. In Proc. of the 9th International Conference on Advanced Learning Technologies, ICALT 2009, 56-58.

[Vignollet, 10] Vignollet, L., Bote Lorenzo, M.L., Asensio Pérez, J.I.: Work Flow Management and Learn Flow Management: commonalities and differences. IEEE Learning Technology, 2010, 12(3):35-37.

[Vogten, 05] Vogten, H., Martens, H.: CopperCore 2.2.4. Heerlen: Open University of The Netherlands, http://coppercore.sourceforge.net/

[Weinberger, 04] Weinberger, A., Fischer, F., and Mandl, H.: Knowledge convergence in computer-mediated learning environments: Effects of collaboration scripts. Proceedings of the 85th Annual Meeting of the American Educational Research Association (AERA), San Diego, CA, USA, 2004.

[WfMC, 94] The Workflow Reference Model, Workflow Management Coalition, TC00-1003, December, 1994.

[Wilson, 05] Wilson, S.: Workflow and Web services, CETIS White paper. See http://www.e-framework.org/resources/SOAandWorkflow2.pdf, last accessed 20th June 20011.