Towards Adaptation Languages for Adaptive Collaborative Learning Support

Alexandros Paramythis[1] and Alexandra Cristea[2]

[1] Johannes Kepler University
Altenbergerstraße 69
A4040 Linz, Austria
alpar@fim.uni-linz.ac.at

[2] Department of Computer Science
The University of Warwick, Coventry
CV4 7AL, United Kingdom
acristea@dcs.warwick.ac.uk

Abstract. Collaborative learning involves interaction between students, whilst adaptive learning traditionally involves individual learning progress. Each of these learning paradigms has its advantages and disadvantages, which are mainly disjunctive. In order to create a more powerful learning experience, in an ideal world, these two paradigms should be integrated, in order to alleviate each others' weaknesses. This would render it possible to allow for the coexistence of collaboration in learning environments, along with adaptation and personalization, but would also enable *personalized collaboration*, as well as *collaborative adaptation* in learning. This however requires extensions to the way adaptation, personalization and collaboration are approached today. In this paper we gradually extract patterns and abstract specifications and finally show how such extensions can be applied to an adaptation language for personalized learning, in order to enable collaborative learning support.

Keywords: collaborative learning; adaptation language; collaborative adaptation language; LAG

1 Introduction

Personalization in learning environments is currently a strong focus of new research developments. The European Union (EU), for instance, strongly supports programmes that build adaptive, personalized learning platforms at the European level (see, e.g., successful EU projects such as PROLEARN[1], PROLIX[2], etc.), as well as encourages commercial uptake of the developed technologies. A widely adopted perspective in such efforts has been that learning is an individual experience, and thus individual traits are essential in customizing the right learning package for each student.

[1] http://www.prolearn-project.org/
[2] http://www.prolixproject.org/

On the other hand, current learning management systems (LMSs, such as Blackboard[3], Sakai[4], Moodle[5], etc.) as well as trendy Web 2.0 environments are based on, and support, a traditional learning paradigm of collaboration, based on constructivist learning. Learning in such spaces is a social activity, flourishing from the interactions with peers, experts, etc.

In this paper we endeavour to unite these two important learning paradigms, taking the position that both are important to enhance the learners' experience.

The remainder of this paper is organized as follows. In section 2 we outline a number of collaborative learning scenarios, which require various types of adaptation to one or more users. Subsequently, in section 3 we extract requirements based on these scenarios, as well as high level patterns and pseudo-code reflecting the scenarios. In section 4, some of the pseudo-code is translated into *adaptation language* code, leading to language extensions for a generic adaptation language, LAG [4]. This work is novel in combining the two worlds of adaptation and collaboration and making the first steps towards a reusable abstraction level for it. Section 5 discusses work that relates (albeit remotely in some cases) to the research outcomes presented herein. Finally, in section 6 we draw conclusions and we outline further work.

2 Collaborative Learning Scenarios

To drive the discussion we are presenting below a few representative scenarios for collaborative learning, that implicitly or explicitly involve adaptive support on the part of the system. These scenarios will be analysed in the following section to extract requirements that refer to both the learning infrastructure, and, more importantly, to adaptation authoring language primitives necessary for expressing the adaptation logic in the scenarios.

Overall Learning Set-up

Students are learning in an online adaptive learning environment. They have access to various online learning materials, and can communicate for the purpose of their study with each other via online tools. Experts in the study area may be available on request, via said online communication tools, but may also be unavailable due to other responsibilities.

In the following, some of the tasks of the stakeholders in the environment as described above are detailed into concrete scenarios.

Scenario 1

A learner, John, is studying about databases, and specifically about the concept of 'Sub-queries'. He struggles with this concept, and would like some help. He requests (from the system) that he be put in touch with experts on the subject. After contacting

---

[3] http://www.blackboard.com/

[4] http://sakaiproject.org/

[5] http://moodle.org/

one or more experts, John can communicate with them via the online communication tools. After receiving the advice he needs, John will continue his personalized study.

Scenario 2

Mary and Anne are chatting about their 'Neural Networks' course. They realize that none of them understands enough about 'Energy functions'. Mary requests (from the system) a list of course participants currently online and likely to be willing to help. The system compiles a list of online persons known to be very active in helping their fellow students. After having a joint chat with their peers, and receiving the explanations they need, Mary and Anne resume their own joint study and chat.

Scenario 3

Samantha is a student of the 'Adaptive Systems' course. She remembers that she has recently spoken to some expert, or one of her peers, about 'Adaptive navigation support'. She would like to contact that person again to ask about 'Adaptive presentation'. She asks the system to show her the last five peers that she has been chatting with recently about her current course. She then contacts Jonathan, who has helped her previously with the topic of 'Adaptive navigation support', and is happy to help her now with 'Adaptive presentation'.

Scenario 4

Julia is a teacher of English literature to a class of 15 students. She teaches this course via the online adaptive collaborative system. She decides that students need to work on assignments in groups. She creates three learning tasks: 'Grammar study', 'Essay writing' and 'Argumentation'; she also decides on three different sets of criteria for (automatically) grouping students for each of the tasks:

The first group division, for learning English grammar, is based on similarity of the student's knowledge levels for the subject matter. Julia decides on having three groups, beginner, intermediate and advanced, for this study task.

The second group clustering, for writing essays on Renaissance poets, is based on complementarities of the student's knowledge of the subject's sub-topics. Students previously studied materials on different poets. In each group, there will be at least one student that read about Shakespeare, one about Marlowe and one about Sidney. The exact number of resulting groups is not strictly defined.

Finally, the last grouping for argumentation is based on previous peer activity, as monitored by the system. Julia asks the system to group together at least two active students with two non-active ones, in order to help the non-active ones to better participate in the course.

Scenario 5

Continuing from the previous scenario, one of the student groups working on the task of 'Essay writing', comprising Lee, Jane and Mustafa, have missed the deadline for their first draft of their essay. The system has recorded many recent discussions between Lee and Jane, but Mustafa hasn't been present in any of them. The system, based on the students' previous track record, proposes as a remedy action a one-week extension to the submission deadlines. It contacts the instructor to receive permission for granting the extension[6]. The system then first modifies the deadline adding one week to it, and then sends a message to the three students, with the notification of the extension, and a recommendation for Mustafa to be more active in the group. The system also informs the group that, if Mustafa will not become active and the group doesn't meet the deadline, Mustafa will be removed from the group, and Lee and Jane will need to submit a reduced draft by themselves. The coordinating instructor has ultimate control over the entire preceding adaptation process, and can reverse any system actions which she considers inappropriate.

3 Requirements based on the Scenarios

To establish the requirements that the above scenarios place on the adaptation language, we will first analyse them in more detail, to get a better understanding of: (a) the information presumed to be available / maintained by the system in its various static and dynamic models, and accessible to the adaptation language, and (b) the facilities (tools, services, etc.) available in the system and, again, accessible to the adaptation language (either for retrieving information, or for effecting adaptations). Following this analysis, we will extract and discuss in more detail the new requirements imposed on the adaptation language itself, in the subsequent section.

3.1 Analysis of the Scenarios

Each of the scenarios is analysed using the following structure: we start with a brief *scenario summary* for ease of reference; we continue with providing a possible *outline of the process* involved in adaptively supporting aspects of the scenario; following that, we identify the *information that needs to be maintained by the system*, as well as the *requirements imposed on the learning / collaboration environment* more generally. Note that, for the sake of conciseness, the analysis of all scenarios after the first is incremental, i.e., we only list the requirements that the scenario exhibits and have not already been identified in previous scenarios.

Scenario 1

Scenario summary:
John requires expert help with his current study topic.

Process outline[7]:
Upon a user's selection of the "requiring expert help" function,
set *current-topic* to the topic of the page currently being read by the user;
identify experts on *current-topic*:

[6] An alternative approach, assuming the instructor has configured the system appropriately, would be for the system to perform the actions described in the rest of the scenario, and offer the instructor an easy way to retract these actions at a later time – something the instructor should be able to do anyway.

[7] This represents high level "pseudo code" for the adaptation workflow.

set *experts* to the list of users whose knowledge of *current-topic* is above a given threshold,

order the *experts* list by: (a) the users' knowledge of "parent topics" to *current-topic*, and (b) the users' knowledge of the study subject matter more generally;

add to the *experts* list users that have been registered as experts on the study subject matter.

For each user on the *experts* list:

determine whether they are online; if not, remove them from the list.

Provide the resulting list to the user, and start / continue monitoring their communication-oriented activities.

Wait until the user has finished communicating with people on the *experts* list, and then automatically return the user to the context / page from which they requested assistance.

Information to be maintained by the system[8]:

- The *current course topic* studied by the learner[9]. This is required to automatically identify the study context within which the user is requesting help.
- The *learners' knowledge per topic*, along with a "threshold" mechanism that determines when someone has become an expert (e.g., a peer that has studied that topic already, and has completed the related self-assessments successfully). This is required to automatically determine topic experts.
- *Registered experts* (of the study subject matter) with the system. If available, this information can also obviously be used to determine topic experts. This information can also alternatively be deduced in an adaptive manner, by extracting, for instance peers with a knowledge level above a certain threshold, or peers with a knowledge level above the current user.
- The *online status of users*, possibly with respect to the current course. This is required to determine whether a user is available to be contacted at a given point in time.

Environment requirements:

- A personalized learning environment, capable of monitoring study activities. This is necessary to build the learners' knowledge model. Also, a way to "remember" study context so that learners can manually or automatically return to it, after having interrupted their study to perform a different task within the system.
- A communication tool capable of supporting at least person-to-person communication, so that John can contact a single expert at a time.
- A manual / automatic facility to start a communication channel (for a communication started by John, or by the system).

[8] Please note that here 'the system' can be a conglomeration of components or modules working together towards a common goal. For instance, adaptation can be provided by an adaptation engine, whilst communication (and its respective tracking) can be provided by a communication tool.

[9] We assume the system employs an overlay model – typical of adaptive educational hypermedia systems –, which would allow for study topics / concepts to be uniquely identified.

Scenario 2

Scenario summary:

Mary and Anne want to discuss a specific study topic with helpful peers.

Process outline:

Upon request for peers to help (from within the chat tool):

Establish the *common current topic(s)* of discussion of the chat participants[10].

Establish a *list of expert peers* on the topic(s)[11].

For each person in the *experts list*:

if the person is not currently online, remove that person from the list.

For each person in the *online experts list*:

set *recent received help sessions* to the number of similar sessions the person has initiated in the last 30 days;

set *recent offered help sessions* to the number of similar sessions the person has participated upon request in the last 30 days;

set *current load* to the number of open tasks for the person that the system is aware of;

compute an *availability index* for the person based on *recent offered and received help sessions* and *current load*[12].

Sort the list of online experts on the basis of the people's availability index.

Send invitations to help to the first two people on the list,

and repeat until two people have accepted, or the list is exhausted.

For each person that accepts the invitation,

extend the chat session and automatically add that person to it.

When the last invited expert leaves the chat session,

revert all used tools to their context before the beginning of the help session.

Information to be maintained by the system[13]:

- Knowledge about the *current topic* / course being discussed by the students. Required to identify students that are already familiar with the topic / course.

[10] This can be done in one of the following ways (in decreasing order of sophistication and demands on system capabilities): (i) if the system supports text analysis, try to infer the topic from the chat text, cross correlated with the content of courses that both students participate in; (ii) if the system supports "co-browsing", then derive the topic from the page currently viewed by both students; (iii) if neither of the above is possible, then: (a) derive the individual topics from the pages that each student is looking at, and (b) if the topic is not the same, search for the first "parent" topic of the two, and (c) if such a "parent" topic does not exist, use the topics in conjunction. Any of these three approaches can be scaled to more than two students at a time as necessary. The last two approaches can also be used in non-textual communication tools, where inferring the context of a conversation may be impossible or forbidding in terms of required resources. A simpler alternative would be to have communication tools implicitly or explicitly associated with courses.

[11] It is assumed that this is done in a way similar to what has been described for Scenario 1.

[12] This is meant only as an example of what is possible. The availability index could take many other things into account as well, starting from the duration of the help sessions, to more "exotic" metrics based on sophisticated behaviour analysis to determine a person's propensity to help specific individuals.

[13] Incremental to previous scenarios.

– *History* of users' interactions with peers (frequency, responsiveness, time involved in helping, etc.) Required to evaluate a person's willingness to help and availability (depending, among other factors, on how busy a user is / has been recently)

Environment requirements[13]:

– A communication tool capable of supporting group communication, so that Mary and Anne can participate in a discussion with one or more of their peers.
– Manual / automatic switch from communication tool to the personalized learning environment (if Mary and Anne should be studying after finishing the conversation).

Scenario 3

Scenario summary:

Samantha wants to ask a peer she has recently communicated with about a different topic of the same course.

Process outline:

Upon request for recent discussion partners on related topics:
  Set *current-topic* to the topic of the page currently being read by the user.
  Set *topic hierarchy* to the list of all parent topics of the *current topic*, up to the level of the enclosing course.
  Set *recent conversation partners* to all users with whom Samantha communicated in the last two months.
  For each person in the *recent conversation partners* list,
    and for each conversation that Samantha has had with that person:
      if the topic of the conversation is included in the topic hierarchy or in sub-topics thereof,
        include that person in the *potential communication partners* list.
  Order the potential communication partners list by:
    the frequency of the most recent conversation,
    the distance of the current topic to that conversation's topic[14].
  Select the top 5 persons on the list, and show Samantha an interactive "contact" list with these persons[15].

Information to be maintained by the system[13]:

– History of previous communication sessions between users, *including knowledge about the topics discussed*.

Environment requirements:

– No new environment requirements in comparison to scenarios 1 and 2 above.

[14] Several different algorithms can be used to calculate that distance (e.g., depth of first common ancestor of both topics). The appropriateness of different algorithms depends mainly on the knowledge domain model, and the organization of course materials.

[15] The list is assumed to contain links, which, when selected, establish a communication session with the respective person.

Scenario 4

Scenario summary:

Julia wants to have her students automatically grouped for three different learning tasks, using three different grouping policies.

Process outline:

Note: For space economy we will only analyze one of the three grouping sub-scenarios, the one where people with complementary knowledge are grouped together – the rest of the cases are similar. We will also assume that Julia, the instructor, has selected the following parameters when setting up the tasks: the grouping policy; the topics on which complementarity should be sought[16]; the number of people per topic that should be present in each group; the system / environment facilities that should be made available to each group.

Upon request for automatically grouping a course's students:
  If the grouping policy is "by knowledge complementarity",
    Retrieve a list of *all course participants*
    For all complementary topics:
      create a (empty) *list of participants that know a topic*
    For each user in the *participants list*,
      and for each of the *complementary topics*:
        If the user's *knowledge of the topic* is above a set *threshold*,
          add the user to the *list of users that know the specific topic*.
    Invoke the grouping algorithm with parameters:
      the list of topics and people that know each of them, and
      the number of people from each topic that should participate in the resulting groups
      (The algorithm returns a set of uniquely named groupings and their participants)
    For each of the *returned groupings*:
      Create a real group in the learning environment;
      add the specified users as participants of the group;
      make available to the group "private" facilities for the members to use for their group work; and,
      notify the users about the creation of the groups, and the availability of the group facilities

Information to be maintained by the system:

Although this scenario doesn't really mandate the maintenance of additional information (other than what we have already seen before), it does make use of information in different ways:

- The students' level of knowledge of a set of topics is used in the first grouping policy to determine how similar students are to each other in this respect.
- The students' coverage of the knowledge domain is used in the second grouping policy to determine whether they complement each other appropriately.

[16] The algorithm described can also work with minor modifications for sets of topics.

– The history of the users' activities within the system (including the studying of materials, communication with peers, use of other system facilities, etc.) is used to determine the users' overall level of activity in relation to a specific course (and possibly also along several courses).

Environment requirements:

– The system may need to have external to the adaptation language grouping / clustering policies and algorithms, which it makes available to the adaptation engine at run-time. This is discussed in more detail in the next section.
– The learning environment must support a concrete organisational unit that can be used for group work. This includes the possibility to assign to the group system facilities that are "private" to the group and cannot be accessed by members of other groups.

Scenario 5

Scenario summary:

The group of Lee, Jane and Mustafa has missed a deadline and the system attempts to mediate the collaboration process.

Process outline:

Upon a group's missing of a submission deadline for a course assignment
If the deadline has not already been extended before
Determine the *average number of activities* performed by group members, in the context of the course (e.g., total number of activities by group members in course / number of group members); designate this as the *average participation* threshold.
Determine *non participative group members* by checking whether their participation is substantially below the group's *average participation.*
Determine *participative group members* – the rest of the people in the group.
If there are more *participative* than *non-participative* members, then
(Provisionally[17]) add one week to the deadline
(Provisionally) introduce an automatic reduction in the group's mark
(Provisionally) modify the model of non-participative users to reflect their contact in this group / assignment thus far
(Provisionally) add a recurring task that will closely monitor the activities of the previously non-participative group members; if those members' participation remains low, this task will eventually trigger the removal of the said members from the group.
Notify instructor of system's intention to mediate the process (by listing all provisional actions described above), and wait for approval[18].
If the instructor approves of the system's proposed adaptive behaviour, then

[17] The term "provisionally" is used here to indicate that the specified actions are "prepared" but not yet "effected" by the system; whether they will actually be applied depends on the instructor's approval, as described later on.

[18] Alternatively to the process described here, the instructor may have set up the system to perform such adaptations automatically, without asking for permission. In that case, this step of the process would be omitted, and the next step would happen unconditionally.

Apply the provisionally decided upon actions from above.
Notify group members of modifications and expected activities on the group members' part.

Information to be maintained by the system:

– "Process models" (or "activity models") of different group activities, possibly at different levels of granularity / strictness, including models of artefacts used / created during collaborative activities. Process semantics are necessary to be able to reason about (semi-)structured group activities, and to be able to intervene in these activities. This refers both to the requirement to have access to a full model of all potential activities a group may engage in, and to track the group's activities with reference to the model.

Environment requirements:

– The facilities and services that make up the educational system must be accessible programmatically, so that changes can be effected by the system directly, as if they were being made interactively by a user with the appropriate permissions. Note that this is different from being able to automatically initiate the use of a facility by the end users, which was mentioned previously.
– The monitoring information of user activities must not only be possible to query (a requirement implied in several of the previous scenarios), but must also be observable directly at run-time (so that different activities can be used as "triggers" for adaptive system behaviour, as in this scenario).

3.2 Synthesis of Requirements for Adaptation Languages

Based on the above analysis we can now identify the most important characteristics an adaptation language must exhibit in order to be able to express the kind of scenarios in the previous sections. Note that the following list is not exhaustive, and as such should not be seen as an enumeration of *sufficient* language features for adaptively supporting collaborative learning, but rather of *required* ones. Also note that the list is not prioritized, and features are presented in related groupings.

Representation of, and Operations on, Groups
To start with, there is the obvious requirement that the language support the explicit representation of groups. This can be further split into: (a) having the means to refer to groups in general, as well as to concretise a reference to a specific group; (b) being able to access the groups a user is participating in; (c) being able to access the members of a group (either through a general or through a concretised reference); (d) being able to differentiate between the models of individual users and the models of the groups in which the users participate; and (e) supporting set operations involving the members of one or more groups.

Workflow- or Process- based Reasoning
To fully support groups in the context of the activities that their members perform, it is necessary to have a model of those activities (much in the same way that to support

individualized navigation through course materials, a system needs to be able to represent and reason about these materials). This can be achieved by following either workflow- or process- based approaches, as long as these approaches support: (a) the representation of actors involved; (b) the representation of activities performed by these actors; (c) the representation of any constraints that apply to these activities (structurally, temporally, procedurally, or otherwise), including ones that define completion / continuation conditions for the activities; (d) the representation of artefacts used or generated during these activities; etc.

A separate matter, but highly relevant to situations such as the one described in Scenario 5, is the possibility for the language to express the definition of "triggers", i.e., conditions that should result in specific instances of adaptation logic being invoked by the system.

Temporal Operations

Although arguably subsumed by the features just discussed (workflow- or process-based reasoning), support for temporal operations is worth discussing separately for two reasons: firstly, they are very important as user activities inevitably have a temporal dimension to them, which is often neglected in adaptive content-oriented hypermedia systems; secondly, temporal operations may serve as a "poor man's" version of simplified reasoning over activities when full support for workflows or processes is not present, or cannot be added to a language.

Support for Expressing or Accessing Grouping and Clustering Policies

Grouping or clustering policies and algorithms are vital when attempting to support adaptive group formation, where there is often the requirement to automatically assemble a group among learners that do not have sufficient familiarity, or knowledge of each other's characteristics, to perform the task themselves.

One possible approach to this type of support is to embed it directly in the language. The main benefit of this approach is that policies can easily be assembled and modified by the adaptation author. However, such policies often involve complicated algorithms which may be difficult to express (let alone comprehend by the average author) in declarative, non-programmatic terms, or with limited programmatic constructs available.

An alternative approach is to have these policies expressed through other means, and introduce in the adaptation language a way of inspecting the required parameters programmatically, and invoking the policies at run-time with provided or adaptively calculated values.

Invoking System Facilities and Manipulating System State

This refers to being able to: (a) "start" facilities on behalf of the users (or at least facilitate the starting of these facilities); and (b) programmatically changing system state information – not necessarily represented in the adaptation models themselves.

The first feature is necessary in cases where the system has to facilitate to initiate communication sessions on behalf of the user, including tasks such as inviting others to participate in a chat session, etc. As we have seen in several of the scenarios, this is an inextricable part of bringing together students without imposing on them the burden of all required interaction steps, and, more importantly, overcoming the quite tangible social inhibitions such actions might incur if they were to be carried out by the users themselves.

The second feature is required when attempting to adaptively support the collaboration process itself. Referring back to the fifth scenario, one can see that there exist situations where the system might need to undertake tasks that affect several parts of a system in order to adjust characteristics of the said process, much in the same way an instructor, or external human observer would

Support for "Provisional" Adaptation Decisions
This refers to the capability to "pre-decide" on adaptations, express them in a human-readable form, and request permission to apply them. This is not necessarily a feature required only by group-oriented adaptation, but it's more pronounced when: the effects of an adaptation may affect several people; the adaptation itself is an embodiment of an educational approach (in which the instructor may play a vital role); several people may need to agree before an adaptive behaviour is performed. It should be noted here that although, semantically, the proposed feature calls for separating between making and effecting adaptation decisions, this separation does not need to exist at the level of the technical implementation. For instance, the employment of techniques such as "continuations" in programming languages, or nested transactions from database systems, could provide a sufficient technical framework for implementing this feature.

4 LAG Extensions for the Scenarios

Adaptation languages are 'vehicles of explicit intelligence in adaptive hypermedia' [14], used to author the adaptive behaviour once and reuse it for various courses, for various systems, and with various users / learners. The LAG *adaptation language* [6], [1] is used for expressing personalization in learning environments. In order to express collaborative interactions, however, such a language would have to be extended with the features detailed in the previous section. In the following, we describe some possible extensions based on a selection of the above scenarios. Due to the lack of space, only one scenario is detailed via the LAG language, and one presented as Annex.

Scenario 1 Code Snippet[19]: *John requires expert help with his current study topic.*

```
// if the user model is set to the first scenario:
IF UM.requestingExpertHelpOnTopic == true THEN (
   FOREACH GM.User DO
   IF enough( // if the next 3 conditions are satisfied
   // if the knowledge of others for the course is
   // greater than the knowledge of the current user:
```

[19] Please note that this is only a snippet of the LAG strategy of adaptation.

```
  UM.GM.User.GM.knowledge > UM.GM.knowledge
  // and if their knowledge is greater also for the
  // current concept
  UM.GM.User.GM.Concept.knowledge >
                            UM.GM.concept.knowledge
  // and they are online
  UM.GM.User.onlineStatus = true, 3)
   // then show this other's chat link
  THEN (UM.GM.User.chat.show = true)
) // record that the 1st scenario has been performed
UM. requestingExpertHelpOnTopic == false
```

Similarly, the other scenarios can be expressed in the LAG language with the appropriate extensions. The Annex presents another scenario. For brevity, here we only analyze what extensions needed to be performed (minimally) on the LAG language, in order to be able to express the collaboration as above.

Extensions:
For expressing collaborative adaptation in the above example, the following extensions had to be made to the LAG language, and the following *new items* had to be introduced:

- **`FOREACH`**: used to go through all topics, or through a list of peers, etc., and to perform actions on all of them. Possible usage is:
  - `FOREACH GM.User`: meaning for each user of the current *GM* (standing for Goal Map, and representing the current lesson).
  - `FOREACH GM.Concept.other`: meaning for each concept other than the current concept within the given lesson.
- **`GM.User`**: used for accessing users other than the current user; this was previously not necessary, as the adaptation only reflected upon the current user in personalized non-collaborative environments. Thus, `GM.knowledge` referred to the current knowledge of the current user about the current lesson, or `GM.Concept.knowledge` referred to the current knowledge of the current user about the current concept in the current lesson, etc.
- **`other`**: to denote actions that are applied on other topics than the current topic. Thus, in the above `GM.Concept.other` we can access also other concepts beside the current concept the student is accessing (or learning about). This is especially useful in collaborative environments, where interaction between users means often a more extensive interaction with the various concepts at a given time. In the following, we compare our work with various related work directions.

5 Related work

The level of abstraction we envision in our work, that can lead to reuse of the collaborative adaptation strategies, is expressed in our paper via an adaptation language. An alternative to this is to express such sequences and interaction via *workflow languages*[20]. However, workflow languages have previously been shown to be insufficient to express personalization at a level of expression as delivered by adaptive hypermedia [10]. Although not shown here, the LAG language can express various personalization strategies (based, e.g., on preferences, learning styles, goals, etc.) and thus is more powerful with the proposed extensions than regular workflow languages.

A popular current growing competitor to adaptation languages and adaptive hypermedia expressivity is *IMS-LD*[21]. Research has shown however that IMS-LD is still not yet capable of delivering all adaptation functionality as defined by adaptation hypermedia [1], [13], and also has serious limitations when it comes to adaptively supporting collaborative learning [12].

Other approaches, such as other adaptation languages exist (see, e.g., LAG-XLS [14]). The later language caters for learning styles, but it would need further extensions to cater for more extensive personalization as well as collaborative aspects.

Adaptation languages are based on the rule-based approach. Alternatively, reasoning mechanisms can be used to express the adaptive behaviour (e.g., description logic). The new developments in the Semantic Web offer new vehicles for reasoning such as RDF[22], OWL[23] (used also by adaptive learning systems, such as the Personal Reader [8]). They may provide viable alternatives for the future, but currently systems based on such mechanisms have serious performance problems when compared to other rule-based systems. Furthermore, whereas such approaches are very good in expressing modelling information, interrelations, etc., they lack support for programmatic constructs required to express most of the behaviours we have discussed in this paper.

Yet another direction of adaptation representation is the family of "assembly-level" adaptation languages, such as used in systems as AHA! [7], Interbook (Word-document-based) [8], WHURLE (LP: lesson plan) [10]. The problem with such languages is that, not only do they lack many of the required facilities as outlined in previous sections, but they are also extremely verbose and difficult for non-experts to express high-level, reusable adaptation strategies in, rendering them a questionable choice for employing as the basis of adaptive collaboration support. They may, nevertheless, serve as an appropriate "end-of-line representation", in essence serving as a possible "output format" for higher level languages such as LAG. The same is true for specifications such as IMS LD already mentioned above.

Our approach is also related to Pattern languages [1]: extracting snippets of adaptive behaviour (here, for collaborative adaptation) that are to be reused in different context (e.g., by different learners, teachers; groups of learners or teachers; with different course materials, etc.).

---

[20] http://www.yawlfoundation.org/
[21] http://www.imsglobal.org/learningdesign/
[22] http://www.w3.org/RDF/
[23] http://www.w3.org/TR/owl-features/

6 Conclusions and further work

In this paper we have analysed, based on a scenario-driven approach, the ways to implement adaptive collaboration and collaborative adaptation in learning environments. Scenarios were transformed to pseudo-code and requirements, and finally to language extensions for a specifically chosen language.

This process is however more general, and can be applied for any system aiming at using the synergetic effects of the combination of the two learning paradigms: personalization and collaboration.

We consider, specifically, the derived requirements to be of high value in deciding the general appropriateness of a language for supporting aspects of adaptive collaboration, or collaborative adaptation. A careful review of these requirements reveals that to ensure full support for the scenarios presented herein, one would have to combine approaches and features from several different "source" fields and language paradigms (e.g., support for workflow- or process- based reasoning on the one hand, and support for expressing grouping or clustering policies on the other).

A danger one has to acknowledge and heed when commencing to extend a language with all these features is to prevent the entry level barrier for language users from getting too high. The goal is not to end up with an adaptation language whose power and complexity are akin to those of a programming language – this would severely limit the target audience. The goal, instead, is to make such additions in a way that allows users with little programming skills, but ample domain knowledge (e.g., instructional designers) to make effective and efficient use of the new facilities. It is argued that this is vital for the uptake of the new approaches by an "audience" in which even simpler adaptive learning technologies may seem daunting, or excessively time consuming.

The work reported in this paper will be continued for extending and refining the LAG language, for immediate use in the ALS EU project (see Acknowledgments), as well as for longer-term use as a vehicle of transport for the specifications of the dynamics of personalized, collaborative learning.

Acknowledgments

This work has been performed in the framework of the ALS EU project 229714-CP-1-2006-1-NL-MPP. For more information please refer to the project's web site: http://www.als-project.org

References

1. Alexander, C.: A Pattern Language: Towns, Buildings, Construction. USA: Oxford University Press. ISBN 978-0195019193 (1977)
2. Berlanga, A., J., Francisco J. Garcia, F., J., Modelling adaptive navigation support techniques using the IMS learning design specification, Proceedings of the sixteenth ACM

conference on Hypertext and hypermedia table of contents, Salzburg, Austria, 148 - 150 (2005)
3. Cristea, A.I., Burgos, D.: Authoring Adaptive Hypermedia and IMS Learning Design: A possible understanding?, Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade, The Netherlands (2006)
4. Cristea, A.I., Verschoor, M.: The LAG Grammar for Authoring the Adaptive Web, ITCC'04 (International Conference on Information Technology) April, 2004, Las Vegas, US, IEEE (2004)
5. Cristea, A.I., Calvi, L.: The three Layers of Adaptation Granularity. UM'03, Pittsburg, US (2003)
6. Cristea, A.I., De Mooij, A.: Adaptive Course Authoring: My Online Teacher. Proceedings of ICT'03, Papeete, French Polynesia (2003)
7. De Bra, P., Smits, D., Stash, N.: The Design of AHA!, Proceedings of the ACM Hypertext Conference, Odense, Denmark, August 23-25, 2006 pp. 133, and http://aha.win.tue.nl/ahadesign/ (2006)
8. Dolog, P., Henze, N., Nejdl, W., Sintek, M.: The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies, AH 2004: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2004.
9. Eklund, J., Brusilovsky, P.: InterBook: An Adaptive Tutoring System UniServe Science News Vol. 12. March 1999. 8--13 (1999)
10. Koch, N., Wirsing, M., Software Engineering for Adaptive Hypermedia Applications?, Third Workshop on Adaptive Hypertext and Hypermedia at the 8th International Conference on User Modeling, Sonthofen, Germany, July 13-17 (2001)
11. Moore, A., Stewart, C.D., Zakaria, M.R., Brailsford, T.J.: WHURLE - an adaptive remote learning framework, International conference on Engineering Education (ICEE-2003), July pp. 22-26, Valencia, Spain (2003)
12. Paramythis, A.: Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet? Proceedings of the Adaptive Collaboration Support Workshop, held in conjunction with the Adaptive Hypermedia 2008 conference, July 29th 2008, Hannover, Germany, pp. 17-29 (2008).
13. Specht, M. Burgos D.: Modeling Adaptive Educational Methods with IMS Learning Design. Journal of Interactive Media in Education (2007)
14. Stash, N., Cristea, A.I., De Bra, P.: Adaptation languages as vehicles of explicit intelligence in Adaptive Hypermedia, In International Journal on Continuing Engineering Education and Life-Long Learning, vol. 17, nr 4/5, pp. 319-336, InderScience, 2007.
15. Stash, N., Cristea, A. and De Bra, P.: Adaptation to Learning Styles in E-Learning: Approach Evaluation, Proceedings of E-Learn 2006 Conference, Honolulu, Hawaii (2006)

Annex

Scenario 5

Scenario summary:

The group of Lee, Jane and Mustafa has missed a deadline and the system attempts to mediate the collaboration process.

Short explanation:

A LAG file consists of two main parts:

– the **initialization**, which describes what the user sees the first time when s/he enters the adaptive system (and contains all the variable declarations and initializations), and
– the **implementation**, which describes the adaptive interaction between user and adaptive system, and which is, importantly, expected to run in a continuous loop. For more information about the LAG language is available online[24].

```
initialization (

// the deadline for the group is set 31st of the month:
UM.group.submission-deadline = 31
// deadline has not been extended yet:
UM.group.submission-deadline-extended = 0
// group size: how many members
UM.group.size = 0
// group participation: how great is the overall group
// participation
UM.group.participation = 0
FOREACH UM.group.User(
  UM.group.User.participation = 0
  UM.group.size += 1 )
)


implementation (

FOREACH UM.group.User (
  IF enough (
    UM.group.User.GM.access, UM.group.User.GM.chat,
    UM.group.User.GM.Concept.chat, 1)
  THEN UM.group.User.participation +=1
)
```

[24] http://www.dcs.warwick.ac.uk/~acristea/mot.html

```
  IF (not(UM.group.submission)) THEN
  ( IF (UM.group.submission-deadline-extended < 1)
    THEN (
      UM.group.n-participative = 0
      UM.group.n-nonparticipative = 0
      FOREACH UM.group.User (
        // If the user's participation was below the
        // group average:
        IF (UM.group.User.participation <
                 UM.group.User.participation /
                 UM.group.size) THEN
        ( UM.group.User.participative =
                 participative;
          UM.group.n-participative +=1 )
        ELSE
        ( UM.group.User.participative =
                 non-participative )
        PM.group.User.participative.show = true
        PM.group.User.show = true
      )
      // If more users are participative than
      // non-participative :
      IF (UM.group.n-participative >
                 UM.group.n-nonparticipative)
      THEN (
        // then extend the deadline:
        UM.group.submission-deadline-extended +=1
        // and notify the teacher:
        UM.group.teacher.notify =
                 'UM.group.submission-deadline + 7'
        // if the teacher approves:
        IF (UM.group.teacher.notify.approve )
        // then the new submission deadline is
        // stored:
        THEN (
           UM.group.submission-deadline =
                 UM.group.submission-deadline + 7
        )
      )
    )
  )


  FORALL UM.group.User.participative = non-participative
    // add monitoring for non-participative members:
    (UM.group.User.monitor = true)

)
```