Comparative Analysis of Adaptation in Adaptive Educational Hypermedia and IMS-Learning Design

M. Hendrix[1], A. I. Cristea[1], D. Burgos[2]

1 The Department of Computer Science, The University of Warwick, Gibbet Hill Road, CV4 7AL, Coventry, UK, {maurice, acristea}@dcs.warwick.ac.uk.

2 ATOS Origin, Research & Innovation, Avda. Diagonal, 188, 08018 Barcelona Spain, daniel.burgos@atosresearch.eu

Abstract— Currently, Adaptive Educational Hypermedia (AEH) and IMS Learning Design (IMS-LD) are separate research areas, with little shared knowledge between them. Their goal, however, is the same: to design, author and implement the best possible learning experience for the learner. This paper addresses the issue of differences and similarities between AEH and IMS-LD with regard to knowledge representation and adaptation and investigates, generically, as well as for the specific case of the Layered AHS Authoring-Model and Operators (LAOS) framework, how these paradigms can benefit from each other.

Index Terms— Information theory, Knowledge acquisition, Human-centered computing, Human information processing

1 INTRODUCTION

AAdaptation and personalization are gradually 'seeping' into various aspects of our daily life via the devices that we use: the Web, the phone, the Palmtop, etc. etc. It is predictable that those who will be involved in this trend at an early stage will be the research leaders of tomorrow. Personalization was added to the Web as an afterthought to solve various legacy problems (such as the 'one-size-fits-all' web, and issues of personal information security, etc.)

Adaptive Hypermedia (AH) [10] research offers solutions to these problems, although other ad-hoc ones appear everywhere - in the commercial domain as well as in the educational one. It is therefore vital to foresee upcoming trends and to prepare for them in advance; and to apply the lessons of the past especially in new developments.

IMS Learning Design (IMS-LD) [29] is a relatively new e-learning specification, with strong pedagogical roots, and a de facto emerging standard, that models pedagogical scenarios, aiming at covering various learning situations and roles. Its forefathers are design methodology and pedagogy. IMS-LD is quickly gaining popularity and is touted as being the future of all educational web-material and interaction.

Adaptive Educational Hypermedia (AEH) [11] adds personalization and adaptation to e-learning. It caters for each learner individually, for example, to her knowledge, needs, preferences, learning styles, etc., conforming to learner-centered education. Its forefathers are hypermedia (any collection of nodes and links), adaptivity (its main strength; based primarily on user models) and finally, distance learning.
The goals of both AEH and IMS-LD are thus the same [17]: to create the best possible learning experience for the learner. In this paper, we provide a more thorough comparison of these two seemingly unrelated paradigms, with respect to the data representation used, and the adaptation they provide for, to establish how they can benefit from each other.

2 REPRESENTATION AND SEMANTICS

In this section we focus on the representation of learning material and semantics in AEH and IMS-LD. In AEH, the LAOS framework [19] is one of the most comprehensive and well used frameworks for representing adaptive content for authoring. LAOS extends previous frameworks such as AHAM [6] in the sense that it introduces an even higher separation of concerns for adaptive hypermedia authoring. Thus, we can claim that comparisons made between IMS-LD and LAOS highlight differences between the IMS-LD and the AEH view in general. Therefore, we can use LAOS for the purpose of this comparison. In the remainder of this section we shall sketch

LAOS and IMS-LD and comment on the main similarities and differences.

3 Comparison Purpose and Criteriad

3.1 LAOS

LAOS [18], [34] is a generic framework for authoring of AEH. It has been used by various researchers in Europe [1], [8], [32], [33] and beyond [28], [7] and extensions have been proposed [26]. LAOS stipulates five authoring layers, corresponding to the basic AEH high-level components: *domain model* (***DM***); *goal and constraints model* (***GM***) (also called *pedagogical model* for the educational domain); *user/learner model* (***UM***); *presentation model* (***PM***) and *adaptation model* (***AM***).

A *DM* is similar to a book or reference manual. It defines e-learning content, structure and meta-data, organizing information. The *GM* filters this large 'book', extracting elements for the learning event, and pre-orders elements; however, adaptation can still change this initial order. Importantly, the *GM* adds pedagogical labels and weights to concepts (e.g., to determine that material is for beginners or advanced learners, etc.). The *UM* stores information about the learner. The PM has information concerning a learner's environment, such as device type (handheld versus desktop), quality of service, etc. The *AM* dynamically uses the above static models, via adaptation strategies (or pedagogic strategies). The *AM* is further detailed in the LAG model [21]. The first four layers are static layers. Their representation is concept-based. I.e., domain knowledge is represented as a concept hierarchy, with several attributes; similarly, the goal model and user model are represented as concept maps. *AM* is the only dynamic layer, describing the adaptive behavior of the system. The *AM* specifies how the 'ingredients' of the other layers are combined with each other into an adaptation 'recipe' – resulting in all possible types of adaptation: content, navigation, presentation.

3.1.1. Common Adaptation Format (CAF)

CAF [22], [23] is a portable XML format, extracting common and extraneous elements related to the way adaptive content is represented in most Adaptive Hypermedia authoring systems, and is used by popular academia systems such as AHA! [5] (as input), MOT [20], [38] (as output) but also by commercial systems such as Content-e/LAOS [49] (as output). CAF is a system-independent instantiation of the *domain model* and *goal and constraints model* in LAOS. Although not an official standard yet, we hope it could well be extended to become one in the future. Below, we show the Domain Type Definition (DTD) definition of the CAF file. A CAF XML file consists of two parts, a domain model part and a goal model part.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT CAF (domainmodel?, goalmodel+)>
<!ELEMENT domainmodel (concept+)>
<!ELEMENT concept (name, attribute*,
                   concept*, relation*)>
<!ELEMENT attribute (name, contents)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT contents (#PCDATA)>
<!ELEMENT relation (name, relationlink+)>
<!ELEMENT relationlink (#PCDATA)>
<!ATTLIST relationlink
                      weight CDATA ""
                      label  CDATA ""
                      type   CDATA "">
<!ELEMENT goalmodel (lesson)>
<!ELEMENT link (#PCDATA)>
<!ATTLIST link
                      weight CDATA ""
                      label  CDATA "">
<!ELEMENT lesson (link*, lesson*)>
<!ATTLIST lesson
                      weight CDATA ""
                      label  CDATA "">
```

Fig. 1 CAF DTD

In the domain model, all domain maps used by the exported goal map are described (please note: there is a multi:multi relation between domain maps and goal maps, as one course, represented by a goal map, can be created from many book-like domain stores, represented by the domain maps). The DTD depicted in Fig. 1 shows that concepts can, in turn, have *sub-concepts*; as well as *attributes* that represent relevant domain concept metadata (e.g., a *title*, *introduction*, etc.); or relations, that represent relatedness to other concepts (forming connections other than the hierarchical connections). Attributes have a name and contents. Relations have a name and a *relation link*. The relation link connects to another concept, with which the concept is thereby associated. Such connections are established by the pedagogical experts, based on the extent of relatedness of the two concepts. In the authoring system MOT [20], for instance, this can be semi-automatically computed by the system, based on keyword frequencies in the various attributes of each domain concept.

As described previously, the *goal model* has the role of extracting (and pointing to) a subset of the exported domain map(s). As Fig. 1 shows, a goal map lesson has one root lesson; which, in turn, can have many sub-lessons (which can have sub-lessons, etc). The 'link' attribute in the DTD highlights the fact that lesson concepts are actually pointers to domain concept attributes. This link can be annotated with a weight and label, to enable adaptive systems to decide (via adaptation strategies) for which users the targeted contents are to be shown. In this way, goal maps can annotate filtered versions of the domain maps with pedagogic labels and weights. They represent therefore the first, filtered, pre-ordered version of the contents, enriched with pedagogic meta-data, and ready to be processed by the adaptation strategy.

The following example (Fig. 2) illustrates a simple DTD usage.

```
<CAF>
 <domainmodel>
<! ...>
   <concept>
    <name>Questions</name>
<! ...>
    <concept>
     <name>Question1</name>
     <attribute>
       <name>question</name>
       <contents>Where is  Hypertext 2008?
  </contents>
     </attribute>
     <attribute>
       <name>answer</name>
       <contents>Pittsburgh</contents>
```

```
      </attribute>
     </concept>
< ! ... following concepts omitted >
    </concept>
 </domainmodel>
 <goalmodel>
  <! ...>
    <lesson weight="0" label="question">
      <link weight="0" label="question">
        Questions\Question1\question  </link>
      <link weight="0" label="answer">
        Questions\Question1\answer </link>
    </lesson>
<! ... following lesson parts omitted>
</goalmodel>
</CAF>
```

Fig. 2 Adaptation content example: questions and answers

Fig. 2 shows a CAF XML file with one domain map only, called 'Questions[1]', with sub-concepts titled 'Question1', etc. All concepts have a title attribute, and the questions have each a question and an answer attribute. The goalmodel part always has one lesson (since CAF only describes a lesson, and the domain maps on which it is based). In this example, the 'lesson' (goal map) is comprised of the above mentioned questions, in the exact same order and structure as in the original domain map. Note that in the goal map, labels ('question' and 'answer', respectively) have been added for the questions and answers.

3.1.2. Layers of Adaptation Granularity (LAG)

LAG stands for 'Layers of Adaptation Granularity' [21]. The LAG model extends the adaptation layer in LAOS [18], and defines three layers of adaptive behavior specification: (1) The *adaptation assembly language*, based on the notion of adaptivity, as outlined by Brusilovsky's taxonomy, consists of IF-THEN rules[2]. Conditions and actions in these rules refer to instances of the variable values from the four static LAOS models. (2) A wrapper around these basic rules forms an *adaptation language*. (3) Finally, at the highest level, reusable *adaptation strategies* can be formed. These strategies can be applied to more than one domain model. Thus, behavior (adaptivity), data and meta-data (contents) are kept apart. This model has been a source of inspiration for many researches on authoring and engineering of adaptive and semantic web (e.g., [4], [30], [32], [41]). The LAG language [24], a language inheriting its name from the LAG model, is an instantiation of the intermediate LAG model layer, the *adaptation language layer*. It allows for the creation of system-independent descriptions of adaptation. Finally, it also enables the creation of *adaptation strategies*, as defined by the LAG model.

Continuing the previous example, in order to add adaptation to the display of questions and answers in Fig. 2, we could use a strategy as in Fig. 3 (written in the LAG adaptation language).

```
// DESCRIPTION OF THE LAG STRATEGY
// This strategy starts by only showing the user
   at first entry concepts which are not answers
   (i.e., questions, other text). Then, it proceeds
   by checking if the question was accessed and
   processed. If that is the case, the answer can
   be shown.
initialization(
  while PM.GM.Concept.type != answer
      // make only non-answers readable
      ( PM.GM.Concept.show = true )
  UM.GM.showanswers = 0//number of questions:
  UM.GM.nq = 3)
implementation(
  if enough(PM.GM.Concept.type == question
             UM.GM.Concept.access == true,
             2)//if a question was accessed:
  then ( UM.GM.showanswers += 1 )
  if enough (UM.GM.showanswers > UM.GM.nq
              PM.GM.Concept.type == answer,
               2)//show answers:
  then ( PM.GM.Concept.show = true )   )
```

Fig. 3 Adaptation strategy example: questions and answers

This example adaptation strategy initially only displays questions; however, at the next access, answers will be made available. The strategy's initialization section is performed only once, before the user starts the lesson. In this section, all concepts that are not labeled as answers are made visible (the rest is invisible by default) and the Boolean *showanswer* variable is created and initialized as false. The *implementation* section is repeated every time the users perform any action. Here, it is checked whether the user accessed a concept of the type question; if so, then answers can be shown, so *showanswer* becomes true. The result is that, after a user selects a question, its answer will show up next time he refreshes the same page. Showing an answer means allowing access to it via the menu (thus, *adaptation navigation*) and actually displaying it at the next visit of the question (thus, *adaptation of content*). If the answer is displayed on a separate page, this becomes pure adaptation navigation, whereas if a menu is not present, it will be pure adaptation of content. Thus, from a conceptual point of view, it is not necessary to separate the two adaptation types; these decisions are of a lower level and to be decided by the presentation adaptation. The strategy presented here can be extended to any number of questions, but is overly simplified here for readability (example based on 'type-based' strategy in [1]). Please note that the simple strategy above does not use the *labels* of the goal model, but the types of concepts, as defined in the domain model (Fig. 2). If we wished to use labels, we would have only to replace PM.GM.Concept.type with PM.GM.Concept.label. This illustrates how easy it is to use the same static data for different adaptation purposes. Please note that the author creating static content can be different from the one authoring dynamic content.

3.2 IMS-Leadning Design

IMS-LD [28] is an e-learning specification for pedagogical scenarios. There are various players [1], [48], [39], authoring tools [47] and engines [40] for IMS-LD. It has three implementation levels of incremental expressiveness (A, B and C). They allow the modeling of Units of Learning (UoL). They describe collaboration, adaptation, adaptability or any other pedagogical method. The division into levels is as follows. Level A provides method, activities and roles. Level B adds properties, conditions, calculations, global elements and a monitoring service. Level C rounds it off with notifications [31], [45]. UoLs are made

[1] Please note that this example is shown for its brevity and for making a parallel between CAF and IMS-LD representation. For tests only, the QTI standard can also be used very efficiently.

[2] IF-THEN rules, also popularly described in adaptive hypermedia context as ECA (event-condition-action) rules, which can be also rewritten as CA (condition-action) rules.

up of a manifest, describing the adaptation, a set of resources and optional external XHML files that can improve a few features, e.g., a specific usage of properties or services. Also, every UoL is written taking a structured metaphor that defines runs, plays, acts and activities. IMS-LD provides a full XML representation model from a lesson plan to a final UoL running online. Its lifecycle is defined in three isolated steps: modeling, publishing and playing. This implies that design-time and run-time are separated. When any modification is required, the author has to re-start designing, publish the new package and play it afterward. This is similar to the authoring & delivery process in AEH. The learning roles are sharply defined, and can be assigned to different people. End-users of IMS-LD based system can be students, tutors or authors. Hence, IMS-LD is a flexible way to represent, edit and execute a variety of pedagogical models. Furthermore, some features in levels B and C allow several types of adaptation. The appropriate use of conditions, global elements, calculations, monitoring services, properties and notifications allows personalized UoL based on flow, content, interface, evaluation, a.o. [14].

The manifest.xml file below defines a similar example to the one previously described for the CAF and LAG representation: it first hides the answers and, only after the questions they belong to have been accessed, it shows them.

```
……<properties>
    <locpers-property identifier="Question1">
      <datatype datatype="boolean"/>
      <initial-value>0</initial-value>
    </locpers-property>
  </properties> ……
<!-- If question1 is accessed show answer -->
<if>
  <is>
      <property-ref ref="Question1"/>
      <property-value>1</property-value>
  </is>
  </if>
  <then>
      <show>
        <class class="Answer1" />
      </show>
  </then> ……
```

Fig. 4 A manifest example for a UOL

3.3 Comparison of high-level features of LAOS and IMS-LD

General Content Representation:

- Comparison: From the point of view of representation and semantics, especially, for general representation, IMS-LD enforces using an XML format to describe properties; and global XHTML files that use these properties for its content. Although AEH also allows XML representations (like CAF), the clear difference is that there is currently not one standardized way of describing the content, and different systems may use different ways of representing the same content. The use on XML and the introduction of a standard makes IMS-LD more portable and allows a high level of re-use. This is desirable for any authoring system for e-learning, and especially for authoring of adaptive material, which is notoriously complex and time consuming. For such authoring, the 'write-once, use many' paradigm is vital.
- Conclusion: The field of AEH would, we believe, benefit from a clearly defined and well thought-through standard, and an unification of approaches. Such an approach is now currently sought after in the new EU FP7 GRAPPLE project.

Content extent:

- Comparison: Next, an IMS-LD manifest is significantly more verbose than the combination of CAF and LAG. A reason for this is that IMS-LD manifests need to specify much more information, which might or might not be relevant to the current application. This allows various enriched functionality, but costs in readability and space. Such enhanced functionality is illustrated by the following. After authoring in AEH, the author does not interact anymore with either the students or the content. This is mainly because the focus has been on creating automatic adaptation, and not on providing a software tool for teachers to communicate in real time with their students. In IMS-LD, provisions for such communication are, however, present. The latter therefore also has specific definitions of various roles, specified via the manifest.
- Conclusion: This shows that IMS-LD and AEH have some complementarities. In terms of educational value of the experience, both IMS-LD, with focus on people and their roles in the learning process, as well as personalization to the learner's needs, as supported by AEH, are necessary. It is conceivable that the two approaches could co-exist together in learning systems. Such issues are explored within the GRAPPLE project in workpackages on authoring and on IMS-LD applicability.

Generic conceptual point of view:

- Comparison: From a generic conceptual point of view, both AEH and IMS-LD use a multi-layered method for describing the *content* and *adaptation*. In AEH, this is done via the authoring model layers, and in IMS-LD, via the different levels. However, the levels in AEH represent a clear separation between content, grouping of content and adaptation, whereas in IMS-LD, the division is based on certain *functionality features*. Past experiments [18] show that a clear separation of the adaptation from the content (such as in LAG) is very beneficial, as it allows re-use of advanced adaptation strategies created by programmers, for people with little or no programming knowledge (for example teachers).
- Conclusion: IMS-LD should allow for a clear separation of adaptation from content, as is

supported by (some of) the adaptive hypermedia frameworks and methodology. This would allow a much more flexible approach to reuse of the authored products.

3.4 Comparison of low-level features of LAOS and IMS-LD

Static content representation:

- Comparison: Looking deeper into the data representation and semantics, we note the following. *'Static' content* (called 'domain content' in AEH) is represented by IMS-LD as XHTML documents, tagged as resources, and stored as separate files (see Fig. 4). In AEH, such content can be represented in various ways (there is no standard). CAF uses the domain concept hierarchy representation, storing all data for a lesson in one file (see Fig. 2). However, the AHA! adaptation engine [5], for instance, interprets CAF data and divides it between several XHTML documents, in a fashion not that different from the IMS-LD representation.
- Conclusion: XHTML representation of static content seems to be the best way to deal with the atomic, indivisible pieces of static information, that build the building blocks of an e-learning system, and that can be reused in various sequences and configurations to allow for personalization to the learner. In our examples we describe a CAF file with a set of questions and their answers, showing the latter only after learners have seen the questions. The way questions and answers are described is not very different in CAF or IMS-LD; yet, the way the adaptation is described differs greatly. For IMS-LD, rules are described via a hierarchical XML structure. In LAG, rules are defined via a dedicated programming language. Most AEH represent adaptation at the level of adaptation assembly language (conform LAG), and formats vary: rules can be encapsulated into concepts (as in AHA! [5]), or kept separately, in XSLT sheets (as in WHURLE [39]).
  Another great difference between LAOS AEH and IMS-LD lies in the issue of *reusability*. AEH LAG strategies can be reused to adapt different content, as long as they are written in general terms rather then for specific concepts. In IMS-LD, however, the adaptation rules are embedded within the content; this makes reuse virtually impossible. Please note that the manifest.xml files could be re-used, but this only defines a set of properties to which to adapt to, it does not specify how to adapt to these properties.

4 ADAPTATION

Both AEH and IMS-LD aim to provide a better learning experience tailored towards the end-users needs (i.e., teachers, learners, administration staff). Thus both provide possibilities for personalization, and, to some extent, adaptation. In this section we will be investigating the differences and similarities between AEH and IMS-LD with respect to *how* adaptation is achieved.

4.1 Adaptation Engineering Taxonomy

Brusilovsky's Taxonomy [10] describes almost exhaustively, from a technical, adaptation engineering point of view, the types of adaptation encountered in Adaptive Hypermedia (AH). The two main adaptation types are *Adaptive Presentation* and *Adaptive Navigation Support*. These are divided into sub-classes which we investigate in turn. Moreover, AH defines the concepts of *adaptivity*, as system-driven adaptation (e.g., adaptation that is induced by a rule-base reasoning system connected to the user interface) and adaptability, as user-driven changes and adaptation (e.g., adaptation that is performed via interfaces allowing users different levels of options, changes in parameters, etc., such as the setting of preferences). All types of adaptation in Brusilovsky's taxonomy can be performed via *adaptivity* and *adaptability*, in general. However, AH systems usually aim for a low cognitive overhead for the user, thus striving towards 'pure' *adaptivity*. Next, we examine to what extent the classes of Brusilovsky's taxonomy can be represented in CAF& LAG as well as IMS-LD.

4.1.1. Adaptive Presentation

Adaptive multimedia presentation

Multimedia adaptivity can only be achieved in current AEH systems at the level of access: different media, such as text, video or others, can be accessed, based on rules and guided by the user model. Such an example is the strategy that caters for visual versus verbal students, written in LAG, 'vis/verb' [1]. Adaptive multimedia presentations are also possible in IMS-LD. In, e.g., the 'e-adventure' project, an adaptive game's story board is specified with the use of IMS-LD [37]. It is then possible to tune the game according to the learner's profile.

Adaptive text presentation

Natural Language Adaptation

IMS-LD as of yet does not provide specific methods for natural language adaptation, nor do most current AEH systems, in spite of some early research in this area.

Canned text adaptation

Canned text adaptation

(inserting/removing/altering/sorting/dimming fragments; stretchtext) is currently the most used type of adaptation in AH systems. A simple example is the Q&A strategy shown in Fig. 3. Canned text adaptation (showing and hiding fragments of text) can be done in IMS-LD as follows. In the XHTML resources of IMS-LD the authors can define their own pieces of text inside DIV layers (division.layers). DIV elements in IMS-LD are just used as placeholders to define ones own custom tags. Authors can thus add conditions, e.g., if a certain element should be shown or not. However, this method is not inherent to IMS-LD, but to (X)HTML in general, and it is extremely low-level (adaptation assembly language in LAG) and time consuming.

Adaptation of modality

Adaptation of modality, such as providing multiple language alternatives, or providing different media (videos, text, audio) could be implemented in IMS-LD, by adding all modalities to the resource files and then selecting which one to show upon certain conditions. Such adaptation is typical, for instance, for browsers, so for AEH it is less of a mainstream research direction. In CAF and LAG, such adaptation of modality can be simulated in a similar way to canned text adaptation (for different languages) or adaptive multimedia presentation (for different version of media).

4.1.2. Adaptive Navigation Support

Direct guidance

In AEH, direct guidance means often providing an (adaptive) 'Next' Button. This can be added via the LAG language with the statement: PM.GM.Next = true. In IMS-LD, direct guidance can be achieved by using DIV layers in the XHTML resources. In effect, it can then emulate menu-adaptation.

Adaptive link sorting

Adaptive link sorting can, in theory, be done in the same way as hiding and showing of text. In IMS-LD, different DIV layers could be defined with the correct sorting, and the right one showing at the right moment. However, links cannot actually appear in different orders, just at different learning stages. In LAG, a simple version of sorting is achieved by using the 'To Do' list, that specifies which links to access next: PM.GM.ToDo = true.

Adaptive link Hiding

Adaptive link hiding can be performed in the same fashion as the hiding and showing of text. In IMS-LD this can be implemented, since hiding and showing can be performed over resources. In LAG, the following expression will hide not only the respective concept, but also the link (e.g., in the 'To Do' list) to that concept: PM.GM.Concept.show = false.

Adaptive link annotation

Again, in IMS-LD, using DIV layers in the XHTML resources of the IMS-LD makes it possible to obtain link annotation. In the conversion to AHA! [5] of LAG, links to concepts in the menu illustrate adaptive link annotation, via bullets colored in the traditional (pseudo-)traffic-light colors (green, for ready to visit; red, for not appropriate yet; white, for already seen).

Adaptive link generation

Adaptive link generation means providing links on the fly where no links were designed before. This can be, for instance, from a database of links, or from the open web. In AEH (thus in CAF and LAG), links to concepts are provided on the fly when the concepts are appropriate for the learner, so from this point of view, they are adaptively generated links. For IMS-LD, adaptive link generation would require the ability to change the XHTML resources at run-time. IMS-LD allows some limited changes on the fly, via a UoL run-time. As long as the possibility has previously been provided for at design-time, (permanent) links can be added at run-time, by either a teacher or a student. This represents adaptable[3] (but not adaptive) link generation.

Map adaptation

Adaptive maps can be achieved in a number of ways. For IMS-LD, the most labor-intensive approach for the author would be to create a picture for every possible version of the map. A smarter method would be to have the map drawn by DIV elements defined in some style sheet. However, this is too demanding for current authoring tools. For AEH there is some new research in this direction [11], but most current systems don't provide it.

4.2 Pedagogy-based Adaptation Classification

In a literature study that analyzes adaptation from the point of view of its pedagogical goal, instead of a technical viewpoint, eight different kinds of adaptation in eLearning systems were defined [15]: *Interface based*, *Learning flow based* [41], *Content based*, *Interactive problem solving support* [36], *Adaptive information filtering*, *Adaptive user grouping*, *Adaptive evaluation*, and *Changes on-the-fly [44]*. This classification was previously used [14], [15] to describe (and defend) the expression power in terms of adaptation for IMS-LD. Therefore, after analyzing how IMS-LD compares in terms of AH and AEH adaptation, we now investigate how AEH systems in general, and CAF & LAG in particular, can cope with the demands of this classification. The results are shown side-by-side with the IMS-LD results. For the sake of knowledge exchange between these domains, we also map this new classification over Brusilovsky's taxonomy.

4.2.1. Interface Based Adaptation

Interface adaptation is based on menu options, navigation facilities and visualization facilities. In such general terms, this type of adaptation would reflect on basically all aspects of Brusilovsky's taxonomy. More specifically, it can reflect on adaptation performed on such interface aspects as menus (items), display options, size of windows, fonts, etc. Thus, this is the most commonly encountered type of adaptation in AEH. Examples can be found at [1], e.g., the 'Beginner/Intermediate/Advanced' strategy uses adaptive navigation support to color recommended links green, non-recommended ones red, etc., guiding learners through these three learning stages (from beginner to advanced).

For IMS-LD, this issue relates to the user interface provided with IMS-LD players such as the CopperCore

[3] thus, from an epistemology p.o.v., an analytic, a priori solution, instead of a (more interesting) synthetic, a posteriori one;

player [48], the Reload Player [4] and Sled [45]. The current generation of these tools does not provide facilities to allow interface adaptation at run-time, although Sled can be customized during the set-up using stylesheets. Current IMS-LD players cannot change the size and position of their panels or working areas, the definition of their windows (adaptation of modality) or any other navigation facility (adaptive navigation support). These players cannot change basic features, like font-size, font-color, font-type or alignment, either. However, limited adaptation is possible inside the Unit of Learning, if we use two resources: *DIV* and *environments*. *DIV layers* can be shown and hidden at run-time by any of the main participants in the learning process (student, teacher, and set of rules). Inside a DIV layer we can define the different options and/or the look and feel of the same content, meaning a de facto interface-based adaptation. Furthermore, environments can provide different set-ups (contents, approaches, views) related to the same Unit of Learning, leading to a final personalized interface. Although neither of these two solutions (DIV layers and environments) is based on the external wrapper/player, they can provide a simulation of interface adaptation.

4.2.2. Learning Flow Based Adaptation

Generally speaking, adapting the learning flow means creating different sequences of learning events for the different learners. This is actually fundamental in AEH, where different paths are generated based on user models. A learning flow example for AEH, where a learner can choose between a visual, verbal or neutral presentation, is the 'vis/verb' strategy at [1].

For IMS-LD, describing an adaptive learning flow uses four (out of the five) Level elements [13]: *properties*, *calculations*, *global elements* and *conditions*. In addition, monitoring services can be added to track the students' behavior and allow the teacher to adapt the flow dynamically[4] : e.g., 'Learning to Listen to Jazz' (all examples are at [35]). A student can learn about four different Jazz styles in a sequential way, and he can choose between a thematic itinerary and a historical itinerary, following different milestones in the course. This would correspond to adaptable navigation support in AEH (i.e., not system-driven adaptation, but user-driven adaptability and control). An additional example is 'GeoQuiz 3' where the activities are defined by the performance of a student after answering an evaluation form. Depending on the final score and the related level acquired, one or another activity is shown. Here, actual adaptivity (system-driven adaptation), in the form of canned-text AEH adaptation is portrayed. Thus, the description of an adaptive learning flow is mainly based on four IMS-LD elements at Level B: properties, calculations, global elements and conditions. All these elements exist in Adaptive Hypermedia as well.

4.2.3. Content Based Adaptation

Content Based Adaptation, as in personalization based on domain content, is also fundamental to AEH systems, and can be performed via adaptability, but mostly, it is performed via adaptivity (system-control). E.g., the 'Roll-out' LAG strategy in [1] allows for different content to be displayed to the same learner when revisiting a page, thus taking into account his new experience and learning state. If content is regarded as activity content, such content adaptation can be achieved in AEH by switching between the showing and hiding of several linked environments. E.g., show or hide an HTML page containing a frame with the linked environment (for example, youtube videos, or websites).

In IMS-LD, the content of an activity needs a resource linked to the element 'Activity Description'. Although this link cannot be changed at run-time, three other elements can be modified dynamically: 1) the content inside an XHTML resource, defining classes and DIV layers that can be hidden and shown based on certain parameters; 2) the content of pre-defined properties/variables, that can be replaced with other content typed-in on-the-fly by the learner; 3) the content of an activity can be adapted by switching, showing or hiding one of several linked environments. Examples are 'Learning Activities with Conditions', where a student decides the granularity level that he wants and 'From Lesson Plan to LD Level B', where again a student takes control and switches on or off the audio support of the UoL [35]. These examples map to AEH adaptability. An additional approach to content-based adaptation is through the modification of contents linked to fixed resources and based on external tools. For instance, a resource linked to a wiki service hosted outside an IMS-LD UoL could adapt its content dynamically, based on students', tutors' or authors' contributions. This also maps to AEH adaptability, based on interaction of several users and user types (thus is not adaptation).

4.2.4. Interactive Problem Solving Support

This type of adaptation involves helping students to solve problems, in a gradual, adaptive way. Adaptive Hypermedia can produce interactive problem solving support, similar to that which we saw in section 2. For example, one could create a set of questions, and gradual hints for the students, in order for them to find the right answers. One of the first AEH systems, the LISP tutor [2], was based on interactive problem solving. Adaptive Hypermedia however does not offer any support for author or teacher interference with the run-time flow. For IMS-LD, the latter type of interaction is more appropriate. I.e., problem support can be carried out 1) by a tutor editing previous hints, 2) by executing specific design-time rules, or 3) by a combination of both mechanisms. An example is 'What is Greatness' [35], where the tutor moderates the contributions of a group of students on an open question, providing access to the next step when the tutor thinks that the current one is finished.

4.2.5. Adaptive Information Filtering

Adaptive Hypermedia relies heavily on information filtering. If we look at LAOS, then the Goal Map collects and filters information from the domain maps. The adaptation then does the lower granularity filtering, to show

[4] Please note that this is different from AEH adaptation, which is predefined, and cannot be interfered with during runtime.

the user exactly what is appropriate, according to his profile. Thus, AEH and some of its methodology (especially the research on open corpus adaptive hypermedia) is related to adaptive information filtering, but, as to date, they are still different areas. IMS-LD is not designed to provide adaptive information retrieval or filtering. Some rudimentary facilities are available through the index-search service.

4.2.6. *Adaptive User Grouping*

In IMS-LD, by using management systems provided by several IMS-LD tools and engines – Coppercore, Reload, CopperAuthor [47], once the UoL is published, the administrator (e.g., the teacher) can add or delete users and assign them to a specific run of that UoL. This means a de facto grouping [12]. However, the dynamic creation of roles after the publishing process is not currently possible. Once a definition of roles or stakeholders is available, and a UoL run is defined, specific users can be added to, or removed from, any of these groups and these users can play the run. There is currently no support for adaptive user grouping in Adaptive Hypermedia, although research is on its way.

4.2.7. *Adaptive Evaluation*

Adaptive Evaluation can be performed in Adaptive Hypermedia. In LAG, for example, scores could be kept in user model variables, and then used to adapt the content the user sees to the scores of previously performed evaluations. In IMS-LD, taking the performance of a student in a Unit of Learning as input, a full set of parameters can be stored in local properties to be used in the adaptation of formative or summative evaluations. In Geo Quiz 3, certain actions and answers of a student can be allocated into variables pre-defined at design-time and they can also be interpreted at run-time following a set of rules. In this way, both the evaluation system and the content itself, and even the interpretation of the results, can change for each student.

4.2.8. *Changes On-the-fly*

In IMS-LD, with the current tools, once a UoL is published it is not possible to change structure, method or definition of basic parameters (such as conditions or properties). However if a UoL is so designed, a tutor is able to change the way a student perceives the course and the flow: 1) the tutor can update the content, based on pre-defined content or on new contributions; and 2) the tutor can also influence the learning itinerary, uploading files, showing and hiding content elements and structure elements, etc. This means that a tutor is able to change things on the run, as long as he had previously defined that possibility in design-time.

After authoring in AEH, the author normally does not interact anymore with either the students or the content. This is mainly because the focus has been on creating automatic adaptation and not on providing a software tool for teachers to communicate in real time with their students. Therefore, adaptation on the fly as in IMS-LD is currently not possible in Adaptive Hypermedia.

5 RESULTS SUMMARY ACCORDING TO OUR COMPARISON CRITERIA

In Table 1 we present a brief summary of our comparison. The table shows the types of adaptation we analyzed, and whether these types can be achieved using IMS-LD or AEH, and formalizes the comparison in terms of inputs, outputs, interaction, restrictions, synchronization. For example, the comment for adaptive multimedia presentation reads: for IMS-LD, it allows user model input, but not output, and there is no synchronous multi-media; for AEH, inputs are allowed from user -, presentation -, domain – and goal model, and outputs can be changes in the user - or presentation model (e.g., a user model variable is updated); also, the only possible synchronization with multimedia is with SMIL [46]. The table gives an overall idea of how IMS-LD and CAF & LAG can benefit from each other. CAF & LAG already convert to many formats (such as RDF [42], IMS-QTI , IMS-CP) and systems (such as Blackboard [9], WHURLE [39], etc.). Therefore, a straightforward step would be to export CAF content and LAG strategies into IMS-LD – at least to the extent to which this is possible, as suggested by Table 1.

Table 1 Comparison of types of adaptation

Type	**IMS-LD**	**AEH: CAF & LAG**	**Comments** (UM,PM,DM,GM: as conform LAOS, section**Error! Reference source not found.**; I/O: input/output; synch: synchronization points; MM: multi-media; R: restriction)
Adaptive multimedia presentation	✘✔	✘✔	**IMS-LD**: ∃I(UM) ^ ~∃O(UM) ^~∃synch(MM) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM) ^ (∃!synch(MM) with SMIL [46])
Natural Language Adaptation	✘	✘	**IMS-LD**: ~∃I(UM) ^ ~∃O(UM) **CAF&LAG**: ~∃I(UM) ^ ~∃O(UM)
Canned text adaptation	✔	✔	**IMS-LD**: ∃I(UM) ^ ∃O(UM),PM ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Adaptation of modality	✘✔	✔	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers)^ ~∃O(menu) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Direct guidance	✔	✔	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM) ^ ∃O(UM)
Adaptive link sorting	✘✔	✔	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM))
Adaptive link Hiding	✔	✔	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Adaptive link annotation	✔	✔	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Adaptive link generation	✘✔	✔	**IMS-LD**: ∃interaction(author, learner) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Map adaptation	✘✔	✘	**IMS-LD**: ~∃I(DM) ^ R(DIV layers)

			CAF&LAG: ~∃I(DM)
Interface Based	✗/✓	✓	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers)^ ~∃O(menu) **CAF&LAG:** ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Learning Flow Based	✓	✓	**IMS-LD**: ∃I(UM) ^ ∃O(UM) ^ (R(DIV layers)vR(learning flow)) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Content Based	✓	✓	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Interactive Problem Solving Support	✓	✓	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Adaptive Information Filtering	✓	✓	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Adaptive User Grouping	✓	✗	**IMS-LD**: ∃interaction(learner,learner) **CAF&LAG**: ~∃interaction(learner,learner)
Adaptive Evaluation	✓	✓	**IMS-LD**: ∃I(UM) ^ ∃O(UM)) ^ R(DIV layers) **CAF&LAG**: ∃I(UM,PM,DM,GM) ^ ∃O(UM,PM)
Changes On-the-fly	✓	✗	**IMS-LD**: ∃interaction(learner,teacher) **CAF&LAG**: ~∃interaction(learner,teacher)

6 RELATED RESEARCH

What IMS-LD and LAOS (with the current instantiation of CAF and LAG) have in common is the fact that they intend to provide *generic* and *flexible languages* for expressing, in the first case, various *pedagogies*, and in the second, various *adaptation forms*. In the following, we will look into other research into such generic and flexible languages, as well as into pattern languages [25] for related purpose.

EML (Educational Modeling Language) is the predecessor of IMS-LD. It has also been developed by the Open University of The Netherlands and it was an early attempt to codify units of study including the roles and interactions involved, in a standard way. EML has now however been succeeded by IMS-LD.

In the domain of AEH, another language, LAG-XLS [43] has been developed, as a special-purpose language to expressing learning styles for AEH. Whilst this language is flexible and XML-based (thus easily transportable), it is not general purpose.

An example of related research in the educational domain is [3]. This paper presents a first attempt to extract design patterns for Adaptive Web-based Educational systems based on Adaptive Hypermedia, detailing especially the user modeling patterns and their relations. These patterns could inform strategies as described by LAG or IMS-LD, for example, although the paper keeps the granularity of the patterns relatively high.

Another example of an adaptation language in a different domain of adaptation is the generic language for adaptation of system services in middleware as described in [26]. This language allows system services to be dynamically adapted to suit the needs of the specific middleware. Commonalities exist at the level of adaptation specification, but, as the domains are quite different, the similarities end there.

In yet another application domain, the research into Pattern Languages [25] at the University of Oregon, for planning its campus layout could be considered related, in the sense that this research uses pattern languages to be able to adapt to changes in campus development. This type of research shows the flexibility of using patterns in defining adaptation, and, as a next step, special purpose languages for adaptation, as in the current paper.

Authoring and pattern languages go hand in hand, as shown for the domain of e-learning [16]. This research is still a work in progress, as is the one in [4], in which a simpler authoring system for reusable rules for IMS-LD is being designed. The latter is attempting to marry IMS-LD with AEH concepts.

However, there is as yet very little research on generic languages that allow authoring and engineering of adaptive learning material in such a way that reuse of material is allowed.

7 CONCLUSION AND FURTHER WORK

What does *adaptation* actually comprise? It is a term (mis)used in many ways: some consider adaptation to be personalization, i.e., changes in a (learning) system that reflect changes in the user (learner). Others look further and include in the term changes related to the (perceived) quality of service of the respective (learning) system, to device used (e.g., handheld versus desktop, etc.). As we saw in section 5, adaptation can also be seen in a much broader sense, e.g., of adapting, for example, campus planning towards its users needs, etc. One of the problems therefore to compare systems that claim to provide adaptation and personalization, generally speaking, but in particular, for the e-learning field, is the fact that the definitions used for the term 'adaptation' vary greatly.

In this paper, we have felt compelled, therefore, to use two definitions of adaptation, one based on the famous Adaptive Hypermedia taxonomy by Brusilovsky [10], and the other, a pedagogy-based adaptation classification, most used to describe adaptation in IMS-LD [15], in order to merge the concept of adaptation in personalized distance learning for these two fields. We have seen that, whilst adaptation is possible to some extent in IMS-LD, it is clear that this field would benefit from the previous findings in AH and AEH in particular. Conversely, AEH doesn't provide for specific features that are required for flexible pedagogical settings, such as on-the-fly changes, multiple user roles, etc. From AEH systems point of view, the current view is that such functionality can be provided externally, by other systems. However, adaptivity, could also be adaptivity to collaborative groups. In such cases, for instance, adaptation and roles are intrinsically related, so a merged solution needs to be found. The precise extent both AEH and IMS-LD support adaptation to *collaboration* still needs to be explored further. Also further research is necessary to find out how exactly the knowledge of the two fields could be combined; as IMS-LD is the de facto emerging standard, the best option might be to extend IMS-LD with the whole range of AEH functionality.

This paper thus merges the ontologies of IMS-LD and CAF & LAG at a high level of semantics. This is an important step in connecting two seemingly unrelated fields, that of *adaptive educational hypermedia* and that of *IMS Learning Design*. For future research, it would be beneficial to experiment with concrete conversions. The current results point to the fact that the AEH field might be able to provide for IMS-LD both the firm connection to adaptivity and personalization that is still weak, and the respective authoring tools for designing the adaptation.

8 APPENDICES

8.1 LAG Grammar

```
PROG → DESCRIPTION VARIABLES INITIALIZATION IM-
PLEMENTATION
DESCRIPTION → // "text"
VARIABLES → // "text"
INITIALIZATION →initialization ( STATEMENT )
IMPLEMENTATION → implementation ( STATEMENT )
STATE-
MENT→IFSTAT | WHILESTAT | FORSTAT | BREAKSTAT | GEN
STAT | SPECSTAT | (STATEMENT)*STATEMENT | ACTION
IFSTAT → if CONDITION then (STATEMENT)+ |
             if CONDITION then (STATEMENT) + else
(STATEMENT)+
WHILESTAT → while CONDITION
                  (STATEMENT)+ [TARGETLABEL]
FORSTAT → for RANGE do (STATEMENT)
                 [TARGETLABEL]
BREAKSTAT → break SOURCELABEL
GENSTAT → generalize((CONDITION)*)
SPECSTAT → specialize((CONDITION)*)
ACTION → ATTRIBUTE OP VALUE
CONDITION→enough((CONDITION)+,VALUE) | PREREQ
RANGE → "integer"
PREREQ → ATTRIBUTE COMPARE VALUE
TARGETLABEL → "text" | ""
SOURCELABEL → "text" | ""
ATTRIBUTE → GENCONCEPTATTR | PECCONCEPTATTR
GENCONCEP-
TATTR→LAOSCONCEPTMAP.CONCEPT.ATTR | LAOSCONC
EPTMAP.CONCEPT.ATTR.ATTRATTR |
LAOSCM.ATTR | LAOSCM.LAOSCM.ATTRATTR | LAOSCM.L
AOSCM.CONCEPT.ATTR.ATTRATTR
SPECCONCEPTATTR→
'\SPECCONMAP\SPECCON\SPECATTR\ATTR'.ATTRATTR
LAOSCM, LAOSCONCEPTMAP → DM | GM | UM | PM |
CM
CONCEPT → Concept | "text"
ATTR→Attribute | title | keywords | text | introduction | conclusi
on | exercise | child | parent | Relatedness | ATTR.ATTR | CONCE
PT.ATTR | label | weight | "text"
ATTRATTR → type | order | next | ToDo | menu | show |
access | visited | "text"
SPECCONMAP → "text"
SPECCON → "text"
SPECATTR → "text"
OP → = | += | -= | .=
COMPARE → == | < | > | in
VALUE → true | false | "text"
```

8.2 LAG Grammar Semantics

PROG: A LAG strategy or procedure, containing a set of instructions (programming constructs) defining the user and presentation adaptation in an adaptive hypermedia environment.

DESCRIPTION: The description of PROG; contains a natural language description of the behavior of the adaptive strategy; it serves as the label (meta-description) for the whole strategy. It is important, as laic (non-programmer) authors should be able to extract from it the necessary elements to make a decision about using this adaptation or not.

VARIABLES: The variables of PROG; contains the list of variables that are used in the adaptive strategy. This information can be used by a laic (non-programmer) author to decide what attributes of the GM (goal and constraints model) should be filled-in for this strategy.

INITIALIZATION: The static initialization part of PROG; here, the initial experience of the user, when entering the adaptive environment, is described. This is so that a user doesn't enter a void environment. Here, all the default decisions are set. Adaptive environments which are adaptable but not adaptive can only render this part.

IMPLEMENTATION: The dynamic implementation part of PROG; here, the interactivity between the adaptive environment and the user is described (e.g., the effect of user clicks).

STATEMENT: The LAG language is a simple language built of a number of programming constructs, or statements, as follows:

IFSTAT: This statement is similar to IF statements in other programming languages, and is used for condition-action rules; the exact syntax is given in the grammar. This is the basic building block of the adaptation language. Any other (higher level) building block is translatable to it, as all adaptive hypermedia environments use this as the basis of adaptation.

WHILESTAT: This statement is similar to WHILE statements in other programming languages, and is used for loops; the exact syntax is given in the grammar.

GENSTAT: This statement uses the hierarchical structure in the DM (domain model) and GM (goal and constraints model) for adaptive navigation. It specifies that more general concepts, higher in the hierarchy than the current concept, will be displayed to the user, given that the condition(s) is (are) fulfilled. It is currently not available for the MOT2AHA conversion; instead, the child-parent relation can be used.

More info at: http://www.dcs.warwick.ac.uk/~acristea/MOT/help/

9 ACKNOWLEDGMENT

The work on this paper has been triggered by the EU ProLearn Project (IST Contract number 507310, www.prolearn-project.org) and is currently supported by the EU Socrates Minerva project ALS (Adaptive Learning Spaces, 229714-CP-1-2006-1-NL-Minerva), and by the GRAPPLE FP7 STREP project (215434).

10 REFERENCES

[1] Adaptation Strategies Examples for MOT and LAOS. http://prolearn.dcs.warwick.ac.uk/strategies.html

[2] Anderson J. R., Reiser B. J., The Lisp Tutor. Byte, vol.10, 159-75, 1985.

[3] Avgeriou, P., Vogiatzis, D., Tzanavari, A., Retalis, S. Towards a Pattern Language in Adaptive Web-Based Educational Systems, Advanced

Technology for Learning, 2004.

[4] Berlanga, A., Garcia, F. J. Towards Reusable Adaptive Rules, AHCW'04 workshop, http://arantxa.ii.uam.es/~rcarro/AHCW04/

[5] De Bra, P., Calvi, L. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, 115-139, Taylor Graham Publ., 1998.

[6] De Bra, P., Houben, G.J., Wu, H., AHAM: A Dexter-based Reference Model for Adaptive Hypermedia, Proceedings of the ACM Conference on Hypertext and Hypermedia, pp. 147-156, Darmstadt, Germany, 1999.

[7] Bhosale, D. AlcoZone: An Adaptive Hypermedia Based Personalized Alcohol Education, MSc Thesis,. http://www.alcohol.vt.edu/alcozone06/pdf/alcozone.pdf

[8] Bieliková, M., Moravèík, M. Modeling the Content of Adaptive Web-Based System Using an Ontology, smap, 115-120, First International Workshop on Semantic Media Adaptation and Personalization (SMAP'06), 2006.

[9] Blackboard LMS, http://www.blackboard.com/us/index.Bb

[10] Brusilovsky, P. Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110, 2001

[11] Brusilovsky, P. and Rizzo, R. Accessing Web educational resources from mobile wireless devices: The Knowledge Sea approach. In: F. Crestiani, M. Dunlop and S. Mizzaro (eds.): Mobile and Ubiquitous Information Access, 2004. LNCS, Vol. 2954, Berlin: Springer Verlag, 54-66.

[12] Burgos, D. Step by step. How to install CopperCore, how to publish and run a UoL, http://hdl.handle.net/1820/327, 2004.

[13] Burgos, D., Koper, R. Practical pedagogical uses of IMS Learning Design's Level B. Proceedings of SIGOSSEE 2005 Conference, Heerlen, the Netherlands, 2005.

[14] Burgos, D. Tattersall, C. and Koper, R. Representing adaptive eLearning strategies in IMS Learning Design. TENCompetence Conference, Sofia, Bulgaria, March, 2006.

[15] Burgos, D., Tattersall, C., Koper, R., How to represent adaptation in eLearning with IMS Learning Design. Interactive Learning Environments, 15(2), 161-170, 2007.

[16] Chen, C.-T., Cheng, Y.C., Hsieh, C.-Y. Towards a Pattern Language Approach to Establishing Personal Authoring Environments in E-Learning. Web-based Education, 2007.

[17] Cristea, A. and Burgos, D. 2006. Authoring Adaptive Hypermedia and IMS Learning Design: A Possible Understanding?, Proceedings of the 6th IEEE international Conference on Advanced Learning Technologies (July 05 - 07, 2006). ICALT. IEEE Computer Society, Washington, DC, 1190-1191.

[18] Cristea, A. and Cristea, P., Evaluation of Adaptive Hypermedia Authoring Patterns during a Socrates Programme Class, International Journal Advanced Technology For Learning 1(2), ACTA Press, 2004.

[19] Cristea, A., De Mooij, A, LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators, WWW'03, Alternate Education track. (Budapest, Hungary 20-24 May 2003). ACM.

[20] Cristea, A., De Mooij, A. Adaptive Course Authoring: My Online Teacher, ICT'03 (International Conference on Telecommunications), Papeete, French Polynesia, IEEE,IEE, ISBN: 0-7803-7662-5, 1762-1769.

[21] Cristea, A.I., Calvi, L. The three Layers of Adaptation Granularity, UM'03, Springer.

[22] Cristea, A., Smits, D. De Bra, P., Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia, A3EH Workshop, AIED'05, 2005, Amsterdam, The Netherlands.

[23] Cristea, A., Smits, D., De Bra, P. Towards a generic adaptive hypermedia platform: a conversion case study, to appear in Journal of Digital Information (JoDI), Special Issue on Personalisation of Computing & Services.

[24] 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.

[25] Frequently Asked Questions about the University's use of the Pattern Language. University of Oregon's Planning Office http://www.uoregon.edu/~uplan/faq/FAQPatternLanquage.html

[26] Hachichi, A., Thomas, G., Martin, C., Patarin, S., Folliot, B. A Generic Language for Dynamic Adaptation. Proc. of European Conference on Parallel Processing, ACM/IEEE/ IFIP, Lisbonne, Portugal, LNCS 3648, 40-49, 2005.

[27] Hava Muntean, C., McManis, J. Fine grained content-based adaptation mechanism for providing high end-user quality of experience with adaptive hypermedia systems, WWW'06, ACM, ISBN:1-59593-323-9 , 53-62, 2006.

[28] Hepworth, M. How is knowledge about the consumer of information being applied in the design and delivery of information products and services? In C. Khoo, D. Singh & A.S. Chaudhry (Eds.), Proceedings of the Asia-Pacific Conference on Library & Information Education & Practice 2006 (A-LIEP 2006), Singapore, 3-6 April 2006, 461-478, Singapore: School of Communication & Information, Nanyang Technological University.

[29] IMS, IMS Learning Design. 2003.

[30] Jacquiot, C. Logical generic modeling of adaptive hypermedia systems, PhD thesis, University Paris-Sud XI Orsay, France (in French). http://www.metz.supelec.fr/metz/recherche/publis_pdf/Supelec253 .pdf

[31] Koper, R. and Tattersall, C. ed. 2005 Learning Design: A handbook on modelling and delivering networked education and training, Germany: Springer Verlag.

[32] Kravcik, M., Gasevic, D. Adaptive hypermedia for the semantic web, Hypertext 2006, Workshop on Adaptivity, Personalization and the Semantic Web, ACM, ISBN:1-59593-453-7 , 3-10, 2006.

[33] Kravcik, M. Specht, M. Authoring Adaptive Courses - ALE Approach, Special Issue on "Innovations in Advanced Technology for Learning: Authoring for Adaptive Educational Hypermedia", Journal of Advanced Technology for Learning, 1(4), 2004.

[34] LAOS definitions. http://wwwis.win.tue.nl/~acristea/ MOT/ help/DefinitionsLAOS.doc

[35] LN4LD, Units of Learning developed by several authors at Learning Network for Learning Design of The Open University of The Netherlands, http://dspace.ou.nl

[36] Merceron, A., Yacef, K. A Web-based tutoring tool with mining facilities to improve learning and teaching. Proceedings of AI-Ed'2003: IOS Press, 2003.

[37] Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J. L., Fernández-Manjón, B. Language-Driven Development of Videogames: The <e-Game> Experience. Proceedings of 5th International Conference in Entertainment Computing (ICEC 2006), 153-164, 20-22 September 2006, Cambridge, UK: Lecture Notes in Computer Science, Springer.

[38] MOT. http://prolearn.dcs.warwick.ac.uk/mot.html

[39] Moore, A., Brailsford, T.J. and Stewart, C.D. (2001) Personally tailored teaching in WHURLE using conditional transclusion. Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, Denmark.

[40] Pacurar, E., Trigano, P. and Alupoaie, S. (2005). A QTI editor integrated into the netUniversité web portal using IMS LD. Journal of Interactive Media in Education. Advances in Learning Design. Special Issue, eds. Colin Tattersall, Rob Koper), 2005/09. ISSN:1365-893X.

[41] Paredes, P., Rodriguez, P. A Mixed Approach to Modelling Learning Styles in Adaptive Educational Hypermedia, Journal (208) Advanced Technology for Learning, 1(4) 2004.

[42] RDF, Resource Description Framework, http://www.w3.org/RDF/

[43] Stash, N., Cristea, A., De Bra, P. Explicit Intelligence in Adaptive Hypermedia: Generic Adaptation Languages for Learning Preferences and Styles, Proceedings of the HT 2005 CIAH Workshop, Salzburg, September 2005.

[44] Van Rosmalen, P., Boticario, J. Using Learning Design to support design- and runtime adaptation. In R. Koper & C. Tattersall (Eds.), Learning Design: A Handbook on Modeling and Delivering Networked Education and Training. Heidelberg, Germany: Springer Verlag, 2005

[45] The Open University, The Sled player. http://sled.open.ac.uk

[46] SMIL, http://www.w3.org/AudioVideo/

[47] Van der Vegt, W., CopperAuthor. Open University of The Netherlands: Heerlen. 2005.

[48] Vogten, H. and H. Martens. CopperCore 3.1. Open University of The Netherlands: Heerlen. http://www.coppercore.org

[49] Wentzler, A., Cristea, A.I., Heuvelman, E., De Bra, P., A Step towards Commercial Adaptive Hypermedia, International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H), Weibelzahl, S. & Cristea, A. (Eds.), AH2006. Lecture Notes in Learning and Teaching. Dublin: National College of Ireland. (ISSN 1649-8623)

Mr. Maurice Hendrix, MSc is a PhD student in the Department of Computer Science at the University of Warwick. His research interests are in improving the design and authoring process for adaptation using. He has previously finished his MsC at the Eindhoven University of Technology, The Netherlands, within the same subject. His work has already lead to several papers at international conferences and journals.

Dr. Alexandra Cristea is associate professor in the Department of Computer Science at the University of Warwick. She is the Head of the Intelligent and Adaptive Systems group. Her research interests include adaptive educational systems, authoring of adaptive hypermedia, user modelling, intelligent tutoring systems, semantic web technologies, concept mapping, and artificial intelligence. She has published more than 100 papers on these subjects. She is leading a new Minerva project entitled ‘ALS’ (Adaptive Learning Spaces) started on October 2006. She has also successfully lead as project manager the ‘ADAPT’ Minerva project, that was recommended as outstanding by the TAO. She was previously working at Eindhoven University of Technology, The Netherlands, as assistant professor. She has graduated ‘Politehnica’ University of Bucharest, Romania, with two degrees, a Masters in Computer Science and one in Economical Engineering. She received her PhD at the University of Electro-Communications, Tokyo, Japan and was previously employed there as research associate. She has been organizer of workshops, co-organizer, panelist and program committee member of various conferences in her research field (including, for example, ED-MEDIA, Hypertext, Adaptive Hypermedia, ICCE, ICAI). This year she is general workshop co-chair of the ICALT 2007 conference in Niigata, Japan. She is executive peer reviewer of the IEEE LTTF Education Technology and Society Journal and she is co-editor of the Advanced Technologies and Learning Journal. She was invited as UNESCO expert for adaptive web-based education at a high-level (Ministry of Education and Educational institutes) meeting of East European countries. She is an IEEE and IEEE CS member.

Dr. Daniel Burgos Has recently started the new Technology Enhanced Learning Unit in ATOS research in pain. Previously he was affiliated with the Open University of The Netherlands from 2004 as a Researcher in Educational Technology, after working fourteen years as a teacher, multimedia developer and academic manager in Europe and South America, also with his own company. As an Assistant Professor, he is mainly focused on IMS Learning Design, Adaptive eLearning, Learning Networks and Educational eGames and he is involved in the research projects Unfold, Learning Network for Learning Design, ProLearn and TenCompetence. He has written thirteen instructional books about Multimedia and Internet and more than one hundred articles for journals, conferences and professional magazines. He holds degrees in Communication, Computer Science, Education and Business Administration.