

# Building a Learning Object Content Management System

Christopher Brooks  
Advanced Research in Intelligent  
Education Systems (ARIES)  
Department of Computer Science  
University of Saskatchewan  
Saskatoon, SK Canada  
cab938@mail.usask.ca

Lori Kettel  
Department of Computer Science  
University of Saskatchewan  
Saskatoon, SK Canada  
kettel@cs.usask.ca

Collene Hansen  
Advanced Research in Intelligent  
Education Systems (ARIES)  
Department of Computer Science  
University of Saskatchewan  
Saskatoon, SK Canada  
cfh928@mail.usask.ca

**Abstract:** While the educational technology community is rife with learning object-based specifications, tools for supporting these specifications (both at an authoring and at a delivery level) are patchy at best. This paper outlines our experiences over the last two years in building a suite of tools to deploy an online course based primarily on IMS specifications. These tools include a learning object sequencing environment, a content delivery engine, and an assessment application. Finally, this paper outlines areas where specification and tool support could use improvement, based on research conducted from members of our lab over the last decade or so.

## Introduction

Learning objects have been heralded as the next big thing in educational technology. Defined broadly by the IEEE as "any entity, digital or non-digital, that may be used for learning, education or training" [6], learning objects are meant to be easy to find, obtain, and integrate into online courses. To this end, researchers, governments, and industry consortiums have created a plethora of standards and specifications outlining how learning objects can be described, packaged, aggregated, and shared. These specifications attempt to bring a kind of order to the chaos that tends to surround educational technology, and promote interoperability between the courseware offerings from various vendors. Regardless of these potentially noble goals, support for major specifications is still very much in its infancy, mostly due to the large overhead tool vendors must undertake when integrating such specifications in their products.

Developing a course to be taught at a distance over the Internet is a difficult process. It requires not only that content be carefully crafted into accessible and informative web pages, but also that the tools to support self-directed learning, the learning community, and assessment be created and used effectively. This all amounts to much time and money being required to create new course offerings. In an attempt to mitigate the technical challenges involved, many universities have been entering into agreements with third party learning content management system vendors such as WebCT<sup>1</sup> and Blackboard<sup>2</sup>.

In the summer of 2002 the Department of Computer Science at the University of Saskatchewan began the development of our first distance education course. This course, designed for students who were not majoring in computer science, is a breadth course that situates computer science historically and focuses on describing entry level concepts using web technologies. A brief survey of available content management systems was done, and a decision was made to develop both the content and the tools to support delivery of the content in-house. The course developed has since been used by 122 distance education students as the primary resource for learning, and has been available as a supplement to more than 1500 students who received traditional face-to-face instruction.

The goal of this paper is to discuss the issues raised in creating such a content management system. It is a product of joint work completed by the laboratory for Advanced Research in Intelligent Educational Systems (ARIES)<sup>3</sup>, and the instructional and technical support groups responsible for providing technology enhanced learning within the

---

<sup>1</sup> <http://www.webct.com>

<sup>2</sup> <http://www.blackboard.com>

<sup>3</sup> [http://www.cs.usask.ca/research/research\\_groups/aries/](http://www.cs.usask.ca/research/research_groups/aries/)

Department of Computer Science at the University of Saskatchewan. The paper is outlined as follows; firstly, a brief overview of the major e-learning specifications used in our development are discussed. Secondly, the design and implementation of the iHelp Learning Content Management System (iHelp LCMS) is presented, focusing on both the environment itself and the process by which an author can construct and deploy a course. This is followed by a general discussion of the open issues that exist with respect to e-learning. The work concludes with a brief description of our future development goals.

## E-learning Specifications

The IMS Content Packaging Specification [7] describes how digital resources can be organized into logical learning units called content packages. These packages are made up of four major parts; a manifest file which outlines metadata for the package as a whole (usually using the IEEE LOM [6], though strictly speaking the metadata can be made available using other vocabularies as well), a set of organization hierarchies describing how smaller units of learning (e.g. lessons, activities) should be arranged, a collection of references to digital assets (e.g. particular files in the package) which in turn can include descriptive metadata, and finally sub-manifests which effectively allow for the nesting of content packages within one another.

Organizations within a content package are particularly interesting, as they break down larger learning objects into smaller pieces. Different hierarchies then dictate different paths a learner can take through the content, given by an in-order traversal of the organization tree. Instructors can then mark these paths as visible if they are to be rendered in a content management system, or invisible if they should not be shown to the learner. Content packages are typically created by instructional designers (using a tool such as RELOAD<sup>4</sup>), then imported into a content management system (such as WebCT, or Blackboard) for deployment. While content packages may be further stored in learning object repositories to be shared with other educators, this is not yet common practice.

While effective at aggregating content together, the content packaging specification does not provide for fine grained control over which users should see which paths through the content, and provides no mechanisms for dynamically altering paths. Instead, the content packaging specification has been augmented by a number of specifications, in particular the IMS Simple Sequencing specification [3] and the IMS Learning Design specification [5]. These specifications are considerably more complex, and allow an author to associate fine grain rules with smaller content pieces to control the flow of content delivery.

The intent of the simple sequencing specification is to provide a base level of adaptation of course material to a particular learner. This is achieved by allowing authors to indicate within a content package organization which items should be displayed to the learner and under which conditions. Conditions available within the specification include assessment conditions (e.g. those based on a given score on an exam), progress conditions (e.g. those based on the content viewed, length of time spent on content, or objectives the learner has met), and general conditions (e.g. whether the content should always be displayed for this learner). The indication of what conditions have been met is supported by associating action rules (referred to as "rollup rules") with content items, and linking these to a tracking model. This tracking model is similar to the general purpose learner model described in user modeling literature, but is restricted to only the vocabulary provided in the simple sequencing specification.

Absent from the simple sequencing specification is a notion of the different roles that users can play, and how different users can interact with one another to achieve certain learning objectives. This then, is the responsibility of the learning design specification, which indicates how learners can collaborate with one another as well as other supporting actors (e.g. staff) in and around learning activities. Activities are represented using the organizations in a content package, and can be mixed with simple sequencing information to attain even finer grain control over the user experience.

While electronic instructional planning has been highly researched by the e-learning research community<sup>5</sup>, there are few commercial products that provide learner-based content sequencing. Even with industry developed

---

<sup>4</sup> <http://www.reload.ac.uk/>

<sup>5</sup> This research is both broad and deep, and can be found in the areas of user modelling, instructional planning, artificial intelligence in education, and adaptive hypermedia to name a few.

specifications, little or no adoption has been done by commercial environments (both authoring environments and content management systems) beyond the trivial. This situation is frustrating – building home grown tools to provide an adaptive experience for a learner is an expensive and error prone process, especially when many universities have already entered into agreements to provide learning content management systems campus wide. However, deploying static learning content is even more frustrating, as over twenty years of e-learning research is now ready and backed by well defined specifications. The next section will outline our experiences in building a basic learning content management system with the expressed goal of attaining simple learner-based sequencing while remaining compliant with current e-learning specifications.

## Design and Implementation of a Learning Object Content Management System

The designed learning object content management system is made up of two parts; an authoring tool (called Retro), where an instructional designer can sequence course material that has been previously packaged according to the IMS Content Packaging specification, and a delivery engine (called iHelp LCMS<sup>6</sup>), which acts as a front end for the learner taking care of both access control and the rendering of learning content.

Retro is a stand alone system written in Java. Similar to the simple sequencing specification, Retro allows for authors to indicate rules that outline how items within organizations in a content package should be sequenced. These rules come in two flavors: availability rules and evaluation rules.

Availability rules correspond roughly to the simple sequencing "rollup rules", and indicate to the content management system what constraints must be met before the learning activity should be rendered as "visible" to the learner. The specification for rollup rules describes a very basic set of conditions for determining when an object is rendered (e.g. objective met, time limit exceeded, learning object attempted, etc). We extended this set to include the following conditions:

- Current date and time: Some classes explicitly withhold a learning object until a certain time passes. For instance, an object that details an assignment solution might be available only after the assignment deadline has passed.
- Chosen path: In some instances it is useful to only display a learning object once a user has made a selection from a set of other learning object. For instance, one could make a learning object on electronic shopping carts available only if a user has previously chosen learning objects that dealt with the World Wide Web and e-commerce.
- Completed: We introduce the general notion of when a learning object is "completed", which is similar to the simple sequencing "objective met" condition, but is determined by the Evaluation rules.

These new conditions, as well as the ones we support defined in the simple sequencing specification, can have a number of Boolean operators applied to them to control the logic even further, such as "and", "or", and "not". A few data specific functions, such as "time greater than", have also been defined. For example, a rule indicating that a learning object should only be viewable if the student has seen the previous object ("Module 0: Intro") and received a score of 80% on a given test ("Test 1"), or a given date has passed ("01/25/2005"), would look like:

```
(viewed("Module 0: Intro")) && (score_gt(80.0,"Test 1")) || (time_gt("01/25/2005"))
```

Evaluation rules are attached to a learning object to determine the value of the "completed" condition for that object. These rules follow the same format as availability rules, and serve as a convenient way to encapsulate the notion of whether a learning object has been finished or not, thus reducing the amount of detail required in availability rules.

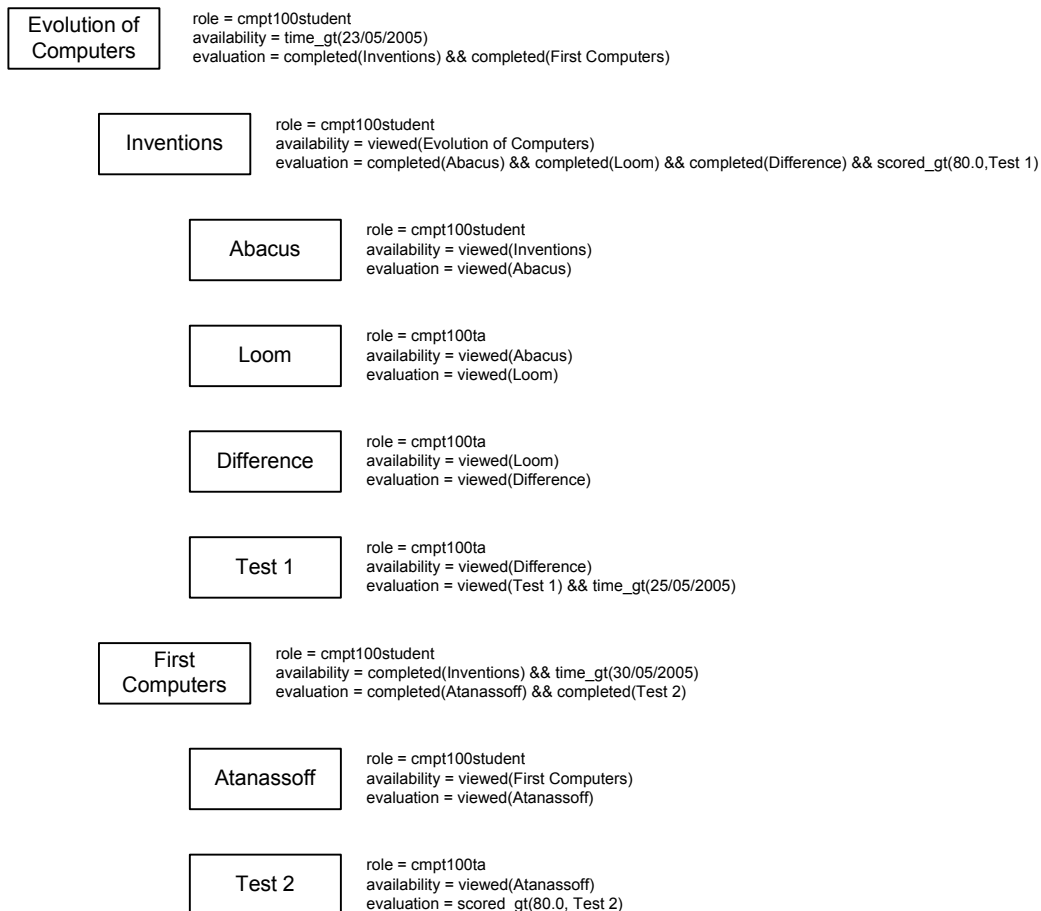
Unlike the simple sequencing specification, Retro allows an instructional designer to specify different rules for the different roles a user may have. A role and permissions structure is integral to any large scale student information system, where users can adopt the roles of instructors, students, tutorial assistants, and markers (to name a few). Further, as learning objects are reused and shared across courses, course-specific roles are required to keep track of which objects should be available to which students. To this end, a rule attached to a learning object can be considered a triple where the first part indicates the role the rule is valid for, and the second part indicates the

---

<sup>6</sup> <http://ihelp.usask.ca>

condition to be checked for availability, and the third part indicates the condition to be checked to determine if the learning object has been completed.

Figure 1 provides a graphical representation of a marked up content package. In this example each box represents a learning object, with indented learning object below indicating a hierarchical example. Availability and evaluation rules are shown for each learning object for one role only, that of a typical entry level student. In this example, the instructional designer has decided that the root object, on the "Evolution of Computers" should be shown to the learner once the date of May 23<sup>rd</sup> has passed. After this date, the object will be rendered, and the child objects will have their availability rules evaluated to determine if they should be shown. The designer has decided to make students traverse the course in a highly structured manner – the "Inventions" learning object will only be shown once the student has viewed the "Evolution of Computers", the "Abacus" learning object will only be shown once the student has viewed the "Inventions" learning object, and so on. Interestingly, the designer has chosen to indicate that the "Inventions" learning object is only completed once all child objects have been completed, and the student has scored a mark of greater than 80% on the first test. A different approach has been taken with the "First Computers" learning object, where the designer has opted to consider the object completed as long as both the "Antanassoff" and "Test 2" objects are completed. While the designer could have restricted the availability further, the rules for the "Test 2" object (perhaps set by another instructional designer), are already fairly strict. This demonstrates the ability to encapsulate and partition complex rules of a given learning object, while retaining the flexibility to override or add to these rules in determining availability.



**Figure 1: Annotated Content Package**

Once created, rules are converted into the Java programming language and are serialized into classes as "if-then" conditional statements. These classes are then compiled and stored in a database in the delivery engine.

When a given user logs into the iHelp LCMS (shown in Figure 2), they are given a three frame view where an outline of the course content is displayed on the left, navigation and tool options are displayed at the top, and learning object content is shown to the right. For each role the user is a member of, the availability rules associated with the root activity in that course are evaluated. This process is a recursive process, and continues down the activity tree, as described in Figure 1. Evaluation down the tree stops either when the availability for a node fails (and thus and children of that node will by default not be made visible), or when the user has the navigation view collapsed (and thus has chosen not to see children of a given activity). The result of this process is a set of trees indicating which nodes are available for each role the user is in. These trees are merged together (using a Boolean OR operation) into a single tree which is rendered to the user.

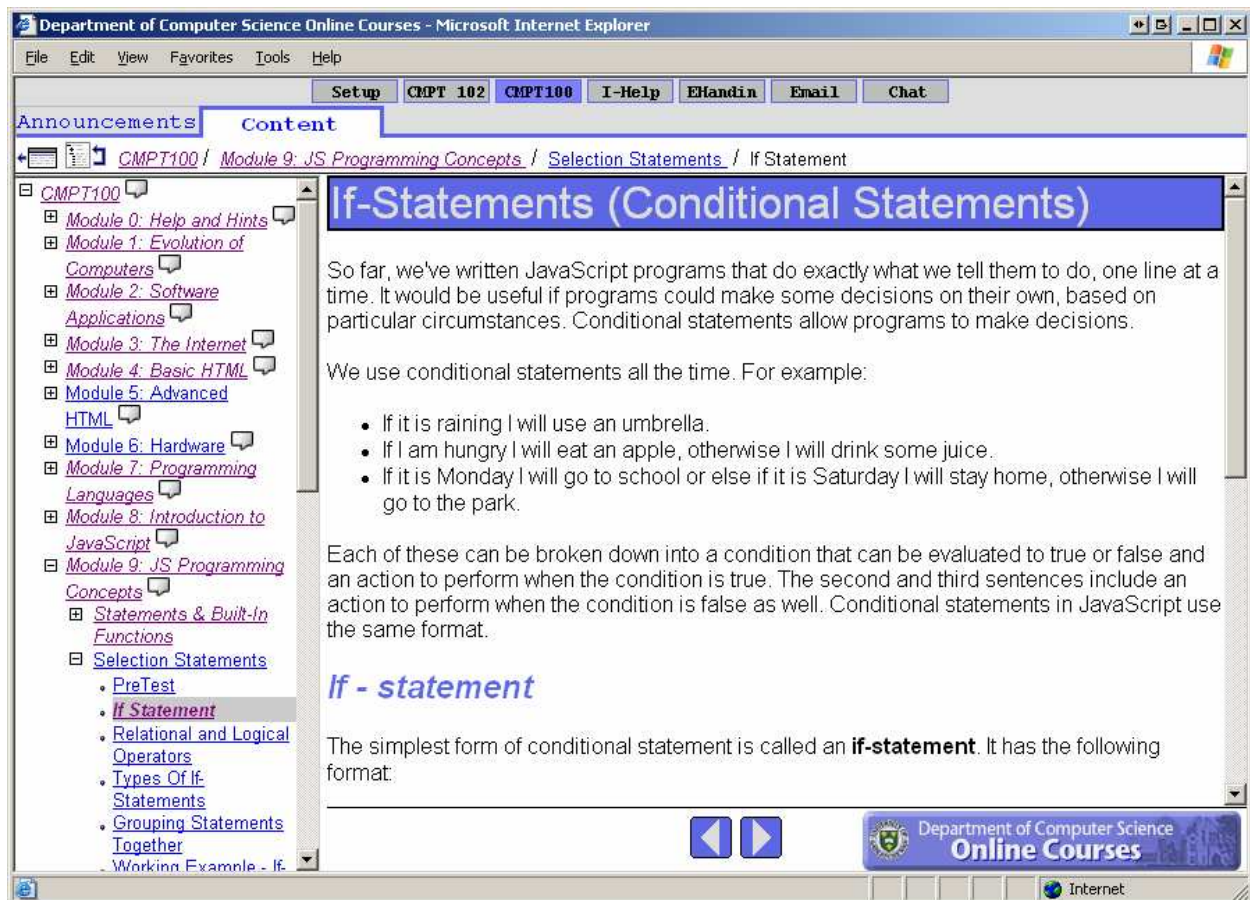


Figure 2: A rendered view of the iHelp Learning Content Management System

By storing the rules as compiled Java statements, they can very quickly be loaded and evaluated using the Java Reflection API. Commonly, students need to know why they cannot access a learning object (e.g. which rule is failing). This tree-based evaluation approach allows us to very quickly determine which rules have failed, and can be mixed with plain English explanations to instruct the learners on what they need to accomplish to continue.

It should be noted that our adoption of the IMS Simple Sequencing specification is only at the abstract data model level, and not with the XML binding that the IMS provides. There are two reasons for this; firstly, by integrating sequencing information within content packages directly some third party tools that are not aware of how it should be handled inappropriately render the contents of content packages. Secondly, the implementation of a component to read and write the IMS Content Packaging format, with our extended simple sequencing data, is a non-trivial process, and poses no benefits until third party applications begin to support this format.

## Open Issues with E-Learning Specifications

While there is a history of over twenty years of e-learning research (including computer based training (CBT) and computer aided instruction (CAI)), accepted industry specifications and standards to support e-learning are only just beginning to emerge. We identify one particular aspect relevant to our own research that we feel is lacking in the current standards landscape; the absence of flexible learner modeling that can be used to support adaptive instructional planning.

In its most general form, a learner model can be viewed as a collection of data which include the learners knowledge of various concepts or topics, and the learners preferences or learning styles [9]. This data can be gathered either implicitly (e.g. by observing the choices or directions the user takes within a learning environment), explicitly (e.g. through the form of a quiz or aptitude test), or by using a mixture of both approaches. Popular e-learning specifications, such as the IMS Simple Sequencing specification, tend to consider a learner model (e.g. the tracking model) to be a set of closed set of values mostly oriented around the activities a student has done, and the marks they may have received on these activities. Other specifications, such as the IMS Learner Information Package [8], provide slightly more detailed user models, which include learner interests, competencies, and administrative information. Despite the large amount of research put into educational objective taxonomies in almost fifty years ([1], [2], etc.), none of the e-learning specifications provides a method of linking content and learners (either through competencies or interests) to both topics and levels of knowledge in a topic.

Content adaptation through automated instructional planning requires both content ontologies (which relate topics within a course to one another) and flexible learner models. The general idea is that an instructional plan outlines a set of content that needs to be learned, and an order in which it should be learnt. Learning material is then annotated with respect to the content it teaches, and can be dynamically chosen by a computer as the student needs it. This is one level of abstraction above the current editing process, and instead focuses instructional designers on providing a clear path of concepts that should be learned instead of a path of learning material. The benefit of this approach is that a piece of automated instructional planning software can then present learners with content that is personalized to their particular needs (learning style, cultural background, etc.) but still teaches the topics that are required.

Our initial method of investigation (described more fully in [10]) of this approach has been to create a domain map covering the content of our online course offerings. This map makes use of Semantic Web technologies, and allows for question/answer pairs from online quizzes to be associated with both concepts within the map, and a depth of knowledge which is demonstrated. It's important to note that since a question can have multiple "correct" answers, as different answers may show different levels of knowledge. For instance, consider the following JavaScript question:

Question: Label the following from this list of options: statement, function, built-in function, parameter, argument, valued function, procedure function, operator, operand, concatenation.

A

prompt( "Please enter your age" , "age");

B

A: \_\_\_\_\_

B: \_\_\_\_\_

**Figure 3: An example question from an online course**

If the student enters "function" for the value of A, then it demonstrates that they have the ability to remember factual knowledge about the topic "Functions". If they enter the value of "built-in function" instead, then they demonstrate both the ability to remember factual knowledge, as well as the ability to classify conceptual knowledge about the same topic.

Quizzes are asked both before and after a learner views a learning object. The difference between the knowledge demonstrated after reading the object and the knowledge demonstrated before reading the object is then associated directly with learning object. For example, if a student tries to view a learning object that discusses the topic of "JavaScript Functions" they are first asked a number of questions about the different kinds of functions that can exist. A detailed learner model is then built that indicates what knowledge the student already has (for instance, they may have surface knowledge of "Built-in Functions" and deep knowledge of the "Alert" function in particular). After answering the pre-test, the learner is then able to read the learning object which ends with another quiz.

As is expected, learners generally perform better on the quizzes they take after reading a learning object. This net gain of knowledge can then be associated with the learning object formally (e.g. by modifying the IEEE LOM data associated with that learning object). With the mapping of content to concepts complete, the instructional planning software can now dynamically determine which pieces of content are acceptable for delivery to learners, and choose from among these based on other attributes of the content (e.g. learning styles, etc.).

## **Conclusion and Future Goals**

This paper has outlined our experiences in building a learning object based content management system for online courseware. Available to over 1500 students, this system implements popular e-learning specifications including the IMS Content Packaging specification for aggregating and packaging learning objects, as well as the IMS Simple Sequencing Specification Abstract Model for the dynamic sequencing of packaged objects.

Despite use of these *de facto* e-learning standards, the system developed is too static and requires that sequences through content be outlined explicitly by instructional designers before the course is delivered to learners. Further, there is no clear way in which rich student model information can be shared between content systems, or used to provide more adaptive instruction. To overcome this, we have begun the process of supplementing these standards with student models that link the content being delivered to both the concepts and competency levels being taught. Instead of taking a top-down approach, where learning objects are annotated explicitly with competencies, we take a bottom-up approach and derive the competencies based on student quiz results. We expect this will lead to a more flexible and adaptive system [4].

Once this is done, our plans include building a pedagogical planning system to automatically sequence content for a given learner. Keeping a standards-based approach while doing this is not trivial. In particular, there is no support within current specifications for representing dynamic sequences, short of explicitly creating each sequence within an IMS Content Package. We hope results from our systems can be used to help inform the next generation of standards in this area.

A longer-term goal of ours is to identify how interactions among learners can be used to more effectively build learning networks. We believe that interactions, both indirect (such as learners having similar experiences with the same learning object) and direct (such as synchronous or asynchronous communication between learners, perhaps creating new learning resources themselves) can be used by the pedagogical planner to better determine what kind of learning object a learner should be delivered at run-time. Integrating the ability to capture and understand these interactions within our learning environment has been of ongoing interest.

## **Acknowledgements**

This project involved a great number of people from both the Department of Computer Science Instructional Support group as well as the ARIES research laboratory. Special thanks go to Jim Greer, Gina Koehn, Jayakumar Srinivasan, Sonia Chiasson, Monisha Shukla, Carrie Demmans, Guus van de Velde, and Gayle Eyolfson who helped in the administration, design, and implementation of this project.

This project was funded in part by the University of Saskatchewan and the Government of Saskatchewan through the Technology Enhanced Learning program, as well as the Government of Canada through the LORNET NSERC research network.

## References

- [1] B.S. Bloom ed., *Taxonomy of Educational Objectives, The classification of educational goals, Handbook I, Cognitive Domain*, David McKay Company Inc., 1956.
- [2] L. Anderson and D. Krathwohls., *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*, Addison Wesley Longman Inc., 2001.
- [3] *IMS Simple Sequencing Information and Behavior Model, Version 1.2*, IMS Global Learning Consortium Inc., 2003
- [4] C. Brooks and G. McCalla, "Flexible Learning Object Metadata" *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL '05)*, 2005. *In submission*.
- [5] *IMS Learning Design Information Model*, IMS Global Learning Consortium Inc., 2003
- [6] *IEEE P1484.12.1-2002, Draft Standard for Learning Object Metadata*, IEEE, Inc., 2002
- [7] *IMS Content Packaging Information Model, Version 1.1.4*, IMS Global Learning Consortium Inc., 2004
- [8] *IMS Learner Information Package Information Model Specification, Version 1.0*, IMS Global Learning Consortium Inc., 2001
- [9] D. Vogiatzis et al., "The Learner's Mirror" *Ninth European Conference on Pattern Languages of Programs*, 2004.
- [10] M. Winter, C. Brooks, and J. Greer, "Towards Best Practices for Semantic Web Student Modelling" *12th International Conference on Artificial Intelligence in Education (AIED 2005)*, IOS Press, 2005.