Adaptive Learning Designs: Bringing together IMS Learning Design and Adaptive Educational Hypermedia Systems

Adriana J. Berlanga
(Open University of the Netherlands, The Netherlands
adriana.berlanga@ou.nl)

Francisco J. García
(University of Salamanca, Spain
fgarcia@usal.es)

Abstract: The purpose of Adaptive Educational Hypermedia Systems is to define methods and techniques to deliver personalized learning support. Even their benefits and potentialities for educational contexts are clear, their use in real learning contexts is still rare. Reasons for this include the complexity of their development, their use of exclusive methods for defining adaptivity and educational elements, and their lack of interoperability amongst courses and applications. A possible alternative to cope with these issues is using as a common notational method the IMS Learning Design specification. This paper attempts to bring AEHS and IMS-LD closer to each other in order to define adaptive behaviour. To this end, we explain how IMS-LD could be used to define personalization properties and adaptive techniques. Based on that, we propose a component called Adaptive Learning Designs, and an authoring tool to create these components. Furthermore, the paper discusses the benefits and limitations of IMS-LD to define adaptive behaviour, and ends suggesting additional research lines.

Keywords: Learning Design; IMS-LD; Adaptive Hypermedia Systems; Adaptive Rules; Authoring Tools
Categories: H.5.2, H.5.4, J.7

1 Introduction

Adaptive hypermedia systems consider users’ characteristics to provide them with hypertext contents and links that they might find useful or adequate to their preference, needs, objectives or knowledge. Certainly, this approach has many benefits for learning environments: providing learners with personalized support based on their characteristics might help them to comprehend the learning material. Educational applications are, indeed, the most popular area of adaptive hypermedia systems. Adaptive Educational Hypermedia Systems (AEHS) try to enhance learners’ understanding of the learning material providing them with paths and contents tailored to their characteristics and preferences. Nevertheless, the use of these systems in real learning environments is rare. Reasons for this include the lack of credible evidence of their benefits [Murray, 04], the pitfalls in their evaluation [Weibelzahl, 05], the absence of general purpose tools and the proliferation of stand-alone systems [Stash, 07], the high cost of production and maintenance of these systems, their complex and time consuming authoring processes, and the lack of mechanisms for
sharing and reusing components, such as adaptive strategies or learning resources, amongst courses and systems.

An additional remark is the tendency to design AEHS from the computer science perspective. This tendency, which is still apparent today, does not involve instructional designers in the description of requirements and design of AEHS. In most cases the result has been content-orientated approaches that do not conceptualize learning as a process where learners interact not only with resources but also with learning activities, teachers and classmates.

An alternative to lessen the AEHS problems is using a common notational method such as the educational modelling language described in the IMS Learning Design specification [IMS-LD, 03]. This paper attempts to bring AEHS and IMS-LD closer to each other in order to define adaptive behaviour. To this end, we propose a component called Adaptive Learning Design (ALD), whose objective is twofold: to permit the definition of the characteristics of the learning strategy –like its learning objectives, prerequisites, learning activities, method of instruction and adaptive behaviour-, and to support the reusability and exchangeability of the defined components amongst learning designs and tools compliant with IMS-LD. The definition of an ALD component is founded on the characterization, techniques, and elements that a number of AEHS take into consideration for performing adaptivity, as well as on the IMS-LD attributes for modelling learning strategies with adaptive capabilities.

The rest of this paper is structured as follows: first, it introduces the concept of learning design and explains the most relevant characteristics of IMS-LD for defining adaptive behaviour. Subsequently, it expands on how IMS-LD can be used to model features for performing adaptivity as well as adaptive hypermedia techniques. Then, it describes how the ALD component integrates IMS-LD and, finally, the paper discusses the limitations of IMS-LD to model adaptive behaviour, and exposes conclusions.

2 Learning design

Learning design is the application of learning design knowledge when developing courses or lessons. Learning design knowledge can be identified by means of instructional design theories, examples of best practices, and patterns and experiences [Koper, 05].

Instructional design theories attempt to define prescriptive design principles that depict the best way for supporting people to learn and develop cognitive, emotional and physical aspects [Reigeluth, 99]. These theories have been conceptualized from different points of view. Traditional models (e.g., [Gagné, 79], [Merrill, 94]), which are suitable for well structured domains, are considered as objectivist and prescriptive approaches, focused on content and learning outcomes. Conversely, constructivism approaches (e.g., [Spiro, 90], [Van Merriënboer, 97]), which are appropriate for ill-structured domains, seek to facilitate learners in constructing knowledge through learning activities. Alternatively, novel approaches move away from prescribing instructional strategies, and propose models to solve problems in the description of learning designs by considering best practices and experiences from experts (http://www.pedagogicalpatterns.org).
From the computing perspective, the challenge is twofold. On one hand, it is necessary to design authoring tools that support the modelling of learning designs without prescribing any particular approach and, on the other, to reduce time and development costs, it is desirable that the resources and elements used in those learning designs could be reused and exchanged between lessons and platforms.

2.1 IMS Learning Design (IMS-LD)

The IMS Consortium identifies and defines an e-learning set of specifications for interoperable learning technology. This framework includes specifications that aim at describing, for instance, learning resources [IMS-LOM, 04], content packages [IMS-CP, 04], learner information [IMS-LIP, 03], questions and tests [IMS-QTI, 05], and learning designs [IMS-LD, 03].

The objective of IMS-LD is to provide a common notation that can be used to describe any learning process in a formal way. This notation, at the same time, aims at meeting requirements for completeness, pedagogical flexibility, personalization, formalization, reproducibility, interoperability, compatibility and reusability.

In short, IMS-LD is a language for modelling units of learning (i.e., the smallest unit that satisfies one or more learning objectives). “A learning design, modelled using the language described in the IMS-LD specification, captures who does what, when and using which materials and services in order to achieve particular learning objectives” [Tattersall, 04, pp. 3].

A learning design is mainly based on the learning and support activities learners and staff need to perform to reach a learning objective. Activities, which can be grouped into activity structures, must be included in a method that defines which activities should be performed by which roles. The definition of these methods follows a theatre-play-like structure: within a play, people (learners, staff) perform several roles (called role-parts) in certain order and sequence (called acts).

Figure 1 shows the hierarchical order of the IMS-LD main elements (the asterisk represents that an element may occur more than once): learning objectives, prerequisites, components (properties, roles, activities, and environments), and a method of instruction that consists of a play (i.e., the way the method will be executed) and conditions.

Figure 2 shows a basic example of an IMS-LD unit of learning notation. The example contains two learning activities (LA-Intro and LA-History) that are grouped in a sequence (AS-Intro), which specifies that both activities have to be completed sequentially (number-to-select="2" structure-type="sequence"). In the method section, this sequence is included in a play (Ply-EducTec) that has to be performed by the role learner (R-learner) in an act (Act-Intro).
IMS-LD has three levels of implementation and compliance that gather together. Level A contains vocabulary for supporting pedagogical diversity, Level B contains attributes and conditions that permit the definition of personalized sequences, and Level C adds notifications. Figure 3 shows the information model of IMS-LD Level A and B. Grey marked classes belong to Level B.

In Level B the definition of personalization characteristics and more elaborated sequences and learning interactions is possible. This level includes elements as properties to store information about users, global elements to set and view the information of the properties, and conditions to manage and change the value of the properties.

Properties can be local or global. In the former case they can have the same value for all the users in a run (local properties), contain information about each role in a unit of learning (role properties), or contain information about each person (personal properties); whereas global properties can contain portfolio information (global personal property), or a single value for all users in all units of learning (global properties).

For personalizing the learning flow, properties are used in conjunction with conditions. Conditions have the following format:

\[
\text{IF [expression] THEN [action]}
\]
Thereby, an action will be performed if an expression is true. Actions can be defined, for instance, to show or hide an element (e.g., a learning activity, a play, etc.), change the value of a property, or send a notification. Expressions can include logical operations (e.g., and, or, greater than, less than, etc.), calculations (e.g., sum, multiply, etc.) or references to a particular element (e.g., role, act, learning activity, etc.).

Finally, there are attributes in elements of Level B that are specific for adapting learning activities, activity sequences, acts and plays, such as <when-property-value-is-set> and <when-condition-true>**. Likewise the element <role-parts>, which describes the activities to be performed by a role in an act, could be used to group learners in stereotypes. In this way, every role-part covers a set of learning activities directed to a specific group of learners.

![Diagram of learning design](image)

**Figure 3. IMS-LD information model Level B [IMS-LD, 03]**

3 AEHS and IMS-LD

Wu et al. [Wu, 00] define two levels to control the adaptation in adaptive hypermedia systems: the author level and the system level. In the former, a person (expert, teacher, designer, etc.) defines and specifies the adaptation rules that will govern the system. In the latter, all the rules defined on the author level are executed by an adaptation engine. Given the elements and characteristics of IMS-LD, we hold that this specification can be used in the author level to model adaptivity behaviour. This level includes the description of the features that are taken into account for
performing adaptivity, such as learner preferences and knowledge, as well as their inclusion into rules for implementing adaptivity techniques [Brusilovsky, 96a].

To support this argument, this section first presents the features for performing adaptivity used in AEHS and explains how they can be modelled in IMS-LD. Then, it describes how these features can be integrated in the definition of adaptive hypermedia techniques using IMS-LD.

3.1 Features for performing adaptivity using IMS-LD

Features that adaptive hypermedia systems take into account for adaptation (in general, not only for AEHS) include [Brusilovsky, 96a] [Kobsa, 01]: user knowledge, user objectives, user experience in other fields of study (profession, experience, etc.), user preferences, demographic characteristics, information about the interaction of the user with the system, and technical information that affects the functionality of the system (e.g., software, hardware, bandwidth, etc.).

In AEHS, learner knowledge is, obviously, the most useful feature for adaptivity. Table 1 summarizes some well-known examples of AEHS. It includes the domain of the AEHS, the features they consider for adaptivity and the adaptive techniques they use (see next subsection).

As explained before, information about users can be modelled using the <properties> element of IMS-LD, which definition is very flexible; it can represent any type of data. This element is useful to include the features AEHS normally use for adaptivity. Consequently, the features for performing adaptivity presented in Table 1 can be modelled using the <properties> element of IMS-LD.

<table>
<thead>
<tr>
<th>AEHS</th>
<th>Domain</th>
<th>Features for performing adaptivity</th>
<th>Adaptive techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterBook</td>
<td>Domain independent</td>
<td>Prerequisites</td>
<td>Link annotation</td>
</tr>
<tr>
<td>[Brusilovsky,96b]</td>
<td>Authoring adaptive contents</td>
<td>Learner knowledge</td>
<td>Direct guidance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning state</td>
<td></td>
</tr>
<tr>
<td>AHA!</td>
<td>Domain independent</td>
<td>Attributes associated with concepts</td>
<td>Inserting/Removing fragments</td>
</tr>
<tr>
<td>[De Bra, 01]</td>
<td>Authoring AEHS</td>
<td></td>
<td>Link annotation</td>
</tr>
</tbody>
</table>
| KBS-Hyperbook  
[Henze, 99] | Domain independent  
Authoring Educational Hypermedia Books | Prerequisites  
Learner knowledge  
Learner preferences | Link annotation  
Direct guidance  
Routes/Project adaptation |
|---|---|---|---|
| TANGOW  
[Carro, 99] | Domain independent  
Authoring of adaptive courses | Learner stereotype  
Learner preferences on learning strategy  
Learning styles (Felder & Silverman approach [Felder, 88]) | Inserting/Removing fragments  
Direct guidance |
| INSPIRE  
[Papanikolaou, 03] | Computer architecture | Learner knowledge  
Learning styles (Honey & Mumford approach [Honey, 92]) | Direct guidance  
Inserting/Removing fragments  
Link annotation |
| ALE  
[Specht, 02] | Learning Management System | Learner knowledge  
Learner preferences  
Learning styles (Felder & Silverman approach [Felder, 88]) | Link annotation |

The features these AEHS use for performing adaptivity can be seen as important features. They are, thus, learning conditions that can be placed into one of the following categories [Koper, 05]: learning objectives, learner characteristics, setting characteristics, and media characteristics. Table 2 shows an ID for each one of them, the options they might include, and an example. For instance, the category “learner characteristics” (LC) includes features such as learner knowledge and learning style of the learner – features that are used, for example, in AEHS such as ALE, INSPIRE or TANGOW. Moreover, the table includes categories such as “learner demographics”, “media characteristics”, and “setting characteristics” that, as mentioned earlier, are reported by [Brusilovsky, 96a] [Kobsa, 01] as features that AEHS consider for performing adaptivity.
Table 2. Classification of the features for performing adaptivity using IMS-LD

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Features</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>[LOB]</td>
<td>Learning Objectives</td>
<td></td>
<td>Knowledge, skills, attitude, competence</td>
</tr>
<tr>
<td>[LD]</td>
<td>Learner Demographics</td>
<td>Age</td>
<td>Spanish, English, Dutch</td>
</tr>
<tr>
<td>[LC]</td>
<td>Learner Characteristics</td>
<td>Learner knowledge</td>
<td>Sensitive/intuitive, visual/verbal, sequential/global [Felder, 88]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning style</td>
<td>[Felder, 88]</td>
</tr>
<tr>
<td>[LP]</td>
<td>Learner Preferences</td>
<td>Level of detail</td>
<td>Basic, medium, high</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning style</td>
<td>Activist, Pragmatist, Reflector, Theorist [Honey, 92]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Level of interactivity type</td>
<td>Linear/interactive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning strategy</td>
<td>Theory/practice; learning by example/learning by doing</td>
</tr>
<tr>
<td>[MC]</td>
<td>Media Characteristics</td>
<td>Technical characteristics</td>
<td>OS, bandwidth, hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Communication</td>
<td>Synchronous/asynchronous</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media type</td>
<td>Video, text, graphic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interactivity type</td>
<td>Linear/interactive</td>
</tr>
<tr>
<td>[SC]</td>
<td>Setting Characteristics</td>
<td>Work type</td>
<td>Individual/group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Work place</td>
<td>Home/school</td>
</tr>
</tbody>
</table>

3.2 Adaptive hypermedia techniques and IMS-LD

Brusilovsky [Brusilovsky, 96a] [Brusilovsky, 01] distinguishes between methods and techniques for adaptive hypermedia. A method describes, from the conceptual point of view, a notion of adaptivity, while a technique is an implementation of that notion. It is possible, then, to implement the same method using different techniques, and vice versa, using different techniques to implement the same method.

Drawing from these concepts, Brusilovsky classifies adaptive hypermedia techniques in two groups: adaptive presentation and adaptive navigation support. Adaptive presentation techniques manipulate the content, whereas adaptive navigation support techniques manipulate the links. Figure 4 shows the taxonomy of these techniques.

The objective of adaptive presentation methods is to provide users with additional, comparative or different explanations of the content, or to present the most significant text fragments by sorting them. Most of the research in this area is focused on adaptive text presentation and, particularly, on canned text adaptation [De Bra, 05]: text fragments are inserted, removed, altered, embedded (called stretchtext), sorted or dimmed in order to adapt the content that is shown to the user. Additionally, adaptive multimedia presentation and adaptation of modality techniques deliver the content selecting, respectively, (canned or automatic) multimedia fragments, or different types of media (text, video, audio, etc.).
Adaptive Presentation

- Adaptive multimedia presentation
- Adaptive text presentation
  - Natural language adaptation
  - Canned text adaptation
    - Inserting/removing fragments
    - Altering fragments
    - Stretchtext
    - Sorting fragments
    - Dimming fragments
- Adaptation of modality

Adaptive Navigation Support

- Direct guidance
- Adaptive link sorting
- Adaptive link hiding
  - Hiding
  - Disabling
  - Removal
- Adaptive link annotation
- Adaptive link generation
- Map adaptation

Figure 4. Adaptive Hypermedia Technologies Taxonomy [Brusilovsky, 01]

The objective of adaptive navigation support methods is to help users to find their way in the hyperspace by providing guidance and orientation support. Guidance is related to suggesting the next step to take, whereas orientation support is related to presenting an overview of the link structure of the hyperspace. Therefore, techniques in this area manipulate the structure of the hypertext links for presenting relevant and appropriate information. Links, thus, can be sorted, hidden, annotated (by using visual clues), or automatically generated. Moreover, links can be manipulated to indicate the best next link to follow (direct guidance technique), or graphically organized to present an overall picture of the link structure (map adaptation).

As mentioned before, Table 1 shows adaptive techniques used by some well-known examples of AEHS. In adaptive presentation the most popular technique is inserting/removing fragments, whereas in adaptive navigation support the most popular ones are adaptive link annotation [De Bra, 05] and direct guidance techniques.

Defining these techniques in IMS-LD could be done in different levels of the learning design. Table 3 shows which IMS-LD elements can be used to model adaptive techniques and which features these techniques can take into account for performing adaptivity. Note that the elements in the last column of Table 3 are equivalent to those defined in Table 2. As adaptive techniques manipulate pages and content, whereas IMS-LD manipulates elements such as plays, acts, role-parts, activity sequences and learning activities, Table 3 also includes a description of the adaptive technique in terms of IMS-LD.

For instance, an adaptive technique that shows or hides links can be defined at the level of a play, act, role-part, activity sequence and learning activities; and it can consider features such as prerequisites, learner characteristics, learner preferences, learning demographics and media characteristics.

Table 3: Adaptive techniques and IMS-LD
The next section explains a component that helped us to test how the features for performing adaptivity and the adaptive techniques can be defined in IMS-LD. To this end, we characterized and developed an authoring tool for creating the so-called Adaptive Learning Designs.

4 Adaptive Learning Designs (ALD)

An ALD is characterized as a learning design that considers learner characteristics (i.e., knowledge, learning styles, etc.) to deliver a personalized learning flow [Berlanga, 05a]. In order to permit their reutilization, ALD are semantically structured according to IMS-LD. Therefore, their intention is to combine the personalization and reusability characteristics of IMS-LD.

Consequently, an ALD is defined by elements such as learning objectives, prerequisites, components (that include roles, learning activities, activity sequences and personalization properties), and a method of instruction (that includes adaptive rules, plays and acts). The definition of these elements, and their sub-elements, follows a Lego metaphor [Berlanga, 05b] where each one of them is defined and stored as a separate object and, as a result, elements, sub-elements and ALD (as a whole component) can be reused in different learning contexts, lessons and courses, or amongst different AEHS, applications and tools. That is to say, a learning activity could be incorporated in different activity sequences, and defining a new method of instruction does not require defining again learning activities or objectives that have been created for other methods already included in existing ALD.

Based on this notion of a Lego metaphor, an ALD authoring tool was developed. This tool, called HyCo-LD, was built up enhancing the functionality of the Hypermedia Composer (HyCo) [García, 06], which is an authoring tool for
hypermedia books structured as chapters and subchapters. These chapters and subchapters, as well as the resources they include can be linked to ALD elements such as learning activities, learning objectives, prerequisites, or feedbacks. Therefore, using the same tool, users can create learning resources and integrate them in ALD. HyCo also includes a metadata editor compliant with IMS LOM. This allows users to export resources and chapters as learning objects.

Figure 5: HyCo - main screen

Figure 5 shows the main screen of HyCo. The pane on the left visualises chapters and subchapters of the book. In the editing section (the central area) users can type the content of the book, and include multimedia resources and bibliographical references. From the main screen users can select the menu instructional design (“diseño instructivo” in the interface), which shows a submenu (see Figure 6) from where users can define or edit:

- Learning objectives (“objetivos de aprendizaje”)
- Prerequisites (“prerrequisitos”)
- Components (“componentes”), which include the definition of roles, learning activities, activity sequences, and personalization properties
- Methods (“flujo de aprendizaje”), which include the definition of adaptive rules, acts and plays
- Adaptive Learning Designs (“diseño de aprendizaje”)
Content packages ("empaquetado") compliant with IMS-CP

Moreover, using the option "player" of this sub-menu, users can execute an ALD using the Coppercore player [Martens, 04] that is integrated into HyCo-LD as a third-party IMS-LD player.

Figure 6: HyCo-LD - instructional design menu

Consistent with the Lego metaphor, in HyCo-LD every element is defined independently from each other and stored in a repository. This facilitates their handling and incorporation into other elements. The definition of each element is presented using a tab structure; it groups the sets of attributes needed to describe the element. As an example of the HyCo-LD interface, Figure 7 shows the interface for defining learning activities.

Figure 7: HyCo-LD - definition of learning activities
To define adaptive behaviour, authors should create personalization properties that, afterwards, can be included in adaptive rules. Personalization properties, which are based on the set of features for performing adaptivity explained before (see Table 3), contain user information, whereas adaptive rules are prescriptions defined by authors that will be taken into account to adjust the learning method. There are two ways of defining these rules. The first one, which is based on the description of adaptive hypermedia techniques explained before, is intended for non-expert users of the specification. The second one is intended for users with deep knowledge of IMS-LD.

4.1 Creating personalization properties

As in IMS-LD, in HyCo-LD users can create different types of properties: local, personal or role. Local properties have the same value for all users, personal properties have different values for every user, and role properties have the same value for every role.

As it has been said before, in IMS-LD the definition of properties is very flexible. To define them, authors have to indicate the title, data type (integer, character, Boolean, etc.) and the initial value of the property. It is also possible to include restrictions, such as the minimum and maximal permitted values. Figure 8 shows the interface for defining role properties.

![Figure 8: HyCo-LD - definition of roles](image)

4.2 Authoring adaptive hypermedia techniques

HyCo-LD has a wizard for supporting users in the definition of basic adaptive techniques. These techniques are managed as separate objects and can be included into different learning methods. The wizard follows the definition of adaptive
techniques presented in Table 3, and the features for performing adaptivity presented in Table 2.

Figure 9 shows the interface for defining adaptive hypermedia techniques. In the first tab “Attributes, type and level” (“Atributos, tipo y nivel” in the interface) the type of the technique (e.g., direct guidance) and its name and level (e.g., play, activity sequence or learning activity) should be indicated. Then, in the second tab, authors should select the feature that has to be taken into account for performing the adaptivity behaviour; the available options are those defined in Table 2. Therefore, when the author selects a category (e.g., learner preferences; “Basado en” in the interface) then, the list box displays the features that the selected category contains (e.g., level of detail, learning style, etc.; “opciones” in the interface). Hereafter, the author should indicate the operation (“operación”), data type (“datos”) and value (“valor”) of the selected element, as well as the property (“propiedad”) from which the value for performing the adaptive technique should be taken.

![Figure 9: HyCo-LD - wizard to create adaptive hypermedia techniques](image)

### 4.3 Authoring advanced adaptive rules

The definition of adaptive rules is guided by an expression-builder tool that gives authors more flexibility for deciding the characteristics and variables that should be defined. This tool uses a formalism [Berlanga, 06] based on the <conditions> element of IMS-LD that allows authors to identify the elements of the learning design structure, as well as both the characteristics of the learner and of the learning activities authors want to include in the adaptive rule.

Figure 10 shows the interface of the expression-builder tool. It contains different boxes that include the operations, properties and elements that can be selected to create an adaptive rule: operators (“operador” in the interface), properties (“property”), roles (“roles”), learning activities (“actividades de aprendizaje”), activity structures (“estructuras de actividades”) and plays (“ejecuciones”). From
these boxes authors select the element they want to include in the adaptive rule. The tool guides them showing only the operator, property or element that can be chosen in each part of the construction of the rule.

![Figure 10: HyCo-LD - expression-builder tool for adaptive rules](image)

4.4 HyCo-LD evaluation, reusability and interoperability

We have conducted an evaluation of the ALD component [Berlanga, 06]. Our objective was to design an ALD for a real course, create it in HyCo-LD, test its adaptive behaviour, and get learner’s feedback about it.

We use as an experimental setting the postgraduate course “Introduction to Learning Technology”, which is part of the PhD programme “Learning Processes in Virtual Contexts” (Procesos de formación en espacios virtuales) of the University of Salamanca. Normally, this course has a learner population with background in different disciplines. In 2006 this programme had 13 students coming from areas such as Computer Science, Humanities and Educational Sciences.

The purpose of this postgraduate course is to acquaint learners with the basic concepts of the Learning Technology field. The course is structured in four lessons that cover one of the following topics: learning objects, metadata, repositories, and learning technology standards and specifications. Lessons follow the same learning flow: learners have to read first the content of the lesson (i.e., explanations) and, then, perform learning activities such as describing learning objects, defining metadata, looking for learning objects in repositories, and discussing the benefits and limitations of learning technologies standards and specifications. Specific learning activities are designed for learners with computer science background.
The prerequisite of the course is that learners should have knowledge about the creation of educational contents for web based environments. If they do not have previous experience on this topic, then an additional lesson is provided. Likewise, if learners want to go deeper on one of the topics, an additional activity is provided.

The implementation of this course as ALD comprised two sequential tasks. The first was modelling conceptually the course as ALD, understanding how the course could be designed using acts, properties, learning activities and so on. The second task was implementing the ALD using HyCo and HyCo-LD.

Conceptually, this course was modelled as ALD using a single method consisting of one play, which included four sequenced acts (one for each lesson). Each act comprised activity sequences that enclosed the content of the lesson and the learning activities designed for each lesson. Additional acts were included for the prerequisite lesson (creation of educational contents) and for the additional activities. Two personalization properties were envisaged: initial knowledge and profile. The former should specify if the learner has initial knowledge, so he/she can skip the prerequisite lesson, and the latter should specify if the learner’s profile is “computer science”, so different learning activities could be displayed. These two properties are needed to define adaptive techniques for initial knowledge evaluation, and for showing specific learning activities according to the learner’s profile.

In the second task, HyCo was used as an authoring tool to create the content of the ALD, which was structured as a hypermedia book. Chapters were defined for each lesson (i.e., act) and, within each lesson, subchapters were created for writing down the content (e.g., textual descriptions, hypermedia texts) of learning objectives, prerequisites, explanations, and learning activities.

Then, using HyCo-LD, the elements and structure of the ALD were created. The first step was to define learning activities for the content and for learning activities. Their textual description was linked to the corresponding subchapter created earlier in the hypermedia book. This connection permitted, when needed, to modify only the definition of the learning activity without having to change its content and vice versa. Then, the learning activities were nested in activity sequences, and properties for the initial knowledge and profile of the learner were created.

Furthermore, three adaptive techniques were included to (1) show the prerequisite lesson, (2) show particular learning activities if the learner’s profile was “computer science”, and (3) show a complementary learning activity if the learner wanted to go deeper on the topic.

To show the prerequisite lesson, we used the HyCo-LD wizard for defining a direct guidance technique that considered the initial knowledge of the learner to deliver the prerequisite lesson. Similarly, other direct guidance technique was defined to present to those learners, whose profile was computer science, a learning activity that required XML knowledge.

In addition, the expression builder tool was used to create an adaptive rule that showed a complementary learning activity if the learner wanted to go deeper on the topic.

After these definitions were ready, acts were created. They were structured and sequenced as follows: first they presented the learning objectives, prerequisites and the content of the lesson and, when the student selected the option “completed”, the
learning activities were displayed. If the learning activities were done, then the next act was presented and the next topic displayed.

Once acts were defined, they were attached into one play. The play was incorporated into a new method that, together with the description of general the learning objectives and prerequisites of the course, formed a new ALD. Finally, a content package of the ALD was generated.

The setup of the experiment was as follows: learners were in the classroom using their laptops to follow the course. While doing the learning activities learners need to write down the outcomes of the learning activities. To sum up the topics, to reflect about learning experience and to ask questions, at the end of the session a round table was organized.

In the experiment 13 learners participated. They were asked to follow the course playing the ALD content package in the Coppercore player integrated in HyCo-LD, which was installed in their machines. In average it took them three hours to complete the course.

When they finished, they were asked to fill in a feedback questionnaire, which included questions regarding the learners’ profile, the suitability of the materials and the learning activities, and their preference to control the learning flow. Out of the 13 learners, 8 answered the questionnaire. The answers analysis established that most of the learners (75%) found the ALD adequate to their initial knowledge and profile. Likewise, most of them judged that the learning activities were helpful to understand the topic. Nevertheless, also most of them (63%) expressed their desire to explore by themselves the existing material and learning activities and, then, decide which learning activities to follow.

Regarding the reusability of the components created in HyCo-LD, we have conducted several use cases. Particularly, using the learning activities created for the experiment mentioned above, we created a new ALD intended for advanced learners with computer background. Almost the same learning activities, activity structures and acts where used, but a new adaptation rule and a more technical learning activity (i.e., programming in XML) were included. This verification showed existing learning activities could be incorporated into new ALD.

Concerning the interoperability of ALD, since HyCo-LD integrates the IMS-LD compliant player Coppercore for running ALD, it is safe to assume that ALD are interoperable components. Besides, other IMS-LD authoring tools have been used to check the interoperability of ALD. Particularly, the well-known IMS-LD editor Reload [Reload, 06] has been used to open and edit existing ALD. Although, the adaptivity rules defined in HyCo-LD cannot be defined in the same manner in Reload, they can be edited and modified.

5 Discussion

IMS-LD has the potential to become a common notational method for developing adaptivity features [Towle, 05] [Burgos, 06]. Specifically, and as it has been argued above, the characteristics of IMS-LD do allow specification of adaptive behaviour for AEHS. In the work we carried out, we analysed the extent with which IMS-LD can be used to model features for performing adaptivity and adaptive hypermedia techniques of AEHS.
The features for performing adaptivity can be easily modelled using the IMS-LD element <properties>. As this element is defined in the specification following a meta-definition structure, any type of feature can be described. As explained before, features for performing adaptivity in AEHS include, for instance, learner knowledge, preferences or learning styles. Furthermore, properties in IMS-LD can be viewed, set, modified and included in adaptive conditions or rules.

Combining IMS-LD elements such as <properties> and <conditions> makes possible to model, to a certain extent, adaptive hypermedia techniques. The element <conditions> has sub-elements such as <show> and <hide> that are ideal to define adaptive navigation support techniques like direct guidance, adaptive link hiding, and adaptive link annotation. However, IMS-LD does not model the learning content; therefore the manipulation of the text is not covered and, as a consequence, adaptive presentation techniques such as stretchtext or dimming fragments cannot be modelled. However, adaptation techniques such as inserting/removing fragments or adaptation of modality can be represented in IMS-LD if fragments or types of media are included in different learning activities that, accordingly to certain conditions, are showed or hidden.

Although the use of IMS-LD can be somehow restrictive, it may bring the following benefits:

- Incorporation of existing annotations (i.e., ontology) to describe learning knowledge and pedagogical strategies into AEHS;
- Assure the separation between pedagogical strategies, learning flows, adaptive logics, and content;
- Feasible reutilization and interoperation of resources, learning elements and learning designs amongst courses and AEHS;
- Quick AEHS prototyping and testing. For instance, a cycle for prototyping a particular adaptive behaviour might consist of creating a prototypical ALD, testing it using IMS-LD tools and verify if the results are as expected, making the necessary changes to the ALD, testing it again, etc. In this way specific components that are needed can be tested and validated without having to make major changes in a course or AEHS.

There are, however, three issues that should be pointed out: the way a unit of learning has to be delivered, the definition of learning objectives, prerequisites and feedback, and the definition of roles.

For delivering a unit of learning, IMS-LD requires to embed the learning design in a manifest compliant with IMS-CP [IMS-CP, 04]. This makes it impossible to change the learning strategies once the learners are interacting with them. Thereby, all the alternatives the learner could follow have to be defined at design time. At run time, the learning flow is controlled only by the system. This makes the authoring process extremely time-consuming and takes the control of learning away from the learner.

The second issue is the definition of learning objectives, prerequisites and feedback. Rather than being specified on the learning level, in IMS-LD the definition of these elements is done on a descriptive level. These elements participate in the learning flow only as information resources described by attached learning objects or URLs; they are not described as triggered elements that, once they have been
completed, viewed or assessed, they could activate additional actions like presenting additional learning activities, activating new actions, or marking an activity as completed. Nonetheless, it is possible to use an artificial alternative and create learning sequences that represent prerequisites. For instance, if the learning activity A is a prerequisite of the learning activity B, an activity structure that contains A and B and sequences them (A then B), has to be defined. However, this is not the optimal solution because the prerequisite of the learning activity B will not be semantically stated and, then, it cannot be included as an element to define adaptive rules nor retrieved automatically.

The last issue is the definition of roles. In IMS-LD the definition of roles is not based on properties. This impedes personalizing the learning flow for roles that have certain characteristics (e.g., knowledge, preferences) or dynamically change the role of the learner.

Finally, it should be stressed that, even IMS-LD is conceptualized as both as an education modelling language and as an interoperability specification, its complexity had provoked that, in practice, it usage and dissemination is limited to the expert users sphere. For non-expert users IMS-LD should not be presented as a “programming language” for creating units of learning or adaptive hypermedia techniques. In this case, this specification should play a “back-stage” role. Definitively, evaluations of the designers’ perception about the creation of ALD and the tools proposed in this paper, as well as current research on visual authoring tools for non-expert users [Tattersall, 07], might bring some insights regarding this matter.

6 Conclusions

It is clear that pushing forward the benefits of AEHS for a wide range of applications and systems requires using a common notational method. As it has been explained in this paper, IMS-LD can be used for this purpose.

We hold, however, that from the pedagogical point of view, the way of delivering learning designs in IMS-LD and the philosophy of the AEHS need to go one step further in order to give learners freedom for building their own learning flows and, as a result, reduce the pre-design workload of tutors and teachers.

Hitherto, the philosophy behind (most of the) AEHS is providing each learner with the content he or she may need according to his/her characteristics. This personalization approach focuses on suggesting the next best resource or learning activity to follow, instead of letting learners to explore the learning material, learn from other peers, and decide by themselves which learning flow they would like to follow.

An alternative approach is showing all the existing learning activities and following learner’s footsteps. This tracing can include, for instance, tracking the learner’s behaviour with the system, his/her interaction with the ALD or unit of learning, his/her successful completed learning activities, or the learning flows peers with the same characteristics have followed. Based on this information, a new emerging ALD can be suggested. As this new ALD will not be pre-designed, it should be created at run time. An engine, thus, should look for the most appropriate learning activities to automatically generate the new ALD or IMS-LD package. The Lego
metaphor explained earlier can be useful to select, at run time and form different servers, which learning activities should be considered.

Tracking learner behaviour and his/her interaction with the learning activities can also facilitate the recommendation of the relevant learning activities to peers that have the same characteristics, as well as provide learners with information about the reasons (e.g., characteristics of the learner, characteristics of the learning activity, highly rated by peers, etc.) of the recommendation provided. Furthermore, this tracking might facilitate the evaluation and analysis of the most followed and efficient paths and, based on that, generate adaptive rules that can be used, later on, for personalizing recommendations and provide navigation support [Hummel, 07]. The features for performing adaptivity and the definition of adaptive rules explained in this paper can be used to generate the IMS-LD structure of these rules and, then, store them as separate objects that can be combined following different conditions to provide further recommendations.

We believe that tracing learner’s behaviour, making recommendation based on peers interaction, providing learners with information about the reasons of the recommendation, and evaluating learning flows, is a bottom-up approach for defining emerging ALD or learning designs that will foster learner control and diminish author’s work load. This is a challenging approach that, certainly, deserves further research.

Acknowledgements

Authors would like to express thanks to the anonymous reviewers for their valuable comments and remarks on the previous versions of this paper. Special thanks to Jorge Carabias for the development of HyCo-LD and, in general, to the other members of the research GRoup of InterAction and eLearning (GRIAL) of the University of Salamanca. Adriana Berlanga thanks the support of the Mexican Council of Science and Technology (CONACyT). This work is partially supported by the Spanish Ministry of Education and Science (ref. TSI2005-00960).

References


[Reload, 06] Reload Learning Design Editor, version 2.1.3 http://www.reload.ac.uk/ldeditor.html


