

# What is wrong with the IMS Learning Design specification? Constraints And Recommendations

Daniel Burgos<sup>1,2</sup>

<sup>1</sup>Atos Origin. Research & Innovation  
Albarracin 25, Madrid 28027, Spain  
daniel.burgos@atosresearch.eu  
www.atosresearch.eu

<sup>2</sup>International University of La Rioja  
Gran Via Rey Juan Carlos I 41, Logroño, La Rioja 26002, Spain  
www.unir.net

## Abstract

The work presented in this paper summarizes the research performed in order to implement a set of Units of Learning (UoLs) focused on adaptive learning processes, using the specification IMS Learning Design (IMS-LD). Through the implementation and analysis of four learning scenarios, and one additional application case, we identify a number of constraints on the use of IMS-LD to support adaptive learning. Indeed, our work in this paper shows how IMS-LD expresses adaptation. In addition, our research presents a number of elements and features that should be improved and-or modified to achieve a better support of adaptation for learning processes. Furthermore, we point out to interoperability and authoring issues too. Finally, we use the work carried out to suggest extensions and modifications of IMS-LD with the final aim of better supporting the implementation of adaptive learning processes.

## 1 A brief description of the IMS Learning Design

IMS Learning Design (or simply IMS-LD) [IMS, 2003] is aimed to transform regular lesson plans into interoperable Units of Learning (UoL). This specification is able to use any pedagogical model to get a UoL run-able and editable in an interoperable way. IMS-LD augments other well-known e-learning specifications aforementioned, like SCORM, IMS Content Packaging, IMS Question and Test Interoperability or IMS Simple Sequencing. Furthermore, IMS-LD provides a language to describe the teaching and learning process in a Unit of Learning. It describes among other things the roles, the activities, the basic information structure, the communication among different roles and users; and all these under the pedagogical approach decided by the teacher and-or the learning designer. In this section, we show what is IMS-LD and how it is structured, as well as how it provides Adaptation within the UoLs

IMS-LD is able to describe a full learning flow with several elements -such as roles, activities, environments or resources- and features -such as properties, conditions,

monitoring services or notifications [Burgos & Griffiths, 2005; Koper & Tattersall, 2005].

The usual life-cycle starts with a lesson plan modelled according to the IMS-LD specification, defining roles, learning activities, services and several other elements, inside an XML document called Manifest. An information package written in IMS Content Packaging [IMSCP, 2001] is used as a container for the resources and links them with the IMS-LD structure. Later, the Manifest is packaged with the nested resources in a compressed ZIP file, meaning a UoL. Several examples available are shown later on.

IMS Learning Design uses the metaphor of a theatrical play to visualize how to model Units of Learning. A play is performed by a number of actors, who may take up a number of roles at different times in the play. Similarly in learning design a learner can take up different roles at different stages of a learning process. At the end of each act the action stops, all the learners are synchronised, and then a something new can begin.

IMS-LD consists of three levels: Level A, with the definition of the method, plays, acts, roles, role-parts, learning activities, support activities and environments. It is the core of the specification, contains the description of the elements that configure IMS LD and the coordination between them. For instance, role-parts define what activities must be taken by a role in order to complete an act and, subsequently, a play.

Level B, adds properties, conditions, calculations, monitoring services and global elements to Level A, and provides specific means to create more complex structures and learning experiences. Properties can be used as variables, local or global ones, storing and retrieving information for a single user, a group or even for all the characters involved. Through these mechanisms the learning flow can be changed at the run time, as decisions can be made taking into account dynamic content. Logically it is the used level to express the most of the pedagogical needs concerning Adaptation, personalization, feedback, tracking and several other usual requests of teachers and learning designers.

Finally, Level C adds notifications to Level B, meaning an email sent and a show/hide command to a specific activity, depending on the completion of another one [Koper & Burgos, 2005].

## 2 IMS-LD and Adaptation

In addition to the basic structure of Level A, the elements in Level B and Level C are actually the key for more expressive UoLs (for instance, based on Adaptation or Collaboration), as they combine several features that encourage and make the content and the learning flow more flexible [Koper & Burgos, 2005]. Furthermore, the combination of these elements allows for the modelling of several classical adaptive methods (i.e. reuse of pedagogical patterns, adaptability, navigational guidance, collaborative learning, contextualized and mobile distributed learning, Adaptation to stereotypes), making use of different structural elements of IMS-LD, like i.e. Environment, Content, User groups and Learning flow [Burgos et al., 2007].

In a literature study, we identify eight different kinds of Adaptation being carried out in eLearning systems [Burgos, 2008]: Interface based, Learning flow based, Content based, Interactive problem solving support, Adaptive information filtering, Adaptive user grouping, Adaptive evaluation, and Changes on-the-fly. All of them use various inputs provided during the learning process and aim to tune the activities and actions of the learner to get the best learning experience as possible [Butz et al., 2003]. A wide and consistent set of rules of dependencies among users, methods and learning objects is needed to describe these eight types of Adaptation, and moreover their possible combinations. If we categorize all these types of Adaptation, we can group them in two clusters [Ahmad et al., 2004; Chin, 2001; De Bra et al., 2004; Baeza-Yates & Ribeiro-Nieto, 1999; Van Rosmalen & Boticario, 2005; Merceron & Yacef, 2003; Romero et al., 2003]. The first one consists of three types of Adaptation:

1. Interface-based (also called adaptive navigation and related to usability and adaptability) where elements and options of the interface are positioned on the screen and their properties are defined (color, size, shadow, etc.); this is closely related to general customization and supporting people with special needs which influence personalization, such as colour impairment or poor hearing, for instance.

2. Learning flow-based, where the learning process is dynamically adapted to sequence the contents of the course in different ways. The learning path is dynamic and personalised for every student, but even also for every time that the course is started (also called run or instance), so that the student can take a different itinerary depending on his performance.

3. Content-based, where resources and activities dynamically change their actual current content, as in Adaptive and Intelligent Web-Based Educational Systems based on adaptive presentation [Brusilovsky & Miller, 2001]. For instance, the information inside a learning activity can be classified in three levels of depth, and every level is shown based on a number of factors.

The first cluster with three types of Adaptation becomes the base for the next one. Additional kinds of Adaptation feed a second cluster: 4) Interactive problem solving support; 5) Adaptive information filtering, 6) Adaptive user grouping; 7) Adaptive evaluation; and 8) Changes on-the-fly.

## 3 Methodology of analysis

This section describes how we have carried out the analysis, as well as the methodology followed to do the research in this paper. Previously, we have described how adaptation is envisaged by IMS-LD and which types of adaptation can be expressed with this specification. Furthermore, we have described, modelled and implemented a number of Learning Scenarios which show features for adaptive learning processes.

First, we have defined, modelled and analysed five Units of Learning (UoLs), which are described as learning scenarios (Table 1). In these learning scenarios, we describe adaptive learning processes and features. Further, we carried out an analysis of a real application case from the ATOS University, where a Unit of Learning (UoL) with adaptation features modelled with IMS-LD, was implemented (Figure 1).

ID	Type of adaptation	Description
1	Adaptive Assessment	adaptation on the learner's performance and knowledge
2	Adaptive Authoring	adaptation on the learning designer's method
3	Adaptive Content	adaptation on the learner's decision
4	Adaptive Mentoring	adaptation on the teacher's decision
5	Combination of adaptive types	Application case on Corporate training

Table 1. Learning scenarios

Last, every learning scenario is analysed and reports on shortcomings and recommendations to improve the expressiveness of IMS Learning Design to achieve a better adaptation process.

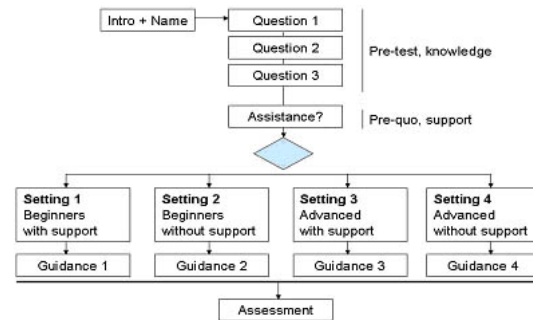


Figure 1. ATOS application case

Our analysis is focused on the main challenges and limitations to performing adaptive learning with IMS-LD. These mainly focus on the need for improving the flexibility and interoperability of this specification, while modelling adaptation.

All of them are available at the GRAPPLE Project website<sup>1</sup>.

<sup>1</sup> www.grapple-project.org

## 4 How IMS-LD expresses Adaptation

In this section, we examine how IMS-LD can be used to represent each of the eight types of Adaptation aforementioned. A combination of the following proposals on Adaptation could support the performance of every role in an eLearning process. Taking the first block (which consists of the three main types), IMS-LD is able to model Adaptation:

### 4.1 Adaptation based on the interface

Interface Adaptation is based on options, navigation and visualization facilities. Interface Adaptation is not possible with today's tools for IMS-LD, such as CopperCore<sup>2</sup> Player, Reload LD Player and Sled. As long as the Adaptation of the interface is based on the tool and not on the Unit of Learning that is interpreted by the player, this is still true. Today's players do not yet provide facilities to change the size or the position of the navigation panels, or even open and close the working areas in the player. Either, these tools cannot change the style sheets related to a HTML file, part of the content, and any of the linked features, as font-size, font-type or background colour, for instance. Although the CopperCore engine provides the appropriate infrastructure, no player uses it so far. Nevertheless, some kind of adaptive interface is possible, using DIV layers and environments.

### 4.2 Adaptation based on the learning flow

The modification of the learning flow as the Unit of Learning is being executed is one of the most often used types of Adaptation. Taking the flow as a base, the Unit of Learning provides different activities, resources and services, depending on these four inputs during execution (user's behavior and performance, user's decision, teacher and set of rules). The activity structure in an IMS-LD UoL is defined using plays, acts, activity structures, learning activities, support activities and environments. We can also use the property of visibility to hide and show these elements and to adapt the learning flow. In these cases the property works as a flag, switching on and off the elements referred to.

### 4.3 Adaptation based on the content

Content Adaptation is based on the information inside an activity that is shown and handled. We know that a learning flow is mainly focused on the sequence of the activities in a Unit of Learning. However, content based Adaptation is focused on the information of every activity, and on the activity itself. There are two main approaches for content based Adaptation in IMS-LD: Flag properties and content of properties. Flag properties hide and show elements like e.g. activities or environments. On the other side, the content of specific properties can be modified on the run, making use of global elements in the specification.

### 4.4 Elements in Levels B and C to model Adaptation

The elements in Level B and Level C providing support to Adaptation in Units of Learning are categorized as a) properties, b) conditions, c) global elements, d) calcula-

tions, e) monitoring services, and f) notifications [Koper & Burgos, 2005; Burgos & Specht, 2006]:

1. Definition, set-up and use of properties: Properties are taken as variables to store values. There are several types of properties: local, local-personal, local-role, global-personal, global. There is also a property-group that is able to compile a number of the others.

2. Conditions: IMS-LD is able to define a basic structure if-then-else, or multiple structure with several chained basic if-then-else in a row, for instance to change the value of a property or to show and hide one element.

3. Global elements: Global elements provide a communication flow between the `imsmanifest.xml`, where the different levels of IMS-LD are set-up, and other XML files. Mainly, they can get an input from the user and they can show a value of a property. Furthermore, they can manage DIV layers in XHTML, for instance to show and hide specific content.

4. Calculations: IMS-LD is able to make some basic arithmetic's (sum, subtraction, multiplication and division) and some combination of a number of them in a row, to get a more complex formula, like a simple average, for instance.

5. Monitoring service: The specification allows monitoring any kind of property assigned to a user or a role, for instance. In order to start this action, firstly the component monitor must be set-up inside an environment and later the property can also be monitored.

6. Notifications: An action is automatically launched depending on the state of a property or a previous action, i.e., when a student ends an assignment an email is sent to the tutor.

## 5 Identification of constraints, gaps and issues to cope with

We use every learning scenario aforementioned as a base to find restrictions, drawbacks and elements to improve within the specification. These resources show how far IMS-LD supports adaptation, when different inputs and roles are involved. We also make links to the integration of UoLs, when needed. Out of the modelling and development of those UoLs we perform an analysis on which features, elements and components are missing or could be modified in order to achieve a more adaptive- and expressive-oriented general definition, with the ultimate aim of improving the specification and bringing it closer to actual needs on eLearning.

In this section, we provide a detailed analysis of what IMS-LD can and cannot model, in its current information model, with regards to adaptation. This analysis concentrates on the weak points and main features of every learning scenario. These remarks will be addressed to produce a set of recommendations (i.e. extensions and modifications) to improve the pedagogical expressiveness on IMS-LD, focused on adaptation, in the next section.

Following, we summarize our main findings. With regards to the specification itself:

1. The definition of properties and the link through several working XML files is too complicated to become useful

2. The relation between layers and actions is not straightforward and it has to be done interlacing files, through global elements and XML

---

<sup>2</sup> [www.coppercore.org](http://www.coppercore.org)

3. The lack of a richer conditional structure makes the editing of the set of rules more complicated on paper than they actually are from a rational point of view

4. Controlled iterations in the activities are not allowed. Furthermore, a closed activity cannot be re-initialized and/or go backwards

5. The monitoring service doesn't cover any kind of user grouping. Therefore, a user (e.g. either a teacher or a learner) cannot follow the performance of several other users at the same time

6. Questions and answers are not personalised for user; they are identical for all users with the same role

7. The communication between teacher and student is little and indirect. They can view the values of properties but there is no other communication service between them

8. There is a lack of flexibility in the input point of changing the itineraries. In the type Sequence, the learning activity with the question appears always at the same place. In the type Selection, the question is always presented after 2 completed learning activities. In case the learning designer/teacher wants to shift this input point, they cannot do so

9. There is no possibility to handle absolute time to start the course and/or a specific activity. Only relative time to the precise time when the instance is created out of the UoL, it is possible

10. There is no chance to make a connection to an already existing database (for instance, to make a query or to import already enrolled students or teachers). The data type of connection is not supported. Therefore, every enrolment has to be done by hand or running a specific tool for that

11. Furthermore, any connection with the external world is impossible. For instance, a real-time effective communication between an LMS and an IMS-LD UoL is not possible so far, so that in fact they cannot benefit each other from mutual services and resources. There is no foreseen dispatcher or service in the specification allowing such connection [Moreno et al., 2007]

12. When an executable module is developed with other technologies (Macromedia Flash and PHP, for instance), it cannot be integrated with IMS-LD in any way. Therefore, we also identified an interoperability problem. Although IMS-LD is not developed with the intention of supporting such interactivity with users, it could allow for a valid integration with external resources using a layer of communication/dispatcher.

13. A file uploaded from the hard disk of a computer is stored in a file-type property inside the internal database of the engine (CopperCore, in this case). There is no possibility to change the default configuration for storing or retrieving resources. There is no facility to manage those uploads either. Although this is an issue concerning tools too, the core documents of IMS-LD do not provide this information and/or service either

14. IMS-LD does not allow saving information into external files or retrieving information from any external source

15. To perform a dynamic user selection in order to create groups is not possible. The teacher can monitor each user, and provide him/her with some feedback on a personal basis. We could set-up a property to be dealt by groups, but these groups should be established before the actual start. However, if the teacher wants to make a dynamic creation of a group of students depending on their

answers, this is not possible so far. To this extent, groups and roles are the same thing

16. IMS-LD does not allow for recording the user's behaviour; in fact, no measures (i.e., Total Time Needed, Time Before First Move) can be restored or retrieved

17. As a consequence, adaptation based on the user's behaviour cannot be developed using the IMS-LD specification. Furthermore, the current state of tooling does not support it either

In addition, with regards with the current engines, we highlight a few issues that would support a more powerful use of the specification:

18. Changes on-the-fly are not possible. In case that the teacher or the learning designer wants to change i.e. the questions, the answers, or the content of the next activity to be carried out, they find that. Every single resource has to be packed in design and publishing time before the actual running of the instance

19. In questionnaires and other forms with fields, the teacher/learning designer cannot modify the number of questions or answers, once the UoL has started

20. There is no option to run the UoL (the whole UoL or a part, such a Learning Activity) twice within the same instance. Once a Learning Activity is closed, the user can read it again but the associated learning flow cannot be executed. For instance, after the question to change the itinerary is made in the historic-route, there is no way to go back

21. There is no flexibility to change the content. When the teacher/learning designer wants to keep the same method and the same structure, but he/she wants to change one single HTML page with some content, the UoL has to be validated and published again. In this case, the learner and the teacher would have to be enrolled and the learning process starts from the very beginning

22. Users cannot be dynamically enrolled within the UoL, once it has started, and they have to be managed by an external tool

## 6 Further analysis

In the next section we show specific recommendations which deal with extensions, modifications of modelling structures, elements and components, as well as with the architecture of IMS-LD. Those recommendations are based on the constraints pointed out in Section 5. However, there is a need for presenting some further analysis, which can bridge both sections, from the constraints to the recommendations, since this in-between step is crucial to understand the rationale. We have organized the analysis as follows:

a) Analysis on general-purpose modelling. These elements will be used as part of others specifically implemented in learning processes, like personalisation. Furthermore, they become a basic set to be re-purposed in different contexts and goals. Therefore, this initial analysis comprises adaptive learning. A few very specific processes cannot be approached with just general structures. They need on-purpose elements which come across on-purpose goals on personalisation

b) Analysis on the integration of Units of Learning and a bi-directional communication with other external resources, systems and standards. When needed, we highlight the need for a way of communication (e.g., a communication layer) although its development is something outside of the scope of this research. We are focused on

the specification itself and how to improve the pedagogical expressiveness, and not on building any *ad hoc* technical artefact to get this aim through.

Out of this analysis, we conclude that specific recommendations should be categorized in three groups:

a) Modelling, that compiles every single extension, modification or addition, general or specific, to the specification and the information model; and b) Architecture, that deals with functional requirements of the spec, with a focus on the interoperability, communication and integration of IMS-LD with other external means. In both cases, we look for the highest performance along with the minimal structural change. Furthermore, we respect the original specification as much as possible and try to make as few changes as possible; on the other side, they all are needed to build the suggested solution, and cope with the overall approach. In addition, c) we reflect some recommendations about the authoring tools. Although they are not responsibility of the specification, they are indeed related to IMS-LD, since the tools which allow the end users to create useful and applicable Units of Learning, can make the process easier or more difficult, and therefore it constraint the actual use and outcomes.

Furthermore, we depict our conclusions within the same two main blocks that we have used to carry out the analysis: modelling (with a special focus on adaptation) and integration. Out of our solution, we also provide a brief note about authoring tools.

## 6.1 Modelling and Adaptation

With regards to general modelling, and modelling focused on adaptive learning we conclude that IMS-LD shows a metaphor difficult to understand. It is not as much to say that people do not understand what a theatre is or how a play is performed. The key issue comes when a teacher needs to translate this well-know structure into specific pedagogical resources and features. This translation process turns not to be so obvious. The conceptual model is clear: play, acts, roles, role-parts, and so on. But all of them, interlaced in a whole structure of learning, become complex. Even the simplest scenario requires some knowledge of the specification in a technical way. And this is far from being user-friendly, moreover when the usual target people consists of non-technical profiles.

The notation itself follows a usual XML Schema and the definition of the several elements and components of the spec can turn too complex, even for skilled programmers. The description of activities, activity structures, environments, and etcetera, and the long cascade of relationships amongst them, makes a difficult-to-trace chain out of a simple scenario. Not to mention when several roles are involved, when some components of Level B are used or when adaptive processes are required. The programming structure is quite easy, but the combination of elements, components and metaphor, makes it hard difficult to implement.

The programming components provided by IMS-LD are quite simple (i.e., simple condition, based arithmetic, visualization of variables, visibility, DIV layers, and etcetera). On the other side, their syntax is long, which hinders the rationale of the modelling process itself.

## 6.2 Communication, interoperability, integration of Units of Learning

We study three ways of communication: 1) simple link between parts, 2) embedded information packages with no information exchange, and 3) full communication of information packages, sharing variables and states. This third solution becomes the most effective one. It implies the development of a communication layer that deals with effective bi-directional exchange of data between information packages. Furthermore, this solution allows for the communication and sharing of services, along with variables, values and states, between IMS-LD and any outside counterpart, i.e., other specifications (e.g. SCORM), languages (i.e. PHP, Java, and Action Script), and LMSs (i.e. LAMS<sup>3</sup>, Moodle<sup>4</sup>, .LRN<sup>5</sup>).

Should this exchange actually happens, it will encourage the re-use of information packages in different contexts, and the development of templates, fostering the re-purpose of Units of Learning within and amongst the several communities of practice (target groups) involved in IMS-LD, beyond the very only technical niche.

In the same line, exportation and importation of Units of Learning is not developed so far; neither does any connection with a database. Once more, no information exchange with other entities is possible so far.

The current two-step working process that makes two isolated parts out of design-time and run-time, makes IMS-LD to be compiled and not interpreted. This distinction stops an on-the-fly visualisation and modification of the learning design, which would improve the interactive personalisation of the learning process. This issue deals with how IMS-LD is interpreted by tools and engines developers and not with how the specification is actually designed.

## 6.3 Authoring

As aforementioned, this research and paper are focused on the specification itself and it does not deal with tools. However, authoring tools largely influence what can be modelled and how. Therefore, we point out a couple of key issues that could support the actual adoption of IMS-LD by the target groups:

a) There is a need for high-level visual authoring tools. Nowadays there are two types of tools: effective but too technical, even for technical profiles; and simple to understand but not powerful, since they usually deal with the very basic Level A. The creation of UoLs should be as far as possible from technical requirements or the underlying elements, components or structure. A more visual approach would encourage the understanding and use of IMS-LD in a broader sense by target groups. Technical low-level editors should live along with the visual high-level ones, though

b) Any authoring tool should allow for an integrated modelling, working with the manifest, the resources and the required external XHTML files with a common interface. It should dependencies and ease setting of properties. This is a hot challenge, not possible so far.

---

<sup>3</sup> [www.lamsfoundation.org](http://www.lamsfoundation.org)

<sup>4</sup> <http://moodle.org>

<sup>5</sup> [www.dotlrn.org](http://www.dotlrn.org)

## 7 Recommendations: Extensions and modifications

This section presents a rich and structured set of recommendations, modifications and extensions to improve the expressiveness of IMS Learning Design on adaptive learning processes. It lays on the aforementioned analysis. The following set of tables show a summary of the constraints, analysis, and recommendations (Table 2). The tables are structured as follows: in the grey-coloured, first row of each table, Column 1 (ID) numbers the constraints and analysis issues. Prefix M relates to issues concerning Modelling, and prefix A relates to issues concerning Architecture. Column 2 (Constraints...) provides a description of those issues numbered in Column 1. The white-coloured row(s) afterwards, presents the recommendation/s in the same couple format: ID and description.

ID	Constraints, analysis and recommendations
[M.01]	Programming structures and resources are very basic (simple condition, simple arithmetic, properties set-up, visibility, DIV layers)
[Rec.01a]	Condition type case
[Rec.01b]	Condition type case with automatic ranges
[Rec.01c]	Conditional loop, type while
[Rec.01d]	Integer loop, type for-next
[Rec.01d]	Modification of the element <calculate>

[M.02]	There is no management of absolute time. There is no synchronization nor input point to work with relative time from
[Rec.02]	Modification of reference to relative time. Addition of reference to absolute time

[M.03]	Notification service, in Level C, is under-used. It only sends an email or plays an activity
[Rec.03]	Extension of the notification service, beyond using <i>sendmail</i> and playing an activity. It can be called from other structures besides the <on-completion> part of a learning activity

[M.04]	There is a blur way to handle the definition and use of properties and links amongst the several XML with global elements
[Rec.04]	Syntax modification, definition and use of elements view-property and set-property, as long as the properties which make use of them

[M.05]	Relationship between DIV layers and the visibility property is difficult to make and follow
[Rec.05]	In principle, the visibility property of any layer is turn off (hide), making simpler the conditional structure which could make use of it

[M.06]	There is no chance for iterations in any of the basic structures of the IMS-LD metaphor (learning activity, support activity,
--------	---

	activity structure, act, play)
[Rec.06]	Extension of the current syntax of every element with a parameter <iteration> which defines a integer loop (type for-next) and-or a conditional loop (type while)

[M.07]	There is no synchronization input point in the manifest
[Rec.07]	Addition of an element GOTO which allows for a direct guiding of the learning flow

[M.08]	There is no chance to assign a specific activity to a selected user
[Rec.08a]	Addition of an element ASSIGN-ACTIVITY-TO-USER which allows for a direct match amongst users, groups and roles, with learning activities and activity structures
[Rec.08b]	Addition of an element ASSIGN-USER-TO-ACTIVITY which allows for a direct match amongst users, groups and roles, with learning activities and activity structures
[Rec.08c]	Addition of an element SWITCH-ACTIVITY which allows for turning on-off activities and activity structures

[M.09]	There is no chance to make groups out of a selection inside the instance
[Rec.09]	Addition of an element CREATE-GROUP which allows for grouping users of the same role

[M.10]	The monitoring service does not allow for monitoring of groups
[Rec.10]	Extension of the monitoring service to trace roles and groups

[A.11]	IMS-LD does not allow for saving or retrieving data in external files, of any kind of format. In addition, connections with external databases or modules developed with other languages are not described or supported within the specification
[Rec.11a]	Addition of the elements EXPORT and IMPORT to handle files with specific parameters (e.g., type TXT) and which is defined in a new property type FILE-IO
[Rec.11b]	Addition of the elements FROM-DB and TO-DB which allows for saving and retrieving data in a database of type MySQL. The connection is defined in a new property type DATABASE

[A.12]	There is no chance to modify the learning skeleton, method, roles definition or any other structural element in run-time
[Rec.12]	Addition of two couples of global elements: a) <i>view-IMS-LD</i> y <i>set-IMS-LD</i> , b) <i>view-resources</i> y <i>set-resources</i> , which allows for the visualisation and modification

of the learning design and the related resources in run-time
--

**Table 2.** Constraints, analysis and recommendations

At the project website pointed out in Section 3, every recommendation is expressed in an XML format, along with a full description, and one example. For instance (Figure 2):

```
<calculate>
<!-- "ID-OP-1" * ("ID-OP-2" + 3) -->
  <property-ref ref="ID-OP-1" />
  <multiply>
    <group-subtotal ref="SUB-1">
      <property-ref ref="ID-OP-2" />
      <sum>
        <property-value>3</property-value>
      </sum>
    </group-subtotal>
  </multiply>
</calculate>
```

and

```
<learning-activity isvisible="true" identifier="activity-1">
  <title>Activity to carry out</title>
  <activity-description>
    <title>First part of the activity</title>
    <item isvisible="true" identifierref="item-1"/>
  </activity-description>
  <iteration>
    <is>
      <property-ref ref="Answer1"/>
      <property-value>A</property-value>
    </is>
  </iteration>
</learning-activity>
```

**Figure 2.** Example snippets of two recommendations

## 8 Conclusions and further work

This paper shows the background about IMS Learning Design and how to model adaptive learning with this specification. In addition, we provide a thorough analysis of a number of learning scenarios and a detailed list of issues to be modified and improved in the specification to better express adaptation. Based on these outcomes we provide recommendations, modifications and extensions to IMS Learning Design in order to improve its expressiveness of adaptive learning.

With these regards, Level A of IMS-LD provides the basic skeleton and a general framework to work with Units of Learning. It makes the 80% of the whole structure. Level C, and above all Level B provide both the spec with stronger and more versatile resources. These two upper levels are the actual responsible means to model some of the current learning and teaching challenges (i.e., active learning, collaborative learning, adaptive learning, runtime tracking).

Furthermore, we examine how to represent adaptive and adaptable Units of Learning with IMS Learning Design in order to model different types of Adaptation. Based on a literature study, a distinction is drawn between eight types of Adaptation that can be classified in two clusters: a) the main group, with interfaced-base, learning-

flow and content-base; b) interactive problem solving support, adaptive information filtering, adaptive user grouping, adaptive evaluation, and changes on-the-fly. Out of this research and modelling efforts we derived a number of findings focused on the limitations that IMS-LD provides. These findings are mainly focused on adaptive learning process. However, since this topic cannot be isolated from the overall approach of the specifications, some of the limitations, and further recommendations, also address other topics, like interoperability, or even authoring tools.

Indeed, IMS-LD will benefit from a re-structure and modification of several elements focused on modelling and architecture. It will also improve the overall pedagogical expressiveness, along with specific features on adaptation of learning processes and integration with other specifications, LMSs, and learning resources. These are two main objectives of the specification: personalised learning and interoperability. At the same time, IMS-LD would increase its level of implementation in real settings and a wider support from Communities of Practice of end users if one or several high-level visual authoring tools are developed. Nevertheless, this issue is out of the scope of this research, and it deals with research groups and companies working on the adoption of IMS-LD.

## Acknowledgments

The research presented in this paper has been partially supported by the following projects: FLEXO (Spanish Plan Avanza, www.ines.org.es/flexo, TSI-020301-2009-9), GRAPPLE (FP7-ICT-2007-1, www.grapple-project.org, contract number 215434).

## References

- [Ahmad et al., 2004] Ahmad, A., Basir, O., and Hasanein, K. Adaptive user interfaces for intelligent e-Learning: issues and trends. Proceedings of The Fourth International Conference on Electronic Business, ICEB2004, Beijing, 2004
- [Baeza-Yates & Ribeiro-Nieto, 1999] Baeza-Yates, R., and Ribeiro-Nieto, B. Modern Information Retrieval. Boston, MA, USA: Addison-Wesley, 1999
- [Brusilovsky & Miller, 2001] Brusilovsky, P., and Miller, P. Course Delivery Systems for the Virtual University. In Tschang F.T. & T. Della Senta (Eds.), Access to Knowledge: New Information Technologies and the Emergence of the Virtual University (pp. 167-206). Amsterdam: Elsevier Science and International Association of Universities, 2001
- [Burgos & Griffiths, 2005] Burgos, D., and Griffiths, D. The UNFOLD Project. Understanding and using Learning Design, 2005. Available at <http://hdl.handle.net/1820/548>
- [Burgos & Specht, 2006] Burgos, D. and Specht, M. Adaptive e-learning methods and IMS Learning Design. An integrated approach, Proc. ICALT2006, 2006. Available at <http://www.ask.iti.gr/icalt/2006>
- [Burgos et al., 2007] Burgos, D., Tattersall, C., and Koper, R. How to represent adaptation in eLearning with IMS Learning Design. Interactive Learning Environments, 15(2), 161-170, 2007

- [Burgos, 2008] Burgos, D. Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning (Extensión de la especificación IMS Learning Design desde la Adaptación y la Integración de Unidades de Aprendizaje). Doctoral thesis. University Carlos III, Leganés, Madrid, Spain, 2008
- [Butz et al., 2003] Butz, M. V., Olivier, S., and Gérard, P. Anticipatory Behavior in Adaptive Learning Systems : Foundations, Theories, and Systems. In. Berlin: Springer Verlag, 2003
- [Chin, 2001] Chin, D. Empirical Evaluation of User Models and User-Adapted Systems. *User Modeling and User-Adapted Interaction*, 11, 181-194, 2001
- [De Bra et al., 2004] De Bra, P., Aroyo, L., and Cristea, A. Adaptive Web-based Educational Hypermedia. In M. Levene & A. Poulouvassilis (Eds.), *Web Dynamics, Adaptive to Change in Content, Size, Topology and Use* (pp. 387-410): Springer, 2004
- [IMSCP, 2001] IMS. IMS Content Packaging Specification. Version 1.1.3, 2001. Retrieved April 28th, 2009, from <http://www.imsglobal.org/learningdesign/index.cfm>
- [IMSLD, 2003] IMS. IMS Learning Design. Version 1. Retrieved February 27th, 2004, from <http://www.imsglobal.org/learningdesign/index.cfm>
- [Koper & Burgos, 2005] Koper, R., and Burgos, D. Developing advanced units of learning using IMS Learning Design level B. *International Journal on Advanced Technology for Learning*, 2(4), 252-259, 2005
- [Koper & Tattersall, 2005] Koper, R., and Tattersall, C. *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Berlin Heidelberg: Springer, 2005
- [Merceron & Yacef, 2003] Merceron, A., and Yacef, K. A Web-based tutoring tool with mining facilities to improve learning and teaching. *Proceedings of AI-Ed'2003*: IOS Press, 2003
- [Moreno et al., 2007] Moreno-Ger, P., Burgos, D., Sierra, J. L., & Fernández-Manjón, B. (2007). An eLearning specification meets a game: authoring and integration with IMS Learning Design and <e-Adventure>. *Proceedings of ISAGA. 2nd international workshop on Electronic Games and Personalized eLearning Processes (EGAEL2007)*. July, 9th-13th, 2007, Nijmegen, The Netherlands
- [Romero et al., 2003] Romero, C., Ventura, S., De Bra, P. D., and Castro, C. D. (2003). Discovering prediction rules in AHA! courses. *Proceedings of 9th International User Modeling Conference* (pp. 25-34), 2003
- [Specht & Burgos, 2006] Specht, M., and Burgos, D. Implementing Adaptive Educational Methods with IMS Learning Design. *Proceedings of Adaptive Hypermedia 2006*, Dublin, Ireland, 2006
- [Van Rosmalen & Boticario, 2005] Van Rosmalen, P., and 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