Adapting Process-Oriented Learning Design to Group Characteristics

Yongwu Miao and Ulrich Hoppe
Institute of Computer Science and Interactive Systems
University of Duisburg-Essen, Germany

Abstract. IMS LD supports the design of personalized learning by adapting activities and other process elements to personal characteristics. This paper presents a new idea to adjust activities and elements to group characteristics for supporting ‘groupalized’ or group-adapted learning. This may help to improve effectiveness and efficiency of group-based collaborative learning. Our approach to formalize adaptive learning designs for groupalization is based on an extension of IMS LD. As a first “proof of concept”, a fully implemented prototype system is presented.

Introduction

Adaptation to a learner’s personal learning objectives, interests, preferences, performances, and other characteristics is a key challenge in many research areas concerning learning technologies such as Intelligent Tutoring Systems [12] and Adaptive Educational Hypermedia [1][11]. Typical approaches to personalized learning are adjusting contents, their structures, and presentations to personal characteristics. At present, there is a trend in learning technologies that the emphases shift from content to activities. The publication of IMS Learning Design [3], an international standard designed to promote exchange and interoperability of content with a focus on facilitating reuse of instructional strategies, can be considered as a positive step forward in this direction. IMS LD provides a "meta-language" which can be used to describe a wide range of pedagogical approaches. A pedagogical approach can be described in IMS LD as a set of structured learning activities in a formalized process model.

One of the objectives of IMS LD is to support personalization [5]. Based on IMS LD several attempts have been made to support personalized learning by adapting activities and other process elements within a unit of learning to personal characteristics/requirements [2][8][9][10]. The basic idea is that a process model can be reused by multiple learners. Learning processes will be adapted to different personal characteristics and profiles of learners. IMS LD provides mechanisms for formalizing adaptive learning process models, which can be automatically executed at run-time system.

In this paper, we propose a new idea -- to support group-based collaborative learning by adapting activities and elements to the characteristics of groups. Corresponding to the term of personalized learning, we use the term ‘groupalized learning’. Groupalized or group-adapted learning is a kind of learning design tailored for individual groups according to the diversity in group characteristics. This leads to a number of new research questions: whether it is generic and significant to develop flexible collaborative learning process models to suit for different groups; how such an adaptive learning process model can be formalized; what a group model should be developed for this purpose; what factors should be taken into account for adaptation; what elements within an activity such as tools and content can be adjusted and how this can be
done; what support is needed in run-time systems; how automatic adaptation and human-involved adaptation can be integrated; what is the relation with personalized learning and so on. Our assumption is that, like personalized learning processes for individuals, groupalized learning processes may help to improve the learning of groups, if adaptive learning designs can be appropriately specified.

In order to go in this direction we have to take the first step. The focus of this paper is on presenting the generic approach and supporting the specification of adaptive learning design for groupalized learning. In the next section, we use a scenario to analyze the characteristics of group-based learning processes and show an example of adaptive learning design. Then, we identify the requirements to specify adaptive learning designs. After presenting our approach and a prototype system to support formalizing adaptive learning designs for groupalization, we conclude our work with indicating future directions.

1. Characteristics of Adaptive Learning Design

In order to help us to analyze the characteristics of adaptive, group-based collaborative learning processes, we will introduce and discuss a scenario that is based on an open issue given by a teacher. The design rationale of this learning design is to create conflicts among students and engage them into interactions to resolve the conflicts and to avoid the situations in which some participants dominate the discussion and others just behave as listeners.

1.1 A Scenario

In a class a teacher gives an open issue to students and requires students using a “pair argue” method. Students are experienced in applying this method because they usually conduct discussion by adopting this method. Each student has a stable partner in the class. Toni and Darina are two students as a pair. First each student writes a position statement independently. When both students in a pair finish writing, they will check whether they have the same position. If having opposite positions, they will argue and try to resolve conflicts. Otherwise, they will exchange position statements with another pair in which both students have a common agreement as well but an opposite position to theirs. Toni and Darina have opposite opinions and after arguing no one can persuade the other. Then, each pair looks for another pair to conduct a discussion according to the following rules: either two homogeneous pairs (both students have the same position) with different positions, or two heterogeneous pairs. If some pairs cannot find an appropriate combination (e.g., all pairs take the same position), the teacher will arrange specific activities for these pairs, for example, assigning some pairs to take the role of objectors and facilitating a debate with an opposite role. After forming a big group, Toni and Darina will continue their debate with an assistant. Finally, the teacher facilitates a debriefing discussion in the class. After the class, each pair has to write a synthesis as homework.

1.2 An Example Model

Figure 1 shows a UML activity diagram that specifies the pedagogical approach, the “pair argue” method described in the scenario. This process model presents an e-learning version of an adaptive learning design with ten activities, two branches, and two sets of artifacts. Among activities, “writing” is an individual activity; “forming groups” is a supportive activity done by an automatic agent; “teacher arranging” is a supportive activity also performed by the teacher; “debriefing” is a session of all students and the teacher. The rest of activities are pair activities. In this diagram, we use the notations (0) and (2) to represent two types of homogeneous pairs,
respectively. The notation (1) represents a heterogeneous pair. The notations \{(0), (2)\} and \{(1), (1)\} represent two kinds combinations: one is the combination of two homogeneous pairs with different positions and the other is a combination of any two heterogeneous pairs. The notation “fail” refers the event that the automatic agent cannot make appropriate combinations for some pairs. This event results in the intervention of a teacher. It is note that some details are ignored for focusing on the control flow and data flow of the model. For example, certain tools may be used in activities such as chat tool, shared whiteboard, shared text editor, issue-based argumentation tool, A/V conferencing tool and so on.

**Figure 1:** An Example Model of Adaptive Learning Designs for Groupalization

1.3 Characteristics of Adaptive Learning Design

From this simple example, we can see some characteristics of adaptive learning design for groupalization.

First, a pedagogical method can be described as a process model that can be repeatedly executed by multiple groups at the same time or at different time. There may be synchronization points in this process. However, it is possible for different groups to take the same process model with different paths at a different pace. The adaptation components are primarily learning activities rather than content. In fact, content is defined within activities. However, in this example there is no content defined in the process model.

Secondly, although there are individual activities and community activities, most activities are group activities. A group as a whole goes through the process from the beginning to the end. Each group activity will be done collaboratively and will terminate when the whole group rather than an individual finishes the tasks. Furthermore, each group will have static and dynamic characteristics while executing the model. In this example, according to the positions of both students in a pair, each pair must fall into one of three categories: (0), (1), and (2). This can be regarded as a kind of dynamic characteristic of the group.
Thirdly, certain characteristics will be used to determine the learning path of each group. In this example, there are two checkpoints and alternate paths in the diagram are based on whether the positions of two students in a pair are the same or not. In addition, multiple alternative activities are available at each checkpoint and each group with certain characteristics will take appropriate activities. In the example, the first branch specifies two options: one for homogeneous pairs to select the “reading” activity and the other for heterogeneous pairs to select the “arguing” activity. In the second branch, the category of pairs is used as a primary factor to determine the path of a pair as well, although it is not the unique factor in this case. Sometimes users may also adapt activities to groups’ characteristics.

2. Requirements for Formalizing Adaptive Learning Design

As mentioned before, the emphasis of this paper is on formally representing adaptive learning design for groupalization. According to the characteristics of adaptive learning designs, a formal process modeling language should have mechanisms to represent the following aspects to support adaptation for groupalization.

2.1 Representing Pedagogical Models

A process modeling language should have mechanisms to represent a whole learning process, not only including content but also including roles, learning activities, services, control flow, data flow, etc. Such a description should be a computer-executable model. The components and their relationships within the model can be used to decide upon adaptation.

2.2 Representing Group Models

A group model used for adaptation should not only maintain generic information about the group (e.g., name, members, creation time, form-policy, etc), but also maintain dynamic information (e.g., active activities, finished activities, intermediate outcomes, etc). Such information captures the characteristics of groups that is used for adaptation.

2.3 Representing Adaptation Models

The process modeling language should provide mechanisms to define the adaptation logic as well as adaptation actions. The former is responsible for relating information available models (e.g., process model, activity model, content model, group model, etc) and assessing whether adaptations are required. The latter refers to specifying the very actions (e.g., showing/hiding activities, forming high-level groups, making configuration of tools, setting property value, making content visible/invisible, etc) that need to be effected by the system for a given adaptation to be achieved. In addition, it must allow the designer, when desired, to pass the control over the adaptation process to users.


IMS LD [3] is a meta-language for modeling learning designs. When trying to use IMS LD to model group-based collaborative learning processes, we see several difficulties. In order to solve the problems, we developed a computer supported collaborative learning (CSCL) scripting language by extending IMS LD. The generic considerations and a whole picture about the CSCL scripting language have been described in [7]. This paper focuses on
discussing an additional issue in detail -- how the extended language can be used to support the formalization of adaptive learning designs for groupalization. Our generic approach is to reuse the mechanisms provided by IMS LD originally for constructing adaptive rules for personalization. Concretely speaking, IMS LD level B and level C introduce mechanisms of properties, conditions and notifications, which can be used to specify arbitrarily complex dynamic behaviors of a system [6]. This section presents how these mechanisms are reused and extended to meets the requirements identified in the last section.

3.1 Reusing IMS LD to Specify Pedagogical Models

Rather than attempting to capture the specifics of many pedagogical models, IMS LD does this by providing a generic and flexible language that is designed to enable express many different pedagogical models. It can be used to express the pedagogical meaning and functionality of the different data elements within the context of a learning design. By using IMS LD, a learning design can be represented in the following way. People with certain roles work individually or collaboratively towards certain outcomes by performing a set of structured activities within associated environments, in which appropriate learning objects and services are available. In addition, IMS LD provides mechanisms to formalize activity models, content models, user models, role model, etc. These models are useful for specifying adaptation. Therefore, we primarily use IMS LD to specify pedagogical models.

3.2 Introducing Groups

The conceptual framework of IMS LD does not include the concept of group. Within the framework, role is an entity relevant to the group. The notation of role can be used to model group in many learning designs. However, mixing up these two different concepts may lead to serious mistakes when modeling some collaborative learning processes. For example, in the scenario if a role is defined for each student pair, how many roles have to be defined in the example learning design model? In fact, in this learning design model we can define a role “student pair”. Then each pair as a whole takes this role. In order to enable the simple and intuitive modeling of group-based collaborative learning, the concept of group is explicitly introduced into the conceptual framework. A group can have individual members and sub-groups. All groups with their person members are structured as a directed-acycle-diagram. A group as a whole can take a role. A group has attributes such as identifier, name, max-size, min-size, person members, super-groups, sub-groups, engaged roles, form-policy, disband-policy, dynamic/static, creation-time and so on. Furthermore, a group model not only encapsulates general information about the group, but also maintains a “live” account of the group’s actions within the system.

In order to support modeling dynamic characteristics or pedagogy-specific characteristics, we add concepts of local group property and global group property in the framework of our process modeling language. Like a local person property, a local group property has a different value for every group in a run. The property is owned by the run of the unit-of-learning, specifying a value per group. In the example model the combination of the pair opinions can be modeled by using a local group property. Like a global person property, a global group property can have a different value for each group, independent of the different executed instances of units of learning. The group entity owns the property specifying the portfolio of the group. For example, the “pair synthesis” produced in the last activity of the example model can be modeled as a global group property, because the value of this property (a synthesis) may need to be stored permanently by the run-time system as a kind of group information. Such information will be used by other learning designs.
Properties can be used to define property-groups as well. Therefore, a group model can be specified with static characteristics and dynamic characteristics.

### 3.3 Enriching Adaptation Logics and Adaptation Actions

IMS LD allows for describing personalization aspects within a learning design, so that the activities and content within a unit-of-learning can be adapted based on the preferences, portfolio, prior knowledge, educational needs and situational circumstances of users. At level B, an adaptive rule can be represented as a condition clause in the following way:

```plaintext
if <condition> then <actions> else <actions>
```

In order to express adaptation logics and adaptation actions, IMS LD provides limited operations on process elements, called element operations in this paper. There are two categories of element operations. The purpose of the first category of element operations is to get the state of a given element at run-time (like the method get() in JAVA) such as datetime-activity-started, users-in-role, when-property-value-is-set, and activity-completed. If a parameter such as an activity, a role, or a property is past to the element operations described above, the element operations will return a value such as a time, a set of user identifiers, or a Boolean. The element operations in the second category will effect on the state of a given element at run-time (like the method set() in JAVA) such as set-property, change-property-value, and show/hide (changing the status of the “isvisible” attribute of the given element). In addition, IMS LD provides `{expression}` schema group to facilitate specifying complex adaptation logics for personalization.

However, the element operations are insufficient to support modeling adaptation logics and adaptation actions for groupalization. As an extension, we introduce new operations: Examples of extended get()-like element operations are users-in-group and roles-taken-by-group. The examples of extended set()-like element operations are assign-group-to-role and remove-user-from-group. In addition some de-/construction operations are added such as create-role and delete-group. Also, we add declaration mechanisms to define complicated expressions and actions. An expression declaration primarily consists of two parts. The first part is a representation of internal operational structure based on the extended IMS LD `{expression}` schema group and element operations. The second part is a user-friendly representation of the expression with a set of parameters. Correspondingly, an action declaration is defined in the same way. We add 'collection' data type and loop control structure to support complicated declarations. A declaration is indeed a procedure writing in the process modeling language, which can be interpreted into executable programming language code based on element operations. After being defined, a declaration can be saved in the modeling environment, and it can be used to define other high-level declarations as well. Then, an expression or an action can be defined by referring to the declaration with parameters without concerning about the internal operational structure. Therefore, we can support learning designers to specify rich adaptation logics and adaptation actions.

In addition, IMS LD adds the capability for a learning designer to specify sending messages and setting new activities based on certain events at level C. We extend such a notification mechanism by introducing the concept of interaction rules. An interaction rule is specified by defining a condition, an agent (e.g., a user, a group, or a role), a permission right, and a set of actions. It will be triggered by certain events and informs an agent to perform actions. The run-time system will provide appropriate user interface for the associated users to perform an action directly rather than just receiving a notification via an email. This mechanism can be used to support human involved adaptation. For example, in the example model, when it fails to form high-level groups, the run-time system should update the user interface of the teacher’s client for the teacher to adjust activities for the remaining pairs.
4. An Authoring Tool

As mentioned before, we developed a CSCL scripting language [7]. One objective of the language is to facilitate formalization of adaptive learning design for groupalization. Based on the language, we developed an authoring tool, called CoSMoS (for “Collaboration Script Modeling System”). It can help designers to understand and specify learning designs (or CSCL scripts), which can be translated from/into XML-formatted files automatically by the tool. The adaptive learning design files can be used by a run-time system to adapt the course during the execution by adjusting the activities to groups’ characteristics and by providing appropriate user interfaces for the group members.

![Figure 2: The User Interface of COSMOS and the Definition of the Example Model](image)

The user interface of CoSMoS is shown in Figure 2. The window of the tool consists of a toolbar and two panels. The left panel is used to define the structure of adaptive learning designs and the right panel is used to create detailed designs for the selected process element. We have applied the tool to defining several CSCL scripts and so far we found that the tool and the underlying CSCL scripting language provide sufficient mechanisms to model adaptive learning designs. Figure 2 shows a definition of the example model described in the section 1. In the structure panel, the ‘pair argue’ script is shown as a tree. Expression declaration nodes, action declaration nodes, and other modeling environment components are listed below the script nodes. The editing panel shows the specification of the first activity. In this panel, an adaptive rule is specified by defining a conditional expression including a local group property “pair category” and two alternative activities. The run-time system will adapt activities according to such a definition.
5. Conclusions and Future Work

The objective of this paper was to outline a framework for an education modeling language that integrates new elements for supporting groupalized learning. The proposed framework is based upon IMS LD, which provides mechanisms to specify adaptive learning designs for personalization. After introducing the group element and adding element operations, declarations, and interaction rules, our CSCL scripting language can meet requirements identified through an analysis of a scenario. The preliminary 'proof of concepts' of the CSCL scripting language was given in using our authoring tool CoSMoS. Preliminary tests show that adaptive learning designs for groupalization can be formalized by using CoSMoS.

As described our approach facilitates the specification of learning designs derived from pedagogical principles without representing deeper reasons for the one or the other choice of method or interaction pattern. Evidently, existing work on intelligent group formation and the management of learning groups (as, e.g., described in [4]) could extend this approach with "expert knowledge".

The validation results of the real experiments will have to look into more detail whether the approach taken is successful. In particular, experiments should be conducted on the corresponding run-time systems to demonstrate the adaptability during the execution of adaptive learning designs. We have confidence in this approach, because IMS LD can support personalization. Therefore, our next step is to develop a compatible execution environment that can interpret CSCL scripts and provide run-time supports. Meanwhile, we will develop more adaptive learning designs to facilitate groupalized learning.

References