QUIS

Work Package 5:

Standards for e-learning

Authored by the partners of the project

QUIS: QUality, Interoperability and Standards in e-learning

Agreement n. 2004 – 3538 /001 – 001 ELE–ELEB14

Contact authors:

M. De Marsico, M. Temperini, A. Bianco

Università *La Sapienza*
Roma, Italia
(demarsico@di.uniroma1.it, marte@dis.uniroma1.it, adewhite@libero.it)

Organization of the document

This section describes the model of description, to be applied to the standards included into the present document.

Each standard is supposed to have a dedicated section, in which a number of aspects will be covered; the selection of such aspects was mainly driven by the opportunity to provide a significant final comparison among the standards and allow for enhanced analysis. Such a comparison is planned to appear in the final section of the document and might be supported by notes interspersed throughout the survey.

The definition of the various aspect of interest, to be met for each standard, was devised also in accord to previous discussions about the categorizations to use, in view of the construction of a topic map based on this report.

In particular, for each standard the following "aspects of interest" should be covered:

1) *Organizations*: a brief survey about the organizations proposing and supporting the standard (it is reasonable to expect links to such official sites);
2) *Goals*: the main goals pursued by the standard (this knowledge should allow to compare different standards basing on the users' goals);
3) *Components*: different components included in the standard, to manage different aspects of the learning material and processes (e.g. packaging, sequencing, assessment, etc.); a short description is expected, to support comparing the power of different standards;
4) *Metadata*: short survey on the categories of metainformation provided by the standard, used to describe different aspects of the learning material; this is to allow to compare the expressiveness of different standards.
   Notice Some metadata are shared among different standards (for example SCORM uses IMS packaging and LOM metadata): we have to decide how to include them in the survey;
5) *Possible authoring tools* available to create standard compliant resources; this is supposed to help comparing the availability of authoring support;
6) *LMS* supporting the standard; this will help comparing the pervasiveness of different standards.
   Notice This can be also bound to results from WP 4;
7) *Repositories* of standard compliant resources; so we can compare the availability of ready-to-use material for users of the standard.

The final section might allow for a global comparison, based on the following parameters:
- Goals;
- Components;
- Metadata;
- # of authoring tools;
- # of supporting platforms;
- # of repositories

Introduction

E-learning Standards

Standards can be defined as "documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose" (ISO, 2002).

Since the advent of personal computer, digital technologies have of course become increasingly common in education -- both in distance and classroom-based education and training settings. However, these technologies have typically been applied in ad hoc and divergent forms. Innumerable courses, course modules and systems for managing and delivering the course material, have been developed independently of one another, quite often at cost of significant effort. Moreover, such content and systems are often partly or even completely unavailable to interchange and/or interoperation. The concept of "Standard" in e-learning addresses those shortcomings.

For systems to interoperate they need to "understand" the data structures they share. Without a standardized specification each "data supplier" or "tools developer" would expect others to conform to its own data structure. This is much similar to a world wide web where each web site asks for a different browser to be enjoyed.

There are four major advantages in standard development and use, specifically in the field of e-learning:

- Durability – no need for modification as versions of system software change.
- Interoperability – operability across a wide variety of hardware, operating systems, web browsers and Learning Management Systems.
- Accessibility – indexing and tracking on demand.
- Reusability – possible modification and use by many different development tools.

These should apply to both the systems themselves and to the contents, data and processes they manage.

1. SCORM

1.1 Organization

The ADL (Advanced Distributed Learning) is a U.S. government-sponsored organization that researches and develops specifications to encourage the adoption and advancement of e-Learning. The purpose of the ADL is to ensure access to high-quality education and training materials.

The most widely accepted ADL publication is the ADL Shareable Content Object Reference Model (SCORM). The SCORM specification combines elements of IEEE, AICC and IMS specifications into a consolidated document that can be easily implemented. The ADL adds value to existing standards by providing examples, best practices and clarifications that help suppliers and content developers implement e-Learning specifications in a consistent and reusable way.

Details about the standard and all related documents can be found at http://www.adlnet.org/scorm/index.cfm

1.2 Goals of the Standard

Goals of this collection of specifications are:

- *Accessibility***:** the ability to locate and access learning content from anywhere and anytime.
- *Adaptability***:** the ability to tailor instruction to individual and organizational needs.
- *Affordability***:** the ability to increase efficiency and productivity by reducing the time and costs involved in delivering instruction.
- *Durability***:** contents will not easily become obsolete and will be preserved even when technology changes.
- *Interoperability***:** different tools and platforms communicate and work efficiently together.
- *Reusability***:** learning contents can be easily modified and used many times by different tools and platforms, so reducing costs.

1.3 Components

SCORM consists of three topics:
- Content Aggregation Model (CAM),
- Run-time Environment (RTE),
- Sequencing and Navigation (SN).
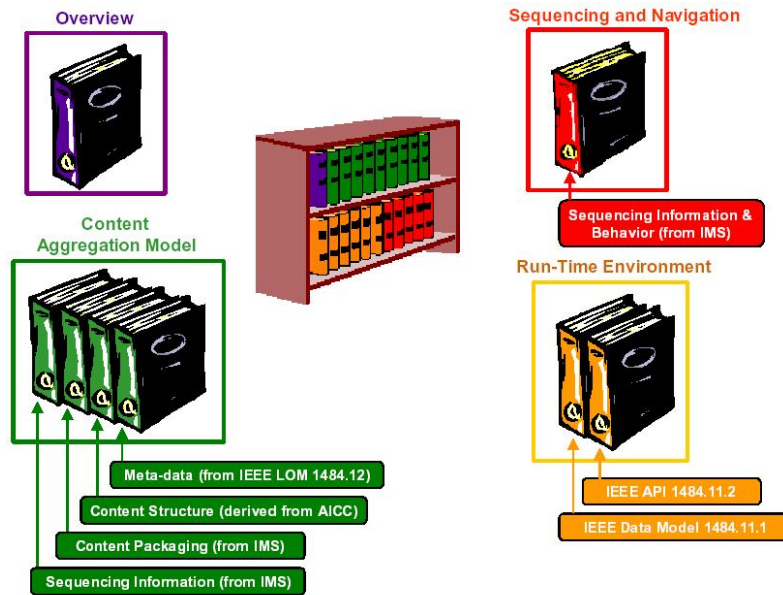
Content Aggregation Model

The Content Aggregation Model (CAM) consists of four components:
- Content Model,
- Content Packaging ,
- Meta-data,
- Sequencing and Navigation.
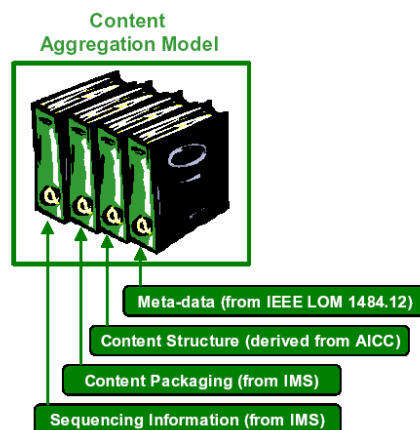
Content Model

The SCORM Content Model describes the SCORM chunks used to build a learning experience, such as a course or module.
Chunks are of three types:

- Asset,
- Sharable Content Object,
- Content Organization.

Asset represents the lower level of learning resource to present to a learner; it may be a Web page, an XML document, an JPEG image, etc…
More than one asset can be collected together to build larger assets.

A SCO is a collection of one or more Assets that utilizes the SCORM RTE to communicate with LMSs (Learning Management System).

A Content Organization defines structured units of instruction- activities- that represent the learning content (for instance a course may be divided in more module and a module in more lessons).

Content Packaging

Content Packaging defines the aggregation process for e-learning resources and a way to associate metadata to components.
SCORM Content Packaging consists of IMS Content Packaging (for details see IMS section) and the requirement to create a package as PIF (Package Interchange File), a compressed archive file in the format PKZip v2.04g(zip).

<div align="center">

Runtime Environment

</div>

SCORM RTE specifies how content should behave once it has been launched by the Learning Management System (LMS). It includes guidelines for the connection to a contents archive, the communication with it and the retrieval of content. In this way, RTE enables reusability of Learning Objects and interoperability between different LMSs.
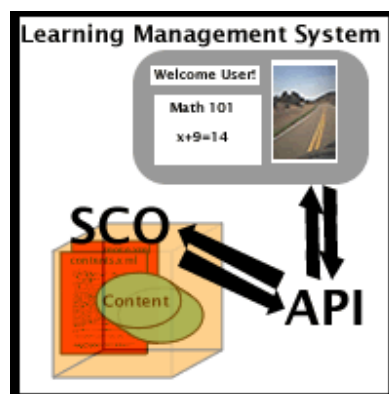
To achieve this, the developers of SCORM have also created a standard Application Programming Interface (API) that provides developers with a common language for implementing and exploiting Shared Learning Objects. The API allows the LMS to initialize and import data and to set and retrieve information defined in the SCO. Among other things, it is so possible for the LMS to track learners' progresses and to recall a user's position within the content.

Run-Time Environment consists of three components:
- Launch: The mechanism for defining a common way for LMSs to start Web-based learning resources. It defines the procedures and responsibilities for establishing the communication between the delivered learning resources and the LMS.
- Application Programming Interface: provides a communication mechanism that allows the SCO to communicate with the LMS. It is assumed that once the SCO is launched it can then exchange information with an LMS. Any communication between the LMS and the SCO is initiated by the SCO. There is currently no mechanism supporting the LMSs in initiating calls to functions implemented by a SCO.
- Data Model: A standard set of data elements used to define the information being communicated, such as the status of the learning resource. In its simplest form, the data model defines the elements that both the LMS and SCO are expected to "know" about. The LMS must maintain the state of required data elements across sessions. The purpose of establishing a common data model is to ensure that a defined set of information about SCOs can be tracked by different LMS environments.

The core of the SCORM RTE is actually its Data Model, which is derived directly from AICC CMI (Computer Managed Instruction) Data Model. AICC CMI Data Model defines sets of standardized data elements that can address most needs of information exchange between learning resources and LMSs.

A SCORM compliant LMS is required to implement the API (Application Program Interface), which consists of 8 functions:
- LMSInitialize()
- LMSFinish()
- LMSGetValue() (retrieves a data element from the LMS)
- LSMSetValue() (writes a data element to the LMS)
- LMSCommit() (called after any values are set to ensure the data is saved)
- LMSGetLastError()
- LMSGetErrorString()
- LMSGetDiagnostic()

This API is called API Adapter. All communication between the content and the LMS is handled by this adapter. It must reside in a window which is parent window or parent frame of the window containing the content. This means that the LMS may launch the content either in a new window or in a frameset.
The API Adapter may be implemented through an ECMAScript (JavaScript) object named "API" accessible though the DOM. The LMS provides an API instance.
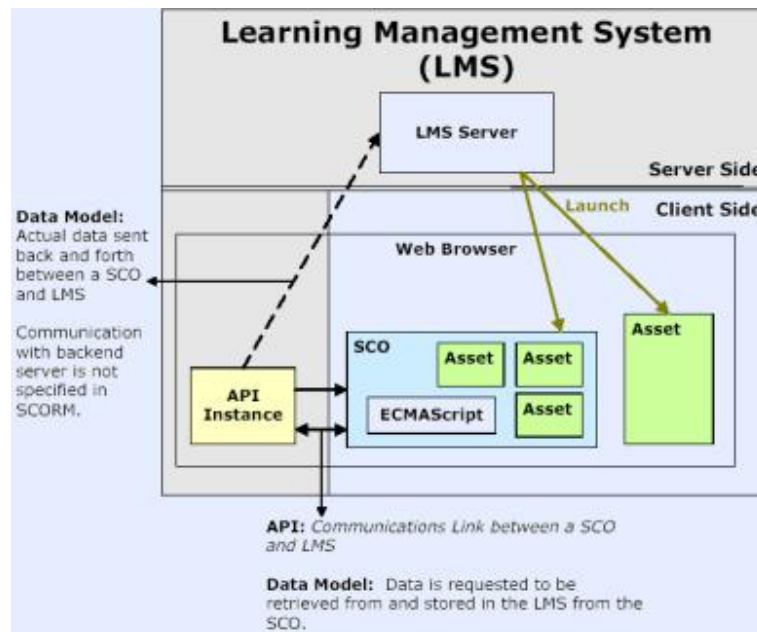
SCORM Conceptual Run-Time Environment.

The use of a common API fulfils many of the SCORM high-level requirements for interoperability and reuse. As we have discussed, it provides a standardized way for SCOs to communicate with LMSs. However, the way the LMS-provided API Instance communicates with the server-side component of the LMS is outside the scope of SCORM.

As it can be noticed, there are several terms that are used throughout SCORM in the above description: API, API Implementation and API Instance. The figure shown below describes the terms and their relationships to each other.
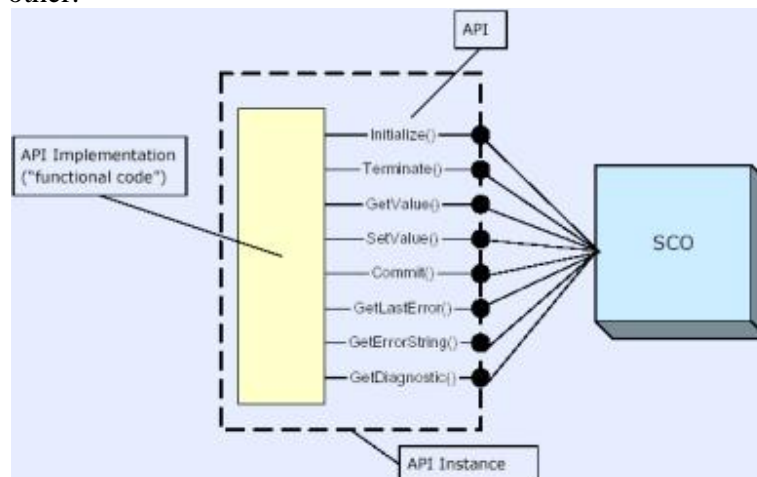
In its simplest terms, the API is merely a set of defined functions that the SCO can rely on.

An API Implementation is a piece of functional software that implements and exposes the functions of the API. How an API Implementation actually works should not matter to a SCO developer, as long as the API Implementation uses the same public interface and adheres to the semantics of such interface. On the other hand, the LMS needs only to provide an API Implementation that implements the functionality of the API and exposes its public interface to the client SCO.

An API Instance is an individual execution context and state of an API Implementation. The API Instance represents the piece of executing software that a single SCO interacts with during the SCO operation.

An API Request represents a single exchange of data between the SCO and the LMS. The objects involved in such exchange of data are those defined by the Data Model.
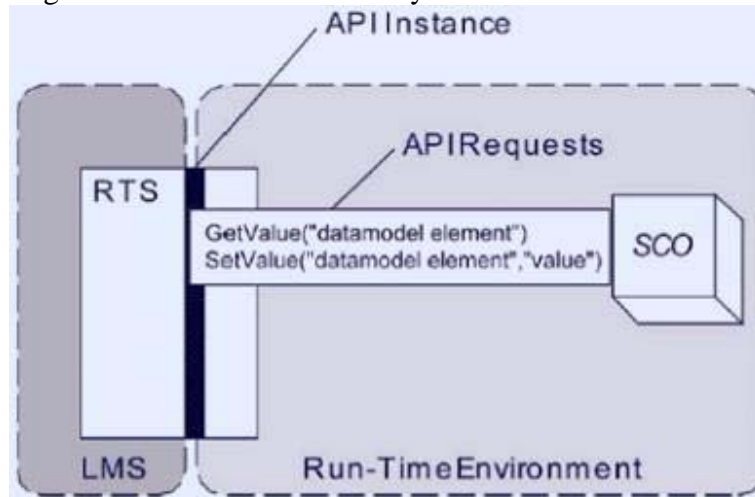
Using the data model with the API

For major details see SCORM RTE book at official SCORM web site.

Sequencing and Navigation

At design time, content developers may define different activities flows based on students' knowledge and skills. The SCORM SN defines a method for representing the activities flow defined by a teacher for different students.

The SCORM SN describes how a SCORM-conformant LMS interprets the sequencing rules expressed by a content developer along with the set of learner-initiated or system-initiated navigation events and their effects on the run-time environment.

1.4 Metadata

SCORM uses Learning Object Metadata (LOM) to describe each Content Model component (Asset, Sco and Content Organization) or package as a whole.

These metadata are defined in an XML file and may be stored within or outside the package.

The LOM Information Model is broken up into nine categories. Such categories are based on the definitions found in the model itself. The nine categories of meta-data elements are:

1. *General* category: can be used to describe general information about the SCORM Content Model Component as a whole.

2. *Life Cycle* category: can be used to describe features related to the history and current state of the SCORM Content Model Component and to those who have affected the component during its evolution.
3. *Meta-metadata* category: can be used to describe information about the metadata record itself (rather than the SCORM Content Model Component that the record describes).
4. *Technical* category: can be used to describe technical requirements and characteristics of the SCORM Content Model Component.
5. *Educational* category: can be used to describe the educational and pedagogical characteristics of the SCORM Content Model Component.
6. *Rights* category: can be used to describe the intellectual property rights and conditions of use for the SCORM Content Model Component.
7. *Relation* category: can be used to describe features that define the relationships between this SCORM Content Model Component and other targeted components.
8. *Annotation* category: can be used to provide comments on the educational use of the SCORM Content Model Component and information on when and by whom the comments were created.
9. *Classification* category: can be used to describe where the SCORM Content Model Component falls within a particular classification system.

(For details see LOM section)

1.5 Authoring tools

AltEd Global Project

AltEd Inc. provides an authoring tool named AltEd 4.0 plus, an IMSmanifest & Meta-data generator named AltEd SCORM Editor 1.2, and a Learning Content Management System(LCMS) named AltEd SCORM LCMS. All the products are based on SCORM standard.
It is not free.

http://www.alted.com/SCORM/index.asp

Design a Course Author

It is a tool which does not require to learn programming or scripting languages.
It provides four different types of questions with feedback and question timing.
It allows to create courseware that can be exported to a SCORM 1.2 or SCORM 2004 compliant LMS.

It is not free.

http://www.mindiq.com/elearning/dac/stand-alone.php

E-learning Consulting SCORM Tools

E-learning Consulting provides:

- **Manifest Maker -**A Free Dreamweaver extension to quickly and easily create a SCORM 1.2 Manifest.
- **SCORM Visualizer-** SCORM Visualizer lets you see how a course and a Learning Management System (LMS) use the SCORM 1.2 Runtime to communicate. It is not free.
- **Flash Course Development Toolkit –** This toolkit allows to create SCORM-based e-learning courses with Flash. It supports SCORM 1.2. It is not free.
- **Test Builder-** This tool allows to create SCORM-based tests and quizzes with true-false, multiple choice, fill-in-the-blank and matching questions. No programming is required.

http://www.e-learningconsulting.com/products/index.html

Exe 0.6 release

It is an open source SCORM/IMS authoring tool.
http://exelearning.org/

E-learning authoring tool

This tool lets you use your favourite HTML editor to create e-learning courses quickly and easily.
The Kit fully supports SCORM 1.2. Courses developed with the Kit will pass the SCORM conformance test. The Kit can be used to create content in multiple languages.
It is not free.
http://www.gold-software.com/eLearningAuthoringTool-review10077.htm

Lersus easyContent

LERSUS is a software product that allows creating interactive tutorials, computer training courses (CBT), and e-learning courses (WBT) according to the existing e-Learning standards.

LERSUS supports the creation of descriptive interactive tutorials intended for e-learning via Internet, local area network, or by means of a Learning Management System (LMS).

LERSUS allows creating tests to check the knowledge and including them into the interactive materials.

It supports IMS QTI and SCORM. It is not free.
http://www.lersus.com/

Macromedia Authorware 7.0

Authorware Learning Object Metadata Editor is available at http://www.macromedia.com/devnet/authorware/articles/metadata_editor.html. It allows to define metadata of SCORM content packaging in standardized format.
It is not free but It is possible to download a trial version.

Metadata Generator Pro

Metadata Generator Pro allows the user to create, import, and export SCORM 1.2 conformant metadata files. Users may also import, edit, and validate any existing SCORM 1.2 metadata file.
Limitations of free version: 5 user trial.
http://downloads-zdnet.com.com/Metadata-Generator-Pro/3000-2383-10297484.html

SCORM 1.2 Player

The SCORM Player is developed at Bolton Institute by Paul Sharples and Phillip Beauvoir.
The current version supports ADL SCORM 1.2 and is free.

http://www.reload.ac.uk/SCORMplayer.html

SCORM 1.2 Resource Kit

The Click2learn SCORM 1.2 Resource Kit is an evolving set of documents and demonstration applications.

The Kit includes:
- SCORM 1.2 Package Aggregator- A visual aggregator that assembles SCORM 1.2 content aggregation packages into larger SCORM 1.2 content packages. It also includes a user-friendly metadata editor for the metadata for the new aggregated package and limited on-line help to support the workflow.
- SCORMisizer - This "SCORMisizer" wizard will turn a single document or other digital resource into a SCORM 1.2 package in few easy steps. No XML editing is required.
- SCORM API Exerciser- The SCORM API exerciser is a tool to exercise and explore the SCORM API. It is a single HTML page, but it works only if launched as a SCO in a SCORM conformant runtime environment.

The kit is free.

http://academiaelearning.com/contenido/SCORM/cooking/index.html#aggregator

Visual Course Builder

The VCB tool has been developed from the ground up with the goal of supplying a powerful multimedia development tool with an intuitive interface that an average Windows user can operate after a short training session.

The VCB has the most common e-learning user interface features pre-designed such as built navigation, progress bar, menus, tracking (AICC / SCORM), graphic editors, interaction editors....

It is possible to deliver the content through any browser and without plug-ins unless external media that requires a plug-in.

It is possible to download a trial version.

http://www.maxit.com/content/view/48/222/

1.6 LMS supporting SCORM

Blackboard

This LMS is SCORM 1.2 compliant. It is a commercial product.
Blackboard is another popular tool to create an online learning course. It was created by Blackboard Inc. Blackboard is entirely web based and the users do not need to install client software. Blackboard offers a number of interesting features to the instructor and students. It allows all users to join the Blackboard communities where they may share their teaching and learning experiences. Blackboard has a student tracking feature which allows the instructor to monitor student progress. It also has assessment tools which help the instructor to create assignments and quizzes and automatically grade them.

For details see the brochure at page
http://www.blackboard.com/docs/AS/Bb_Learning_System_Brochure.pdf.

http://www.blackboard.com

Contento LMS

Contento supplies an integrated system for the assembly and distribution of training content and the management of users connected to the system.
It is particularly easy to use and does not require any specialized technical knowledge. It isn't free.

It is SCORM 1.3 compliant.
http://www.bitness.it/Company.htm

DataUfficio

This platform is Aicc, SCORM and Ims compliant and allows test delivery and management.
It offers these services:
- Forum,
- E-mail,
- On-line tutoring.

Courses registration isn't free and may be made both online and by fax.
It is possible to access a demo.

The function "Personal record-book" allows a user to know his/her complete situation (courses and module attended, etc..).
www.dataufficio.com

DLMS

It is a web-based system for courseware delivery implemented by Education spa.
It can deliver any kind of multimedia content conforming to the supported standard.
It offers:
- delivery environments
- collaborative environment (chat, forum),
- compatibility with the more popular e-learning standards,
- portability on different operating systems (Windows, Linux),
- student's personal area,
- user tracking.

It is SCORM/AICC/IMS compliant.
www.e-ducation.it/html/dlms.html

DoceboLMS

This LMS (known like SpaghettiLearning) is SCORM 1.2 compliant. It is open source.
It manages different formative models: from collaborative didactic to self-training.
DoceboLMS is written in Php and MySql, can work on whichever server (Linux, Windows, Mac OSX).
http://www.docebolms.org/

Dokeos

Dokeos is an open source LMS developed using PhP, Apache and MySQL technologies.
Dokeos is translated in 31 languages.

It allows the trainer to create educational content, to structure activities in learning paths, to interact with students and to follow their evolution through a reporting system.
http://www.dokeos.com/

Edo

It was implemented in 2004 by Didael spa, and is an interactive solution to learn English. It addresses all types of users. It allows to make interactive exercises, to see a video, etc…
Students can:
- verify own knowledge and choose start level (Placement Test),
- plan own study times (Study Planner),
- known own advancement state.

Teachers can:
- define courses,
- control access to resources,
- control students' activity.

It isn't free but it is possible to access it for 15 days through a registration to a "guided visit".
It' s SCORM 1.2 compliant.
www.didael.it

E-learning Consulting LMS

This LMS manages the delivery of self-paced, e-learning courses. It lets publish courses and place them in an online catalogue. The LMS provides online reports for each course and learner.
It is browser-based and fully supports SCORM 1.2.
It is not free but It is possible to download a 14-day trial version.

http://www.e-learningconsulting.com/products/learning-management-system.html

Inquisiq EX

This LMS is SCORM 1.2 and SCORM 1.3 (2004) Compliant.
It is not free. It is possible to download a 30 days trial version at page
 http://www.icslearninggroup.com/products/inquisiq/learning-management-system.htm

Ilias

It is an open source web-based LMS. It consists in tools for creation and learning of contents. It allows users to create, edit and publish learning and teaching material in an integrated environment with their normal web browsers.

ILIAS has reached SCORM 1.2 Conformance.
Some features:

- Learning environment with personal annotations, tests, glossaries, print function, search engine and download
- SCORM 1.2 and AICC compliance
- Course management system
- Communication features like mail system, forums and chat
- Support of metadata for all levels of learning objects
- User and system administration interface
- System languages: Albanian, Bulgarian, Chinese, Czech, Danish, Dutch, English, French, German, Hungarian, Italian, Polish, Portuguese, Romanian, Spanish, Serbian, Ukrainian.

www.ilias.de

LearnerWeb

LearnerWeb is 100% browser based with enterprise level scalability and support for emerging/open standards including AICC, SCORM, XML, SQL and Web Services.
LearnerWeb can manage and track any type of training an organization may offer including on-line, classroom, on-the-job, off-site seminars, lending library, and general learning resources at an unlimited number.
LearnerWeb conforms to the following standards/models: AICC 1, AICC 2 and SCORM 1.2.
It is not free.

http://www.maxit.com/content/view/30/224/

Learning Space

It is a complete and flexible platform, implemented by IBM. It allows courses delivery in both asynchronous and synchronous form through a browser such as Internet Explorer or Netscape.
Courses access is regulated by an internal registration process.
The student can:
- access subscribed courses through authentication,
- manage email through an internal platform service,
- send files (for instance in discussion forums),
- participate to online test sessions,
- exploit self-assessment activities,
- search a particular course content ,
- communicate with others users through chat, e-mail and forum.

The teacher can:
- plan courses to publish,
- define and publish tests,
- verify students performances.

LearningSpace is Ims and SCORM compliant.

It is a commercial product.
http://www.lotus.com/lotus/offering3.nsf/wdocs/learningspacehome

Moodle

It is an open source LMS and it is SCORM compliant.
Some features:
- forum
- easy update of course's contents
- personalization of the layout of specified courses
- the teacher only sets up access option to courses.

http://www.moodle.org

TrainCaster LMS

TrainCaster Learning Management System (LMS) is conformant to SCORM Version 2004. The learning management system has been tested using the ADL SCORM Conformance Test Suite Version 1.3.2.
It is not free but it is possible to download a 30 day trial version.

http://www.traincaster.com

TinyLMS

TinyLMS is a learning management system for SCORM compliant learning content.
In particular It is conformant to SCORM 1.2.
Tutorials generated with TinyLMS can be used online and offline.
Runtime requirements for tutorials are low: HTML, JavaScript and Cookies are all that is needed. No server-side software is needed.
TinyLMS supports layered organization of learning content in addition to the hierarchical organization specified by SCORM.
It is free.
http://www.randelshofer.ch/tinylms/download.html

WebCT

This LMS is SCORM 1.2 compliant. It is a commercial product.

http://www.webct.com

TopClass E-learning Suite

The TopClass e-Learning Suite includes both a Learning Management System and a Learning Content Management System that is SCORM Version 1.2.compliant.
It is not free.

http://www.vr.com.au/elearn_prod_lms.html

1.7 Repositories of SCORM resources

IntraLibrary

This web-based application is not free. It is designed to allow storage and retrieval of learning objects.
It allows IMS/LOM metadata editing, import/export of IMS/LOM metadata and export of Dublin Core metadata, import / export of learning objects conformant to IMS and SCORM specifications.

A demonstration is available at page http://dewey.intrallect.com/products/intralibrary/demo.htm.

Courseware Repository

The Courseware Repository is an 'Open Library' where it is possible to publish courses.
The courses can be used stand-alone or integrated with an ADL/SCORM conforming Learning Management Systems.
http://www.vusonline.com/elearning-appliance-interface/Content_Repository/Index.html

E-learning Repository

E-learning repository is a tool for the storage and retrieval of learning objects.
It includes tools for meta-data tagging and taxonomy browsing. Courses are stored with IMS / IEEE LOM meta-data. The repository is designed to use learning objects that adhere to the international specifications from IMS and SCORM.

http://www.teknical.com/products/elearning_repository.htm

2. IMS

2.1 Organization

The IMS (Instructional Management Systems) Global Consortium is an international USA association of vendors universities and implementers who focus on the development of XML-based specifications for learning resources. These specifications describe the key characteristics of courses, lessons, assessments, learners and groups.

Details about the standard and all related documents can be found at http://www.imsglobal.org

2.2 Goals of the Standard

IMS has two fundamental goals:
- define specific guidelines which guarantee interoperability between applications and services in e-learning;
- support guidelines application in international products and services.

2.3 Components

The IMS standard currently consists of the following subparts:
- **Meta-data** (the core elements of Learning material)
- **Content Packaging** (packaging of courses or parts of courses),
- **Question & Test Interoperability** (quizzes),
- **Learning Design**
- **Simple Sequencing**

Meta-data

IMS is based on LOM and Dublin Core metadata. For details see LOM and Dublin Core sections.

Content Packaging

The IMS Content Packaging Specification defines how to describe (with metadata) and organize learning resources in packages.
An IMS Content Package contains two major components:
- A special XML document called the manifest file (imsmanifest.xml) describing the content structure and associated resources of the package,
- The physical files making up the content package.
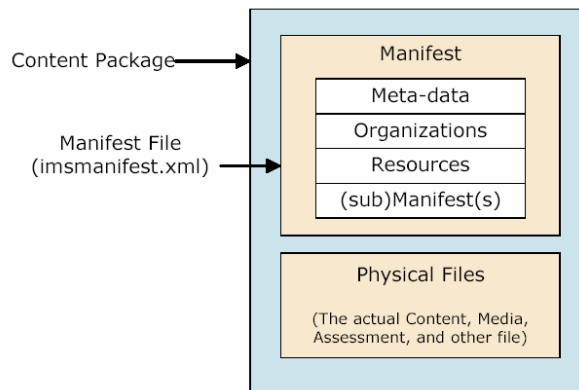
The Manifest is a description of the resources included in the package. It is divided in four sections:
- Meta-data – describes file manifest as whole,
- Organizations- describes one or more ways to organize contents,
- Resources- describes resources, i.e. the actual physical files contained in the package
- (sub)Manifest(s) – nested manifest(s).

Question & Test Interoperability

IMS QTI is a set of guidelines which describe how to share both single tests and complete evaluation tests and data. It allows representation of different type of questions, definition of feedback and compute of results.
The main types of supported questions are:
- Multiple response
- Single response
- True or false
- Fill in the blank.

Learning Design

The IMS Learning Design specification is a language for modelling units of study.
The purpose of the IMS LD is to provide a kind of abstraction for the description of learning processes, offering constructs suited for different pedagogical approaches. So the language enables pedagogical diversity to be supported through the implementation of a single engine. Moreover, it is designed to promote exchange and interoperability of e-learning materials, associating educational content with information describing its instructional strategy.

The IMS Learning Design Best Practice and Implementation Guide states that the core concept of the Learning Design Specification "is that regardless of pedagogical approach, a *person* gets a *role* in the

teaching/learning process, typically a *learner* or a *staff* role. In this role he or she works toward certain *outcomes* by performing more or less structured *learning* and/or *support activities* within an *environment*. The environment consists of the appropriate *learning objects and services* to be used during the performance of the activities. Which role gets which activities at what moment in the process, is determined by the *method* or by a *notification*."

IMS Learning Design Specification is based in the **OUNL EML** (Educational Modelling Languages), a notational system developed by the Open University of the Netherlands, intended to describe a variety of instructional models, course material and almost any kind of pedagogical model. It provides a generic and flexible "meta-language" that supports the definition of learning designs according to different pedagogical approaches. These are captured by describing a *method* and prescribing *activities* for participant *roles*. The method is essentially a *play* where the teaching/learning process is described by specifying which roles perform which activities in what order. A method can include *conditions* for directing learning activities, using *properties* from the learner's portfolios. Conditions and properties enable personalization and more elaborate sequencing.

Activities describe the actions the role has to undertake within a specified environment, which is composed of learning objects and services. Activities can be of the two types: learning and support activities.

A *learning activity* is directed at achieving a learning objective per individual user, and includes a single activity-description (which could be text, audio, or video).

A *support activity* is meant to facilitate a role performing learning activities.

More than one user can be assigned to a role during run-time.

The primary use of IMS LD is to model units of learning. Such modelling is performed by including an IMS Learning Design in a content package, which could be an IMS Content Package. A unit of learning, then, is an IMS Content package with the IMS LD included as an *organization* element in the package's manifest.

The learning design specification does not detail how the course material itself is represented; it rather specifies how to package up the overall information into a structure that is modelled on a play, with acts, roles (actors) and resources. So LD can replace the simpler Content Package adding more meaning to it.

There are three levels of LD:

- Level A – this is the simplest form and covers the activities, roles, acts and the environment involved in a learning design
- Level B – this introduces the notion of properties and conditions. It is at this level that LD becomes more useful, as it allows what-if conditions and storing properties (such as student performance) to allow for multiple paths through learning material
- Level C – this supports notification or messaging between system components, which allows for a more dynamic workflow and personalisation.

Simple Sequencing

The IMS Simple Sequencing specification describes how to organize the learning objects and present them to learners. The specification should guarantee major control about learning course.

It is possible to specify, for example, least score to reach in a test to continue to higher levels or to define mandatory material consultation before to choose the course between formative contents.

2.4 Metadata

IMS is based on LOM and Dublin Core metadata. For details see LOM and Dublin Core sections.

2.5 Authoring Tools

Alfanet QTI Tools

ALFANET QTI Tools consists of two tools: Author and Engine.
ALFANET QTI Author Tool provides course designers with a facility to define metadata associated with pre-exiting items (QTI compliant) and generate dynamic assessments (tests) using selection & ordering specification.

ALFANET QTI Engine is an interpreter of IMS-QTI Specification. It provides an integrated tool for the presentation of tests (assessments) integrating one or more questions (items), and their evaluation providing the corresponding results reports.
It is free.
http://rtd.softwareag.es/alfanetqtitools/index.html

Elicitus Content Publisher

This tool supports SCORM 1.2 and AICC.
It is possible to download a free evaluation copy usable for 30 days.
http://www.elicitus.com/

Exe 0.6 release

See SCORM authoring tool section.

IMS Assesst Designer

IMS Assesst Designer allows the authors to easily create, modify, delete and rearrange questions in assessments. This complies with open standards such as the IMS Project Question and Test Interoperability (QTI) Specification.
Assessment data can be exported and imported in zipped files according to IMS QTI specification.
It is free.
(http://www.gold-software.com/IMSAssesstDesigner-review14971.htm)

Lersus

See SCORM authoring tool section.

wizBuilder

Using this tool it is possible to create SCORM 1.2 compliant course content.
http://www.cyberwisdom.net/en/html/products/authoring_tool.asp

Komposer

It is an authoring tool for editing IMS Learning Design Content Packages, and plans to extend this to IMS Learning Design. It is free.
http://www.unfold-project.net:8085/UNFOLD/general_resources_folder/tools/komposer/komposeraug04

IMSEVIMSE: the IMS editor VIMSE

ImseVimse is a full-fledged graphical IMS metadata editor written in Java. It is free.

http://kmr.nada.kth.se/imsevimse/

RapidExam™ v2.5 Deluxe

It allows to create exams in a variety of different media formats, including EXM (native file format), XML, EXE, IMS QTI, and SCORM.
It is possible to download an evaluation copy for Windows.
http://www.xstreamsoftware.com/rapidexam_description.htm

2.6 LMS supporting IMS

ATutor

It is an open source web-based Learning Content Management System (LCMS) .
It allows to export content from ATutor as IMS/SCORM conformant Content Packages. Entire courses, or individual course units can be packaged for viewing or redistribution. Content from other compliant systems can be imported into ATutor.
Some features:

- Direct interface for administrator
- Forum and chat
- Contextual help
- Possibility to create a glossary
- Tracking of accesses.

In particular it is possible to create three different types of courses:

- Private (accessible only by users with an account and with teacher's approval)
- Protected (accessible only by users with an account)
- Public (accessible by anyone).

http://www.atutor.ca/atutor/index.php

Claroline

Claroline is a free application based on PHP /MySQL.
It is translated in 30 languages. The software is released under Open Source licence (GPL).
Claroline is compatible with Windows, Macintosh and Linux environments. It integrates the current standards like SCORM and IMS for a fast and simple integration of learning contents in learning sequences.
http://www.claroline.net

DataUfficio

See SCORM section.

Learning Space

See SCORM section.

OpenLMS

OpenLMS now supports English (core language) and Norwegian.
It is SCORM and Dublin Core compliant.
It is open source.

This system supports group collaboration, file sharing, distribution of lectures, distributing lecture notes to groups of students, and also facilitates collaboration for groups of students and teachers.
http://openlms.sourceforge.net/

2.7 Repositories of IMS resources

COL Learning Object Repository

It is an Open Source Learning Object Repository.
It includes learning objects which are IMS-compatible (IMS Schema 1.2.2).
http://www.col.org/lor/

E-learning Repository

See SCORM repository section.

Iconex Project

The ICONEX project is investigating interoperability and re-use of learning objects specifically through learning management systems. They are working with a learning management system called xtensor which has an IMS compliant repository. The repository manages the resource and holds IMS Metadata in a database.
http://www.iconex.hull.ac.uk/

SeSDL (Scottish Electronic Staff Development Library)

This repository of learning objects allows to upload content, tag it with IMS Metadata and then download it in an IMS content package.
http://www.sesdl.scotcit.ac.uk/

3. IEEE-LOM

3.1 Organization

The IEEE (Institute for Electrical and Electronics Engineers) is an international organization that develops technical standards and recommendations in technical areas ranging from computer engineering, biomedical technology and telecommunications, to electric power, aerospace engineering and consumer electronics, among others. IEEE LTSC (Learning Technology Standards Committee) developed the homonymous standard for learning material.
The standard specifies the syntax and semantics of Learning Object Metadata, defined as the attributes required to adequately describe a Learning Object.

The most widely acknowledged IEEE LTSC specification is the Learning Object Metadata (LOM) specification, which describe learning resources. The IEEE LOM standard has been approved in June 2002. Both IMS and ADL SCORM use the LOM elements and structures in their specifications.

Details about the standard and all related documents can be found at http://ieeeltsc.org/wg12LOM/

3.2 Goals of the Standard

A "learning object" designates something that is both an informational or interactive object and that has an evident educational application. Derived from the world of object-oriented programming, the term "object" connotes a resource that is modular, reusable, and capable of being integrated with other objects.

The word "learning" appearing in the term "learning object" implies that a learning object must have at least one specifiable educational purpose or context.

A learning object could conceivably take the form of any of a number of digital media. These media can include Java applets, Flash animations, and audio and video clips; but they can also take the form of more exclusively "informational" materials like Web pages, Web sites, PDF documents, or PowerPoint presentations. Any of these resources could conceivably be used by teachers to augment classroom or online lessons, by students for remedial or independent study, by instructors or designers to construct online courses, or by administrators for purposes of curriculum coordination.

Learning objects are building blocks that can be combined in nearly infinite ways to construct collections that might be called lessons, modules, courses, or even curricula.

There is no standard for the size (or *granularity)* of a learning object. Larger learning objects are typically harder to reuse, and smaller learning objects require more work to be combined and reused.

Goals of the Standard are:

- To enable learners or instructors to search, evaluate, acquire, and utilize Learning Objects.

- To enable the sharing and exchange of Learning Objects across any technology supported learning systems.
- To enable the development of learning objects in units that can be combined and decomposed in meaningful ways.
- To enable computer agents to automatically and dynamically compose personalized lessons for an individual learner.
- To enable, where desired, the documentation and recognition of the completion of existing or new learning & performance objectives associated with Learning Objects.
- To enable a strong and growing economy for Learning Objects that supports and sustains all forms of distribution: non-profit, not-for-profit and for profit.
- To enable education, training and learning organizations, both government, public and private, to express educational content and performance standards in a standardized format that is independent of the content itself.
- To provide researchers with standards that support the collection and sharing of comparable data concerning the applicability and effectiveness of Learning Objects.
- To define a standard that is simple yet extensible to multiple domains and jurisdictions so as to be most easily and broadly adopted and applied.
- To support necessary security and authentication for the distribution and use of Learning Objects.

3.3 Metadata

LOM defines a **Base schema** that specifies a hierarchy of data elements for learning objects metadata. At the top level of the hierarchy there are nine categories:

1. The **General** category groups the general information that describes the learning object as a whole.
2. The **Lifecycle** category groups the features related to the history and current state of this learning object and those who have affected this learning object during its evolution.
3. The **Meta-Metadata** category groups information about the metadata instance itself (rather than the learning object that the metadata instance describes).
4. The **Technical** category groups the technical requirements and technical characteristics of the learning object.
5. The **Educational** category groups the educational and pedagogic characteristics of the learning object.
6. The **Rights** category groups the intellectual property rights and conditions of use for the learning object.
7. The **Relation** category groups features that define the relationship between the learning object and other related learning objects.
8. The **Annotation** category provides comments on the educational use of the learning object and provides information on when and by whom the comments were created.
9. The **Classification** category describes this learning object in relation to a particular classification system.
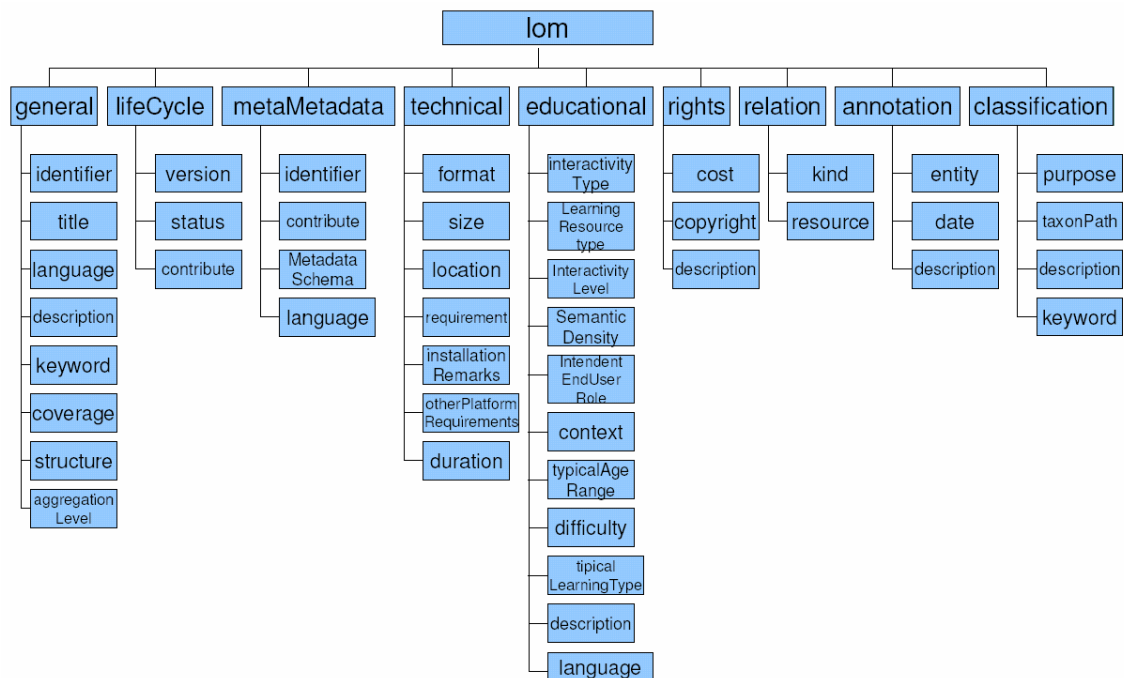
General

The *General* category contains the following child elements:
• <identifier> (represents a mechanism for assigning a globally unique label that identifies the learning object)
• <title> (represents the name given to the learning object)
• <language> (represents the primary human language or languages used within the learning object to communicate to the intended user.)
• <description> (represents a textual description of the learning object being described by the meta-data)
• <keyword> (should be used to define common keywords or phrases that describe the learning object)
• <coverage> (should be used to describe the time, culture, geography or region to which the learning object applies)
• <structure> (should describe the underlying organizational structure of the learning object such as atomic or collection)
• <aggregationLevel> (should describe the functional granularity of the learning object, such as lesson, module or course).

LifeCycle

The *Life Cycle* category contains the following child elements:
• <version> (should describe the edition of the Learning object)
• <status> (should describe the completion status or condition of the Learning object such as draft, final, etc..)
• <contribute> (should be used to describe those entities -i.e., people, organizations- that have contributed to the state of the Learning object during its lifecycle -creation, edits, reviews, publications, etc...).

Meta-Metadata

The *Meta-Metadata* category provides elements that describe the meta-data record itself and not the Learning object the record is describing. This category describes how the meta-data instance itself can be identified, who created the meta-data instance, how, when and with what references.

The <metaMetadata> element contains the following child elements:
• <identifier> (represents a mechanism for assigning a globally unique label that identifies the meta-data record that describes the Learning object.)
• <contribute> (should be used to describe those entities -i.e., people, organizations- that have affected the state of the meta-data (not the Learning object being described) instance during its development lifecycle)
• <metadataSchema> (represents the name and version of the authoritative specification used to create the meta-data instance)
• <language> (represents the language of the meta-data instance).

Technical

The Technical category describes all of the technical characteristics and requirements of the Learning object.
The <technical> element contains the following child elements:
• <format> (represents the technical datatype(s) of all of the components used in the makeup of the Learning object and should be a MIME)
• <size> (represents the size of the digital Learning object in bytes)
• <location> (is a string that specifies the location of the Learning object described by the meta-data.)
• <requirement> (expresses the technical capabilities necessary for using the Learning object.)
• <installationRemarks> (is used to represent any specific instructions on how to install the Learning object)
• <otherPlatformRequirements> (is used to represent information about other software and hardware requirements of the Learning object. This element should be used to describe requirements that cannot be represented or expressed with the other Technical elements.)
• <duration> (represents the time a continuous Learning object takes when played at intended speed. This element is useful for sounds, movies or simulations.)

Educational

The Educational category describes the key educational or pedagogic characteristics of the Learning object. This category is typically used by teachers, managers, authors and learners.
The <educational> element contains the following child elements:
• <interactivityType> (represents the dominant mode of learning supported by the Learning object such as expositive)
• <learningResourceType> (represents the specific kind of the Learning object such as exercise, simulation, slide etc…)
• <interactivityLevel> (represents the degree of interactivity characterizing the Learning object. Interactivity in this context refers to the degree to which the learner can influence the aspect or behaviour of the component)
• <semanticDensity> (represents the degree of conciseness of the Learning object.)
• <intendedEndUserRole> (represents the principal user(s) for which the Learning object was designed)

• <context> (represents the principal environment within which the learning and use of the Learning object is intended to take place such as higher education or school)
• <typicalAgeRange> (represents the age of the typical end user)
• <difficulty> (represents how hard it is to work with or through the Learning object for the typical intended target audience.)
• <typicalLearningTime> (represents the approximate of typical time it takes to work with or through the Learning object for the typical intended target audience.)
• <description> (should be used to comment on how the Learning object is to be used)
• <language> (the <language> element represents the human language used by the typical intended user of the Learning object).

Rights

The Rights category describes the intellectual property rights and conditions of use for the Learning object. This element should be used to describe any and all digital rights of the SCORM Component (cost for use, copyright, etc.).

The <rights> element contains the following child elements:
• <cost> (represents whether or not the Learning object requires some sort of payment.)
• <copyrightAndOtherRestrictions> (describes whether copyright or other restrictions apply to the use of the Learning object.)
• <description> (allows for comments on the conditions of use of the Learning object.)

Relation

The Relation category defines the relationship between the Learning object and other components, if any. The Relation element is allowed to be repeated. To define multiple relationships, one could create several instances of this category. If there is more than one target component, then each target should have a new relationship instance.
The <relation> element contains the following child elements:
• <kind> (describes the nature of the relationship between the Learning object and the target component identified by the Resource such as "is part of" or "is a version of")
• <resource> (element describes the target Learning object that this relationship references)

Annotation

The Annotation category provides comments on the educational use of the Learning object and information on when and by whom the comments were created. This category enables educators to share their assessments of Learning objects, suggestions for use, etc.
The <annotation> element contains the following child elements:
• <entity> (identifies the entity or entities that that have created the annotation)
• <date> (identifies the date the annotation was created)
• <description> (should be used to represent the contents of the annotation).

Classification

The Classification category describes where the Learning object falls within a particular classification system. The Classification category is typically used to link to a controlled vocabulary or classification system.

The <classification> element contains the following child elements:

• <purpose> (defines the purpose for classifying the Learning object)

• <taxonPath> (describes a taxonomic path in a specific classification system. Each succeeding level is a refinement in the definition of the proceeding level)

• <description> (contains a description of the Learning object relative to the stated Purpose - <purpose>- of the specific classification, such as discipline, idea, skill level, educational objective, etc.)

• <keyword> (contains keywords and phrases descriptive of the Learning object relative to the stated Purpose -<purpose>- of this specific classification, such as discipline, idea, skill level, educational objective, etc.).

3.4 Authoring Tools

LOM-Editor

At page http://www.multibook.de/lom/en/index.html a free LOM editor is available.

TreeLom

TreeLom is one of the LOM editing tools developed in .NET platform. It supports LOM Draft Standard. Currently two bindings models have been defined for LOM. They are XML and RDF binding of LOM. TreeLom is an editor that applies XML binding of LOM with a tree view editing functionality.
It is free.
[TreeLOM]
http://cebeciz.cu.edu.tr/tools/treelom/

Automatic Metadata Generation Framework

This open source framework is written in Java. It may be used as a web application or standalone.
If used standalone it allows to create metadata in LOM and DUBLIN CORE format.
http://www.cs.kuleuven.ac.be/~hmdb/amg/builds.php

3.5 LMS supporting LOM

Currently we haven't found anything yet.

3.6 Repositories of LOM resources

Currently we haven't found anything yet.

4. Dublin Core

4.1 Organization

The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the adoption of interoperable metadata standards. It develops metadata vocabularies.

Details about the standard and all related documents can be found at http://dublincore.org

4.2 Goals of the Standard

The mission of DCMI is to make it easier to find resources using the Internet through the following activities:

1. Developing metadata standards for discovery across domains,
2. Defining frameworks for the interoperation of metadata sets,
3. Facilitating the development of community specific metadata sets that are consistent with items 1 and 2.

4.3 Metadata

Dublin Core Metadata defines 15 optional elements to describe digital materials on the web. Elements are:

- Title (the name given to the resource by the creator or publisher),
- Creator (people or organizations that have contributed to create the resource),
- Subject (the topic of the resource, or keywords, phrases, or classification descriptors that describe the subject or content of the resource),
- Description (textual description of the resource),
- Publisher (the entity responsible for making the resource available in its present form, such as a publisher, a university department, or a corporate entity),
- Contributor (person(s) or organization(s) in addition to those specified in the creator element who have made significant intellectual contributions to the resource, but whose contribution is secondary to the individuals or entities specified in the creator element.),
- Date (date of creation or publication of resource),
- Type (category of resource),
- Format (data format),
- Identifier (string or number used to uniquely identify the resource such as URN),
- Source (the work, either print or electronic, from which this resource is delivered)
- Language (used language),
- Relation (contains identifier of a related resource),
- Coverage (time and space characteristics of resource)

- Rights (defines property rights and conditions of use).

The Dublin Core Metadata Initiative provides access to schemas defining DCMI term declarations represented in various schema languages such as XML and RDF.
http://dublincore.org/schemas/

4.4 Authoring Tools

DC-assist

DC-assist is a utility for metadata written in Javascript. It provides examples, comments and various links. It is free.

http://www.ukoln.ac.uk/metadata/dcassist/

DC-dot

It is a template to retrieve a Web page and automatically generate Dublin Core metadata, either as HTML <meta> tags or as RDF/XML.

It is available at http://www.ukoln.ac.uk/metadata/dcdot/

Dublin Core Metadata Template

A simple template to creation of Dublin Core metadata is available at page http://www.lub.lu.se/cgi-bin/nmdc.pl

EUC metadata tool

It is a template which allow to produce Dublin Core standard compliant metadata.

http://www.lub.lu.se/EUC/tool/euc.pl

Metadata Generator

With this tool it is possible to create meta tags based on the Dublin Core Specifications and to simply add information to fields.
http://www.artsit.unimelb.edu.au/metadata/

Meta Tag Editor

It allows editing of Dublin Core meta tags.

It isn't free. Some screenshots can be viewed at page

http://net-promoter.com/pagepromoter/help/meta_editor.html.

Mozilla Firefox Dublin Core Viewer Extension

This extension adds a button to the browser's status bar (and, optionally, the toolbar) to access an overview list of Dublin Core Metadata embedded in HTML/XHTML documents with META and LINK elements.

It is free and available at http://www.splintered.co.uk/experiments/73/

MyMetaMaker

It is a template which allows to produce Dublin Core standard compliant metadata to markup websites of Research Groups and Institutions. The output of the provided scripts produces the html source for the header of the webpage.

http://www.marenet.de/MareNet/mmm_groups.html

Reggie - The Metadata Editor

It is an applet which requires the use of the latest browser from either Netscape or Microsoft. http://metadata.net/dstc/

TagGen Office - Dublin Core Editor

It is a part of TanGen Offic,e a complete software for developing metadata. A trial version can be downloaded at page http://www.hisoftware.com/fact_sheetcc.htm.

The EULER Dublin Core Metadata Template

It is a template which allows to produce metadata compliant to Dublin Core standard.

http://euler.lub.lu.se/mdc/creator.cgi

Automatic Metadata Generation Framework

See LOM authoring tools.

4.5 LMS supporting Dublin Core

Currently we haven't found anything yet.

4.6 Repositories of Dublin Core resources

Currently we haven't found anything yet.

5. Ariadne Metadata

5.1 Organization

ARIADNE Foundation for is a not-for-profit Association and is involved in work related to technical specifications, most notably in the area of metadata.

Details about the standard and all related documents can be found at http://www.ariadne-eu.org/

5.2 Goals of the Standard

Standard's goals are:
- Simplicity,
- Understandability,
- Adaptability.

5.3 Metadata

Metadata are grouped into six categories:
- General (general information that describe the learning object such as title and language),
- Semantics (groups elements that describe the semantic classification of the learning object like the science type, main discipline, etc…),
- Pedagogical (describes the pedagogic and educational characteristics of the learning object such as semantic density and interactivity level),
- Technical (describes the technical requirements and characteristics of the learning object such as OS version),
- Indexation (describes the general information about the metadata itself of the learning object such as metadata creation date and creator),
- Annotations (describes people or organizations notes about the learning object).

5.4 Authoring Tools

Automatic Metadata Generation Framework

See LOM authoring tools.

5.5 LMS supporting Ariadne

Currently we haven't found anything yet.

5.6 Repositories of Ariadne resources

Ariadne Knowledge Pool System

Ariadne KPS contains metadata instances for learning objects. It's a distributed repository for learning objects; it encourages the share and reuse of such objects.

6. AICC

6.1 Organizations

Created in 1988, the AICC (Aviation Industry CBT Committee) is an international group of technology-based training professionals. They create CBT-related guidelines for the aviation industry. They publish a variety of recommendations and computer-managed instruction (CMI) guidelines.
The AICC's mission is to provide and promote information, guidelines and standards that result in the effective implementation of CBT and WBT.

Details about the standard and all related documents can be found at http://www.aicc.org

6.2 Goals of the Standard

The objectives of the AICC are as follows:

1. Assist airplane operators in development of guidelines which promote the economic and effective implementation of computer-based training (CBT) media.
2. Develop guidelines to enable interoperability.
3. Provide an open forum for the discussion of CBT (and other) training technologies.

6.3 Metadata

AICC/CMI Guidelines for Interoperability (CMI-001)

These guidelines are contained in CMI-001 document and describe the interoperability between CMI (Computer Managed Instruction) systems where a CMI System, in the terminology of AICC, is the equivalent of an LMS (Learning management System). Interoperability means that a system is able to manage CBT lessons with different origin and that a CBT course is able to exchange information with different CMI systems.

Guidelines address three aspects of interoperability of Computer Managed Instruction systems:

- Communication between a CMI system and a lesson
- Moving a course between different CMI systems
- Storing lesson evaluation data

A number of files correspond to each of these aspects where guidelines for the format and content are also described.

The AICC has identified seven files (some optional) that can be used to describe a course's content and structure. Additionally, the AICC guidelines define five levels of complexity in describing the course structure. Increasing the level of complexity should result in:

- Less effort to review and modify the CMI system after importing the data.
- More complete description of the designer's intended usage of the course material.

The level of complexity determines the number of files required and the amount of information required in each file. The following table briefly describes the contents or purpose of each file.

Course Description File	Information about the course as a whole including a textual description of the course, and general makeup of the course -- the number and type of elements.
Assignable Unit Table	Information about the assignable units (AUs) in the course. Each AU has its own record (or row in the table). The information includes the name of the AU, its ID, and the mastery score for that AU.
Descriptor Table	A complete list of every course element in the course including: AUs, Blocks, Objectives, Complex Objectives. It is used as the basic cross reference file showing the correspondence of system-generated IDs with user-defined IDs for every element.
Course Structure Table	The basic data on the structure of the course including all of the AUs and blocks in the course, showing how they are organized. Finally, it implies the order in which these should be taken.
Objectives Relationships File	Objectives have complex and variable relationships to other elements of a course. This file defines all of these relationships. This file is optional, depending on the level of the course description.
Prerequisite Listing	Sometimes it may be desirable to prevent a student from entering a lesson until he has met certain prerequisites. This file allows that sort of constraint to be placed on each block or AU in a course. There are three levels of complexity that may be used in describing prerequisites: - a single prerequisite AU or block to be defined for each element in the course - prerequisites to be defined in the form of a logic statement (with "ands" and "ors") - the definition of prerequisites for each mode (Review, Browse, Normal) for the lesson
Completion Requirements	While lesson and objective status is determined within the lesson by the logic designed into it, this is not true of blocks. Blocks are created specifically to describe a course structure. Similarly Complex Objectives are defined in terms of other structure elements. Therefore, block and complex objective status must be determined by the CMI system. The Completion Requirements file is designed to allow the explicit specification of when a block or objective is complete when it does not conform to the defaults for completion. It is essentially an exception file.

Data that can be communicated is subdivided in data elements. Such data elements are arranged hierarchically and are mainly related to the Communication Data Model and to the Course Structure Data Model specified by the standard. Files exchanged to describe a course content refer to such data elements, which in this way play a role similar to metadata.

The full list is quite long, so the interested reader is invited to look at the updated documentation provided on the AICC site (http://www.aicc.org) in the page devoted to white papers and technical reports.

6.4 Authoring Tools

Istruendo Kit Authoring

It is a module of the package Istruendo Authoring.
It provides tools to create AICC compliant contents.

It is not free.

http://www.fabbricadigitale.it/attivita/elearning-offerta-Authoring.html

Course Authoring Tool

The Course Authoring Tool, a quick, easy-to-use course-authoring tool, enables training managers and instructors to develop new courses or modify existing ones. It's web-based and compliant with AICC/SCORM standards for content management.

It's not free.

http://www.emind.com

Authoring Pro

Authoring Pro is an authoring and publishing software. It's universally interoperable with learning management systems based on SCORM, AICC and LRN standards.
It is not free but it is possible to download a trial version.

http://www.pro-ductivity.com/authoringpro/

AICC File Editor

This tool allows to open and edit any set of AICC descriptor files in a graphical interface. The program also contains explanations of each AICC field.
It is free.
http://www.leftbrainmedia.com/tools_intro.html

Lectora

Lectora is an AICC, LRN, SCORM conformant authoring tool.
It's not free but a trial version is available.
View http://www.lectora.com/product_info_overview.html for major details.

SyberWorks Web Author

It's a Web-based training course authoring tool. Using SyberWorks Web Author it is possible to create Web-based AICC/SCORM compliant training courses with audio, video, software demonstrations, and much more.

Some features:

- Easy to use
- Requires no programming knowledge
- Automatically builds pre-test, post-test, and lesson feedback sections
- Allows to insert, delete, re-design, and re-order lessons

It is not free but it is possible to download a trial version.

http://www.syberworks.com/product_sa.htm

6.5 LMS supporting AICC

Pathlore

Pathlore is an LMS certified by the AICC.
Pathlore LMS is a functionally rich, secure, web-based application that automates all processes associated with learning management administration.

http://www.pathlore.com/products/index.asp

University360 Learning Management System

University360 Learning Management System is SCORM and AICC compliant.

This platform works alone or as a component of University360 to help you create and manage e-learning solutions.

It is not free.

http://www.rwd.com/products/elearningmanagementsystems/university360lms/

WBT Manager

WBT Manager is a Learning Management System (LMS) designed exclusively to organize, deploy, and track web-based training content.
It's AICC compliant.
It's available in different languages.
It's possible to download a trial version.

http://www.ielearning.com/wbt/index.cfm

LearnerWeb

See SCORM LMS section.

DataUfficio

See SCORM LMS section.

Ilias

See SCORM LMS section.

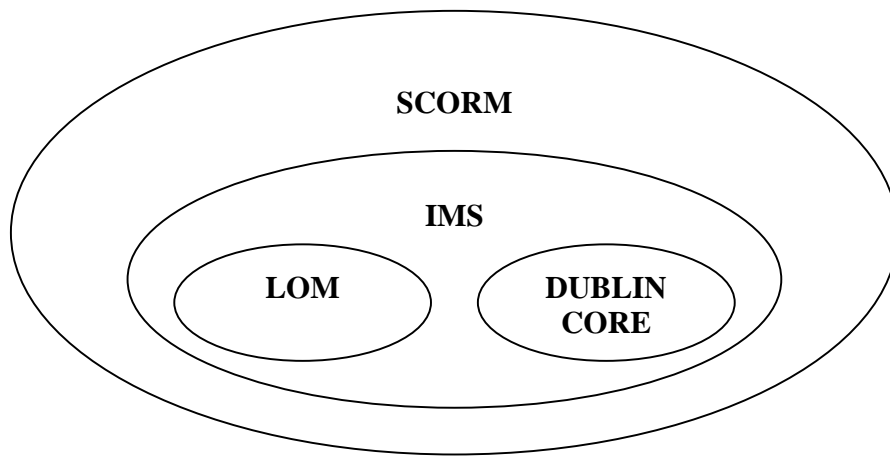# 6.6 Repositories of AICC resources

Currently we haven't found anything yet.

7. A comparison of standards

The analyzed standards are: SCORM, IMS, LOM, DUBLIN CORE and ARIADNE.

Metadata comparison:
- SCORM metadata include IMS metadata and LOM metadata.
- IMS metadata include LOM and DUBLIN CORE metadata.

More comparisons:

Standard	# Tools	# LMS	#Repositories
SCORM	5 free+ 8	14	3
IMS	5 free+4	5	3
LOM	2 free		
Dublin Core	9 free +2		
ARIADNE	1 free		1

Note: generally Dublin Core open source tools are templates.

We compare LOM standard (so its nine categories) with Dublin Core and ARIADNE:
- Educational category is present only in ARIADNE

- Classification category is present only in ARIADNE
- Annotation category is present only in ARIADNE
- Relation category is present only in DUBLIN CORE
- Rights category is present only in DUBLIN CORE
- Technical category is present in both standards
- metaMetadata category is present only in ARIADNE
- General category is present in both standards
- Lifecycle category is present only in DUBLIN CORE.

Generally, LOM standard has more tags that other standards, it's more general, the others more specific.

From an initial analysis, it seems that a lot of metadata is devoted to administering and reusing resources, but educational information should be extended, in order to provide more expressiveness in describing educational contexts and targets to which such resources could be addressed.

A more complete set of data elements is provided by AICC, but the use of such standard seems much more complex.

SCORM and AICC specify the communication interfaces (how learning resources exchange information):
- SCORM defines the RTE (including run time communication API),
- AICC offers two web-based approaches for courseware interface with an AICC-compliant LMS: the API implementation and the HACP (Hypertext AICC Communication Protocol) implementation. The API implementation is based on a JavaScript approach, while the HACP implementation is based on a traditional Hyper Text Markup Language approach.

The SCORM and AICC API specifications operate in the same manner. The main difference between the two views of API communication interface is that AICC specifies a larger set of data elements(data model), while SCORM uses a subset of such data.

8. Appendix A: LRN

The Microsoft Learning Resource iNterchange (Microsoft LRN) is a specific implementation (the first commercial implementation) of the Instructional Management Systems (IMS) content packaging specification, introduced in February 2000; it incorporates the IMS metadata and provides a toolkit.

It address the description, packaging, and runtime execution of learning resources in order to enable interchange and interoperability.

In particular, it is an implementation of IMS Content Packaging 1.1 and Metadata 1.2 specifications by the Global Learning Consortium. LRN also supports the Sharable Content Object Reference Model (SCORM) 1.2 reference model developed by the ADL.

LRN is an XML-based schema that defines course content, allowing organizations and e-learning providers to easily create, share and manage compatible online learning content.

A toolkit is provided for working with content packages and metadata using Microsoft's LRN implementation. It allows to transform PowerPoint presentations in SCORM (or IMS) compliant packages. Such toolkit is available for Windows 98, 2000 and XP, and includes an editor for creating IMS manifests (for content packaging and metadata). The most interesting part is the LRN Viewer: "The LRN Viewer generates a Dynamic HTML (DHTML) view of LRN content, adding a syllabus and navigation features, and provides a framework that supports the SCORM run-time environment. The LRN Viewer uses the Microsoft XML Parser with Extensible Stylesheet Language (XSL) to generate the DHTML view."

9. Appendix B: LMML

The Learning Material Markup Language (LMML) is developed in Germany.

The difference with the other standards presented in this document is that in this case metadata are not applied externally to the educational resources, like an header or an envelope, but rather are integrated into the content. This implies that it is possible to supply information for each conceptual component of the resource, fostering more powerful mechanisms of information retrieval and composition. The drawback is that LMML resources are written in XML language, though allowing insertion of pieces of multimedia content. Also existing learning contents in other electronic form, such as MS Word or MS Powerpoint documents, can be directly integrated in LMML modules as media objects.

LMML is a standard for the XML based description of e-learning, based on the conceptual Passau Teachware Model.
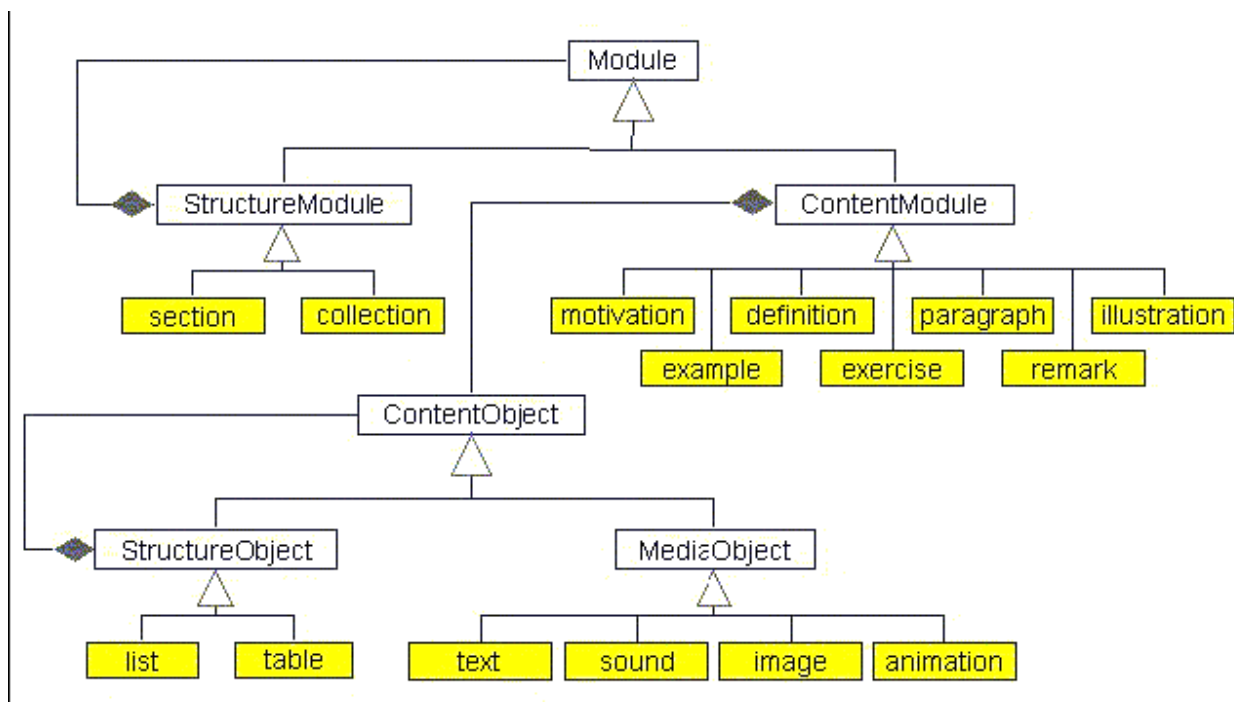
The e-learning content is organized in modules of different types. Each module may itself contain further modules and may be associated with other LMML modules in different ways. In this hierarchy the basic conceptual units are called Content Modules.

1. ContentModules: small units like definitions or examples. They are composed of ContentObjects, that can be structured using StructureObjects.

2. MediaObjects: specializations of ContentObjects, like text, images or sounds; they refer to the actual multimedia content, possibly declaring its type to indicate the visualization tool (e.g. an external plug-in.

3. StructureObjects: specializations of ContentObjects, they allow to structure any kind of ContentObject. For example, list and table elements can be recursively nested.

4. StructureModules: section and collection provide the ability to structure ContentModules. In general, also StructureModules can be recursively nested.

LMML provides a framework that can be instantiated to different domains of application, because it is extensible and can provide specific elements for different educational fields. As a matter of fact, the LMML framework is the XML binding of a general teachware model. The instantiations of specific fields are XML bindings of the appropriate models. For this reason, LMML actually consists of a family of XML markup languages for e-learning contents.

LMML describes only the teachware content, while visualization style is strictly separated and can be personalized. It is marked up with XML itself.
LMML is extendable with different XML-languages. For example, in mathematics contents LMML uses elements out of the language MathML to describe mathematical expressions and formulas.

Metadata

Each kind of LMML element is identified by a specific tag. Moreover, LMML provides a basic set of metadata, applicable to any element:
- general (such as author, title, catchwords),
- time-related (such as creation, modification, release and expiration date),
- educational (such as difficulty, abstraction).

Similar to the LMML elements that represent learning contents, also the metadata representing information about learning contents is adaptable to different specific fields. The target group as well as the level of details can be described, for example basics, structure or specialization. Didactical-pedagogical comments are also possible, in order to describe the content in a detailed way.

The modular structure of LMML teachware allows the fragmentation of a specific field content in any granularity, based on the identified conceptual units. This makes it possible to reuse LMML modules, starting from single multimedia elements like images, sound or animation, via content modules with a specific meaning like motivation, definition or simple text paragraphs, till to course paragraphs, course chapters or even whole courses. A course cannot only be reused as a whole course, but as a paragraph of another course, or a single paragraph can be reused as a whole course. Courses and paragraphs are both treated as modules in this modeling. This makes a smooth transition between the structure of a course and the structure of its contents. The separation of content and visualization further facilitates the flexible reuse.

LMML teachware shows a hierarchical structure, which reflects a possible hierarchical organization of the projects concerning the creation of learning material. Authors of media objects create multi media contents, e.g. images or animations; authors of content modules use them for instance in illustrations, motivations; the course designer finally composes the course. Besides this bottom-up procedure, it is also possible to create modules in a top-down way. Similar to the creation of traditional documents it is possible to create a classification at first. Afterwards the classification can be filled step by step with contents. The strict separation of content and visualization is the base allowing, for example, that a definition can be used within a LMML module and at the same time as part of a glossary. Glossaries or content indexes can be created automatically.

LMML documents are XML documents. The documents can be created with any text editor, though it is recommended to use special XML editors, possibly supporting the author with syntax highlighting, validating of LMML documents, project management, integration of XSL transformations and so on.

One possibility to publish LMML learning contents in the WWW is shown by the LMML tutorial.

A short introduction in the LMML tutorial is available at
http://daisy.fmi.uni-passau.de/pakmas/lmml/11/doc/en/html/

Official web site:http://www.lmml.de