Project no. 034595

SELF
Science, Education and Learning in Freedom

Instrument: SSA - Specific Support Action
Thematic Priority: IST-2005-2.5.5 – Software and Services

D2: Learning content standards and guidelines
Workpackage 3

Due date of deliverable: November 2006
Actual submission date: <November 28th, 2006>

Start date of project:
July 1st, 2006

Duration:
2 years

Universitat Oberta de Catalunya
David Megias

Revision [**final**]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

1 About this document

This document describes the digital format candidates to be used in the SELF platform. These digital formats are related to the learning object (LO) contents as it will be stored in the SELF repository. This document also describes some e-learning standards that can be used internally by the SELF platform to manage learning materials. In this version the LOM and SCORM standards and the IMS LD specification are presented as references for the development of the internal representation for learning content in the SELF Platform.

2 Introduction

This section describes the issues we have taken into account in order to choose and define the different standards which will be supported by the SELF platform. First of all, we must provide with an appropriate definition of the term "open standard" which will be basic for the development of the platform.

Prior to providing a definition of the term "Open Standards", the concept of "**format**" must be introduced [1]. First of all, it must be taken into account that computers store and transmit information in encoded form. These used to be very simple representations where certain numerical values stand for a certain character, for instance. And while their complexity has been increasing steadily with the power and complexity of computers, certain basic rules always apply. These codes used to store information are referred to as "formats".

The first important rule is that any such choice of encoding is an arbitrary, and not a natural choice. The number 33 may represent the letter 'a' or 'z' depending on the convention for this standard. There is no right way of doing this, there are only possible ways. The second important rule is that once data has been encoded in a certain format, it can only be read by software that implements such format, and implements it exactly. Even slight deviations from the conventions of the format will easily cause massive data corruption. A common and mostly harmless form of this is lost or broken formatting in text processing software. In the worst case the data will be unrecoverable.

All formats and protocols are fundamentally arbitrary in nature, but must be followed precisely for the data that was stored in them to be recovered. Customers who saved their data in one format quickly can find themselves unable to choose another vendor that was not able to implement the same format, or unable to implement it well enough. If the only way to migrate is to loose years of data there is a very effective vendor lock-in that practically makes it impossible to choose software according to its merits.

The solution of this situation is to choose those formats which are implemented as **Open Standards.** These formats are defined verifying some conditions which ensure that any software producer has enough information to implement them and, thus, guaranteeing the persistence of information. There are various definitions for what should or should not be considered an Open Standard. For example, as stated in the European Interoperability Framework for Pan-European eGovernment Services [2], the following are the minimal characteristics that a specification and its attendant documents must have in order to be considered an open standard:

- The standard is adopted and will be maintained by a not-for-profit organisation, and its ongoing development occurs on the basis of an open decision-making procedure available to all interested parties (*e.g.* consensus or majority decision).
- The standard has been published and the standard specification document is available either freely or at a nominal charge. It must be permissible to all to copy, distribute and use it for no fee or at a nominal fee.
- The intellectual property – *i.e.* Patents possibly present – of (parts of) the standard is made irrevocably available on a royalty-free basis.
- There are no constraints on the re-use of the standard.

Another definition is provided in the motion B103 [3] of the Danish parliament. According to this motion a format can be considered as an open standard as far as it is:

- well documented with its full specification publicly available,

- freely implementable without economically, politically or legal limitations on implementation and use, and
- standardised and maintained in an open forum (a so-called standards organisation) through an open process.

This is relatively similar to the definition of an Open Standard by the European Commission in its European Interoperability Framework, provided above. Following these guidelines, the formats supported by the SELF platform will be chosen according to the following criteria:

1. Is there a public specification of the format?
2. Are there any patents claimed on the format? If so, is the patent irrevocably available in a royalty free basis (or at a nominal charge)?
3. Is the format supported or created and maintained by some standardisation organism?
4. Is the format implemented by (one or more) free software applications?

The answer to these four questions will help us to choose the appropriate open standards for the representation of the SELF contents. According to these questions, the SELF platform will provide support to two kinds of formats:

1. If the answer to questions 1 – 4 is "yes", then the format is an open standard *de iure* and will be fully supported.
2. If the answer to the third question is "no", but the answer to questions 1, 2 and 4 is yes, then the format is not a pure *de iure* standard. However, it will be considered open enough for SELF purposes and it will be called a standard *de facto*.

Standards *de iure* will be always preferred for its use in the SELF repository, although standards *de facto* with an affirmative response to questions 1, 2 and 4 will be supported.

The SELF platform will provide a repository of contents and open standards will be used to store these contents. The use of open standards within the SELF platform must be considered at different levels:

- Firstly, all the digital documents stored in the SELF platform must be saved in some open standard format. If some material is to be included in SELF, it should be given (or converted to) in an open standard format.
- Secondly, for each document, many different output formats may be possible. For example, one may choose to view some document using a web browser or may want to produce a high quality printed version of some text. All of these output (or presentation) formats (which are not often editable) must also be given in an open standard format. The question whether the different presentations are stored (precomputed) or generated each time by the platform will be taken into consideration at the design and implementation stages. With the first alternative (precomputed output), the platform would require more storage space, whereas an output generated on demand would increase the computational load of the platform.
- Thirdly, all the documents in the SELF platform must be organised according to a complex structure. Contents will be provided in different languages, different educational levels, different objectives, and so on. Because of this, the organisation of the contents will be performed using some learning content standard. SELF will store the so-called "metadata", which is a set of complimentary information associated with each particular content.

The term “**Learning Object**” (LO) is defined in IEEE LOM [13] to denote any entity, digital or non-digital, that may be used for learning, education or training. In the SELF Platform context, a LO is a **digital entity** used in the SELF learning contents. A course, a lesson, a chapter, an image, an audio file are examples of learning objects in the SELF Platform. In principle, all LOs will be stored as independent files in the SELF repository. SELF must refer to some learning content standard to describe the LO structure of its contents.

The following two sections deal with the different standards which will be used in the SELF platform. Section 3 describes the digital formats use to store individual documents, whereas Section 4 discusses different reference standards and specifications which may be used to define the LO structure of the SELF repository.

Another relevant issue is that of “competences”. There have been different internationals educational moves towards the definition of courses and programmes in terms of competences rather than contents, such as the principles of the European Higher Education Area. At this stage, SELF has been conceived to provide contents and, thus, it is clearly a content-oriented project. However, this does not imply that competences are ignored by the project, since all contents are provided as a means for the accomplishment of some ability or competence. The inclusion of a description of competences in the LO stored in SELF repository will be thoroughly considered during the design stage of the platform, and specific data structures and tagging must be defined in order to support the inclusion of competences. The initiatives towards the creation of a unified specification for the description of the competences should be followed closely.

3 Digital formats

As discussed in the previous section, the choice of the standard formats used for individual LO in the SELF platform is based on the answer to four questions for each candidate format. The following subsections consider different format categories with some candidate formats for each category.

It must be taken into account that this Section is not intended to provide with an exhaustive list of all formats which will be chosen as SELF standards (the list of supported formats). This is just a first analysis of a few formats and the subsection 3.12 provides a first list of supported formats. Such a list will be open to other formats. In addition, the analysis performed in this document is subject to a further study by the legal experts of the SELF project which will be undertaken during the execution of Work Package 4 (Licensing). Although the team of WP4 has also contributed to this document, it is possible that a deep study of the patent claims or licensing issues related to some of the standards accepted in the list of supported format (subsection 3.12) leads to the conclusion that such a format is not open enough for the purposes of the SELF consortium. Hence, the discussion provided in this document is not closed and may be subject to revision under the direction of the learning content standards executive committee.

In short, this Section presents the process according to which a tentative list of SELF-supported formats are chosen, which is based on the answer to the four questions detailed in Section 2. Such a process will be undertaken the same way in the future whenever a new format is considered for inclusion in the SELF-supported list.

3.1 Unformatted or plain text

This is the most basic form of text which is given as is, without any formatting information (such as bold face, italic, underlining, etc.) The following table shows the answers of the four relevant questions for two selected formats:

Format	**Public specification?**	**Patented?** **If so, royalty free?**	**Standardisation organism?**	**Implemented by a free software application?**	***De facto / de iure* standard?**
ASCII (American Standard Code for Information Interchange)	Yes	Not patented	Yes, American National Standards Institute (ANSI)	Yes (for example gedit)	*De iure*
ISO/IEC 8859	Yes	Not patented	Yes, International Organisation for Standardisation (ISO) / International Electrotechnical Commission (IEC)	Yes (for example gedit)	*De iure*
Unicode	Yes	Not patented	Yes, Unicode consortium	Yes (for example gedit)	*De iure*

3.2 Formatted text

As defined in Wikipedia [4] **Formatted text**, **styled text** or **rich text**, as opposed to plain text, has styling information beyond the minimum of semantic elements: colours, styles (boldface, italic), sizes and special features (such as hyperlinks). Many different applications have been developed to help writing formatted text, and most of these applications work with their own formats. However, some of these formats have been standardised and, thus, are candidates for being accepted as SELF formats. The following table shows the answers of the four relevant questions for two selected formats:

Format	**Public specification?**	**Patented?** **If so, royalty free?**	**Standardisation organism?**	**Implemented by a free software application?**	***De facto* / *de iure* standard?**
Open**D**ocument Format for **T**ext processing (ODT)	Yes	Patented by Sun Microsystems Inc., but Sun is irrevocable covenant not to enforce any of its enforceable U.S. or foreign patents against any implementation of its specification	Yes, Open Artwork System Interchange Standard (OASIS)	Yes (OpenOffice.org 2.x and many others)	*De iure*
DocBook (XML/SGML)	Yes	Not patented	Yes, OASIS	Yes (for example Emacs in xXML mode)	*De iure*

3.3 Scientific or mathematical text

In fact, this could be considered a subcategory of formatted text, but the edition of mathematical formulae is important enough to consider it as at a different category. Here is the table for the three selected formats:

Format	**Public specification?**	**Patented?** **If so, royalty free?**	**Standardisation organism?**	**Implemented by a free software application?**	***De facto* / *de iure* standard?**
TeX / LaTeX	Yes	Not patented	No	Yes (LaTeX: LaTeX Project Public License 1.2)	*De facto*
Open**D**ocument Format for **F**ormulas (ODF)	Yes	Patented by Sun Microsystems Inc., but it is the same case as ODT (see Section 3.2).	Not yet. It is an extension of the OpenDocument format which will be included in the OpenDocument standard (OASIS)	Yes (OpenOffice.org 2.x)	*De iure* when it is included in the Open document format of OASIS
MathML (XML)	Yes	Not patented	Recommendation of the World Wide Web Consortium (W3C)	Yes (many, for example OpenOffice.org and Mozilla)	MathML is a recommendation of the W3C. It can be considered standard *de iure* as far as it is XML-based

Note that one of the formats introduced in this table is TeX / LaTeX. It must be pointed out that TeX / LaTeX will be fully supported by the SELF platform **not only for mathematical processing, but also as a standard for formatted text**. However, since one of the main advantages of TeX / LaTeX is to produce high quality mathematical content, we have decided to classify this format in the scientific text category.

3.4 Raster images (bitmaps)

A raster graphics image, digital image, or bitmap, is a data file or structure representing a generally rectangular grid of pixels, or points of colour, on a computer monitor, paper, or other display device [5]. The following formats have been analysed for raster images:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
JPEG (**J**oint **P**icture **E**xpert **G**roup) lossy compression	Yes	There is a patent claimed by Forgent Network, but it expires in 2006	Yes, ISO/IEC	Yes (many, for example the Gimp)	*De iure*
Portable **Bit**map (PBM), **P**ortable **Pix**map, (PPM), **P**ortable **Gray**map (PGM), **P**ortable **A**ny**m**ap (PNM)	Yes	Not patented	No	Yes (many, for example the Gimp)	*De facto*
Windows **Bitmap** (BMP)	Yes	Not patented	No (Microsoft, proprietary)	Yes (many, for example the Gimp)	*De facto*
Portable **N**etwork **G**raphics (PNG)	Yes	Not patented	Yes, ISO/IEC	Yes (many, for example the Gimp)	*De iure*
Graphics **I**nterchange **F**ormat (GIF)	Yes (proprietary)	Not patented. Patents have expired in US, Japan and Europe	No (Compuserve proprietary)	Yes (many, for example the Gimp)	*De facto*

3.5 Vector graphics

According to Wikipedia [6] vector graphics (also called geometric modelling or object-oriented graphics) is the use of geometrical primitives such as points, lines, curves, and polygons, which are all based upon mathematical equations to represent images in computer graphics. It is used by contrast to the term raster graphics, which is the representation of images as a collection of pixels (dots). In this case, we have found two formats which can be considered standard:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
Scalable **V**ector **G**raphics (SVG, XML for describing 2D vector graphics)	Yes	Not patented	Yes, W3C	Yes (web browsers, the Gimp, Blender)	*De iure*

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
Open**D**ocument for **G**raphics (ODG)	Yes	Patented by Sun Microsystems Inc., but it is the same case as ODT (see Section 3.2).	Yes, OASIS	Yes (OpenOffice 2.x)	*De iure*

3.6 Video (Motion images)

A video format describes how one device sends a video pictures to another device, such as the way that a DVD player sends pictures to a television, or a computer to a monitor. More formally, the video format describes the sequence and structure of frames that create the moving video image [7]. The following formats have been taken into account:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
OpenEXR	Yes	Not patented	No	Yes (for example Blender)	*De facto*
Theora	Yes	Patented by On2, but royalty-free	No	Yes (for example FreeJ)	*De facto*
Resource **I**nterchange **F**ile **F**ormat (RIFF) **A**udio **V**ideo **I**nterleave (AVI)	Yes	Not patented	No (Microsoft proprietary)	Yes (for example Mplayer)	*De facto*

3.7 Printer-oriented formats

This section is devoted to formats which are designed in order to produce high quality printed version of documents. These formats are usually just final (presentation) and are not very suited for editions or modification. Two formats have been considered in this category:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
Portable **D**ocument **F**ormat (PDF)	Yes	Patented by Acrobat Systems Inc., but royalty-free	Yes, ISO PDF/X: ISO 15929, ISO 15930 PDF/A: ISO 19005-1 (pending)	Yes (for example OpenOffice.org and Xpdf)	*De iure*
Post**S**cript (PS)	Yes	Not patented. Includes LZW compression, but all known patents (US,	No	Yes (for example LaTeX)	*De facto*

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
		Canada, Japan, Germany, France, Britain and Italy) expired in 2004.			

It must be taken into account that the latest versions of PDF can contain Adobe proprietary extensions. Also, tools like OpenOffice.org and xpdf only support a subset of the PDF definition. The Legal Group of the SELF project will help us define which specific PDF subset and version requirement will be supported by SELF. This analysis will be performed taken into account licensing issues and the different PDF specifications and will require a deep study of the documentation about the PDF format.

3.8 Hypertext

In computing, hypertext is a user interface paradigm for displaying documents which, according to an early definition (Nelson 1970), "branch or perform on request." Hypertext is a way of organizing material that attempts to overcome the inherent limitations of traditional text and in particular its linearity [8]. Just one format is analysed in this category:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
Hyper**T**ext **M**arkup **L**anguage (HTML)/ e**X**tensible **H**yper**T**ext **M**arkup Language (XHTML)	Yes	Not patented	Yes, ISO/IEC HTML: ISO/IEC 15445:2000 (subset of HTML 4.0)	Yes (OpenOffice.org 2.x and many others)	*De iure*

3.9 Presentations

Presentation is the process of presenting the content of a topic to an audience. Presentation software such as Microsoft PowerPoint, Keynote or OpenOffice.org Impress is often used to illustrate the presentation content [9]. A format has been analysed in this category:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
Open**D**ocument Format for **P**resentations (ODP)	Yes	Patented by Sun Microsystems Inc., but it is the same case as ODT (see Section 3.2).	Yes, OASIS	Yes (OpenOffice 2.x)	*De iure*

3.10 Audio

An audio format is a medium for storing sound and music. The term is applied to both the physical medium and the format of the content – in computer science it is often limited to the audio file format, but its wider use usually refers to the physical method used to store the data [10]. The following table presents two different formats which can be considered as SELF standards:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
Vorbis	Yes	Not patented	No	Yes (Vorbis Ogg ACM Codec 1.0)	*De facto*
Free **L**ossless **A**udio **C**odec (FLAC)	Yes	Not patented	No	Yes (FLAC codec)	*De facto*
RIFF Windows Audio (WAV)	Yes	Not patented	No (Microsoft & IBM proprietary)	Yes (for example Audacity)	*De facto*

3.11 Others

Finally, this subsection provides with an overview of some formats which are not included in the categories provided above.

Some of these formats are for data compression. In computer science **data compression** or **source coding** is the process of encoding information using fewer bits (or other information-bearing units) than an unencoded representation would use through use of specific encoding schemes. Compression helps reduce the consumption of resources. The design of data compression schemes involve trade-offs between various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompress the data [11].

The **Extensible Markup Language** (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data. In other words, XML is a way of describing data **[12]**.

The following formats have been considered as standards:

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
7z	Yes	Not patented	No	Yes (7-Zip)	*De facto*
Tape **ar**chive (tar)	Yes	Not patented	Yes, ISO/IEC 9945 (**P**ortable **O**perating **S**ystem **I**nterface for Uni**X**, POSIX)	Yes (GNU tar)	*De iure*
gzip	Yes	Not patented	Yes, Internet Engineering Task Force (IETF: RFC 1950, 1951 and 1952)	Yes (Gzip)	*De iure*

Format	Public specification?	Patented? If so, royalty free?	Standardisation organism?	Implemented by a free software application?	*De facto / de iure* standard?
E**X**tensible **M**arkup **L**anguage (XML)	Yes	The company Scientigo publicly asserted that two of its U.S. patents which cover the transfer of "data in neutral forms" apply to the use of XML, but some legal experts believe it would be difficult for Scientigo to enforce its patents through litigation.	Yes, W3C	Yes (many editors and browsers)	*De iure*

3.12 Supported formats

According to the analysis provided in the previous subsections, the first list of supported formats is provided below. The preferred formats (open standards *de iure*) are written with boldface fonts and underlined:

Category	*Supported formats*	*Category*	*Supported formats*
Unformatted text	**<u>ASCII</u>**, **<u>ISO 8859</u>**, **<u>Unicode</u>**	**Formatted text**	**<u>ODT</u>**, **<u>DocBook</u>**
Scientific text	**<u>ODF</u>**, **<u>MathML</u>**, Tex/LaTeX	**Raster images**	**<u>JPEG</u>**, **<u>PNG</u>**, PNM, GIF, BMP
Vector images	**<u>SVG</u>**, **<u>ODG</u>**	**Video**	OpenEXR, Theora, RIFF AVI
Printed-oriented	**<u>PDF</u>**, PS	**Hypertext**	**<u>HTML</u>**, XHTML
Presentation	**<u>ODP</u>**	**Audio**	Vorbis, FLAC, RIFF WAV
Others	**<u>tar</u>**, **<u>gzip</u>**, **<u>XML</u>**, 7z		

4 Learning content standards and specifications

This section discusses three alternatives which may be used as open standards or specifications to define the internal representation of the LO in the SELF repository, namely, the Learning Object Meta-data Standard (LOM), the Shareable Content Object Reference Model (SCORM) and IMS Learning Design (IMS LD) which are described in the following three subsections.

4.1 LOM

IEEE 1484.12.1 – 2002, Learning Object Meta-data Standard (LOM) is an international, open and multi-part standard that specifies the structure of meta-data to describe learning objects [13]. A learning object for LOM is any entity, digital or not, that may be used for learning, education or training.

The Standard defines a conceptual data schema (LOM v1.0 Base Schema) for a learning object meta-data instance, that is composed of data elements. Data elements are grouped into categories:

1. **General**: Groups the general information that describes the learning object as a whole.
2. **Lifecycle**: Groups the categories related to the history and current state of the learning object and those who affected it during its evolution.
3. **Meta-Meta-data**: Groups information about the meta-data instance itself.
4. **Technical**: Groups the technical requirements and technical characteristics of a learning object.
5. **Educational**: Groups the educational and pedagogic characteristics of a learning object.
6. **Rights**: Groups the intellectual property rights and conditions of use for a learning object.
7. **Relation**: Groups features that define the relationship between a learning object and other related learning objects.
8. **Annotation**: Provides comments on the educational use of the learning object and provides information on when and by whom the comments where created.
9. **Classification**: Describes a learning object in relation to a particular classification system.

LOM does not restrict the type of file format of learning objects. Data element 4.1 within Technical category, **Format**, specifies the data types of all the components (1 or more) a learning object. For digital data types MIME is used. For non digital data types, “non-digital” string value is set in this element.

LOM provides a way to describe the type of learning object related to its educational use. Data element 5.2 within Education category, **Learning Resource Type**, contains the specific kind of the learning object. The value space for the data element provides the following alternative values: *exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment,* and *lecture*. This set of values may not be enough to specify all resource types in the SELF project. New values can be added to the Learning Resource Type vocabulary, but, as said in IEEE 1484.12.03 (LOM XML Binding), LOMv1.0 base schema vocabulary data type extension is allowed. If vocabularies (set of values for a data element) are extended, then the source of additional vocabulary tokens should be identified and shall not be LOMv1.0

IEEE 1484.12.3 specifies how LOM records should be represented in XML [12]. The creation of meta-data for learning objects is usually equivalent to create XML files that contain the data elements of the meta-data instance. There are some tools to create and edit LOM meta-data [15]. Some of them are below:

Tool	***Link***	***License***
eRIB Metatagging Tool	http://demo.licef.teluq.uquebec.ca/eRIB/	LGPL
LomPad	http://helios.licef.teluq.uquebec.ca:8080/LomPad/en/index.htm	GPL
RELOAD	http://www.reload.ac.uk/index.html	MIT Open Source License

The creation of meta-data with any of these tools can require the selection of a profile because there are different meta-tagging standards. If there is not the IEEE LOM profile it's possible to generate IEEE LOM meta-data using IMS LRM profile [16]. IMS LRM (Learning Resource meta-data) v1.3 is aligned to LOM standard and this is because it's equivalent to refere it as LOMLOM. IMS LRM v1.3 provides a Best practice and Implementation Guide, and a XSL transform that can be used to migrate from earlier versions of IMS LRM XML binding to IEEE XML LOM binding.

Some developers at the Universidad de Extremadura are working on a contribution to the learning management system Moodle [17] to generate meta-data to describe the courses as LOM metadata.

The LOM standard can be implemented using XSL transforms. No specific player for LOM meta-data is known.

About multilingual environments, meta-data instance contains some data elements name **Language**:

- Data element 1.3, under General category, specify the primary human language or languages used within the learning object to communicate to the intended user.
- Data element 3.4, under Meta-Meta-data category, specify the language of a meta-data instance.
- Data element 5.11, under Educational category, specifies the human language used by the typical intended user of the learning object.

The use of LOM in a multilingual environment must be analysed because it seems that LOM is not enough to create a multilingual repository of learning objects. The platform should implement a system to select instances of learning objects from a specific language. This learning objects can have an aggregation relation with another learning object, the generic learning object. Or the software can have a database that relates data in a language to learning objects in the repository.

To store a learning object a XML structure can be used. IEEE 1484.12.3 specifies how LOM records should be represented in XML [14]. Furthermore, IEEE 1484.12.4 specifies how LOM records should be represented in RDF [18].

4.2 SCORM

Shareable Content Object Reference Model (SCORM) is a s a collection of standards and specifications for web-based e-learning developed by the Advanced Distributed Learning Initiative

(ADL) [19], launched by the Department of Defense (DoD) of the United States and the White House Office for Science and Technology Policy (OSTP).

SCORM describes a **Content Aggregation Model (CAM)** and a **Runtime Environment (RTE)** for instructional objects to support adaptive instruction based on a learner's goals, preferences, prior performance and other factors. SCORM also describes a **Sequencing and Navigation** model for dynamic presentation of content based on learner needs [20].

In SCORM a Learning Management System (LMS) is a suite of functionalities to deliver, track, report on and manage learning content, learner progress and learner interactions. LMS is a server-based environment that has the capacity for managing and delivering content to learners. SCORM does not define the LMS but the interface points between instructional content and LMS environments.

The following are underlying standards for SCORM:

- IEEE Data Model For Content Object Communication [21].
- IEEE ECMA Script Application Programming Interface for Content to Runtime Services Communication [22].
- IEEE Learning Object meta-data (LOM) [13].
- IEEE Extensible Markup Language (XML) Schema Binding for Learning Object Meta-data Model [14].
- IMS Content Packaging [23].
- IMS Simple Sequencing [24].

SCORM integrates technology developments from groups such IMS, AICC, ARIADNE and IEEE LTSC.

The Content Aggregation Model defines:

- Meta-data from IEEE LOM 1484.12.
- Content Structure derived from AICC [25].
- Content Packaging from IMS.
- Sequencing Information from IMS.

The Runtime Environment uses:

- IEEE Data Model 1484.11.1 [21].
- IEEE API 1484.11.2 [22].

The Sequencing and Navigation model defines:

- Sequencing Information and Behaviour from IMS.

The SCORM CAM (Content Aggregation Model) Book [26] describes the components used in learning experience, how to package them to share between systems, how to describe them to enable search and discovery, and how to define sequencing information for the components. SCORM CAM is made up of:

- Content Model: Nomenclature defining the content components of a learning experience. Content Model is made up of Assets, Sharable Content Objects (SCOs), Activities, Content Organization and Content Aggregations.

  - Asset is the basic building block of a learning resource. It is an electronic representation of media. In general, it is any piece of data that can be viewed by a learner using a web client (text file, audio file, image,...). An asset can be built from one or more assets. Asset can be described with meta-data to allow for search and discover within repositories.
  - Sharable Content Object (SCO) is a collection of one or more Assets that represent a single launchable resource that uses the SCORM RTE to communicate with an LMS. This is the lowest level of granularity of a learning resource that is tracked by an LMS using the SCORM RTE Data Model. SCOs should be independent from its learning context to be reusable. A SCO can be described via meta-data.
- Content Packaging: Defines how to represent the intended behaviour of a learning experience (Content Structure) and how to aggregate activities of learning resources for movement between different environments (Content Packaging). The purpose of Content Packaging is to provide a standardized way to exchange learning content between LMSs, development tools and content repositories. IMS Content Packaging Specification is used here.
- Meta-data: A mechanism for describing specific instances of the components of the content model. meta-data facilitates the search and discover components across systems. SCORM defines its meta-data based on IEEE 1484.12-1.2002 LOM standard and IEEE 1484.12-3 Standard for Extensible Markup Language Binding for LOM.
- Sequencing and Navigation: A rule-based model for defining a set of rules that describes the intended sequence and ordering of activities. The activities may or may not reference learning resources to be delivered to the learner. Sequencing strategies are encoded in XML as specified in IMS Content Packaging.

The SCORM Runtime Environment Book [27] describes the Learning Management System (LMS) requirements in managing the runtime environment, and the SCOs requirements related to their use of a common Application Programming Interface (API) and the SCORM Runtime Environment Data Model.

The RTE management describes the launching of content objects (SCOs and Assets), management of communications with a SCO and Runtime environment data model management.

The API describes LMS API requirements, SCORM communication requirements and communication error conditions.

The SCORM RTE Data Model describes data model management and behaviour requirements, and data type requirements.

The SCORM SN Book [28] describes how SCORM conformant content may be sequenced to the learner through a set of learner or system-initiated navigation events.

File formats for Assets are not restricted to a type of file. It's only required that Assets related media can be viewed in a web client.

There are two types of SCORM related software: authoring tools and players or LMS. The following table shows some of the available free software tools.

Authoring tools	
Reload Editor [29]	Content Package and meta-data editor. Creates IMS and SCORM content packages.

dokeos [30]	e-Learning and course management web application that includes export to SCORM functionality.
aTutor [31]	Learning Content Management System that adopted the IMS/SCORM content package specifications.
Players	
Reload SCORM Player [29]	Player for SCORM 1.2 packages.
Moodle [17]	Learning Management System, course-oriented, where course activities can be in SCORM format.
dokeos [30]	e-Learning and course management web application that includes import SCORM functionality.
aTutor [31]	Learning Content Management System that includes a SCORM 1.2 RunTime Environment for playing SCORM based SCOs.

The use of SCORM in a multi-language environment depends on the use of language fields in meta-data, as in LOM.

4.3 IMS LD

IMS Learning Design (IMS LD) is a specification for a metalanguage wich enables the modelling of learning processes [32]. It is maintained by the IMS Global Learning Consortium [33]. It must be taken into account that IMS LD is not a standard, since it is not certified by any official standard making body. However, the IMS LD specification is a very good candidate to gain the status of standard in the future, and it has been considered by the SELF Learning Standards Expert Group due to its relevant pedagogical properties.

IMS LD [34] describes a framework of elements that can depict any design of a teaching-learning process in a formal way. The specification defines the framework vocabulary, its syntax expressed in terms of its information structures, and its semantics.

The core concept of the IMS LD Specification [34] is that regardless of pedagogical approach, a **person gets a role** in the teaching-learning process (i.e. learner or staff role). In this role she works to reach certain **goals** by performing learning and/or support **activities** within an **environment**.

IMS LD specifies three levels of implementation and compliance.

- Learning Design Level A: It includes all the core vocabulary needed to support pedagogical diversity. Levels B and C add three additional concepts and their associated capabilities in order to support more sophisticated behaviours.
- Learning Design Level B: Adds Properties and Conditions to level A, which enable personalization and more elaborate sequencing and interaction based on learner portfolios.
- Learning Design Level C: Adds Notification to level B.

IMS LD is an XML based specification composed by:

- IMS Learning Design Information Model: Describes a model for learning design composed by:
  - Conceptual model: Vocabulary, functional relationships between the concepts, and relationships with IMS Content Packaging.
  - Information model: IMS Learning Design elements for levels A, B and C.
  - Behavioural model: Set of runtime behaviours that delivery systems must implement.
- IMS Learning Design XML Bindings for level A,B, and C.
- IMS Learning Design Best Practice and Implementation Guide.

IMS Learning Design is related to the following specifications:

- IMS Content Packaging
- IMS Simple Sequencing
- IMS/LOM Meta-data
- IMS Question and Test Interoperability (QTI)
- IMS Reusable Definition of Competency or Educational Objective (RDCEO)
- IMS Learner Information Package
- IMS Enterprise
- SCORM

IMS LD documents are XML based files. They can refer to any type of document, as set in LOM learning objects.

The following is a table with some free software tools that can work with IMS LD, either as authoring tools or as players:

Software	***Description***
Moodle [17]	Learning Management System, course-oriented, where course activities can be in IMS-LD format.
LAMS [35]	Tool for designing, managing and delivering on line collaborative learning activities. It is inspired by IMS Learning Design. Imports from IMS LD level A with LAMS LD extensions format, and exports to IMS LD level A format.
Reload [29]	Reload Project provides a Learning Design Editor and a Learning Design Player, based on IMS LD.
Coppercore [36]	J2EE runtime engine for IMS LD, that supports A, B and C levels. It's oriented to developers, to

Software	*Description*
	incorporate IMS LD in applications.
SLeD [37]	Web front-end player that uses Coppercore as back-end engine.
COLLAGE [38]	High-level learning design authoring tool for collaborative learning. It's IMS LD level A compliant.

The use of IMS LD in a multi-language environment depends on the use of language fields in meta-data, as in LOM.

4.4 SELF supported Learning Content Standards or Specifications

Following the discussion presented in this section, SELF will define an internal representation of Learning Objects which will not be **identical** to any of the three presented standards and specifications, but will be based on them. More precisely, the internal representation of the SELF LO will be mainly based on the SCORM and LOM, although some of the ideas of IMS LD will also be incorporated.

In order to provide compatibility with other e-learning platforms or tools, SELF will offer import/export filters from/to standards (or specifications). In other words, it will be possible to convert SCORM materials to the SELF internal representation and conversely. The first version of the SELF platform will provide a SCORM 2004 import/export filter. An IMS LD filter will not be provided in the first release, because IMS LD is not a standard but just a specification. In any case, import/export filters for other standards or specifications may be developed in the future, specially taking into account that the SELF platform itself will be released as free software together with its source code. This, includes, for example the forthcoming Content Cartridge specifications from IMS. However, it must be noted that in order to support some functionalities of other standards or specifications changes could be needed in the internal representation of the SELF LO. In any case, these changes may be introduced by an interested expert in the future since SELF will be a free software application.

5 Conclusion

In the above sections, we have outlined requirement for the SELF Platform not only to collect contents stored in open standard formats, but also to follow learning content standards for its own operation. This ensures that all the content collected into the Platform, as well as what is distributed from it, will conform to the widely used open standards.

One of the serious implications of adopting the learning content standards or specifications, such as LOM, SCORM or IMS LD, is that the data model of the platform gets specified almost completely. The suggested standards and specifications comprehensively describe not only the learning resources, but also provide an indication for the possible functionality (collection, authoring, management and delivery) of the content. Thus, the didactic specification, at once becomes both an enabling factor, and also a constraint. Once we have a frozen data model for a very dynamic cultural activity like the process of education, it may become a serious constraint.

The advantage of adopting a standard, on the other hand, is immense. Several other platforms or e-learning environments that are already deployed conforming to the above mentioned standards can make use of the content distributed out of the SELF platform. Thus, inter-operable distribution becomes almost impossible without conforming to a standard. Therefore it is mandatory that the platform delivers content in an open standard. The arguments given in the second section, of why SELF platform uses content that conforms to open standards exclusively, applies also to the organisational and delivery mechanism described within the platform itself.

The standards that are recommended for adoption are well researched and there exists sufficient flexibility in their design. However, the process these standards attempted to model is a dynamic cultural process, namely, education. Therefore, it is necessary to follow a strategy that provides scope for flexibility, and support for alternative models of education. Let us illustrate the problem.

The platform and its content can be used for multiple ways in which education happens in society. A formal education model requires the learner to follow a programmed schedule of delivery, examinations and evaluation. But, for in-service training, the mechanism should be more flexible, since it is not convenient for the teachers on job to take a rigid programme. Also the requirements for upgrading the knowledge differs from teacher to teacher. Again, some others who are interested in learning a subject entirely due to personal motivation would opt for a very flexible mechanism, so much so that they may never complete every module of the course, or undergo the evaluation process. Currently, collaborative mode of learning is becoming increasingly popular both in the classroom as well as on-line environments. Each of the above situations demand a different model of delivering the content.

How to deliver the same content for different kinds of learners is a technical challenge. Traditional classrooms cannot adapt more than one process of delivery, but a computer based school can, in principle, provide what the user wants. Therefore, it is recommended that the SELF platform provides a mechanism to model different styles of delivering content. It is even more desirable if the platform provides a mechanism to creatively model the process of education. Considering that the platform will adapt a semantic web model, these possibilities can be addressed by our platform. The semantic web offers the possibility to package and distribute the same resources in several ways without disturbing the resources. It is therefore recommended to explore the possibilities of providing the freedom of how a learner wants to learn. This will also keep the spirit of "learning in freedom", the motto of the SELF project, embellished in the design of the platform.

6 Bibliography and web links

[1] Greve, G. Sovereign Software Open Standards, Free Software, and the Internet:

http://www.intgovforum.org/Substantive_1st_IGF/SovereignSoftware.pdf

[2] European Interoperability Framework for Pan-European eGovernment Services: http://ec.europa.eu/idabc/servlets/Doc?id=19529

[3] http://www.ft.dk/Samling/20051/beslutningsforslag/B103/index.htm

[4] Formatted text in Wikipedia: http://en.wikipedia.org/wiki/Formatted_text

[5] Vector graphics in Wikipedia: http://en.wikipedia.org/wiki/Vector_graphics

[6] Raster images in Wikipedia: http://en.wikipedia.org/wiki/Raster_graphics

[7] Video formats in Wikipedia: http://en.wikipedia.org/wiki/Video_formats

[8] Hypertext in Wikipedia: http://en.wikipedia.org/wiki/Hypertext

[9] Presentation in Wikipedia: http://en.wikipedia.org/wiki/Presentation

[10] Audio format in Wikipedia: http://en.wikipedia.org/wiki/Audio_format

[11] Data compression in Wikipedia: http://en.wikipedia.org/wiki/Data_compression

[12] XML in Wikipedia: http://en.wikipedia.org/wiki/Xml

[13] Final 1484.12.1-2002 LOM Draft Standard: http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html

[14] IEEE 1484.12.3 – Standard for Learning Technology-Extensible Markup Language (XML) Schema Definition Language Binding For Learning Object Meta-data Model: http://ltsc.ieee.org/wg12/par1484-12-3.html

[15] CanCore: LOM Tools: http://www.cancore.ca/editors.html

[16] IMS LRM: http://www.imsproject.org/metadata/

[17] Moodle: http://www.moodle.org

[18] IEEE 1484.12.4 – Standard for Resource Description Framework (RDF) Binding For Learning Object Meta-data Model: http://ltsc.ieee.org/wg12/par1484-12-4.html

[19] Advanced Distributed Learning: http://www.adlnet.gov/index.cfm

[20] SCORM® 2004 3rd Edition Overview Version 1.0 DRAFT

[21] IEEE Data Model for Content Object Communication: http://ltsc.ieee.org/wg11/par1484-11-1.html

[22] ECMAScript API for Content to Runtime Services Communication: http://ltsc.ieee.org/wg11/par1484-11-2.html

[23] IMS Content Packaging Specification: http://www.imsproject.org/content/packaging/

[24] IMS Simple Sequencing Specification: http://www.imsglobal.org/simplesequencing/index.html

[25] AICC: http://www.aicc.org/

[26] SCORM® 2004 3rd Edition Content Aggregation Model (CAM) Version 1.0 DRAFT

[27] SCORM® 2004 3rd Edition Run-Time Environment (RTE) Version 1.0 DRAFT

[28] SCORM® 2004 3rd Edition Sequencing and Navigation (SN) Version 1.0 DRAFT

[29] Reload Project: http://www.reload.ac.uk/

[30] dokeos: http://www.dokeos.com

[31] aTutor: http://www.atutor.ca/

[32] IMS LD in Wikipedia: http://en.wikipedia.org/wiki/IMS_Learning_Design

[33] IMS Global Learning Consortium: http://www.imsproject.org

[34] IMS Learning Design Specification: http://www.imsglobal.org/learningdesign/index.html

[35] LAMS: http://www.lamsinternational.com/

[36] Coppercore: http://coppercore.org/

[37] SLeD: http://sled.open.ac.uk/web/

[38] COLLAGE: http://gsic.tel.uva.es/collage

[39] IMS LD resources at UNFOLD project: http://www.unfold-project.net:8085/UNFOLD/general_resources_folder/

[40] List of tools at UNFOLD project: http://www.unfold-project.net:8085/UNFOLD/general_resources_folder/webs/general_resources_folder/tools/currenttools

[41] IMS LD Learning Networks: Moodle course on IMS LD: http://imsld.learningnetworks.org/

[42] Implementing Learning Design to support web-based training: http://ausweb.scu.edu.au/aw04/papers/refereed/mcandrew/paper.html