2010 International Conference on Intelligent Networking and Collaborative Systems

Study of the dataflow problem in complex adaptive collaboration learning scripts

Osmel Bordiés
Electrical Engineering Faculty
Oriente University, UO
Santiago de Cuba, Cuba
obordies@fie.uo.edu.cu

Yannis Dimitriadis
School of Telecommunications Engineering
University of Valladolid, UVa
Valladolid, Spain
yannis@tel.uva.es

***Abstract*—The flow of artifacts or dataflow in learning design has not been adequately tackled by the IMS-LD specification or other existing Educational Modelin Languages. Several proposed approaches, including the *LeadFlow4LD*, aim to handle the main dimensions of the dataflow problem, i.e. automation, consistency and reuse. However, considering the adaptive and complex nature of real collaborative learning scripts may have a significant impact in the existing solutions to the dataflow problem. This paper presents an analysis of the potential/limitation of *LeadFlow4LD* in a real case study, in which a script based on CLFPs and that requires adaptation has been used. Findings of such an analysis suggest the use of workflow templates in order to overcome the detected problems of real reusable designs.**

***Keywords*-IMS-LD, Data Flow problem; workflow; reuse; adaptation;**

I. INTRODUCTION

Learning design [1] and the parallel field of scripting have drawn considerable attention in the last few years. Modeling pedagogical strategies and scaffolding the learning processes has shown to be beneficial in terms of effectiveness and efficiency, especially in complex pedagogies supported by Information and Communication Technologies (ICT), such as computer supported collaborative learning (CSCL) [2].

The IMS-LD specification, the *de facto* standard for the learning design community allows for modeling and providing a computational representation of teaching/learning scenarios. Far from SCORM standard, IMS-LD not only delivers educational content but also pedagogical strategies as processes describing teaching and learning aspects [3]. The learning design concept comprises both the definition of learning and data flow [4]. For instance a *Peer-Review* scenario design implies the definition of the sequence of activities but also how the user will exchange artifacts among them. However the data flow in IMS-LD is person oriented and lacks formal definition for both automatic learning tool invocation and management of artifact flow among tools [4], [5]. Such limitation could be the origin of error prone situations especially in CSCL scenarios with many users. IMS-LD has also been critized for other limitations on modeling dynamic behavior in which artifact flow is also involved [6].

The "dataflow problem" [7] in the learning design or scripting fields can be characterized by several facets, such as the need for automation, consistency checking and design re-use. Several existing solutions [6], [8] do not consider the need for interoperability with IMS-LD. *LeadFlow4LD* design methodology on the contrary, provides a interoperable composition-based solution to the dataflow problem [9]. A composition-oriented approach combines IMS-LD for the actvitity flow definition and a workflow language for handling the flow of artifacts among tools. In run time learning and data flows are executed according to a master-slave coordination mechanism in which the IMS-LD based learning flow acts as the master. Automation and consistency are supported by the use of workflow engine and workflow language respectively. Support for reuse and adaptability of complex CSCL scripts may be limited since the current workflow standards do not allow for the definition of the instantiation semantics [10].

Adaptation of complex scripts, as those built based on the collaborative learning flow patterns (CLFP), should be considered and analyzed, since it may respond to real educational needs [11]. Such an adaptation is very important, since it can contribute to a better equlibrium between coercion and specific characteristics of the context during the script design or enactment. Among several efforts, in this paper we will focus on the adaptation patterns proposed by Karakostas et al. [12] which may be combined with the CLFP in order to provide realistic scripts that may serve as a basis for studying existing solutions to the dataflow problem.

This paper aims to provide initial evidence related to the effect of using *LeadFlow4LD* along adaptive CSCL scripts based on CLFPs. This analysis is made with respect to a real case study, whose design was based on CLFP. The original design has been enriched through the use of an adaptation pattern. Several aspects of the dataflow problem, such as reuse or dataflow adaptation mechanism definition, were studied using the resulting complex adaptive script from the application of the *LeadFlow4LD* methodology. Initial evidence suggests paths to provide a new solution for the dataflow problem in adaptive complex CSCL settings.

The following section describes the concepts of collaborative learning flow and adaptation patterns and the way

978-0-7695-4278-2/10 $26.00 © 2010 IEEE
DOI 10.1109/INCOS.2010.97

that they were used in order to define the case study. The dataflow problem and the *LeadFlow4LD* approach are briefly described in section three, while the effects of using the *LeadFlow4LD* approach in the new adaptive complex setting are presented in section 4. Conclusions, discussion and future work are presented in the last section.

II. CLFP-based CSCL scripts

In many scripts, each group member is required to play a specific role. Four basic elements (students, groups, class and roles) define the social structure of any script [13]. This simple structure enables the definition of phase transitions as matrix transformations based on generic operators. According to a pedagogical model, during a CSCL script execution several structures can be transformed by sequenced operators [13]. They transform one data structure matrix (social, resources or product) into another to induces effective interactions (e.g. one *products matrix to social matrix* transformation can be applied when new actual groups are formed according to the quality of the group contributions during the previous phase). But even a well formed script, with the adecuate set of operators may be incomplete, because its success could be affected by other parameters that might not be available at design time (e.g. personality of both students and professor or another contextual parameters). That's why it is necessary to constantly adjust or regulate the script during runtime applying adaptation processes or providing the appropiate scaffolding [14].

However, it is well known that design of an effective CSCL script is not trivial since many parameters have to be taken into account in order to mantain the equilibrium between coercion and flexibility [6]. At least five attributes have to be specified at each CSCL script phase to express not only the pedagogical method but also to achieve those effective interactions (e.g task to be executed by each student, group composition or the way the task is distributed among students) [15]. The Collaborative Learning Flow Patterns (CLFP) have been proposed as a means to help educational practitioners to design and implement effective CSCL scripts [11]. Those patterns reflect *best practices* on structuring collaborative learning activities flows and follow the *recurrent solutions to recurrents problems* philosophy. CLFP contribute also to communicate these *best practices*, establishing a common conceptual ground for teacher and learning system's developers, while they also promote the software reuse [16]. For instance, the *Jigsaw* pattern responds to the situation in which one problem can be divided into several sub-problems, promoting positive interdependence and individual accountability. Such a pattern enables the computer formalization of a script that can be progressively particularized.

III. Adaptive collaboration scripts and Adaptation patterns

As it was menctioned in the previous section, the use of CLFPs or other design methods increases the probabilities of CSCL script success, but there this is no guarantee of such success. For that reason many adaptation approaches have been reported in literature in order to provide mechanisms to educators so that they may regulate the execution of the scripts and modify predefined parameters based on certain determinants observed during the enactment. Those mechanisms can be specified in design time or alternatively they can be directly implemented during enactment. The recently proposed adaptive collaboration scripts [14] include such adaptation mechanisms that comply to the intrinsic script characteristics, while they may adapt the extrinsic ones [17]. Following an approach analogous to that of CLFP, the Adaptation Patterns (AP) also aim to communicate the *best practises* taken from adaptation process expertise [12]. For instance, "Group heterogeneity" pattern refers to the idea of forming heterogeneous groups based on partners' prior domain knowledge. In case of using a *Jigsaw*-based activity flow structure, it is possible to form the "expert's" group taking into account sub-problem individuals characteristics *observed* in the previous phase.

Both CLFP and AP can be implemented using IMS-LD. Thus, teachers may generate complex adaptive CSCL scripts based on best practices that can be executed using a common standard educational modeling language.

IV. Data Flow Problem in Learning Design

The data flow problem in Learning Design reflects limitations of the IMS-LD specification on supporting automatic invocations of learning tools and the artifact flow among them [4], [5], [18]. The data flow specified in IMS-LD is person oriented and therefore artifacts transference must be managed by the users involved in learning situations. In case of collaborative scenarios where data flow is implemented based on role-properties and monitoring services it can be even confusing for participants. For instance, in the case study reported in [19] it has been necessary to guide the participants by explicit natural language orientations included in the IMS-LD script. A clear outcome of this case study consists in the need to reduce the cognitive load for participants and the instructional designer. *Automating* several data flow operation, checking for *consistency* and providing design *re-use* are therefore the main facets that need an answer.

In the composition-based solution [19] the data flow automation and consistence dimensions are tackled by combining IMS-LD with a workflow language. Thus, the responsibility of defining, executing and managing the data flow and tools' invocation fall into workflow language scope, while IMS-LD is used to define the learning activity flow. According to this approach both learning and data flows are executed in separate engines following a master-slave coordination mechanism. The reuse requisites requires an adequate separation between the levels of declarative and instantiation design. At declarative level generic structures are created but at instantiation level the basic design is particularized (e.g. users, groups) and the specific relation between artifacts and tools instances is defined. However, the principal limitation of this approach comes from the declarative nature of workflow and the inability to define such instantiation semantics [10]. One variant of existing solutions to this limitation is described in the following section.

V. The *LeadFlow4LD* Design Methodology

The *LeadFlow4LD* (*Learning and Data Flow composition-based solution for Learning Design*) design methodology [9] can be considered as a conceptual proposal to validate the composition-based approach to the data flow problem. One variant of this design methodology recommends the use of a non-standard specification for defining the semantics of the instantiation of both the learning flow and the data flow. The creation sequence of corresponding documents is explained as follows (see figure 1):

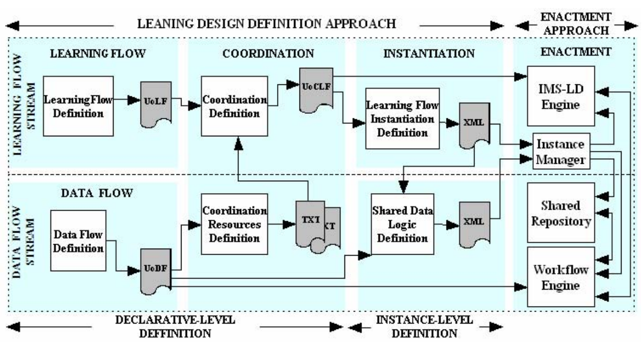

Figure 1: *Leadflow4LD* design methodology

Learning flow structure document: It specifies through IMS-LD the activity sequence, the roles to be played by participants in each activity and the resources (tools and learning objects) available for each role in each activity. The resources are understood as tools and learning objects. The language used is IMS-LD as it was mentioned before.

Data flow structure document: It specifies the sequence of workflow activities as tool invocation activities, the roles associated to each workflow activity, and the resources associated to them. As previously the workflow roles defined must be identical to those specified in the learning flow document and the resources refer to Input/Output artifacts that will be used by the tools involved in each learning activity. *LeadFlow4LD* methodology proposes the use of a standard language for coding the data flow structure document (e.g. BPEL, XPDL).

Both streams, learning flow and data flow are executed in a coordinated manner following a master-slave mechanism. Learning flow plays the *"master"* role and the workflow, the *"slave"* one. In consequence it is necessary to **add coordination resources** in both flows to specify when the flow control switches from learning flow to the workflow and backward.

Instantiation document: It reflects the particularities of the actual learning scenario in which learning and data flow should be enacted: users, groups and assignment of users to roles. Each dataflow sub-process should be instantiated creating several copies for each user or group sharing the same tool. This document (*iLeadFlow4LD*) does not follow a standard specification and therefore affect the solution interoperability may be affected.

VI. MOSAIC case study

The aforementioned design methodology has not been adequately analyzed in the context of adaptive complex CSCL scripts. The use of a real case study might shed some light to such an analysis.

The case study presented in this paper is an adapted version of one performed in March 2007, in which 12 PhD students from three Spanish universities were involved [19]. The main instructional goal of this experience was the interactive creation of a conceptual map on the topic of Grid services and service oriented computing. The original learning design was created based on the selection and hierarchical combination of *Pyramid, Jigsaw* and *Peer-Review* CLFPs based on *contextual* characteristics (see Figure 2). At the first level of the *Pyramid* the *Jigsaw* pattern is employed together with its three logical phases. At first phase the students are supplied with 3 basic readings about the subject (4 students read the same paper) in order to construct the first conceptual map. At the second phase the students who had read the same paper are joined as *experts* groups to analyze the prior map (review) and produce a new version of the conceptual map. At third phase, the students are reorganized again to form 3 *Jigsaw* groups each with 4 experts students following similar orientations. Then at the second level of the *Pyramid* the 4 groups of the second phase are combined to form 2 groups doing the same to reach two more sound conceptual maps. Finally the students arrive at the third level of the *Pyramid* where they review the past two previous maps and reach a final consensus.

The actual case is enriched by adding one adaptation mechanism based on the "group heterogeneity" AP [20]. In this case the teacher evaluates the contributions of the *Jigsaw* groups, so that she may select the most appropriate predefined settings and therefore improve interactions in the following phase. The overall structure of the actual case is depicted in figure 2 (left) as well as a simplified version of activity diagram related to the data flow instance diagram in figure 2 (right).

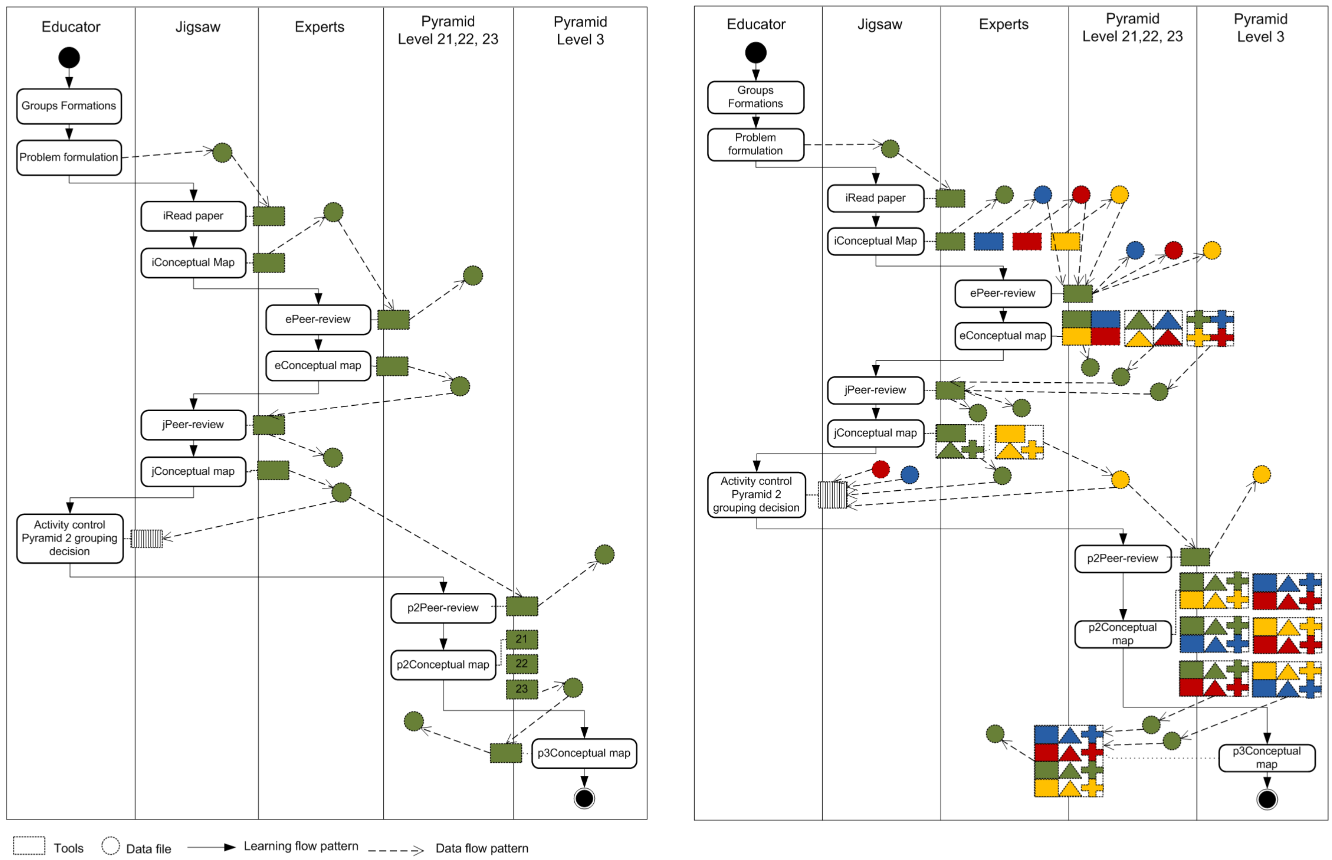

Figure 2: A simplified version of the activity diagram of case study comprising the learning flow and the data flow (left) at declarative-level (right) at instance-level

A. *LeadFlow4LD methodology on MOSAIC case*

The first two documents corresponding to *LeadFlow4LD* methodology should be specified as shown at the UML activity diagram of figure 2 (left). Note that the effect of applying the adaptation pattern is reflected in the roles of the second level of the pyramid (`21, 22` and `23`) associated to the `P2Conceptual map` activity. These roles correspond to alternative groupings that have been included in the design document.

In correspondence with learning flow document, the data flow should be specified as a workflow activity sequence with a similar structure (except the first support activity performed by the teacher, all others are additionally supported by a learning tool). The roles associated to workflow actities are similar to those specified in the learning flow document. Then as a third step, the coordination resources must be incorporated at learning flow and data flow structures enable a synchronized execution of both flows during enactment. This synchronized execution can be achieved by matching the learning activities to the corresponding tool invocation activities at the workflow side.

Finally, the instantiation document reflects the particularities of the context in which the design should be enacted. For simplicity Figure 2 (right) depicts the context through the outline of only one workflow process instance (*Green*) including also the relationship between itself and the other eventuals "group partners" instances at data flow level (*Red*, *Yellow* and *Blue* boxes and circles; Figure2).

VII. Discussion

It is relevant to highlight some important issues related to the aforementioned design description.

First of all is remarkable the propensity to know some context information before initiate the pattern based CSCL design. This situation compromise an effective reusability and creation of CSCL design and alter the timeline of the design methodology process because there is no clear separation between instructional designer and the real performer, the educator. This is an important aspect taking into account the CLFP-based learning flow structure is not fixed. For instance, in the progressive process of particularization it can be structured with certain number of phases depending of contextual characteristics and the selected teaching strategies. In our case it was known beforehand that 12 students would participate in the experience. Another important aspect is a current limitation in IMS-LD based CLFPs models to incorporate some predefined adaptation mechanisms which it is an open discussion on research community.

But the main concern refers to the dataflow instantiation specification which does not fully cover reuse dimension. This design methodology requires separation of the structural and instantiation aspects. In these case data flow is specified using a workflow language, and due to its prescriptive nature it is not possible to define the data flow instantiation. The non-standard specification *iLeadFlow4LD* allows to associate users with their respective roles and groups for learning and data flows, as well as to define the creation of corresponding tools instances and specify the crucial relationship between the data instances and tools instances. This specification is not enough to overcome the adaptive data flow specification reuse issue. Note that the adaptation mechanism used in the case can be different for each new particular context. Changes due to the adaptation mechanisms could be different taking into account the pedagogical strategy selected, the groups' formation policy and the number of users per group. For instance, the activities `ePeer-review` and `Activity control` *`Pyramid`* `2 grouping decision` structurally depend on the number of times the same activity has to be performed, in this case *reviewing* the last contributions or artifacts. Following the same analysis, different grouping combinations affect the number and type of alternative routes in the workflow schema.

Finally, the tool characteristics may have an effect to data flow structural design, especialy when context data are taken into account.

Thus, one may observe that the current *LeadFlow4LD* design methodology does not support full reuse of the CSCL scipts. The user should have to move back and forth between the structural design and instantiation/configuration phases. Such a process increases the cognitive load taking into account that teachers should be technically skilled on workflow modeling to achieve their goal. An overview of the current methodology can be seen in Figure 3. The use of more abstract representations would be more helpful regarding the use of workflow schemas. This way, particular aspects specification may be relegated to the configuration/instantiation phase.

VIII. Related works

The main issues discussed here are caused by the decision of using workflow schemas to specify the generic data flow structures as basis for reuse. Since it is well known that collaborative processes can be modeled using workflow patterns but they cannot be reused. [21].

Miao et al. identify in [6] a new limitation on the IMS-LD model and proposes a proprietary CSCL scripting language in which core elements are explicitly defined such

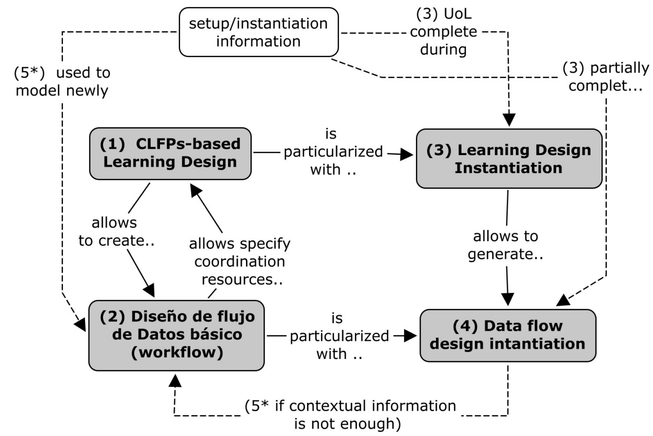

Figure 3: *Leadflow4LD* methodology and implications (additional steps) from complex adaptive designs

as artifact, groups, and dynamic featured related ones. This approach pays attention to reusability of artifact flow issue and adaptation processes. However, it does not allow for an interopeable solution with respect to the current standard.

In [8], Vantroy et al propose COW (*Cooperative Open Workflow*) driven by certain requirements like the support of collaborative activities, the dynamic redefinition of learning path and the reuse of existing courses and activities models. However the proposal requires extensions (e.g. workitem notion) that would disable interoprability with XPDL and IMS-LD standards. The main limitation refers to interoperability with XPDL standard and again the technological distance from the community accustomed to use of IMS-LD as the main language to describe their courses.

Thus, we propose to employ abstract representations in the initial phases in order to support reusable and adaptive CSCL scenarios. In this case generic workflow templates can be useful to create global workflow instance(s) for the particular scenery data flow [22].

IX. Conclusion

This paper has analyzed in a rather complete case study the performance of a solution to the dataflow problem in CSCL scripts. The combined use of adaptation and collaborative learning flow patterns has allowed us to define a realistic case study for such an analysis. The discussion presented in the paper suggests that the *LeadFlow4LD* methodology does not adress properly the reusability issue in such complex scenarios. Thus, we propose the use of abstact workflow templates to handle the creation of concrete instances of dataflow, and therefore we suggest important changes of the design methodology. Several aspects should be analyzed in the near future, such as the effect of the tool features, the analysis of different combinations of CLFP and AP, as well as the development of a prototype solution of the alternative methodology.

Acknowledgment

This work was financially supported by a MAEC-AECID fellowship to the first author, as well as the Spanish Ministry of Research and Innovation (TIN2008-03023/TSI) and the Castilla and León Regional Government (VA106A08) SOFOCLES projects.

References

[1] R. Koper, "The learning design specification," in *Learning Design, A Handbook on Modelling and Delivering Networked Education and Training*. Springer–Verlag Berlin Heidelberg, 2005, pp. 1–20.

[2] Weinberger, I. Collar, Y. Dimitriadis, K. Mkitalo-Siegl, and F. Fischer, *Technology-Enhanced Learning. Principles and Products*. Springer Verlag, 2009, ch. Computer-supported collaboration scripts: Theory and practice of scripting CSCL. Perspectives of educational psychology and computer science., pp. 155 – 174.

[3] R. Koper and Y. Miao, *Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies*. IDEA Group, 2007, ch. Using the IMS LD standard to describe learning designs, pp. 16–48.

[4] L. Vignollet, C. Ferraris, C. Martel, and D. & Burgos, "A transversal analysis of different learning design approaches," *Journal of Interactive Media in Education*, vol. 26, p. 11, 2008.

[5] J. R. Dalziel, "Lessons from LAMS for IMS Learning Design," in *ICALT '06: Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 1101–1102.

[6] Y. Miao, K. Hoeksema, H. U. Hoppe, and A. Harrer, "CSCL scripts: modelling features and potential use," in *CSCL '05: Proceedings of th 2005 conference on Computer support for collaborative learning*. International Society of the Learning Sciences, 2005, pp. 423–432.

[7] L. Palomino-Ramirez, A. Martinez-Mones, M. Bote-Lorenzo, J. Asensio-Perez, and Y. Dimitriadis, "Data flow between tools: Towards a composition-based solution for learning design," in *Advanced Learning Technologies, 2007*, vol. 0, Los Alamitos, CA, USA, Jul 2007, pp. 354–358.

[8] T. Vantroys and Y. Peter, "COW, a flexible platform for the enactment of learning scenarios," in *Proceedings of the Ninth International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2003)*, ser. Lecture Notes in Computer Science, no. 2806, Grenoble, France, 2003, pp. 168–182.

[9] L. Palomino-Ramírez, M. Bote-Lorenzo, J. Asensio-Pérez, and Y. Dimitriadis, "LeadFlow4LD: Learning and data flow composition-based solution for learning design in cscl," *Groupware: Design, Implementation, and Use: 14th International Workshop, CRIWG 2008, Omaha, NE, USA, Sept 14-18, 2008, Revised Selected Papers*, pp. 266–280, 2008.

[10] G. Decker and J. Mendling, "Process instantiation," *Data Knowledge Engineering*, vol. 68, no. 9, pp. 777–792, 2009.

[11] D. Hernández Leo, Y. Dimitriadis, and J. I. Asensio Perez, "Computational Representation of Collaborative Learning Flow Patterns Using IMS Learning Desing," *Journal of Educational Technology and Society*, vol. 8, no. 4, pp. 75–89, 2005.

[12] A. Karakostas and S. Demetriadis, "Adaptation patterns in systems for scripted collaboration," in *CSCL'09: Proceedings of the 9th international conference on Computer supported collaborative learning*. International Society of the Learning Sciences, 2009, pp. 477–481.

[13] P. Dillenbourg and F. Hong, "The mechanics of CSCL macro scripts," *International Journal of Computer-Supported Collaborative Learning*, vol. 3, no. 1, pp. 5–23, Mar 2008.

[14] S. Demetriadis and A. Karakostas, "Adaptive collaboration scripting: A conceptual framework and a design case study," in *CISIS '08: Proceedings of the 2008 International Conference on Complex, Intelligent and Software Intensive Systems*, F. Xhafa and L. Barolli, Eds. Los Alamitos, USA: IEEE Computer Society, 2008, pp. 487–492.

[15] P. Dillenbourg, "Over-scripting cscl: The risks of blending collaborative learning with instructional design," in *Inaugural Address, Three Worlds of CSCL. Can We Support CSCL?*, P. A. Kirschner, Ed. Heerlen: Open Universiteit Nederland, 2002.

[16] D. Hernández-Leo, J. Asensio-Pérez, Y. Dimitriadis, M. Bote-Lorenzo, I. Jorrín-Abellán, and E. Villasclaras-Fernández, "Reusing IMS LD formalized best practices in collaborative learning structuring," *Advanced Technology for Learning*, vol. 2, no. 4, 2005.

[17] P. Dillenbourg and P. Tchounikine, "Flexibility in macro-scripts for computer-supported collaborative learning," *Journal of Computer Assisted Learning*, vol. 23, no. 1, pp. 1–13, 2007.

[18] S. Wilson, "Workflow and web services," CETIS, Tech. Rep., 2005. [Online]. Available: http://www.e-framework.org/Portals/9/Resources/SOAandWorkflow2.pdf

[19] L. Palomino-Ramrez, M. Bote-Lorenzo, J. Asensio-Prez, L. de la Fuente-Valentn, and Y. Dimitriadis, "The data flow problem in learning design: A case study," in *Proceedings of the International Conference on Networked Learning 2008*, Mayo 2008, pp. 285–292.

[20] A. Karakostas and S. Demetriadis, "Adaptation patterns as a conceptual tool for designing the adaptive operation of cscl systems." 2010, Educational Technology Research and Development. (in press).

[21] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow patterns," *Distributed Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003.

[22] Y. Gil, P. Groth, V. Ratnakar, and C. Fritz, "Expressive reusable workflow templates," in *Proceedings of the Fifth IEEE International Conference on e-Science,*. Oxford, UK: IEEE Computer Society, Dec. 2009, pp. 9–11.