Modeling Collaborative Learning Services - A Case Study

Surya Bahadur Kathayat and Rolv Bræk
Department of Telematics, NTNU, Norway
surya,rolv.braek@item.ntnu.no

ABSTRACT

This paper presents an approach to model collaborative learning activities. Interactions among collaborating participants in a collaborative activity are encapsulated in a special kind of building block. Such building blocks are then composed together in order to specify the ordering and causality among them resulting a learning activity-flow model. Using such activity-flow models, one can design a learning activity by applying various collaborative learning patterns such as Jigsaw and Pyramid. Our approach is illustrated using a case study of a city learning activity.

KEYWORDS: Collaborative Learning Activities, Modeling

1. INTRODUCTION

Emerging information and communication technologies are supporting teaching-learning process in various ways. Learners are becoming more mobile and at the same time they are interacting and collaborating more with co-located or distributed learners having similar interests or learning objectives. Emerging technology is also opening opportunities for informal learning activities besides traditional classroom based learning. Technology is being accessible to more and more users and educational practitioners such as learners are becoming more active in creating or customizing technological solutions on their own according to their needs. However the challenge is to provide proper notations and tools supporting flexible, reusable and customizable collaborative learning activities designs.

In this paper, we take the perspective that learning comes, not only from sitting at a desk and consuming content such as books, web pages and other materials, but also comes from being active and participating in collaborative activities with other learners and learning objects. Learning areas may be informal - such as in city wide collaborative learning as discussed in [7] where learners learn about a city by being in the city, and by interacting with other co-located or distributed learners. In such situations, learners collaborate and learn in social settings using groupware to support their activities.

We look into the ways to simplify the design and development of such collaborative learning activities (or interchangeably will be called services). In particular, we focus on modeling collaborative learning services and to automatically produce executables from there. Currently, IMS learning design specification [11] can formally describe any design of learning activities i.e., teaching learning processes for wide range of pedagogical approaches. There are some editors and tools supporting IMS learning design specification, however several authors have pointed out that IMS learning design is insufficient (discussed in Sect. 6) in modeling group based and collaborative learning activities. We use UML [23] for modeling such learning services.

By using the concepts of UML 2.x collaborations, it is possible to model the structure of collaborating entities in a service. UML activity diagram can then be used to specify their collaborative behavior i.e., how they interact with each other. Detailed interactions among interacting entities can be encapsulated into collaborative building blocks and such building blocks can then be put into a library and reused to compose a new and larger learning service. Note that some of the services in a system runs in the background while others may interact with users. This means that user-interface concerns (the behavior of a user while interacting with a service via graphical user interfaces) plays significant role while designing a complete system specifications. Interestingly, using a similar approach as services, it is also possible to model and compose user-interface concerns on a high level of abstraction. From such models, service can be implemented in an automated way using code generation techniques.

The modeling style has a number of interesting properties for the development of the collaborative learning services. The building blocks lead to an incremental specification style, since they can be developed and analyzed in isolation, and serve as interfaces between experts of different domains. For example an user-interface expert may design the UI blocks and put into the library. Similarly, a service developer may develop service blocks and put into the library. Educational

978-1-61284-639-2/11/$26.00

Table 1. Collaborative services and associated roles

Service identified	Collaborating entities	Description
Login Service	User, Server	allows an user to login to the system
Social Matching	User, Server	find the matched users
P2P Chat Service	User, User	allows peer to peer chat between users
Group Chat/Discussion Service	User, Group	allows facilitates group chat, discussion among group members
Quiz Service	User, Group, POI	quiz about a particular point-on-interest
Positioning Service	User, Group	sharing location information within a group
Configuration Service	Teacher, Server	allows to configure a system

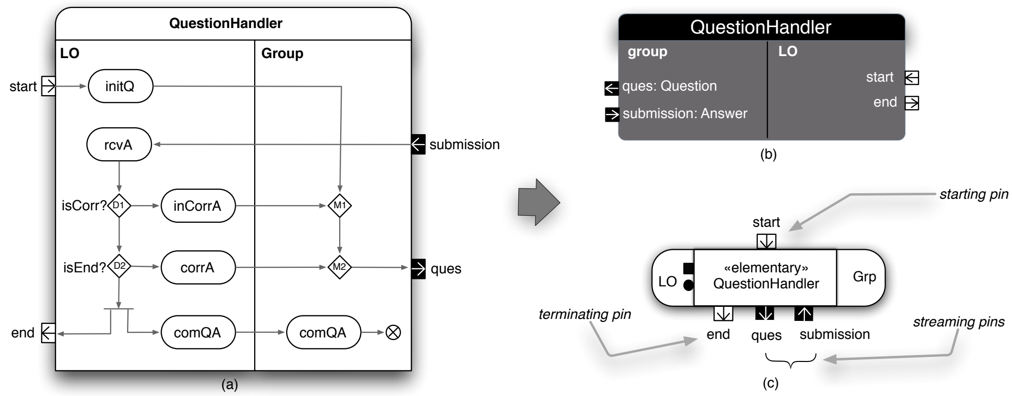


Figure 1. Question Handler Service Model

practitioners such as teachers may then compose a new service according to their needs putting together the available services and UI blocks in the library. Note that educational practitioners may use different collaborative learning patterns (c.f., Sect. 4) as guidelines while composing learning activities together.

The structure of the paper is following. Sect. 2 discusses a case study that will be used to illustrate the approach. Designing learning activities and user-interface concerns is discussed in Sect. 3. Design of collaborative learning patterns is discussed in Sect. 4. Implementation issues are discussed in Sect. 5. Related work is discussed in Sect. 6 and conclusion is given in Sect. 7.

2. A CASE STUDY

City learning activity: New students, called users in the following, are just arrived in a city and are interested to collaboratively learn about different POIs in the city.

Detailed description: The Users log in to the system (optionally configured by a teacher or an instructor) using their GPS enabled handheld devices. Users may create their profiles specifying their preferences and interests. Based on their locations and profile information, the users may be able to participate in different groups and social activities and learn about different POI around the city. The user may also opt to learn individually and share the knowledge with other users thereafter. When a user is trying to find a particular point of interest (e.g., historic building in a city), a user may interact and collaborate with co-located or distributed other users having similar interests. During the collaboration, users may for instance chat, have a group discussion and may participate in a quiz about a particular point of interest. Table 1 shows some services that collaborating users may use during a learning activity [13]. Note that collaborating entities represent domain entities which play part in the service. They are also called roles in a service.

Some learning activities may be implemented by a elementary service i.e., those which can not be decomposed further, whereas some learning activities may be implemented as a composite service where a service is composed from smaller services. Some of the services shown in Table 1 are elementary, while some are composite. In the following section, we discuss the design and composition of elementary and composite services.

3. DESIGNING LEARNING ACTIVITIES

In this section, we start discussing the modeling of elementary collaborative activities (services) and then we discuss the modeling of associated UI concerns and the composition of UI and service models. We take a quiz service as an representative example in the following.

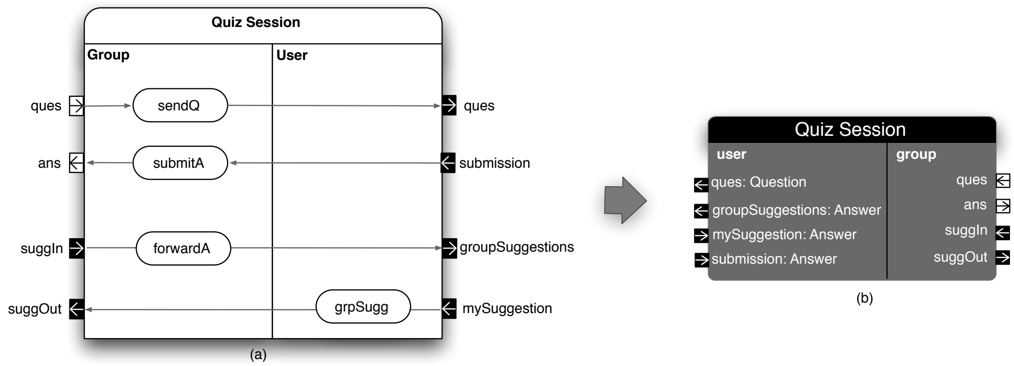


Figure 2. Quiz Session Service Model

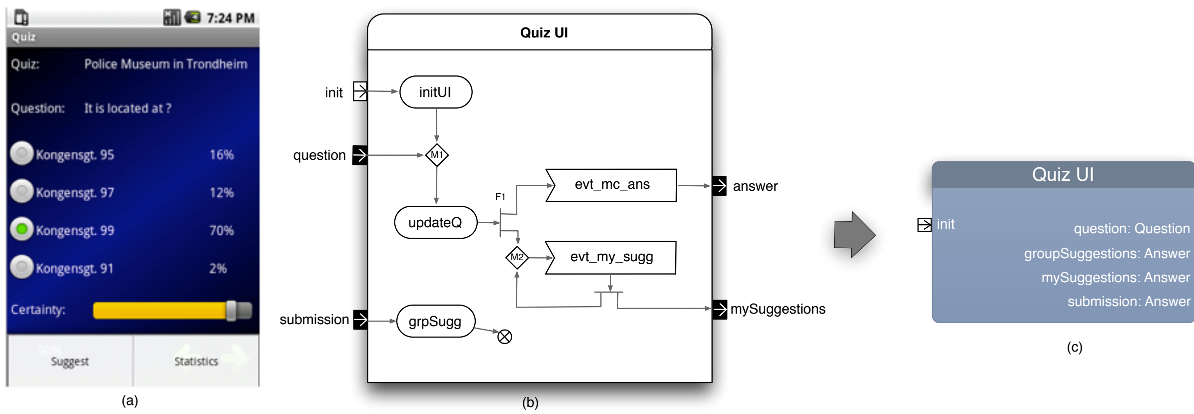


Figure 3. Quiz Session User Interface Model

We use UML 2.x activity diagram [23] to formally encapsulate the interactions among the collaborating entities in learning activity. Participating entities are represented as UML *ActivityPartition* elements and shown as swim-lanes for example in Figure 1. A collaborative activity is represented as special kind of building blocks having different types of input and output pins for instance streaming, starting and terminating types.[1] These pins are used to connect the building blocks together.

3.1. Designing Elementary Learning Activities

A quiz service is a composite service: it consists of two elementary services called *QuestionHandler* and *Quiz Session*. In a Question Handler service, a group and a learning object (LO) collaborate. A specification of a question handler service using UML 2.x activity diagram is shown in Figure 1(a). A *QuestionHandler* service is started with a *start* starting pin. Questions are then initialized in LO and sent to the group. These questions will be forwarded to the group members via streaming pin *ques*. When the group receives an answer via *submission* streaming pin, the group forwards it to LO. The correctness of the received answer is checked at LO and then accordingly either next question is sent to the group or completion of question-answer is indicated via *end* terminating pin.

Note that building block representations of *QuestionHandler* service are shown in Figure 1(b) and 1(c). Detailed interactions among collaborating entities are hidden in building blocks and they have pins at the boundaries which will act as interfaces or connecting points while composing two or more services together. The building blocks of type in Figure 1(b) are supported in our service engineering tool Arctis [19] where one can see that the service pins are local to particular role (or collaborating entity). Building blocks of type in Figure 1(c) are more abstract representation (an extension to Arctis, proposed in [15]) where pins are not local to any participating entities, but are owned globally by a service itself. Starting and terminating service roles are represented by black filled circle and square box respectively.

1. An Initiating pin will start the called activity if it is not started yet. An Terminating pin will terminate the called activity. An Streaming pin can pass tokens while the called activity is active, i.e. a flow between two activities connected by streaming pins allows the called activities to interact without being stopped.

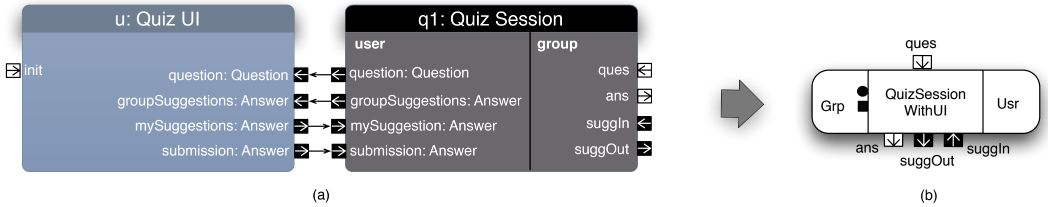


Figure 4. Composition of Quiz UI with Quiz Session Service Model

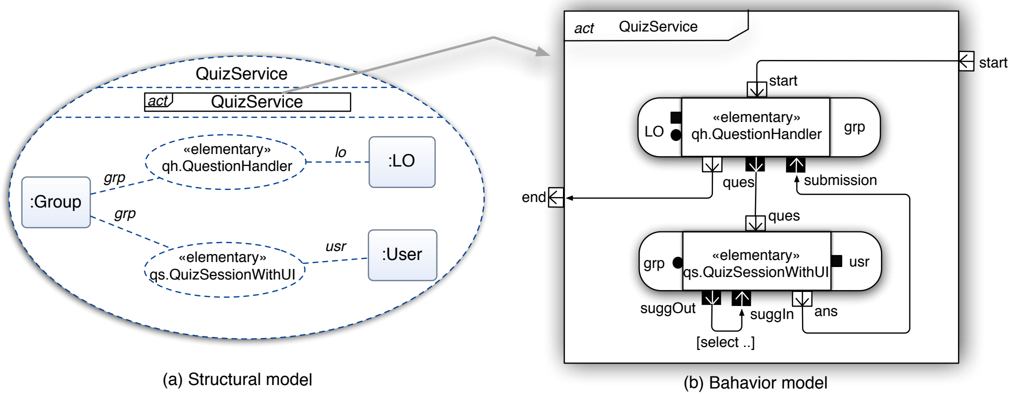


Figure 5. Quiz Service Model as Composition of Quiz Handler and Quiz Session services

Another sub-service of a quiz service is *Quiz Session* between group and users. The specification of this service is given in Figure 2(a) and the building block representation is in Figure 2(b). The service starts when a question is received at the group via *ques* starting pin. The received question is then forwarded to a user. The user may then suggest an answer to the group (via *mySuggestion* to *suggOut* pins) or receive suggestions from other members of the group (via *suggIn* to *groupSuggestions* pins). Finally the user may submit an answer (via *sumbission* to *ans* pins).

3.2. Designing User Interfaces Concerns

Note that users interact with *Quiz Session* service via graphical user-interfaces for example view the questions, submit suggestions and answers as shown in Figure 3(a). Such user-interface concerns can be encapsulated in a building block using UML activity diagram, using a same approach as we model services. Figure 3(b) shows the user-interface model that captures the user-interfaces concerns of a *Quiz Session* service. One can see in Figure 3(c) that there is only one activity partition (unlike in a service model where there are more than one partitions) representing that it is a local block.

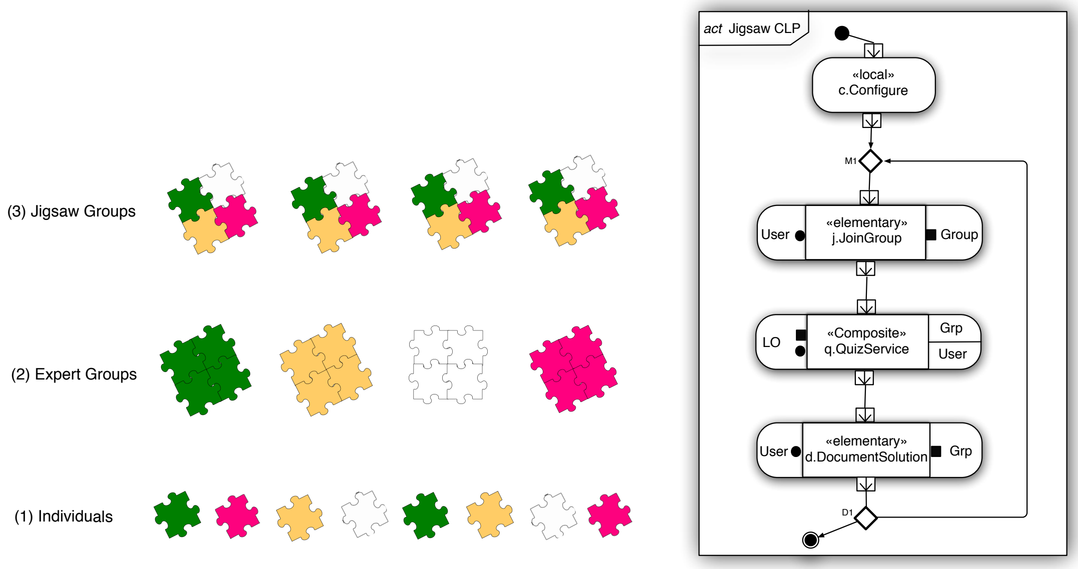
3.3. Composing UI Concerns with Service Models

As shown in Figure 4(a), user-interface (UI) block *u.Quiz UI* and service block *q1.Quiz Session* are composed together by connecting their pins together. The pins can be connected together either manually or semi-automatically based on their name and associated data type. Refer [17] for more details on designing comprehensive UI blocks for Java J2SE and Android platforms and composing UI blocks with service blocks.

Figure 4(b) shows the abstract representation of the composite block i.e., service and UI blocks composed together. In the following section, we assume that such abstract blocks are in the library and will be used to compose composite collaborative learning services.

3.4. Composing Learning Activities

Services building blocks can be composed together by connecting their pins together by specifying a flow between them. For example a composition of a *QuizService* which is composed from two elementary services *QuestionHandler* and *QuizSessionWithUI* is shown in Figure 5. Structural model, representing structure of the collaborating participants and the services they involved in, is shown in Figure 5(a). The service behavior model is shown in Figure 5(b). It starts from a starting pin *start* and a flow then goes to the starting pin of *QuestionHandler*. The question handler service will send the question via *ques* streaming pin to the starting pin of *QuizSessionWithUI* service. Users collaborate in the *QuizSessionWithUI* service, solve the problem and submit an answer. When *QuizSessionWithUI* service terminates, which is indicated by a terminating pin, a flow goes to



(a) Jigsaw Collaborative Learning Pattern

(b) Choreography Model of Jigsaw Learning Activity

Figure 6. Jigsaw Collaborative Pattern and City Learning Activity

incoming streaming pin of *QuestionHandler* service. As discussed in Sect. 3.1, the *QuestionHandler* service then checks the correctness of the answer and accordingly either send the new question or terminate the service which is represented by the flow line from terminating pin of *QuestionHandler* to terminating pin of a composite *QuizService*.

4. LEARNING PATTERNS

Collaborative learning patterns are defined as the formalization of good practices in structuring the sequence of collaborative (or not) learning activities or services [20]. We therefore take them as guidelines while describing the composition i.e., the activity-flow of learning activities. In the following, we discuss *Jigsaw* and *Pyramid* collaborative learning patterns while composing a city learning activity.

4.1. City Learning Activity with Jigsaw CLP

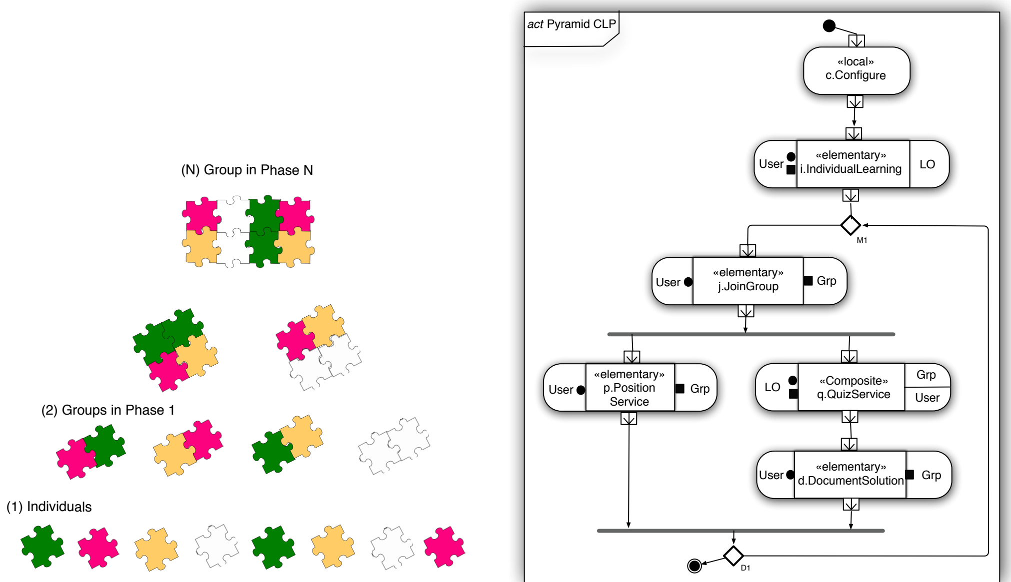
A Jigsaw collaborative learning pattern [5] is shown in Figure 6(a). In Jigsaw collaborative learning pattern, *individuals* initially join *expert groups* where they collaborate and become experts on a particular subject or topic. (Note that different colored Jigsaw pieces represent individuals having different pieces of knowledge.) Thereafter, experts from different expert groups form a new group called *Jigsaw groups* where they share their knowledge to each other. In this way, all the users learn about all the topics to be learned.

A city learning activity based on the Jigsaw pattern can be organized as following: First, students will be divided into *expert groups* where group members learn about particular POI and become experts i.e., knowing detailed information about it. In each expert group, students collaboratively learn together - identify the location of the POI, solve the quizzes about that particular POI, make presentation and report together. Once students in a expert group solved their assigned tasks, they become experts and then they will be reassigned to *Jigsaw groups*. In Jigsaw groups, each member are experts on some particular POI as they come from their original expert groups. The task of each expert in Jigsaw group is to share their knowledge about the POI of which that they are experts and to learn about other POI from other experts. Once all the experts have shared their knowledge, all the students will have learned about all the POI in a city.

Figure 6(b) shows the learning activity-flow model as the composition of services supporting the city learning activity based on Jigsaw collaborative learning pattern. The users join in a group using *j.JoinGroup* service. Group members then collaborate, for example using *q.QuizService* and find a common solution about a particular POI and document it using *d.DocumentSolution* service. Thereafter, members in a group depart from their original expert group and join to a Jigsaw group using *j.JoinGroup* and group members collaborate and know about the point-of-interests shared by each members of the Jigsaw group. (It is assumed that nature of the quiz service will be the same in jigsaw groups and expert groups.) The process may continue until all the students know about all the point-of-interests to be learned which is indicated by last decision node D_1 in Figure 6(b).

4.2. City Learning Activity with Pyramid CLP

Pyramid collaborative learning pattern [5] is shown in Figure 7(a) where each each student initially studies the



(a) Pyramid CLP

(b) Choreography Model of Pyramid Learning Activity

Figure 7. Pyramid Collaborative Pattern and City Learning Activity

problem individually. Thereafter they join to the larger group and share their knowledge. The smaller groups join to the bigger one and share their knowledge until all the users are in the same larger group and develop a shared knowledge.

A city learning activity based on the Pyramid pattern can be organized as following: Each individual student studies the problem i.e. locate and acquire more detail information about a particular point-of-interest(s). Thereafter some students form a group and share the knowledge about the POIs they have learned, compare their solutions and develop a common shared knowledge. After that, those smaller groups join in larger groups, collaborate and acquire the knowledge about the new POIs that other groups have initially learned. In this way, at the end all the students will develop a shared knowledge about all the point-of-interests to be learned in a city. Figure 7(b) shows the learning activity-flow model of a city learning activity as the composition of the services based Pyramid collaborative learning pattern. User individually learn themselves about a particular POI using *i.Individual Learning* service. Thereafter they join in a group using *j.JoinGroup* service and collaboratively learn about any POI using *q.QuizService* and document common shared solution using *d.DocumentSolution* service. At the same time, they can use other services such as *p.PositionService* which is shown in Figure 7(b), chat service and etc. The process may continue (i.e., groups may join to the another group and the collaborate) until all students are in a same group.

5. IMPLEMENTATIONS

We use a model-driven development (MDD) [6], [8], [16] approach for the development of collaborative learning services. Our development approach starts from abstract models which are close to the problem domain and are understandable by end users or domain experts. Such abstract models are transformed into more detailed models and so on until one can automatically generate application code. Our learning activity-flow models, which are described as the composition of services, represent models with highest level of abstraction. They represent the work-flow of learning activities. Such models uses the semantics of UML activity diagram with small notational extension which is provided through UML profile in [15]. These learning activity-flow models are platform independent and specify the pure functionality of a particular solution or an applications. Such models however may contain different types of design faults leading so called *realizability problems* in [14]. Therefore learning activity-flow models can be model-checked for such errors, identified errors can be resolved (if resolvable for example rules in [14]) or informed to the learning activity-flow designer about the potential point of problems which need to be manually resolved. After that the abstract learning activity-flow models can be automatically transformed into more detailed models using the rules in [14]. The resulting models can then be imported in to Arctis tool-suit [19]. The Arctis tool can then be used to model-check for other realizability problems, and to generate application code for different platforms such as Java and Androids [17], [18]

6. RELATED WORKS

Various standards and specifications for learning design are proposed and being used in the area of computer supported learning, in particular e-learning, mobile learning,

Table 2. Comparative summer of work work and IMS learning design

	IMS Learning Design	Our Approach
Development Approach	Top down	Bottom up
Semantics	formal (XML)	Semi-formal (graphical)
Notation	None	UML activity diagram
Learning Activities	Local activities	Local and collaborative activities
User Interfaces	No	Yes
Reusability	No	Yes
Workflow Design	Sequencing of local activities	Sequencing of local and collaborative activities
Who is it for	Developers/Teachers trained to IMD LD	Teachers knowing bases of work flows

and computer supported collaborative learning. IMS learning design [11], [12] (grounded on EML [10]) is relevant work in the context of this paper i.e., modeling the work flow of learning activities. IMS learning design (LD) provides a framework to specify the ordering and work-flow among the learning activities in the form of unit of learning (UOL) which specifies who does what, when and with which facilities in order to reach the learning objectives. Note that creation of unit-of-learning also involves (besides the flow of activities) the bundling of all associated resources such as files, web references, learning materials, learning service configurations and etc.

Several authors have pointed out that IMD LD is insufficient to model collaborative learning activities, and accordingly propose extensions to IMS LD. Caeiro et. al. [24] proposed the extension to meta-model of IMS LD by introducing the concepts of *community* to support collaborative activity. The *community* has local activities within it to support collaboration for multiple individuals. Hernandez et al in [9] proposed an extension to the IMS LD *service* descriptions defining a special type of service called *group service*. Authors in [25] however pointed the some of the limitations of the proposed preliminary concept of *group service* such as on the limited awareness and on the specification of privileged roles. Miao et. al. [22] also discussed some of the limitations of IMS learning design, in particular on modeling groups, complicated control flows and various forms of social interactions. There are some learning design editors such as: RELOAD [1], CopperAuthor [2], CoSMoS [21], and MOT+ editor [3]. RELOAD, CopperAutor and CoSMoS presumes that learning designer have sufficient knowledge about IMS learning design constructs and specifications. Collage tool [9] is a graphical tool, based on RELOAD, for authoring collaborative learning activities. MOT+ editor in addition provides some graphical representations for facilitating authoring tasks to some extent. LAMS editor [4] uses a set of predefined (learning) activities in the library and a new activity is designed just by dragging and dropping such activities and connecting them together. It is however not compliant to IMS LD. Unlike most of the approaches, we use UML activity diagram which has intuitive and rich flow-constructs for modeling activities. For modeling collaborative learning activities, we encapsulate interaction among collaborating entities in collaborative building block and later these blocks can be composed together. This gives more flexibility compared to existing IMS LD based techniques, specially in composing collaborative activities together.

Our approach also allows multiple individuals to interact together in collaborative activities using a concept of session [16]. Summary of the comparison of our work and IMS learning design related works is shown in Table 2. Unlike IMS LD based learning design, our approach provides the reusability of learning activity-flow models or unit of learning. We also support the modeling of user-interfaces in a similar ways as services or activities. Due to the graphical/visual nature of notations, we believe that our approach is intuitive for educational practitioners who has bases of workflows but need not be trained with formal semantics of IMS LD.

7. CONCLUDING REMARKS

We use UML activity diagram to model a work-flow of learning activities. With this, we aim to implement the concept of learning design which are not specifically intended for IMS learning design specifications. Such learning activities flow models capture requirements of educational practitioners and then one can go all the way down to the automatic generation of application code. Reusability and flexibility are the major benefits of our approach where services are designed and put into the library and can be reused, flexibly composed in other contexts while making a composite service. With the proposed approach, one can model (and then compose) user-interfaces and service concerns in a unified way. Our notations are intuitive for representing collaborative activities as well as UI blocks as one can see the information about the participating entities, their roles (starting or terminating or participating) and the interfaces (as connecting points) in a abstract service notation. Our approach is partially tool supported as well.

In the future work, we aim to transform our learning activity-flow models to IMS learning design (LD) compliant models.

Since our models use UML semantics, there may be straight forward way to transform such UML models into IMS LD scripts, some related works are [12], [25]. For that extensions in IMS learning design may be necessary in order to support the concept of collaborative activities .

REFERENCES

[1] RELOAD Project Website, http://www.reload.ac.uk/.

[2] CopperAuthor Project Website, http://www.copperauthor.org/.

[3] MOT+ Editor Website, http://www.licef.teluq.uqam.ca/.

[4] LAMS Project Website, http://lamsfoundation.org/.

[5] J.I. Asensio, Y.A. Dimitriadis, M. Heredia, A. Martinez, F.J. Alvarez, M.T. Blasco, and C.A. Osuna. Collaborative learning patterns: assisting the development of component-based cscl applications. In *Parallel, Distributed and Network-Based Processing, 2004. Proceedings. 12th Euromicro Conference on*, pages 218 – 224, 2004.

[6] Rolv Bræk and Oystein Haugen. *Engineering real time systems: an object-oriented methodology using SDL*. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1993.

[7] C. I. Canova and M. Divitini. Reflections on the role of technology in city-wide collaborative learning. *International Journal of Interactive Mobile Technologies (iJIM)*, 3, 2009.

[8] Humberto Nicolás Castejón. *Collaborations in Service Engineering: Modeling, Analysis and Execution*. PhD thesis, Norwegian University of Science and Technology, 2008.

[9] D. Hernández-Leo, E. D. Villasclaras-Fernández, J. I. Asensio-Pérez, Y. Dimitriadis, I. M. Jorrín-Abellán, I. Ruiz-Requies, and B. Rubia-Avi. COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology & Society*, 9(1):58–71, 2006.

[10] H. Hummel, J. Manderveld, C. Tattersall, and R. Koper. Educational modelling language and learning design; new opportunities for instructional reusability and personalised learning. *Int. J. Learn. Technol.*, 1:111–126, November 2004.

[11] IMS. IMS Learning Design Version 1.0 Final Specification. http://www.imsglobal.org/learningdesign/.

[12] IMS. IMS Learning Design Best Practice and Implementation Guide. http://www.imsglobal.org/learningdesign/.

[13] S. B. Kathayat and R. Bræk. Platform support for situated collaborative learning. In *International Conference on Mobile, Hybrid, and On-line Learning, 2009. ELML '09.*, 2009.

[14] Surya Bahadur Kathayat and Rolv Bræk. From flow-global choreography to component types. In *System Analysis and Modeling (SAM)*, volume 6598 of *Lecture Notes in Computer Science (to appear)*. Springer - Verlag, 2010.

[15] Surya Bahadur Kathayat, Simon Han, Hien Nam Le, and Rolv Bræk. Choreography model transformation by graph transformation - a case study. In *International Conference on Evaluation of Novel Approaches to Software Engineering (sumbitted)*, Beijing, China, 2011.

[16] Frank Alexander Kraemer. *Engineering Reactive Systems: A Compositional and Model-Driven Method Based on Collaborative Building Blocks*. PhD thesis, Norwegian University of Science and Technology, 2008.

[17] Frank Alexander Kraemer, Surya Bahadur Kathayat, and Rolv Bræk. Unified modeling of service logic with user interfaces. In *MoSE+DQS '09: Proceeding of the first international workshop on Model driven service engineering and data quality and security*, pages 37–44. ACM, 2009.

[18] Frank Alexander Kraemer, Vidar Slåtten, and Peter Herrmann. Engineering Support for UML Activities by Automated Model-Checking — An Example. In *Proceedings of the 4th International Workshop on Rapid Integration of Software Engineering Techniques (RISE)*, November 2007.

[19] Frank Alexander Kraemer, Vidar Slåtten, and Peter Herrmann. Tool support for the rapid composition, analysis and implementation of reactive services. *Journal of Systems and Software*, 82(12):2068 – 2080, 2009.

[20] D. H. Leo, J. I. Perez, Asensio, and Y. A. Dimitriadis. IMS learning design support for the formalization of collaborative learning patterns. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, pages 350–354, Washington, DC, USA, 2004. IEEE Computer Society.

[21] Y. Miao. Cosmos: Facilitating learning designers to author units of learning using IMS LD. In *Proceeding of the 2005 conference on Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences: Sharing Good Practices of Research, Experimentation and Innovation*, pages 275–282, Amsterdam, The Netherlands, 2005.

[22] Y. Miao, K. Hoeksema, H. U. Hoppe, and A. Harrer. Cscl scripts: modelling features and potential use. In *Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, pages 423–432. International Society of the Learning Sciences, 2005.

[23] OMG. Uml 2.2 superstructure specification, February 2009.

[24] Manuel Caeiro Rodriguez, Luis Anido Rifon, and Martin Llamas Nistal. Towards ims-ld extensions to actually support heterogeneous learning designs. a pattern-based approach. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, pages 565–569, Washington, DC, USA, 2004. IEEE Computer Society.

[25] D. Yu and X.M. Chen. Creating computer supported collaborative learning activities with IMS LD. In Julie Jacko, editor, *Human-Computer Interaction. HCI Applications and Services*, volume 4553 of *Lecture Notes in Computer Science*, pages 391–400. Springer Berlin / Heidelberg, 2007.