Data Flow between Tools: Towards a Composition-Based Solution for Learning Design

Luis Palomino-Ramírez, Alejandra Martínez-Monés, Miguel L. Bote-Lorenzo, Juan I. Asensio-Pérez and Yannis A. Dimitriadis
University of Valladolid, Spain
lpalomin@ulises.tel.uva.es, amartine@infor.uva.es, {migbot, juaase, yannis}@tel.uva.es

Abstract

Data flow between tools cannot be specified using the current IMS Learning Design specification (LD). Nevertheless by specifying this data flow between tools, several degrees of activity automation may augment the system intervention opportunities for data flow management. Service automation, data flow automation and data flow validation may enhance the continuity of the learning design realization, reduce the student's cognitive load and obtain system-support for error prone situations. In this paper a novel approach based on the composition of LD and a standard workflow technology is proposed. Unlike other current approaches, our approach maintains interoperability with both LD and workflow standards. Then an architectural solution based on the composition approach is presented.

1. Introduction

Formalization of teaching-learning processes expressed as scripts in an educative modeling language, is an essential step towards process automation in e-learning. Exchange and design of computer-interpretable scripts could be of great help for educators. For this purpose LD has recently emerged as an XML-based and activity-based process oriented approach [1] [2]. Nevertheless, the learning process automation has not been entirely accomplished. Tools and services automation is still an open issue in LD [3] [4]. In LD there are no means to specify that the output data from one tool be the input data for another tool [5]. Instead, LD supports an activity-oriented data flow model [1], where the user is responsible for managing the input and output data for tools. Throughout this human-oriented data flow model, not only the Learning Management System (system) cannot fully automate a tool or service, but it is not even able to validate the data flow between tools while the user is managing it. Although several authors have reported this problem in literature [5] [3] [4], just Miao et al. have proposed a concrete solution to the problem [5], but without maintaining interoperability with LD.

The aim of this paper is to develop a proposal to address the data flow problem between tools, without affecting the LD interoperability. To reach this goal and following the workflow contributions for LD [6] [5] [7], in this paper we state that workflow and LD can be seen as complementary technologies, which could be used in a coordinated manner to specify both human-oriented learning flow as well as system-oriented data flow. On one hand the standard LD can focus on specifying a learningflow among people; and on the other, a standard workflow technology like XPDL [8], could focus on specifying the data flow between tools. So, the coordination of the execution of learning design and workflow scripts enables an extension of the current scope of LD while maintaining the interoperability with LD and XPDL standards.

The rest of the paper is structured as follows. Section 2 introduces the data flow problem in LD which is illustrated by an example. It provides arguments to state that the data flow problem is related to the so-called wide-scope LD problem. In section 3 a solution space of current approaches is categorized and discussed in order to introduce the proper approach. In section 4 a concrete architecture of the composition approach is proposed. Finally, the paper presents the conclusions, contributions and future work.

2. Data Flow Problem in Learning Design

A learning design defines a flow of activities with students and teachers playing roles and using services and tools with the aim of accomplishing their learning goals [2]. The core concept here is that students participate in a learningflow which is just one part of

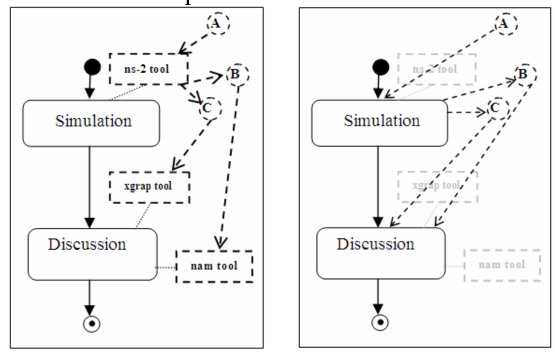
Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)
0-7695-2916-X/07 $25.00 © 2007 IEEE

the learning design; the other part is the data flow. Nevertheless, LD lacks the management of the data flow [9], since LD has no means to specify the data flow between tools [4] [5].

Consider the learning scenario shown in figure 1a in order to illustrate the expressivity problem of the data flow model for LD. This scenario corresponds to a real data network simulation lab exercise in order to illustrate how network protocols work. First, students setup the simulation model using a Tcl/Tk script (data A). This script is then interpreted by the network simulator called ns-2 [10], which produces two output files: a file (data B) to be visualized in an xy-graph tool; and a file (data C) to be visualized in a network animation tool. As illustrated in figure 1a, whereas the learningflow specifies the sequence of the activities, the data flow specifies the sequence of tools as well as their input and output parameters. Unfortunately this data flow between tools cannot be specified in LD because LD has no means to specify the relation between data and tools [5]. Instead, LD specifies a relation between data and activities as illustrated in figure 1b. Then the user through his interaction with the system performs the real data flow between tools while he follows the activity description. For example, the educator has to specify to the system those *properties* (data files), which are related for some activity and also has to instruct the user on which are the input and output files for the tools (via the activity description). Then the user, not the system, is the party responsible for matching data with tools. This means that the activity-oriented data flow model for LD actually becomes human-oriented when related to data flow between tools.

If it were possible to specify the data flow between tools as illustrated in figure 1a, it would open a range of new possibilities for system intervention in data flow management. For example, a support activity could be completely automated; or even in a learning activity it would be possible to keep some degree of user participation: an activity could be performed by people but the system would automatically manage the data flow; or an activity could be fully performed by people but the system would validate the data flow. By defining a gradient of activity automation the student's performance may improve through the learning design realization. Next, this statement will be illustrated throughout the aforementioned example.

Let's consider the *Simulation* activity automation. In such case the learning design execution would be more continuous, the student cognitive load would be lower, and the data flow management would be error free, if the ns-2 service was automatically executed by the system. Unfortunately LD does not consider the initialization parameters of a service nor the results of a service to be exported to any other place [3]. Nevertheless through the specification of the data flow between tools it is possible to address this issue.



(a) (b)

Figure 1. Comparison of the data flow expressivity (a) between tools (b) at activity level

Now consider the *Discussion* activity semi-automation, which means just the automation of the data flow. So if the user starts the *Discussion* activity, both the animation and visualization tools would be automatically launched by the system, uploading the files users need to work with. The main advantage of this degree of activity automation is to avoid the data flow managed by the user, which is error prone. In such case the user has to interpret the activity description, which could be misunderstood, badly applied, ignored or completely forgotten [11]. The system has no means of verifying that the right data is being used with the right tool, thus questioning the achievement of the learning goals. Now consider the same example but extended to a collaborative learning scenario, with multiple concurrent groups interchanging outcomes in order to discuss them. Matching the output files from the ns-2 service corresponding to one group with the input files of the graphical and visualization tools from another group is complex and error prone. The learning goals will not be accomplished if an animation file discussed by one group is not the right one. Nevertheless by specifying the data flow between tools, it is possible to address this issue too.

But data flow automation is not always worthwhile in learning design, just like when the data flow is part of the learning activity itself. So the student has to be aware of the data flow. In this case the activity description compels the student to upload the simulation script to the ns-2 service. Then a typical LD activity has to be specified, but it is still error prone. So the solution proposed is to specify what we call a system-validated activity, which is the activity fully performed by the user but that is validated by the system at run time. By specifying the data flow between tools it is possible to address this issue again.

Although the data flow problem has been reported in the literature, it has been treated only in an isolated manner. We state here that the data flow problem is related to what we call wide-scope LD problem.

2.1. Wide-Scope Learning Design Problem

LD is a new and emergent approach in e-learning. Nevertheless, LD keeps an intrinsic relationship with the workflow paradigm [1] [7] [6] [9] [3] [5]. Then, what really makes them different, beyond their application domain, is their own orientation. While workflow technologies are oriented to the system-supported activities automation in order to optimize the enterprise business process [12], LD is oriented to learning activities performed by people in order to accomplish their learning goals [2]. This comparison between both approaches is illustrated in the diagram of figure 2, which corresponds to Dayal's workflow characterization [13]. In this diagram, an activity-based process is represented within a spectrum characterized by two dimensions, namely the type of process structure and the degree of task automation.

As illustrated in figure 2, whereas LD is mainly human-oriented, workflow technologies are mainly system-oriented. This distinction becomes relevant when it is related with the research found in workflow literature. There, Sheth suggests that the challenge in the work activities coordination within the context of wide-spectrum processes is to find ways to embody human work with automated activities [14], which corresponds to the dotted ellipse shown in figure 2. This means that Sheth already suggests, even before the proposal of LD, that the challenge for workflow technologies is that they should be not only system-oriented, but human-oriented too. This statement has been also exposed by Dayal, who assures that one of the challenges in process management has been to provide end-to-end support for wide-spectrum processes from non-structured activities focused on people to structured activities focused on software [13]. From this argumentation and due to the workflow nature of LD, we state that LD should be a wide-scope approach in order to provide support to learning activities performed by people, as well as support activities executed by computer systems. Therefore, one can see in figure 2, that an emergent opportunity area arises in order to extends the current LD scope along its task-automation dimension. This area corresponds to the already mentioned gradient of activity automation.

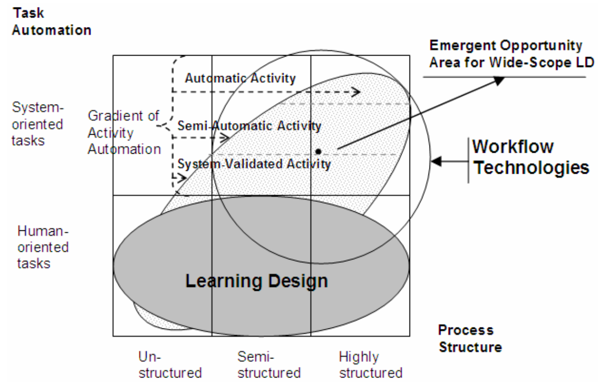


Figure 2. Complementarity between LD and workflow used to extend the current LD scope

3. Current Solution Approaches

In this section, current approaches concerning the solution of the data flow problem in LD are categorized and discussed, aiming to introduce the proper approach.

3.1. Substitution Approach

An immediate solution for the LD data flow problem is to use the so-called substitution approach: partial or complete LD modification. At one end of the spectrum by substitution, a feasible solution could be as simple as replacing the LD data flow model, thus making possible the desired relationship between data and tools. At the other end of the spectrum by substitution a completely new LD specification could also be proposed. This is the case of Miao et al’s proposal [5]. Miao proposes a new LD meta-model which actually merges elements from the current LD and XPDL standard meta-models. Furthermore other solutions by substitution are also possible, such as a schema integration solution, or other ad hoc combinations of LD elements and any other workflow standard. Nonetheless a substitution approach is not suited for a community of practice, which is developing tools around the standard, if there is not a bridge of interoperability with it.

3.2. Mapping Approach

Another workflow trend that has been applied to LD, is what we call mapping approach, which has been inspired from Vantroys’ work [7]. Vantroys proposes executing LD models through mapping LD to XPDL. Thus, a mapping approach for the LD data flow problem would consist of using LD for learning design, and a standard workflow technology for data flow. However, mapping between LD and a workflow technology is not completely feasible since both are complementary technologies as shown in figure 2. For example, in order to carry out such mapping, XPDL should be extended to a wide-spectrum workflow language. Thus we would incur again to a new interoperability problem, since the standard XPDL should be substituted by a new incompatible one.

4. An Interoperable Solution Approach

In this section we propose not only to solve the LD data flow problem, but also to extend the current LD scope while maintaining interoperability with LD and workflow standards.

4.1. Composition Approach

Taking advantage of the complementarity between LD and workflow technologies, a composition-based approach is proposed. Then, one of them may be used to support the other where one of them fails. In this case LD is used to specify the learning design as usual. Then a standard workflow technology is used to specify the data flow between tools, because this is where LD fails. Therefore, instead of substituting LD or mapping LD by using a non-standard technology, we propose the use in a coordinated manner of both standard technologies. This approach not only addresses the LD data flow problem, but also extends the current scope of LD along its task-automation dimension while maintaining interoperability with both LD and workflow standards.

4.2. A Solution based on the Composition Approach

A composition-based solution to the LD data flow problem will be illustrated through an architectural proposal shown in figure 3, as well as through the simulation example described in section 2. During a first design phase, the educator produces the *learning design script* using an LD-compliant authoring tool. Note as shown in figure 4a, that the *learning design script* contains both learningflow and data flow, but the latter specified at activity level as it actually is in LD. The educator then completes the second design phase when the data flow between tools has to be specified. For this purpose an XPDL-compliant authoring tool could be used. Note as shown in figure 4b, that the so-called *workflow script* actually details the data flow between tools that has not been specified in LD.

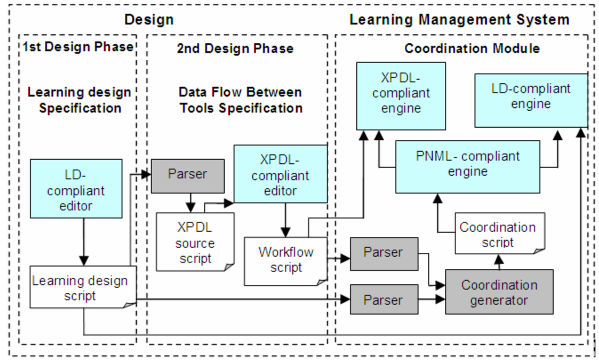
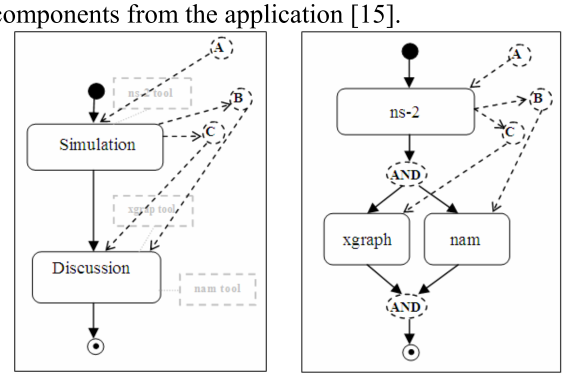


Figure 3. Architecture of a composition-based solution for the data flow problem in LD

Subsequently both learning design and workflow scripts enter into the coordination module in order to be interpreted, executed and coordinated. For interpretation and execution of each script, an LD-compliant engine and XPDL-compliant engine are respectively used. The problem now is to solve where and how both engines should be coordinated. The coordination should not be hardcoded within the system because this is contrary to the flow-independence principle characteristic of the workflow paradigm: separate the flow between an application's components from the application [15].



(a) (b)

Figure 4. (a) ***Learning design script*** **(LD)**
(b) ***workflow script*** **(XPDL)**

So, we propose to specify the coordination of both engines within a third script called *coordination script*, and a Petri Net Markup Language (PNML) [16] could be used for this purpose. Then, in order to automatically generate the coordination script, the learning design and workflow scripts are first parsed, resulting in a series of control points (transitions in Petri nets) that indicate where to start, pause and resume each execution engine. For example, the LD engine starts its execution at the beginning of the *Simulation* activity. Then, due to the ns-2 tool belongs to that activity, it causes a pause in the LD engine at the same time that the XPDL engine starts its execution. Pausing LD engine means that the learningflow cannot continue to perform the next activity until the XPDL engine has released the control. Similar case occurs when the *workflow script* attempt to perform the next *xgraph* and *nam* tasks, which actually belong to the *Discussion* activity, not the *Simulation* one. It causes a pause in the XPDL engine meanwhile the control is released to resume the LD engine.

5. Conclusions

The lack of system-support for data flow management in LD is not an isolated problem, but it is related to the wide-scope LD problem: LD should be not only human-oriented but system-oriented too, as it has been declared in workflow literature these technologies should be. In this paper we state that LD and workflow could be seen as complementary technologies. Thus, a workflow-based solution could support the LD specification for data flow between tools. Then, the coordinated execution of both learning design and workflow scripts becomes the core of a composition approach behind our architectural proposal, on which a prototype is currently under development. Unlike other current approaches that do not maintain interoperability with LD or workflow standards, our approach maintains interoperability with both of them. This composition approach is the first step towards understanding what a wide-scope LD problem is, and to know what kind of activity automation degrees are really feasible in LD. Furthermore, this composition approach may be the in-between approach that will provide a framework for future integration of LD and workflow streams.

6. Acknowledgements

This work has been partially funded by the EU e-Learning project EAC/61/03/GR009, EU Kaleidoscope NoE FP6-2002-IST-507838, Spanish Ministry of Education and Science project TSI2005-08225-C07-04, Autonomous Government of Castilla y León, Spain, projects VA009A05, UV46/04 and UV31/04, Tecnológico de Monterrey Campus Guadalajara, and Fundación Carolina. The authors would also like to thank the EMIC/GSIC research group at the University of Valladolid for their support and ideas.

7. References

[1] R. Koper, B. Oliver, and T. Anderson, "IMS Learning Design Information Model, version 1.0," *IMS Global Learning Consortium, Inc*, 2003.

[2] R. Koper, *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*: Springer, 2005.

[3] S. Wilson, "Workflow and web services," *CETIS White paper. Last retrieved Jan. 22 2007 at www.e-framework.org/resources/SOAandWorkflow2.pdf*, 2005.

[4] J. Dalziel, "Lessons from LAMS for IMS Learning Design," *The 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*, pp. 1101-1102, 2006.

[5] Y. Miao, K. Hoeksema, H. U. Hoppe, and A. Harrer, "CSCL scripts: modelling features and potential use," *Proceedings of th2005 conference on Computer support for collaborative learning: 2005: the next 10 years!*, pp. 423-432, 2005.

[6] M. Caeiro, L. Anido, and M. Llamas, "A Critical Analysis of IMS Learning Design," *Proceedings of CSCL*, pp. 363-367, 2003.

[7] T. Vantroys and Y. Peter, "COW, a Flexible Platform for the Enactment of Learning Scenarios," *Proceedings on the 9th International Workshop on Groupware (CRIWG 2003)*, pp. 168-182, 2003.

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

[9] Y. Peter and T. Vantroys, "Platform Support for Pedagogical Scenarios," *Educational Technology & Society*, vol. 8, pp. 122-137, 2005.

[10] S. McCanne and S. Floyd, "ns-2 Network Simulator," *Obtain via: http://www. isi. edu/nsnam/ns*.

[11] P. Dillenbourg, "Over-scripting CSCL: The risks of blending collaborative learning with instructional design," *Three worlds of CSCL. Can we support CSCL*, pp. 61-91, 2002.

[12] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distributed and Parallel Databases*, vol. 3, pp. 119-153, 1995.

[13] U. Dayal, M. Hsu, and R. Ladin, "Business Process Coordination: State of the Art, Trends, and Open Issues," *VLDB Conference*, pp. 3–11, 2001.

[14] A. Sheth, D. Georgakopoulos, S. M. M. Joosten, M. Rusinkiewicz, W. Scacchi, J. Wileden, and A. L. Wolf, "Report from the NSF workshop on workflow and process automation in information systems," *ACM SIGSOFT Software Engineering Notes*, vol. 22, pp. 28-38, 1997.

[15] D. A. Manolescu, "Micro-workflow: a workflow architecture supporting compositional object-oriented software development," University of Illinois at Urbana-Champaign, 2001.

[16] E. Kindler, "Software and Systems Engineering-High-level Petri Nets," *Part2: Transfer Format. International Standard ISO/IEC 15909-2. WD Version 0.9.0*, 2005.