Deliverable 4.1

Specification of SLOM

Semantic Learning Object Model

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)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	**INTUITEL**
Project Full Name:	Intelligent Tutoring Interface for Technology Enhanced Learning
Grant Agreement No.:	318496
Programme	FP7-ICT-2011.8, Challenge 8.1
Instrument:	STREP
Start date of project:	2012-10-01
Duration:	33 months
Deliverable No.:	D4.1
Document name:	D4.1 Specification of SLOM
Work Package	4
Associated Task	1
Nature [1]	R
Dissemination Level [2]	PU
Version:	0.12
Actual Submission Date:	2013-09-30
Contractual Submission Date	2013-09-30
Editor: Institution: E-mail:	Peter A. Henning, Florian Heberle HSKA peter.henning@intuitel.eu, florian.heberle@intuitel.eu

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2011.8, Challenge 8.1) under grant agreement n° 318496.

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **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)

Change Control

Document History

Version	Date	Change History	Author(s)	Organization
0.01	2013-05-03	Document drafted	Florian Heberle, Peter Henning	HSKA
0.02	2013-06-13	1st revision	Peter Henning	HSKA
0.03	2013-07-01	2nd revision	Peter Henning	HSKA
0.04	2013-07-03	Model hierarchy	Peter Henning	HSKA
0.05	2013-07-11	Updates	Florian Heberle, Peter Henning	HSKA
0.06	2013-07-23	Updates	Florian Heberle, Peter Henning	HSKA
0.07	2013-08-06	Updates	Florian Heberle, Peter Henning	HSKA
0.08	2013-08-29	Updates	Florian Heberle, Peter Henning	HSKA
0.09	2013-09-13	Updates	Florian Heberle, Peter Henning	HSKA
0.10	2013-09-17	Updates	Florian Heberle, Peter Henning	HSKA
0.11	2013-09-23	Updates	Florian Heberle, Peter Henning	HSKA
0.12	2013-09-25	Final updates	Florian Heberle	HSKA
1.0	2013-09-30	Finalization and Approval	Peter Henning	HSKA

Distribution List

Date	Issue	Group
2013-05-03	Request for comments	WP04
2013-07-01	Request for comments	WP04
2013-07-03	Request for comments	WP04
2013-07-24	Request for comments	WP04
2013-08-06	Request for comments	WP04
2013-08-29	Request for comments	WP04, WP02
2013-09-17	Request for closing comments	WP04, WP02
2013-09-23	Request for verification	WP04, WP02

List of Contributions

Date	Organization(s)	Person(s)	Contribution

2013-05-03	HSKA	Florian Heberle, Peter Henning	Initial formulation
2013-06-13	HSKA	Peter Henning	Annotation properties
2013-06-25	UNIR	Luis de la Fuente	Comments reg. file format and connections with SCORM and IMS-LD
2013-07-02	HSKA	Florian Heberle	Overall comments
2013-07-09	ELS, FZI,UVA	Elisabetta Parodi, Jürgen Bock, María Jesús Verdú	Overall comments
2013-07-15ff	FZI, UNIR, UVA, URE	Stefan Zander, Jürgen Bock, Luis de la Fuente, Elena Verdú Pérez, María Jesús Verdú, Daniel Thiemert	Overall comments
2013-07-26	UVA, HSKA	María Jesús Verdú, Peter Henning	Discussion regarding new Media Types
2013-07-29	URE	Daniel Thiemert	Section about IMD-LD transformations
2013-07-29ff	URE, FZI, UNIR, UVA, HIT	Daniel Thiemert, Jürgen Bock, Luis de la Fuente, Elena Verdú Pérez	Comments
2013-08-20	FZI	Stefan Zander	Chapter SMW Transformation
2013-08-28	TIE	Oscar Garcia Perales	Overall comments
2013-09-03	UVA	María Jesús Verdú, Elena Pérez Verdú	Corrections and comments
2013-09-11	UVA, UNIR	Juan Pablo de Castro, Luis de la Fuente	Comments regarding geo aspects
2013-09-10f	URE, UNIR	Daniel Thiemert, Luis de la Fuente	Transformation chapters SCORM and IMS-LD
2013-09-13	HSKA	Peter Henning	Chapter Unstructured Transformation
2013-09-13	FZI	Stefan Zander	Chapter SMW Transformation
2013-09-16	URE, UNIR	Daniel Thiemert, Luis de la Fuente	Transformation chapters SCORM and IMS-LD
2013-09-18	FZI	Stefan Zander	Usage of n-ary relations and Geospatial Semantic Web
2013-09-18f	FZI, TIE	Jürgen Bock, Oscar Garcia Perales	Overall comments
2013-09-24	FZI, TIE	Stefan Zander, Oscar Garcia Perales	Closing remarks

Table of Contents

1 Introduction ... 11

1.1 Dependencies ... 11

1.2 SLOM as Objective of INTUITEL ... 11

1.3 The Two Tasks of SLOM ... 12

1.3.1 Metadata for the enhancement of courses ... 12

1.3.2 Packaging of Content and Metadata ... 13

1.4 SLOM related system components ... 13

1.4.1 INTUITEL Editor ... 13

1.4.2 INTUITEL Merger ... 14

1.4.3 Repositories ... 14

2 Model Hierarchy and Coherences ... 16

3 Directory Structure of a SLOM Package ... 20

3.1 ILIAS specific example: Course on *Cooking* ... 21

4 SLOM Metadata ... 22

4.1 Basics ... 22

4.2 Overview ... 24

4.3 KO property descriptions ... 26

4.4 Specification of Learning Pathways ... 28

4.4.1 Specification of macro Learning Pathways ... 28

4.4.2 Specification of micro Learning Pathways ... 29

4.5 Geospatial aspects in SLOM ... 30

4.6 Future Extensions ... 32

5 Transformations ... 34

5.1 Transforming Semantic MediaWiki into SLOM ... 34

5.1.1 Semantic MediaWiki Introduction ... 34

5.1.2 Transformation of SMW Learning Content and Annotations to SLOM ... 34

5.1.3 Transformation of SLOM to the Semantic MediaWiki Format ... 35

5.1.4 Transformation of Semantics ... 35

5.2 Transforming SCORM into SLOM ... 37
5.2.1 SCORM Introduction ... 37
5.2.2 Transformation of Semantics ... 39
5.3 Transforming IMS-LD into SLOM ... 39
5.3.1 IMS-LD Introduction ... 39
5.3.2 Transforming of Semantics ... 40
5.4 Transforming Unstructured Data into SLOM ... 41
6 Adjustments ... 43
6.1 Pedagogical Ontology ... 43
6.2 Data Model ... 44
7 Appendix ... 46
7.1 IMS-LD Level A Element Descriptions ... 46

List of Figures

Figure 1: Metadata context – What do the respective models describe? 13
Figure 2: Illustration of the content of the two SLOM repositories 15
Figure 3: Illustration of the statements subsuming the Semantic Learning Object Model 18
Figure 4: Graphical representation of the SLOM and its terminological coherences 19
Figure 5: SCORM Manifest Elements Hierarchy 38
Figure 6: IMS-LD Elements Hierarchy 40

List of Tables

Table 1: Overview of dependencies of this deliverable 11
Table 2: Hierarchy of models in SLOM 17
Table 3: INTUITEL Consortium suggestion for standard namespace prefixes 23
Table 4: Metadata and their OWL-realizations per Learning Object type 25
Table 5: Overview of KO-specific metadata and their descriptions 27
Table 6: MIME-Type and their resulting Media Type in unstructured data transformations 42
Table 7: Overview of adjustments of the PO 44
Table 8: IMS-LD Level A Elements Description 49

Codelisting index

Codelisting 1: Demonstration of the structure of a SLOM package 20
Codelisting 2: Minimal example of a Knowledge Object definition 28
Codelisting 3: Example of a Knowledge Object definition 28
Codelisting 4: Example for creating a new macro LP relation and using it to link two CCs 29
Codelisting 5: Example for creating a new micro LP relation and using it to link two KOs 30
Codelisting 6: Example root element for a message originating from the LPM 44
Codelisting 7: Example root element for a message originating from the LMS 44
Codelisting 8: Example root element for a message originating from the SLOM toolset 45
Codelisting 9: Example root element for a general message without a specific origin 45

List of abbreviations

Term:	Explanation:
DOW	Description of work
C	Content Creator (role)
C[x]	Content Creator number x (different roles of content creators)
CC	Concept Container
CM	Cognitive Model
CCM	Cognitive Content Model
DM	Domain Model
GUI	Graphical User Interface
KD	Knowledge Domain
KO	Knowledge Object
KT	Knowledge Type
LMS	Learning Management System
L	Learner
LO	Learning Object
LORE	Learning Object Recommender
LP	Learning Pathway
LPM	Learning Progress Model
MT	Media Type
OWL	Web Ontology Language
PO	Pedagogical Ontology
REST	Representational State Transfer
SCORM	Sharable Content Object Reference Model
SLOM	Semantic Learning Object Model
TUG	Tutorial Guidance
USE	User Score Extraction
WP	Work package
WS	Web Service

Glossary

Term:	Meaning:
Course	The topmost cognitive container of an INTUITEL enhanced system, equivalent to the Knowledge Domain (KD).
Content creator	A person, usually some kind of lecturer, a domain or ontology specialist, who fulfils one or more tasks in the workflow to INTUITEL-enable course, e.g. creating learning material or entering metadata via the INTUITEL Editor.
Cognitive Content Model (CCM)	A Cognitive Content Model is the INTUITEL description of an eLearning course using the terms and relations as specified in the Pedagogical Ontology.
Cognitive engineer	Person that is responsible for creating a CM, possibly but not necessarily a teacher. In general, a person or a group of persons that possess expert knowledge of a domain and is able to model this in an INTUITEL-compatible way (e.g. with the INTUITEL Editor).
Cognitive Model (CM)	A Cognitive Model is the INTUITEL description of a domain of knowledge using the terms and relations as specified in the Pedagogical Ontology.
INTUITEL-enabled course	An INTUITEL-enabled course is a course in a LMS for which the Back End has all necessary metadata to create recommendations.
INTUITEL-enabled LMS	An INTUITEL-enabled LMS is an eLearning platform, which includes the INTUITEL-specific interfaces to exchange messages with the INTUITEL components, while also conforming by providing and presenting all necessary information.
Knowledge Domain (KD)	Topmost cognitive container of an INTUITEL enhanced course. The KD element encapsulates a set of Concept Containers to create an outline for a domain of knowledge. Such CCs are e.g. thematically related collections of pages (learning modules), tests, surveys, files, media objects.
Knowledge Object (KO)	In INTUITEL a Knowledge Object (KO) is an item of knowledge, which typically corresponds to one screen page of content and to an estimated learning time of 3-10 minutes for the average learner.
Learner	A learning person using the INTUITEL enhanced system.
Learning Object (LO)	Learning Object (LO) is in INTUITEL the umbrella term for the various types of learning containers, such as Knowledge Domain (KD), Concept Container (CC) and Knowledge Object (KO).
Learning Object Recommender (LORE)	The Learning Object Recommender (LORE) is a module in INTUITEL, which enables the Back End to send personalised learning recommendations to the LMS and to display them in a way that is suitable for the specific LMS.
Learning Progress Model (LPM)	The Learning Progress Model (LPM) is an element in INTUITEL, which acts as a preliminary stage in the reasoning process by managing, adapting and providing all necessary data.
LMS integrator	An organisation or person(s) that implement the INTUITEL interfaces in an LMS in accordance to the INTUITEL (Data Model and architectural) specifications.
Pedagogical Ontology (PO)	The Pedagogical Ontology is an OWL-ontology, which provides terms and relations to model courses and learning pathways in a semantically rich way, in order to enable the INTUITEL Back End to automatically create learning recommendations.

Position in a Cognitive Model	A learner’s position in a Cognitive Model is the set of Concept Containers that contain Knowledge Objects on which the learner has worked on without completing all of the Knowledge Objects of the learner’s Learning Pathway in that particular Concept Container. When also including the CC internal aspects, the position is more precisely defined by the last worked on KO in that particular CC in regard of the learner’s individual Learning Pathway.
Semantic Learning Object Model (SLOM)	The Semantic Learning Object Model (SLOM) is a file format specification on how different eLearning content and additional INTUITEL specific metadata can be combined in one container file.
Tutorial Guidance (TUG)	The Tutorial Guidance (TUG) is a module in INTUITEL, which enables direct exchange of text, audio and video information between INTUITEL and the learner via the LMS UI, in order to give direct feedback or to request additional information.
User Score Extraction (USE)	The User Score Extraction (USE) is a module in INTUITEL, which supplies the Back End with information about learner performance and their current environmental situation.

1 Introduction

INTUITEL is devoted to the development of intelligent tutorial interfaces for technology enhanced learning. Learning content for electronic Learning Management Systems (LMS) is enhanced by certain metadata, such that an INTUITEL-enabled LMS may use this enhanced content for tutorial guidance of learners, to adapt itself to their personal needs and therefore to present them the most individual learning content.

Task 4.1 will see the specification of an abstract data model for storing the necessary educational metadata together with the learning content, the

Semantic Learning Object Model SLOM

and its implementation as a container format. It also defines a two way transformation of XML formats (and XSD models) like SCORM / IMS-LD into SLOM and back to the syntactic level (i.e. an XML based serialization). A prototypical specification of this transformation is given in chapter 5 of this document and will be implemented in the related INTUITEL tasks 4.2, 4.3 and 4.4.

1.1 Dependencies

This deliverable is built upon four previous deliverables of INTUITEL. The following table gives an overview of them and shortly describes what they are about.

D#	Title	Description
1.1	XML Schema for USE/TUG/LORE	An xml schema and documentation which describe the messages that are exchanged between INTUITEL and LMSs.
2.1	Pedagogical Ontology	Ontology and documentation outlining the structure, properties and didactic foundation of how learning material is described in INTUITEL.
3.1	Overall System Design	Textual description of the interplay of the various technical and pedagogical components of INTUITEL.
3.2	Learning Progress Model	Documentation and software component that acts as a preprocessing and management entity for the INTUITEL Engine in the Back End.

Table 1: Overview of dependencies of this deliverable

1.2 SLOM as Objective of INTUITEL

Presently the eLearning market faces the challenge that many standards (like SCORM, EML, etc.) are expressed in terms of XML Schema Definition (XSD). Today, semantic technologies are mature enough to act as a pivotal technology to integrate sources of heterogeneous provenances and meanings.

INTUITEL therefore contains the development of a new model for the semantic enhancement of learning content by pedagogical and factual metadata, called the Semantic Learning Object Model -

SLOM. In particular, these metadata shall comprise the Cognitive (Content) Model (CM/CCM) and allow storing this along with the learning content itself in the form of a so called 'SLOM package'.

In the INTUITEL environment, SLOM consequently serves as embodiment of (i) an ontology-based metadata model for learning content, and (ii) a file format specification for storing INTUITEL-related metadata together with the learning content in the form of a content package.

1.3 The Two Tasks of SLOM

SLOM basically fulfills two tasks. Firstly, SLOM is used to manage the course-related metadata. Secondly, SLOM is used as a container format to store and exchange INTUITEL-enabled courses, i.e. metadata and content. Both of these tasks are part of this deliverable. They are shortly introduced in the following two sub-sections and the respective means to fulfill them are described in more detail afterwards.

1.3.1 Metadata for the enhancement of courses

The core objective of SLOM is to provide a format that allows specifying the metadata in a way that is compatible with the INTUITEL Back End. This data is the basis for the deductive processes that ultimately lead to learning recommendations (LORE) and learning feedback (TUG). It relies on two main elements, the abstract, domain-specific Cognitive Model (CM) and the concrete, course-specific Cognitive Content Model (CCM).

A Cognitive Model is the INTUITEL description of a domain of knowledge using the terms and relations as specified in the Pedagogical Ontology (PO). It is created by a domain expert and specifies the Concept Containers (CCs) of that particular Knowledge Domain (KD). Additionally, didactically meaningful CC sequences, i.e. the macro Learning Pathways (LPs), are defined here. The description of how a CM should be specified is part of WP02, but is relevant for SLOM because the information is mandatory for the INTUITEL Back End. With SLOM being the only way to feed this into the reasoning apparatus, CMs need to be included in SLOM.

To complete the knowledge basis for the deductive process, SLOM constitutes an extension to the Pedagogical Ontology to semantically enhance learning content, the so called Cognitive Content Model. This is the INTUITEL-compatible specification of the Knowledge Objects (KOs), i.e. the actual elements that should be learned by the learner, of the respective course, for which recommendations should be produced. This also contains a variety of metadata that defines what the respective content in the LMS is about. It, for instance, specifies the Knowledge Type (KT) and the Media Type (MT) of each KO as well as the available micro LPs.

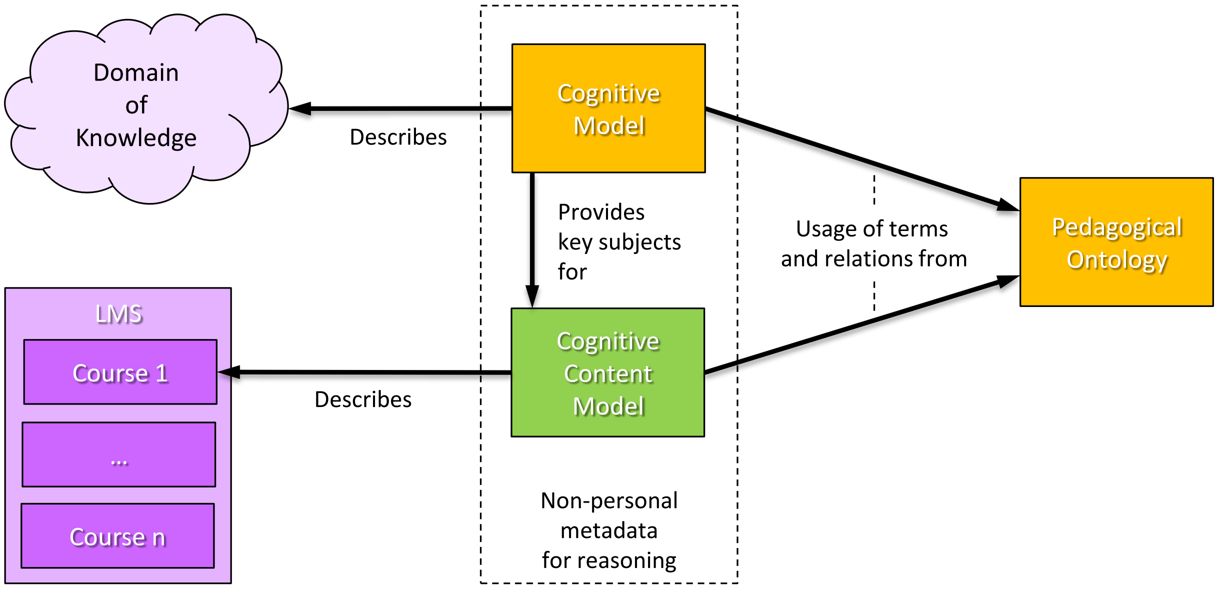

Figure 1: Metadata context – What do the respective models describe?

1.3.2 Packaging of Content and Metadata

SLOM specifies a file format that contains all relevant data to exchange, store or version complete INTUITEL-enabled courses (i.e. courses from the LMS that have a (semi-)complete set of INTUITEL metadata). It contains eLearning course material in a form similar to IMS-CP and to the packaging of SCORM, but with additional annotations according to the pedagogical and domain-specific ontologies. This container format allows for a compact side-by-side storage of learning content and metadata. In other words, the SLOM metadata is stored externally to the Learning Objects.

This allows using INTUITEL-enhanced course material also in Non-INTUITEL systems (albeit without the functionality provided by INTUITEL).

1.4 SLOM related system components

Apart from the specification of the SLOM metadata format and the SLOM package, INTUITEL also comprises software that is used to manage different aspects of its administration in context of the project. This is provided by a set of tools and components that allow users to enter and change the respective data in an intuitive and user-friendly way. In the following, a short overview of these modules is given (see deliverable 3.1 – Overall System Design – for a detailed introduction).

1.4.1 INTUITEL Editor

The INTUITEL Editor (WP07) is a software tool, which provides means to comfortably enrich learning material with the semantic annotations relevant for the course and domain-specific ontologies. Its core task is thus to provide a user interface that allows the addition of the SLOM metadata to any learning content in context of authoring and editing of CCMs.

As a core component of the INTUITEL system, the INTUITEL Editor will be developed and usable independently of any LMS. It will write its output into the SLOM Metadata Repository, which serves as a central place for the INTUITEL Back End to access this information.

1.4.2 INTUITEL Merger

This building block is a new conversion/transformation concept and its concrete implementation. It transforms several SLOM input streams (i.e. learning material in a supported format – cf. chapter 5) into a common representation according to the SLOM container format, which can be stored in the SLOM Repository. The INTUITEL Merger (WP06) is a software tool used independently of other building blocks.

The resulting SLOM package may be edited with the INTUITEL Editor. This will be necessary in most cases, because current eLearning formats will probably not contain all data that bear relevance for INTUITEL. In order to create better, i.e. more fitting/personalized, recommendations and especially in context of defining LPs, using the INTUITEL Editor afterwards to complete the data is required.

1.4.3 Repositories

The DoW and previous deliverables are naming two repositories that are used to store SLOM-related information. Those are the

- SLOM Repository - A repository to keep rich semantically enhanced learning material, i.e. the SLOM packages. It is managed by the INTUITEL Merger, which provides access mechanisms to store the SLOM packages here. They are stored in the same manner, independent of whether they may have already been in the SLOM format, or may have been converted (e.g. XSLT-transformed) through the scripts developed as part of INTUITEL, or may have been obtained by automatically merging different SLOM input formats.

and the

- SLOM Meta Repository - A metadata repository where annotations (CMs and CCMs) are held and which may be linked to the learning content through Xlink/XPointer[3]. Each KO being described in a CCM either links to an actual content object in a SLOM package, to an item in the LMS (i.e. a LO that is part of a course) or both. The SLOM Meta Repository acts as central storage for the INTUITEL-specific metadata in an INTUITEL instance and provides this data for the Learning Progress Model (LPM).

In other words, the SLOM Meta Repository stores metadata but not content; the content remains stored in the LMS or in a SLOM package.

However, the conclusions that have been drawn so far suggest that it is technologically reasonable to merge those repositories. Since the metadata is present as an OWL-file and at the same time also as part of a SLOM package, separating them would result in unnecessary redundancy. To avoid this and

[3] These are XML applications that allow storing hypertext references to document parts - sentences, paragraphs, even single words - outside of the text and without modifying it.

to work more efficiently, the repositories will be implemented as one. The original tasks of the repositories are not affected by this decision.

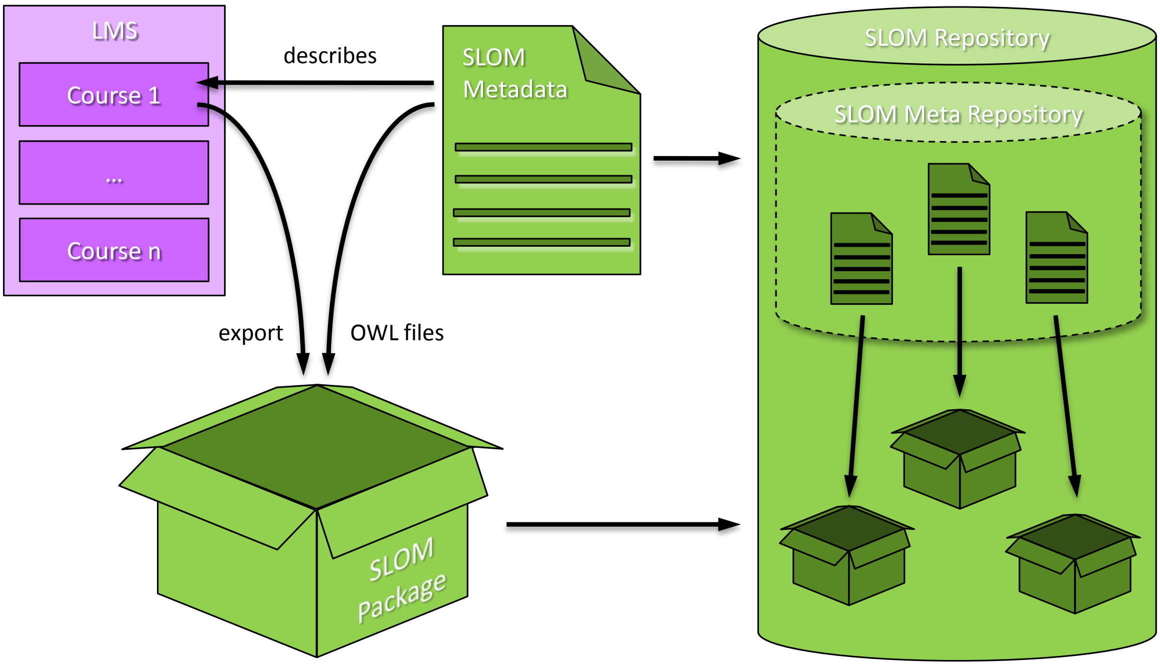

Figure 2: Illustration of the content of the two SLOM repositories

2 Model Hierarchy and Coherences

Within INTUITEL, three layers of abstraction regarding the INTUITEL content and SLOM metadata have been developed for the system. In the framework of SLOM, the Consortium deems it necessary to clarify the different layers, their role and the personal involvement of content developers somewhat. This is achieved in the following table:

Pedagogical Ontology – PO	
Author:	INTUITEL Consortium
Method of authoring:	No authoring necessary, created and maintained by the INTUITEL Consortium
Availability:	Available on the public internet via the official INTUITEL website, currently as http://www.intuitel.eu/public/intui_PO.owl
Coherences:	The PO is imported by the CM and indirectly also by the CCM. This has to be done by referencing on the desired version of the PO on the official INTUITEL website. In case that the public internet is not available (e.g. for security or policy reasons), the INTUITEL Engine will automatically detect this and use a local copy of the PO.
Role in SLOM:	Provision of concepts and vocabulary to describe the teaching process and the teaching content in the CM and CCM.
Additional information:	- The PO is the topmost level of abstraction in INTUITEL. - The PO does not describe, nor reference, nor contain any learning content.
Cognitive Model – CM	
Author:	Knowledge Domain Expert, not necessarily part of the institution operating the Learning Management System that hosts the course in scope.
Method of authoring:	Usage of the INTUITEL Editor or by manually editing OWL code. The latter is less recommended as it needs considerable technical and INTUITEL expert knowledge.
Availability:	A CM needs to be created for a specific domain of knowledge and might be available from a public source or have to be custom made. A CM should be course-agnostic to allow reusing it.
Coherences:	A CM imports the PO, while the CM is imported by the CCM.
Role in SLOM:	A CM provides a concrete instantiation to the abstract classes of the **PO for a given domain of knowledge**, answering the following questions: - What are the necessary Concept Containers? - What are the domain-specific macro Learning Pathways? - What is the metadata for each of these CCs?
Additional information:	- A CM does not contain any learning content. - A CM is written as an OWL file, which actually is a concretization of the PO in the sense that it adds assertional statements about individuals and relations

	(macro LPs) to the ontology. - A CM rests on a particular version of the PO, which is therefore referenced in the CM in a non-ambiguous way by specifying the version via the URI. - A CM can either be public or private (e.g. in a proprietary scenario). The four example CMs produced in INTUITEL are open to the consortium members and to the EC representatives. - In general, a CM will be owned by the knowledge domain expert (resp. his or her company).
Cognitive Content Model – CCM	
Author:	Cognitive engineer, a didactically skilled worker (usually a teacher or tutor) who created the course in scope.
Method of authoring:	a) Usage of the INTUITEL Editor. b) Usage of the INTUITEL Merger to import data from a pre-existing file (e.g. SCORM). This is only possible to initially set up a new CCM.
Availability:	A CCM must be locally available and needs to be custom made for a specific course.
Coherences:	A CCM imports a suitable CM and thus also imports the PO indirectly.
Role in SLOM:	A CCM is the most concrete instantiation of the abstract classes of the PO. It asserts properties, which link the KO instance to, for example, a Media Type or specifies further metadata like a title (cf. section 4.3). Four elements are especially relevant: - Content ID: links KO with its content counterpart in the LMS (IDs generated by LMS as provided via the mapping service – data property). - Package ID: links KO with its content counterpart in the SLOM package (ID as relative reference in accordance to SLOM package structure – data property). - CC IDs: links KO with the CC in the corresponding CM (ID as reference on the CC in the Cognitive Model – object property). - Micro LP: links KO with another KO from the same CC via a micro LP-relation to create KO sequences for recommendation (object property).
Additional information:	- A CCM contains references to the learning content such that it is identifiable in the LMS when INTUITEL creates a recommendation. - A CCM is written as an OWL file, which actually is an extension of the CM in the sense that it adds *more* statements about (new) individuals to the ontology. - A CCM imports a CM. A separate PO import is therefore not necessary as this is indirectly achieved via the CM. - Technically, a CM import is done by a reference on the respective URI. - Distribution of a CCM alone is less reasonable, as it only answers a purpose in combination with a particular course and CM.

Table 2: Hierarchy of models in SLOM

The models described in this section build on one another and gradually become more concrete until the level of describing the actual course content is reached (i.e. the factual metadata). The summation of them allows the INTUITEL Back End to gain a full view on the learning material from three perspectives:

1) A theoretic, didactic, educational sciences perspective.

2) A topical perspective that describes the focus points of the domain.

3) A content perspective which provides detailed information of what the LOs are about.

When reformulating the table above and their meaning for SLOM in a way that is as short as possible, the following four statements result:

- The PO provides the vocabulary and relations.
- The CM provides the abstract topical knowledge.
- The CCM provides the concrete knowledge about the content.
- The combination of PO, CM and CCM is summarized as the Semantic Learning Object Model.

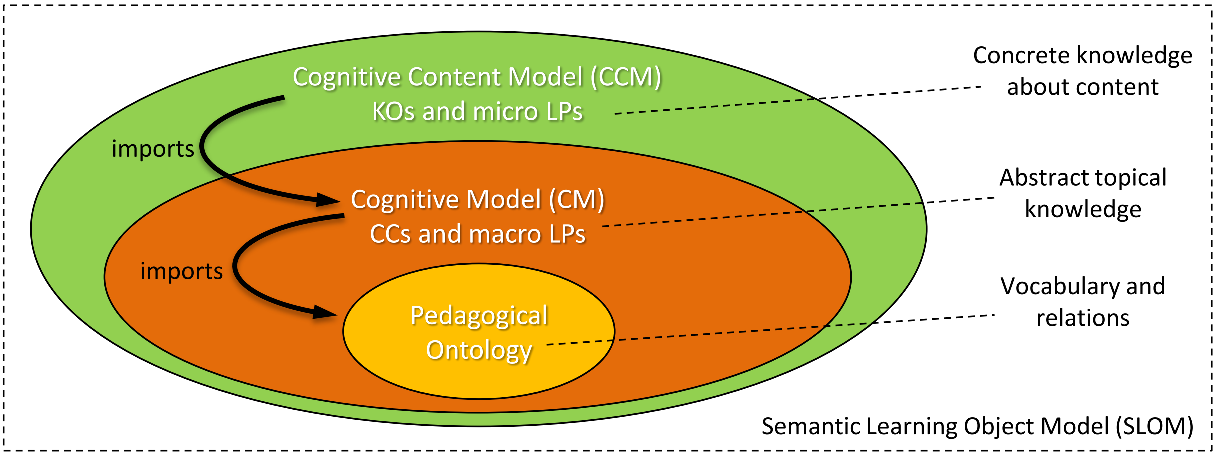

Figure 3: Illustration of the statements subsuming the Semantic Learning Object Model

So far, this only describes the course from a meta-level. However, SLOM is also a format for exchanging this data. Therefore, the content can be merged with the metadata in form of the so called SLOM package. This container format comprises all information that is relevant for INTUITEL to exchange INTUITEL-enabled courses between different LMSs and thus allows a high level of interoperability.

If visualized, the coherences between these elements can be outlined as in the following figure:

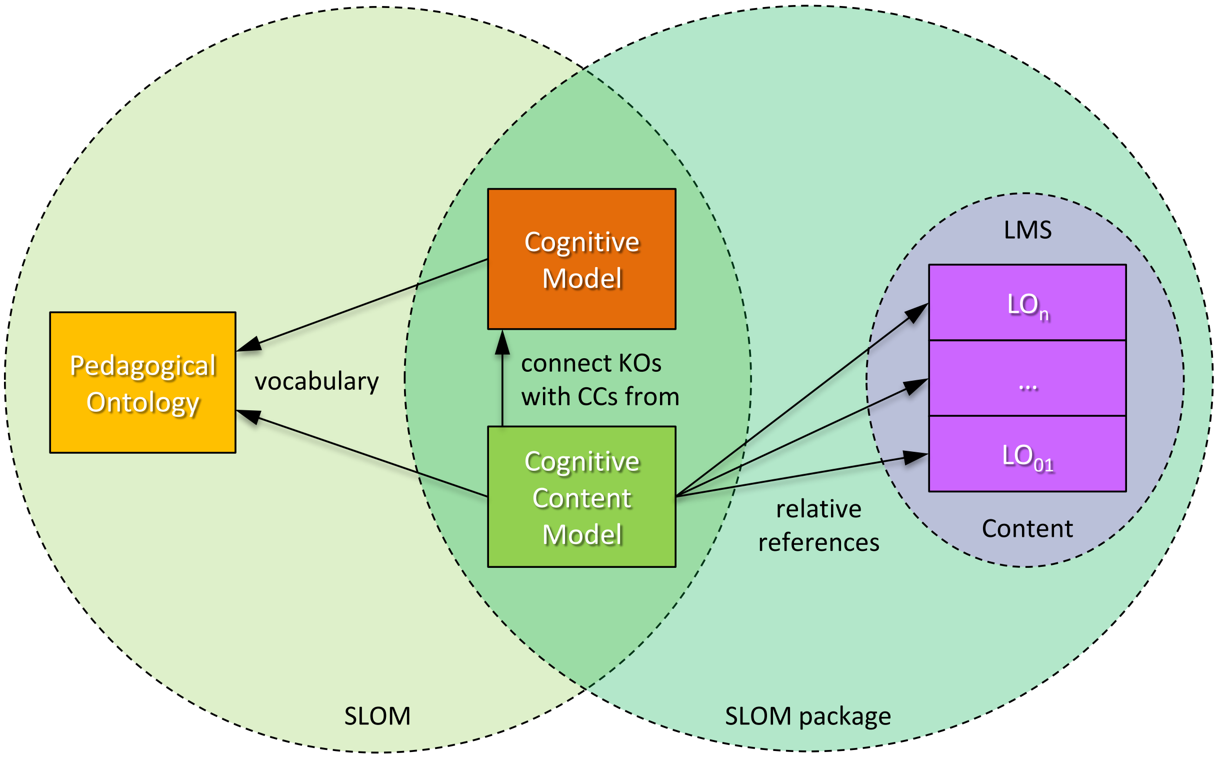

Figure 4: Graphical representation of the SLOM and its terminological coherences

Please note that INTUITEL does not force users to export the Learning Objects (LOs) out of the LMS. The reasoning process can take place without creating a SLOM package, because only the metadata contained in CM and CCM is needed.

As a final remark in this chapter, it shall be noted that it is not technically necessary that the ontologies import each other. The desired effect could also be achieved by referencing. Nevertheless, the INTUITEL Consortium decided to use imports for the development process as this makes it more comfortable to create CMs and CCMs with Protégé. When a more efficient method of authoring these models is available via the INTUITEL Editor, this might be changed. A final decision whether CMs and CCMs should use imports or references has been postponed till then. It could also be possible that both approaches, i.e. importing and referencing, will be accepted in order to let the user decide how he or she wants to create these models.

3 Directory Structure of a SLOM Package

The basic feature of a SLOM package is its *separability* into

- Learning Content, which may be either LMS-independent (e.g. SCORM compatible) or LMS-dependent, and
- INTUITEL metadata, which is the main input for recommendation creation.

In technical terms, this separation is achieved by storing this content in a specific directory structure. INTUITEL therefore relies on the fact that any modern learning content comes along as a collection of files and subdirectories. This content is uncompressed in first order, i.e. any content-internal directory structure is visible. Many export formats for learning content allow distributing single file content, where this collection is zipped into a single compressed file.

A SLOM package structure mainly relies on the content, which the INTUITEL metadata extends. The original format of the content is preserved, i.e. will be used as is, but extended with an additional top-level directory. This **intuitel**-folder holds the following items:

- An OWL file *intui_CM.owl* containing the domain-specific Cognitive Model which imports a specific version of the PO from the official INTUITEL website.
- An OWL file *intui_CCM.owl* containing the course-specific Cognitive Content Model which imports the CM of this folder. The individual definitions of the KOs thereby refer to the IDs of the LOs in the content structure of the SLOM package.
- Optionally, an XML Schema file *intui_DM.xsd* of the Data Model for the USE/TUG/LORE interface to allow formal verification of USE/TUG/LORE messages.

If delivering INTUITEL content as a single file, the whole data collection including the **intuitel** directory is zipped into a single compressed file. On uncompressing this zip file, the presence of the directory **intuitel** will be used for testing compatibility with the INTUITEL system.

The *separability* of SLOM also works in reverse direction, because it allows rewriting any learning content of any LMS as SLOM file.

The following textual representation of a directory demonstrates how a SLOM package could be structured. The parts in square brackets are placeholders that are dependent on the source material.

```
[package name].slom
|-- [learning content files]
|-- [learning content directories]
|    |-- [learning content files]
|    `-- [learning content directories]
`-- intuitel
     |-- intui_CM.owl
     |-- intui_CCM.owl
     `-- intui_DM.xsd #optional
```

Codelisting 1: Demonstration of the structure of a SLOM package

3.1 ILIAS specific example: Course on *Cooking*

Consider the example of an instance of the LMS ILIAS that holds a course fragment on the topic *cooking.* This lecture consists of 112 separate pages, which are linearly aligned. If this course is exported into SCORM using standard features of the LMS ILIAS, a file having a name similar to *1367592206__0__lm_50753.zip* will be created.

On uncompressing this file, the following structure will result:

- A file imsmanifest.xml describing the learning content.
- Several files with ending .xsd, containing XML Schema definitions for the SCORM data model.
- A directory named **res** which contains[4]:
    - 112 separate HTML-pages of the course, plus additional ones for indexing and as table of contents.
    - Several directories for images and other related material.

If this course is now being INTUITEL-enhanced by the INTUITEL Merger, a new folder called **intuitel** is created. This is necessary to store the relevant metadata alongside the actual learning material, i.e. the previously exported files.

- A directory **intuitel**, which holds:
    - The file intui_CM.owl containing the domain-specific Cognitive Model.
    - The file intui_CCM.owl containing the course-specific Cognitive Content Model.
    - The file intui_DM.xsd holding the Data Model schema definition file (optional).

Please note that a LMS will, in the standard workflow, not come into direct contact with this **intuitel** folder. Its sole purpose is to enable the storage and exchange of complete INTUITEL-enhanced courses, including their metadata. If such a course is imported via the INTUITEL Merger at a later point in time, the information will be split again so that the metadata can be stored in the SLOM Meta Repository and the learning material can be imported into a LMS.

[4] Note, that this example is ILIAS specific, as in general the content is not necessarily written in a res-Folder. The example only has the purpose to illustrate the appearance of the **intuitel** folder.

4 SLOM Metadata

In principle, the basic set of metadata the INTUITEL Back End needs for creating recommendations is available in the Pedagogical Ontology. However, in order to enable a more personalized reasoning, additional information is necessary to provide a more profound data basis. The SLOM metadata elements described in this section are thus realized as an extension of the Pedagogical Ontology. Challenge of this task in the SLOM work package is to identify and define the metadata that is yet missing and to embed it into a framework with the SLOM structure (which has been outlined in chapter 2).

The following sections therefore provide an introduction to the metadata part of SLOM and how it is described in OWL.

4.1 Basics

The previous tasks in INTUITEL have produced a solid set of insights that outline what information is needed to specify learning material in an INTUITEL-compatible way, i.e. what metadata is needed in SLOM:

- The foundational principle of LP metadata necessary for recommending Learning Objects has textually been specified in deliverable D2.1 of INTUITEL.
- The main pedagogical properties of Learning Objects have been cast into the Ontology Web Language (OWL), see file INTUITEL_318496_D2_1_PedagogicalOntology.owl.
- Content-related properties of Learning Objects have been formulated in a table of the document file INTUITEL_318496_D2_1_PedagogicalOntology.pdf, section 2.5, and were further analyzed in the Learning Progress Model (deliverable INTUITEL_318496_D3_2_LearningProgressModel.pdf).

The question that needs to be answered in context of this deliverable is what is needed additionally, what needs to be adjusted and what could even be removed. This in particular includes the implementation of the items needed for the Didactic Factors of the LPM specification (D3.2). The sections 4.2 and 4.3 therefore provide a detailed description of CCM-related data and how they are incorporated into the PO.

To standardize the description of SLOM, the INTUITEL consortium further agreed on namespaces and common prefixes. They should be used generally and help to make the ontologies better readable on a code basis. It should also ease the implementation of software that works with these files.

The foundational namespace prefix “**intui**” has been selected as abbreviation for INTUITEL. In its two representations as **intui**: and &**intui**; it will point to:

http://www.intuitel.eu/public/intui_PO.owl

This URL always contains the most up-to-date version of the Pedagogical Ontology, while other version can be addressed individually by specifying the release date. This is done by adding year and month to the URL as in the following example:

http://www.intuitel.eu/public/2013/09/intui_PO.owl

This pattern will also be applied to the other INTUITEL-specific ontologies and XML Schema files that are hosted on the official INTUITEL website. A complete overview of the standard prefixes that have been agreed on in INTUITEL are listed in the following table[5]:

Prefix:	Written-out:	Prefix for…:
intui	INTUITEL Pedagogical Ontology	… addressing of elements specified in the Pedagogical Ontology
intuidm	INTUITEL general message	… general messages as specified in the Data Model without a specific point of origin.
intuilpm	INTUITEL LPM message	… messages as specified in the Data Model specifically originating from the Learning Progress Model (LPM).
intuilms	INTUITEL LMS message	… message as specified in the Data Model specifically originating from the Learning Management System (LMS).
intuislom	INTUITEL SLOM message	… message as specified in the Data Model specifically originating from the SLOM toolset.
intuilm	INTUITEL LM Ontology	… addressing elements of the LM ontology which contain the definition of the Didactic and Rating Factors.
intuicm	INTUITEL Cognitive Model	… addressing elements of a specific Cognitive Model which contains the definition of fitting CCs.
intuiccm	INTUITEL Cognitive Content Model	… addressing elements of a specific Cognitive Content Model which contains the definition of KOs for a particular course.

Table 3: INTUITEL Consortium suggestion for standard namespace prefixes

To identify elements in SLOM unambiguously LOs need to be uniquely identifiable via their URI. This means that a SLOM repository must not contain two LOs with the same URI. Two different KOs with the same name (i.e. URI fragment, e.g. '#KO01') are thus not allowed to have the same authority (e.g. 'intuitel.eu') and path (e.g. '/public/example/intui_CM.owl').

Furthermore, in order to enable an easier processing of SLOM metadata, the INTUITEL URIs have to be named in accordance to the SLOM package format. I.e.:

- A CM has to be named "intui_CM.owl" in a URI.
- A CCM has to be name "intui_CCM.owl" in a URI.

For example, a URI for a CM in the form "http://www.intuitel.eu/public/example/myCM.owl" would not be INTUITEL conformant. To differentiate between courses, the INTUITEL Consortium suggests applying to the following URI pattern:

```
[some base uri]/[courseName]/[version]/intuitel/[filename]
```

E.g.:

```
http://www.intuitel.eu/public/exampleCourse/0001/intuitel/intui_CM.owl
```

[5] Cf. section 6.2 for more information about the fragmentation of the DM as it is described in this table.

4.2 Overview

This section gives an overview of the data items, i.e. the information that can be added to a specific LO in order to describe it, containing SLOM-related information. All these elements are or can be part of LO definitions in context of a CM or CCM.

Please note that SLOM also uses the Dublin Core Standard (DC) to add further semantics to the description of ontologies in INTUITEL. For information on the DC, please consult the official DC website under:

http://dublincore.org/documents/dces/

This particularly includes the following items for the annotation of CMs and CCMs and INTUITEL standard ontologies:

- dc:creator – Person(s) and/or organization(s) that are mainly responsible for the creation of the respective ontology.
- dc:contributor – Person(s) and/or organization(s) that contributed to the creation of the respective ontology.
- dc:description – A (short) description of the ontology that specifies what is it about and what its intention is.
- dc:title – The (formal) name of the ontology.

Furthermore, the OWL-property “versionInfo” is used to specify the version of the respective ontology.

The following Table 4 contains the metadata per Learning Object based on the INTUITEL deliverable 2.1 and 3.2. Some changes and additions have been made to reflect the gained insights since the initial specification of the Pedagogical Ontology. The first column specifies the Learning Object under focus, column number two names the metadata item and the last column describes how it is realized within SLOM.

For a more detailed description of the KO metadata items, please see the subsequent section 4.3. The specification of LPs (i.e. the relations used to link LOs with a LP) is omitted here as this should not only be seen from a technological aspect and as the adjustments made in this context need further explanation. This and other minor changes are described in section 6.1.

LO:	Metadata element:	SLOM-realization:
Knowledge Domain (KD)	Title	Annotation property <dc:title>
	Description	Annotation property <dc:description>
	Reference on top-level CC	Object property <intuit:containsConceptContainer>
Concept Container (CC)	Title	Annotation property <dc:title>
	Description	Annotation property <dc:description>
	Reference on KD	Object property <intui:isContainedByKnoweldgeDomain>

	Reference on contained KO	Object property <intui:containsKnoweldgeObject>
Knowledge Object (KO)	Title	Annotation property <dc:title>
	Description	Annotation property <dc:description>
	Reference on element in LMS	Data property <intui:isLinkedWithLmsElement>
	Reference on element in SLOM package structure	Data property <intui:isLinkedWithSlomPackageElement>
	Reference on CC in the CM	Object property <intui:isContainedByConceptContainer>
	Estimated Learning Time (ELT)	Data property <intui:hasEstimatedLearningTime>
	European Qualifications Framework (EQF) level	Data property <intui:hasEqfLevel>
	Difficulty	Data property <intui:hasDifficultyLevel>
	Recommended age	Data property <intui:hasRecommendedAge>
	Minimal width (pixel)	Data property <intui:hasMinimalWidth>
	Minimal height (pixel)	Data property <intui:hasMinimalHeight>
	Recommended width (pixel)	Data property <intui:hasRecommendedWidth>
	Recommended height (pixel)	Data property <intui:hasRecommendedHeight>
	Size (digital)	Data property <intui:hasSize>
	Available language	Data property <intui:hasLanguage>
	Knowledge Type	Object property <intui:hasKnowledgeType>
	Media Type	Object property <intui:hasMediaType>
	Subtitles availability	Data property <intui:hasSubtitles>
	Suitable for deaf learners	Data property <intui:isSuitableForDeaf>
	Suitable for blind learners	Data property <intui:isSuitableForBlind>
	Suitable for mute learners	Data property <intui:isSuitableForMute>
	Preferred location	Object property <intui:hasPreferredLocation>

Table 4: Metadata and their OWL-realizations per Learning Object type

The two data properties “dc:title” and “dc:description” are used across all LOs and store information for the INTUITEL Editor. When creating or referencing on LOs in this tool, the cognitive engineer will want to work with meaningful names for elements. Automatically generated unique IDs are not suitable for that and thus the property “dc:title” is used for displaying purposes. Further, a declarative text can be added via the “dc:description” property to explain in detail what this element is about. This allows also other persons to understand the intentions of the creator of the respective element.

4.3 KO property descriptions

To provide better insights into the usage and intentions of the individual metadata items of KOs in SLOM, the following sections provide a more detailed look on them. This serves as basis to demonstrate the expressivity of SLOM in a CCM for the reasoning in the INTUITEL Back End.

In general, if a certain metadata item is already available in a LMS or in the learning material itself (e.g. in the SCORM metadata), the respective service or transformation method should directly provide it for SLOM. It will then be used as an initial suggestion for the cognitive engineer in context of the INTUITEL Editor to ease the workload of entering information manually. If the SLOM file already contains the respective information, i.e. the property has already been defined in an earlier step, the information will be ignored.

The properties in the following table originate from the Pedagogical Ontology unless otherwise noted by specifying another namespace. The column "Obl." (obligatory) states whether the data item is mandatory for the reasoning process.

Please note that LPs are not considered in this table. As this matter is more complex, it will separately be explained in section 4.4. Although not explicitly stated here, their definition and usage in KOs and CCs are necessary to be regarded in the recommendation creation.

Property:	Description:	Obl.:
<dc:title>	String specifying a meaningful name that is being used in context of the INTUITEL Editor to label a KO.	Yes
<dc:description>	String providing a (short) description of the KO to outline its meaning. Used in the INTUITEL Editor.	No
isLinkedWithLmsElement	String referencing on a Knowledge Object inside the LMS. This ID is created by the LMS in context of the mapping request. A KO can only be regarded in the reasoning process when this ID is available. (i.e. mandatory for the creation of recommendations)	Yes
isLinkedWithSlomPackageElement	Relative reference linking the metadata to a Knowledge Object in the content regime of a SLOM package. This ID is irrelevant for the reasoning process but must be specified in a SLOM package to preserve coherences.	No
isContainedByConceptContainer	Object property referencing on the corresponding CC to which the KO is assigned in the CM.	Yes
hasEstimatedLearningTime	ELT is given in ISO 8601 format either as hh:mm or as hh:mm:ss. It is the time allocated to the learner for learning this Knowledge Object, e.g. for absorbing it once without repetition.	No
hasDifficultyLevel	A number stating the difficulty level of the KO. The value can either be 1 (Beginner), 2 (Intermediate) or 3 (Advanced).	No
hasEqfLevel	The EQF-level stating the required qualification level of the KO in terms of a number between 1 and 8 in accordance to the EQF	No

	standard[6].	
hasRecommendedAge	Number specifying the recommended age of a learner for this KO.	No
hasMinimalWidth	Size in pixels the device of the learner should at least have in horizontal dimension to display the KO properly.	No
hasMinimalHeight	Size in pixels the device of the learner should at least have in vertical dimension to display the KO properly.	No
hasRecommendedWidth	Size in pixels that the device of the learner should have horizontally in order to view the content optimally.	No
hasRecommendedHeight	Size in pixels that the device of the learner should have vertically in order to view the content optimally.	No
hasSize	Size of the KO content in terms of kB.	No
hasLanguage	Language in which the content of the KO is given.	No
hasKnowledgeType	The KT as specified in the Pedagogical Ontology (D2.1)	Yes
hasMediaType	The MT as specified in the Pedagogical Ontology (D2.1)	Yes
hasSubtitles	Boolean value specifying whether a KO containing a video also provides subtitles.	No
isSuitableForDeaf	Boolean value specifying whether a KO is suitable for deaf learners (i.e. does not contain mandatory audio components).	No
isSuitableForBlind	Boolean value specifying whether a KO is suitable for blind learners (i.e. does not contain mandatory visual components).	No
isSuitableForMute	Boolean value specifying whether a KO is suitable for mute learners (i.e. does not contain mandatory spoken components).	No
hasPreferredLocation	Reference to a location specification where the KO should best be completed. Described in accordance to the SLOM-specific definition of locations (see 4.5).	No

Table 5: Overview of KO-specific metadata and their descriptions

In accordance to this list, a minimal definition of a Knowledge Object in a CCM has to look like in the following example to be considered valid for the reasoning process:

```
<intui:KnowledgeObject rdf:ID="#KO00001">
   <dc:title>Course introduction</dc:title>
   <intui:isLinkedWithLmsElement>LO1201</intui:isLinkedWithLmsElement>
   <intui:isContainedByConceptContainer rdf:resource="#CC00001"/>
   <intui:hasKnowledgeType rdf:resource="#ktOverviewOrientation"/>
   <intui:hasMediaType rdf:resource="#mtVideoPresentation"/>
</intui:KnowledgeObject>
```

[6] See http://ec.europa.eu/eqf/home_en.htm for more information on the European Qualification Framework.

Codelisting 2: Minimal example of a Knowledge Object definition

A more detailed KO definition could look like the following example:

```
<intui:KnowledgeObject rdf:ID="#KO00001">
   <dc:title>Course introduction</dc:title>
   <intui:isLinkedWithLmsElement>LO1201</intui:isLinkedWithLmsElement>
   <intui:isContainedByConceptContainer rdf:resource="#CC00001"/>
   <intui:hasKnowledgeType rdf:resource="#ktOverviewOrientation"/>
   <intui:hasMediaType rdf:resource="#mtVideoPresentation"/>
   <dc:description>Introductory video explaining what this course is
     about and what is required of students to pass.</dc:description>
   <intui:isLinkedWithSlomPackageElement>.../res/LO1201.html
     </intui:isLinkedWithSlomPackageElement>
   <intui:hasEstimatedLearningTime>00:03</intui:hasEstimatedLearningTime>
   <intui:hasEqfLevel>2</intui:hasEqfLevel>
   <intui:hasDifficultyLevel>1</intui:hasDifficultyLevel>
   <intui:hasRecommendedAge>12</intui:hasRecommendedAge>
   <intui:hasMinimalWidth>400</intui:hasMinimalWidth>
   <intui:hasMinimalHeight>300</intui:hasMinimalHeight>
   <intui:hasSize>5000</intui:hasSize>
   <intui:hasLanguage>en</intui:hasLanguage>
   <intui:hasSubtitles>true</intui:hasSubtitles>
</intui:KnowledgeObject>
```

Codelisting 3: Example of a Knowledge Object definition

4.4 Specification of Learning Pathways

4.4.1 Specification of macro Learning Pathways

In order to define a certain macro LP, a cognitive engineer has to create a new sub-property of a fitting macro LP relation from the PO in the respective CM. The Pedagogical Ontology therefore provides seven different base properties. Each of them describes a certain way of how the CCs are ordered and thus encodes didactic information that can be used to determine optimal LPs:

- hasMacroLevelRelation: Most basic property for macro LPs that do not fit in a certain schema.
- hasChronologicalLikeRelation: Property for macro LPs that somehow describe a chronological like relation between Concept Containers.
- hasFromNewToOldLikeRelation: Concretization of chronological like macro LPs specifying that CCs should occur in a new to old sequence. In other words, CC01 is temporarily before CC02.
- hasFromOldToNewLikeRelation: Sibling to the previous relation but in reverse order, i.e. CCs should be ordered from old to new. In other words, CC02 is temporarily after CC01.
- hasHierarchicalLikeRelation: Property for macro LPs that order CCs in accordance to some kind of underlying hierarchical order.
- hasBottomUpLikeRelation: Sub-property of the property above, specifying that the respective hierarchy is traversed in a bottom up manner.

- hasTopDownLikeRelation: Property for hierarchical macro LPs that traverse a CC hierarchy in a top-down-like way.

In general, cognitive engineers should always use the most specific property so that also the relation between LP properties bears meaningful information.

The underlying reason for this approach instead of just using the already available relations is that cognitive engineers might want to use multiple macro LPs of the same type (e.g. to display different hierarchical coherences), but using the same property for two paths induces problems as they become indistinguishable for the INTUITEL Back End. Therefore, by simply creating a new sub-property, the semantics are preserved while the possibilities for modelling the content are increased at the same time.

The following example firstly shows how a new macro LP relation is set up and secondly how it is applied in a CC:

```
<owl:ObjectProperty rdf:about="#hasFlatHierarchicalNext">
   <rdfs:domain rdf:resource="&intui;ConceptContainer" />
   <rdfs:range rdf:resource="&intui;ConceptContainer" />
   <rdfs:subPropertyOf rdf:resource="&intui;hasTopDownLikeRelation" />
   <dc:title>Flat hierarchy LP</dc:title>
   <dc:description>LP expressing a rather flat hierarchy between CCs so
     that the overviews about the rather high-level topics are given before
     presenting the more detailed topics. </dc:description>
</owl:ObjectProperty>
<!-- ... -->
<intui:ConceptContainer rdf:about="#CC00001">
   <!-- ... -->
   <hasFlatHierarchicalNext rdf:resource="#CC00002" />
   <!-- ... -->
</intui:ConceptContainer>
```

Codelisting 4: Example for creating a new macro LP relation and using it to link two CCs

Via the usage of the annotation properties dc:title and dc:description, a cognitive engineer can also add a meaningful plaintext name for the new macro LP relation and add a descriptive text to outline his or her intention regarding this particular LP.

4.4.2 Specification of micro Learning Pathways

The procedure for micro LPs is in this context fully compatible to macro LPs, except that it uses different properties to base the LPs on. They are grouped in accordance to different educational strategies, namely multi-stage and inquiry-based learning, as well as in accordance to Media Types. These base models allow it to interpret didactically meaningful micro LPs in different ways while preserving the semantics of them.

In total, there are eleven object properties that can be used for the creation of micro LPs in a CCM:

- hasMicroLevelRelation: Most basic property for micro LPs that do not fit in a certain schema.

- isMoreAbstractThan & IsMoreConcreteThan: Two different approaches that model micro LPs in respect to a certain order of Media Types.
- isAfterGoodPracticeMultiStage & isBeforeGoodPracticeMultiStage & isAfterSimulatedMultiStage & isBeforeSimulatedMultiStage: Different versions and LO orders of multi-stage learning.
- isAfterOpenInquiryBased & isBeforeOpenInquiryBased & isAfterStructuredInquiryBased & isBeforeStructuredInquiryBased: Different versions and LO orders of inquiry-based learning.

The technical realization is analogous to the macro LPs:

```
<owl:ObjectProperty rdf:about="#hasMultiStageNext">
   <rdfs:domain rdf:resource="&intui;ConceptContainer" />
   <rdfs:range rdf:resource="&intui;ConceptContainer" />
   <rdfs:subPropertyOf
      rdf:resource="&intui;isAfterGoodPracticeMultiStage"/>
   <dc:title>Multi-stage learning</dc:title>
   <dc:description>LP expressing an exemplary order of elements in
      accordance to multi-stage learning.</dc:description>
</owl:ObjectProperty>
<!-- ... -->
<intui:KnowledgeObject rdf:about="#KO000001">
   <!-- ... -->
   <hasMultiStageNext rdf:resource="#KO000002" />
   <!-- ... -->
</intui:ConceptContainer>
```

Codelisting 5: Example for creating a new micro LP relation and using it to link two KOs

For a more detailed elaboration on the nature and meaning of inquiry-based and multi-stage learning, please refer to INTUITEL deliverable 2.1.

4.5 Geospatial aspects in SLOM

SLOM contains means to describe a preferred location for Knowledge Objects, meaning a place where it should best be completed. This is especially interesting for courses that require certain learning material (e.g. lab equipment) or include illustrative material (e.g. buildings in architecture courses).

This is far from trivial, since geospatial aspects are generally a quite complex topic and can get considerably difficult when taking into account that, for instance, different geospatial reference systems (SRS/CRS[7]) could be used to specify a location. There are multiple standards in the field of computer science (for and among different programming languages) and using them usually results in multiple advantages in the long run (e.g. possibility to use third-party software to evaluate the defined coordinates). It consequently is also advisable to use a respective standard for defining locations/coordinates in INTUITEL and SLOM.

[7] A spatial reference system (SRS) or coordinate reference system (CRS) defines a location on the earth and also formulae to convert between each other. A commonly used SRS/CRS is the WGS84, which is used by GPS satellites.

The Consortium therefore selected two ontologies, which go hand in hand - the Basic Geo ontology[8] of the W3C and the GeoNames ontology[9], a free geographical database under a Creative Commons attribution licence.

It has further decided to represent location-based information *as first-class citizens* by making use of concepts from the Geospatial Semantic Web, the central idea of which is to make every physical point, that can be represented by geo-coordinates, identifiable using a URI, i.e., represent locations as web resources. The URIs of those web resources are *dereferenceable* over the HTTP protocol in order to retrieve additional information that describe and characterize a specific location and make it addressable via its URI. This approach allows for annotating locations with metadata and for defining explicit relationships between different locations in an axiomatic way. Moreover, other types of web resources such as people, buildings, places, events etc. can be related to locations, respectively to the URIs that identify them in meaningful and well-defined ways based on the ontologies that are used for describing the relations.

For a first version, SLOM makes use of a basic set of vocabulary elements for describing location-related information, the elements of which have been defined on the basis of both the W3C geo ontology and the GeoNames ontology. As INTUITEL yet evaluates only one geo-aspect, this solution offers an appropriate trade-off between descriptiveness, expressiveness, complexity and extensibility. Especially the latter is a very relevant aspect for the annotation of location-related information and for the addition of further evaluation criteria for Knowledge Objects, but also for the INTUITEL Editor, which could use this to increase its usability.

On possible extension could, for instance, be reverse geo-coding. This would allow cognitive engineers to specify a concrete location (i.e. an address), which will then be converted into a format that is suitable for INTUITEL.

Technically, this has been implemented by adding the following new elements:

- A new class “Location” has been added, which is a subclass of “Feature” from the GeoNames ontology. The Feature class allows annotating locations with information about what a specific location represents, e.g., the type of amenity that a location resource refers to. This adds additional meaning to locations and also allows for a classification and identification of location-based information respectively the resources that refer to them.
- A new object property “hasPreferredLocation” has been added that links a Knowledge Object with an instance of “Location”.
- Three new data properties have been added, which specify the location of the respective “Location” instance.
    - “hasPreferredLocationLatitude” defines the latitude and is implemented as sub-property of wgs84_pos#lat of the Basic Geo ontology. (This already encodes a SRS – WGS84.)

[8] http://www.w3.org/2003/01/geo/
[9] http://www.geonames.org/ontology/

- “hasPreferredLocationLongitude defines the longitude and is implemented as sub-property of wgs84_pos#long of the Basic Geo ontology. (This also already defines WGS84 as SRS for the coordinate.)
- “hasPreferredLocationMargin” defines an tolerance margin (in meters) around the location. This is necessary as location-based services can usually not be assumed to provide 100% accurate information at any time, due to, for instance, a weak GPS signal.

Summarizing, to define a location where a learner should best complete a certain KO, a new instance of “Location” needs to be created, which specifies the coordinates according to the WGS84 reference system as well as a tolerance margin for measurement inaccuracy. By linking the respective KO with this instance, the INTUITEL Back End is able to evaluate spatial information as send by a LMS.

4.6 Future Extensions

The SLOM principles, as they are realized in INTUITEL and especially with the usage of OWL in SLOM, allow extending the knowledge basis even more. While this is not in the focus of the present INTUITEL project, the Consortium nevertheless believes it important to summarize some of the options here to stimulate further discussion and to increase the impact of INTUITEL.

The INTUITEL Consortium has come up with some ideas that could be implemented in future versions of the PO and SLOM. As the project firstly strives after a functional version, these points have been postponed. However, in order to give an impression about the possibilities of the present approach, some of these ideas are described in the following:

- Templated CCMs – Building on the assumption that some CMs will be reused across different courses or even institutions, it might be advantageous to include implementation hints in CMs. It would thus be possible to convey the basic ideas concerning the intentions the original creator had when implementing certain structures or elements. Such statements could, for instance, be textual (e.g. “CC01 should contain an orientation KO”) or technical (e.g. by pre-defining certain KO statements) and could ultimately be used to define templates for a CCM.
- Conditional statements in a LP – Certain parts of a course might require that the system makes sure that the learner has completed previous elements to a certain degree. It should thus be valuable to add conditional statements to LPs. These should be machine-readable and express that, for instance, KO10 should not be recommended until the KO09 has not been completed with a score of at least 50%.
- Optional parts of a LP – Some cognitive engineers might want to add parts to a LP that include LOs with background knowledge, which is nice-to-know or helps to better understand the material as a whole, but are not mandatory for the course. Adding the possibility to mark these segments of a LP as optional would make it possible to specify this in a CM and CCM respectively.
- LO segments on a LP – The basic approach of INTUITEL includes that a learner follows a certain LP. Automatically omitting parts of a LP in a recommendation is critical as this might destroy internal coherences. To provide more recommendation alternatives for a learner, LPs could be extended to specify LO chains, i.e. sequences of LOs that consists of coherent LO groups. This

would allow the reasoner to analyse segments individually and thus to create more recommendations of which a learner could choose from.

5 Transformations

This chapter provides prototypical specifications of how transformation from various formats into the SLOM format will be carried out. The actual implementation is afterwards carried out in context of the tasks 4.2, 4.3 and 4.4.

5.1 Transforming Semantic MediaWiki into SLOM

5.1.1 Semantic MediaWiki Introduction

Semantic MediaWiki was created as an extension to the popular MediaWiki software, which provides the basis for numerous collaborative authoring systems and tools on the Web, where the free encyclopaedia Wikipedia is the most prominent example of a wiki-based system. Wikis in general are well known for their capabilities to collect and share knowledge within and across communities. Although wikis provide facilities to store and retrieve individual facts, their capabilities in expressing relationships between facts and in aggregating factual knowledge are limited.

Semantic MediaWiki, abbreviated as *SMW* in the following, extends the MediaWiki software with semantic features that allow for an annotation of wiki content with machine-processable semantic information.

This chapter provides an overview of the strategies used for the transformation of SMW content into SLOM and vice versa. It outlines transformation semantics and the different transformation approaches from a technical perspective, and can be regarded as a starting point for the detailed SLOM/SMW transformation specification created in Task 4.2. It further discusses aspects together with characteristics of different import and export approaches and also outlines their consequences with respect to the INTUITEL system specification.

5.1.2 Transformation of SMW Learning Content and Annotations to SLOM

For the export of SMW authored learning content into SLOM, a distinction between two different technical approaches should be made:

1) Make use of the INTUITEL Merger as external mapping tool to combine (and map) the SMW ontology elements (properties and categories) to elements of the PO, CM, CCM, and SLOM. The INTUITEL Merger creates the SLOM package and also maintains the transformation and mapping rules.
2) Make use of SMW facilities such as specific extensions or special pages directly incorporated in SMW for the export. This approach allows for more control over the SMW elements to be exported and about the mappings between individual SMW terms and INTUITEL-specific vocabulary elements. This approach also yields a maintenance advantage as SMW terms, the imported vocabulary elements, and the mappings defined between them are hosted in and controlled by one single SMW instance.

One fundamental problem that needs to be considered is that of *SMW namespaces*; usually, terms and properties are defined and used locally within an SMW instance (they exhibit an SWM instance-specific namespace). This embodies consequences when exporting SWM terms (properties, concepts, and categories) in the RDF format since the elements' URIs are defined relative to the base URI of an SMW instance, i.e., they embody a namespace that is local to the SWM instance and thus should only be used in relation to the SMW instance (see [https://semantic-mediawiki.org/wiki/Help:Import_vocabulary] for an example). However, the vocabulary import mechanism built into SMW addresses this issue as it allows for the specification from where a specific semantic property was imported from. This provides SMW authors with the possibility to explicitly declare which of their SMW terms correspond to imported terms. When making use of external mapping tools, namespace mapping semantics need to de defined for each SMW instance separately.

5.1.3 Transformation of SLOM to the Semantic MediaWiki Format

SMW allows to import and reuse terms that are defined in external vocabularies in a controlled way by associating vocabulary elements to SMW terms. In an SMW instance, the associated terms can be used as normal SMW terms. I.e., when the ontology is exported, the SMW exporter directly uses the terms from external vocabularies for describing exported wiki content as defined in the mappings. This way, users are able to author instance data on the basis of external vocabularies using SMW software. When importing properties, they can be associated with a data type, the object a statement refers to.

As SLOM metadata is based on an adapted version of the PO and describes what data are annotated where (CM, CCM and their relation), the elements in SLOM can be mapped to or transformed into SMW properties in a rather straightforward way, although at the costs of losing some of the restrictions defined in those ontologies. As those restrictions cannot be expressed axiomatically in SMW, additional instructions in a human-readable form have to be given to SWM content creators so that they are able to use the INTUITEL ontologies correctly and consistently for creating and annotating INTUITEL-enabled learning material.

One possibility would be to add those instructions to the INTUITEL ontologies in form of annotation properties the content of which can then be integrated as natural texts in the respective element wiki pages during the ontology import. Although this requires that content creators have to read and understand the wiki pages of the imported ontology elements, it can, at least to a certain degree, be ensured that the semantics and restrictions defined in the PO are not lost during the SMW import.

5.1.4 Transformation of Semantics

The following paragraph briefly discusses how the SLOM metadata elements presented in chapter 4 can be imported and used within a SMW. These paragraphs are intended to provide a general idea of how specific aspects of SLOM might be transformed into SMW terms and concepts. The transformation rules and mapping semantics will be specified and elaborated on in more detail in Task 4.2.

- INTUITEL namespaces

  When importing the INTUITEL ontologies using SMWs import facilities, INTUITEL namespaces are preserved and will replace the local SMW namespaces of those terms for which mappings have been defined (see Section 5.1.2). The SMW vocabulary importer takes care of the correct namespace mappings and applies them during the export.

- Definition of Marco Learning Pathways

  SMW supports the definition of semantic properties and sub properties. Custom macroLP properties defined in the CM as sub properties of intui:hasMacroLevelRelation can be directly mapped to SMW compliant properties (that might share identical labels). The range of those SMW compliant macroLP properties are SMW pages that belong to the category CC.

- Integration of Dublin Core Metadata Terms

  The SMW vocabulary importer is capable of handling the import of external vocabulary elements such as dc:creator, dc:contributor, dc:description, and dc:title from the Dublin Core vocabulary (see Section 4.2) and makes them available for an SMW instance. Dublin Core annotation properties can be mapped to individual SMW annotation properties in order to encode annotations for SMW learning content.

- Learning Objects (LO)

  Under the assumption that data about each LO type (KD, CC, KO) always corresponds to one distinct SMW page, i.e., data pertaining to a specific LO type can be represented by one SMW page, respective SMW categories could be defined for each LO type. For instance, SMW pages that represent the knowledge domain (KD) of a SLOM content package are assigned to the Category:KD to distinguish them from pages pertaining to other categories such as Category:CC or Category:KO. In summary, SMW allows us to define separate categories for each INTUITEL-specific LO type and thus distinguish between them in a straightforward way using built-in SMW facilities.

- Knowledge Domain (KD)

  Meta data pertaining to the Knowledge Domain (KD) can be directly transformed into SWM properties. The URL of a wiki page, which is unique by default, can serve as element identifier under the assumption that a knowledge domain is represented by one wiki page. Alternatively, the element identifier can be represented by a comparable SMW property the value of which holds the identifier string given by the INTUITEL Editor.

- Concept Container (CC)

  The LO type Concept Container can also be realised as SMW category;

  A CC is treated in a SWM instance as a single wiki page in order to uniquely identify it and make it referencable by other SMW properties. A CC page contains the metadata annotations as well as a list of all KOs it includes and the CCs it links to (can be auto-generated using a semantic template).

The values of some properties (e.g. reference on KD) might be set automatically by inference.

- Knowledge Object (KO)

  Knowledge and Media Types can either be realized via respective SMW categories or as SMW properties the values of which refer to the SMW page that represents and describes a specific Knowledge or Media Type. While the latter is the more intuitive and, in terms of SMW, flexible approach, the former approach defines a separate category for each Knowledge and Media Type instance, which would result in complex category schemas and would allow to assign different KT and MT categories to one KO. In order to keep category schemes small and manageable, it is advantageous to represent KTs and MTs as values of corresponding SMW properties that also allow for a greater control of allowed values and for greater degree of extensibility.

- The transformation of the other specified metadata elements is quite straightforward and can be realized with built-in SMW facilities.

- CCM-specific metadata

  All metadata elements can be mapped to SMW properties in a straightforward way:
  - Object properties refer to SMW pages (or are realized via SMW #subobjects)
  - Data type properties are mapped to SMW properties with defined restrictions on their value spaces to control the set of allowed values for a specific property and to guarantee the interpretation of their values as members of a specific data type. As side effect, SMW realizes that it does not need to create a separate SMW page for certain property values.

5.2 Transforming SCORM into SLOM

5.2.1 SCORM Introduction

The Sharable Content Object Reference Model (SCORM) defines standards and specifications used in eLearning environments to ensure compatibility and exchangeability of learning content between different systems. Three sub-specifications are defined within SCORM:

- Content Packaging,
- Run-Time and
- Sequencing.

As Content Packaging is the most important with respect to SLOM and its transformations, this chapter will focus on this in the following paragraphs.

SCORM prescribes the storage of learning content within one ZIP file called Package Interchange File (PIF). This file contains amongst other the Sharable Content Objects (SCO) and, more importantly, a manifest file named imsmanifest.xml. This file defines the structure of the course as well as how SCOs can be launched and the respective metadata. An overview of the manifest file is given in the figure below.

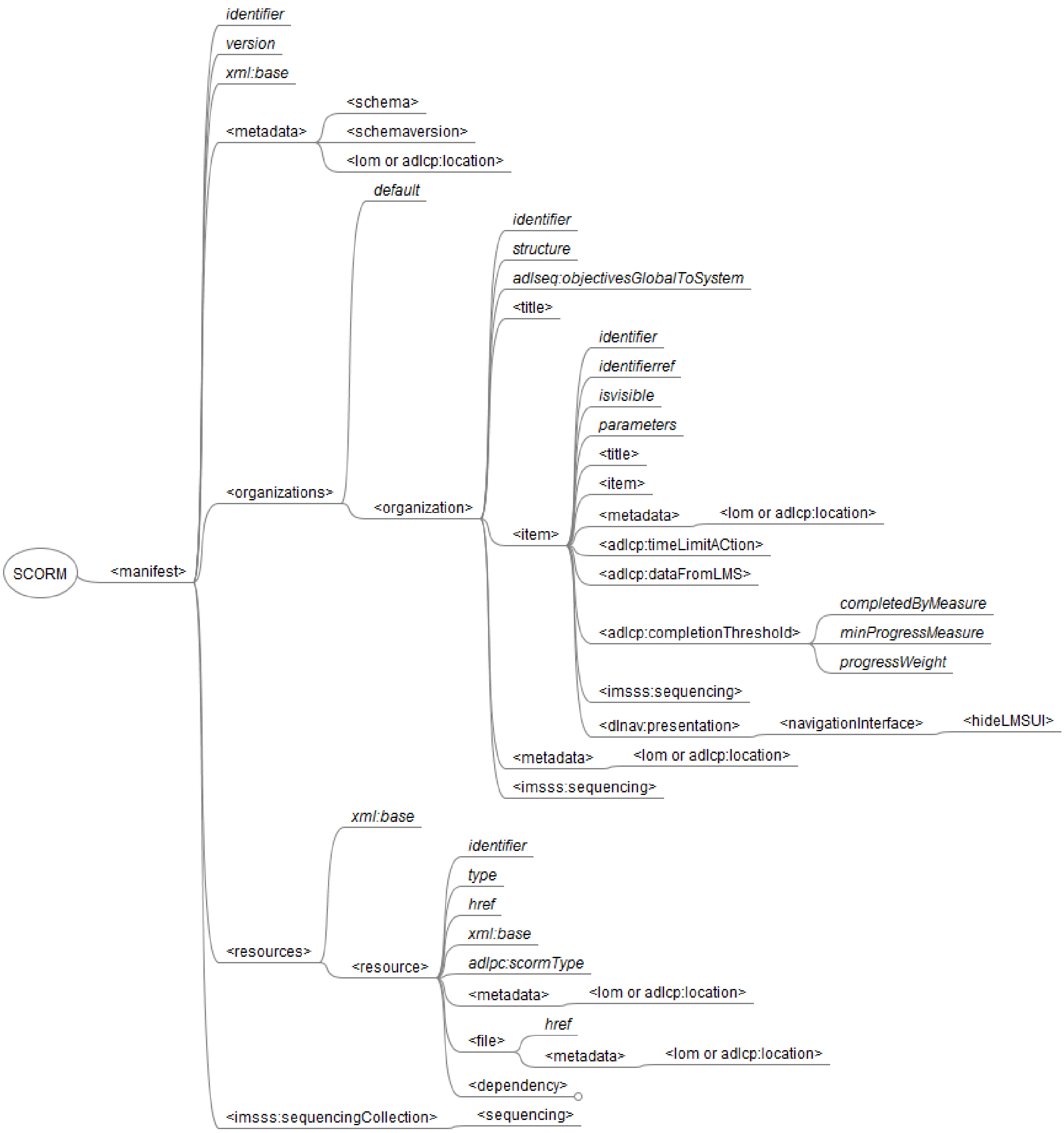

Figure 5: SCORM Manifest Elements Hierarchy[10]

Within the manifest, resources include SCOs and/or assets (SCOs communicate with the LMS whilst assets are static content). Organizations group (a part of) the course into a hierarchical structure. As more than one structure of the same content, this in essence allows for the definition of Learning Pathways as per the INTUITEL PO.

[10] a detailed description of the manifest elements can be found here: http://scorm.com/scorm-explained/technical-scorm/content-packaging/manifest-structure/, 2009, accessed: 06/09/2013 10:45 GMT+1

5.2.2 Transformation of Semantics

Similar to IMS-LD (see section 5.3), transforming SCORM manifests to SLOM and vice-versa can be achieved through XSL Transformations. The detailed specification of the element mapping is outside the scope of this document, but will be dealt with in task 4.3. Here a list of the conceptual mapping is presented:

- Macro-Learning Pathways

  Learning pathways can be achieved either through the organizations elements, i.e. giving the activity tree (even more than one for the same content) or through the SCORM-inherent sequencing capabilities.

- Knowledge Domain

  Each PIF file can represent in essence one Knowledge Domain; thus, a direct mapping can be achieved.

- Concept Containers / Cognitive Content Model.

  Within SCORM, the notion of SCOs best relates to Concept Containers, which is merged with the Cognitive Content Model of INTUITEL (the representation of the Knowledge Objects).

- Knowledge Objects

  It is important for the Knowledge Objects to be executable in a web browser. This might perhaps be the most challenging transformation to be achieved in the mapping of SCORM to SLOM (and vice-versa) as INTUITEL does not prescribe the content delivery platform but aims to be LMS agnostic (to a certain extend). However, simple tools for embedding the content in web sites are available and can be adapted for this purpose.

5.3 Transforming IMS-LD into SLOM

5.3.1 IMS-LD Introduction

IMS Learning Design (IMS-LD) was designed based on the Educational Modelling Language (EML), which was developed by Open University of the Netherlands. IMS-LD is a framework to model and deploy learning processes independent of the eLearning tools and environments as well as the learning domain.

IMS-LD is divided into three levels (A, B and C), each level extending the previous level with further elements, as follows:

- Level A describes the core elements of IMS-LD.
- Level B extends Level A by including generic properties and conditions.
- Level C extends Level B (and thus Level A) by providing means to use notifications.

The course structure in IMS-LD follows the theatrical metaphor, being the activities organized into sequential acts, and played by different roles. The following diagram provides an overview of the core elements of the IMS-LD data model and the hierarchy of these elements[11]:

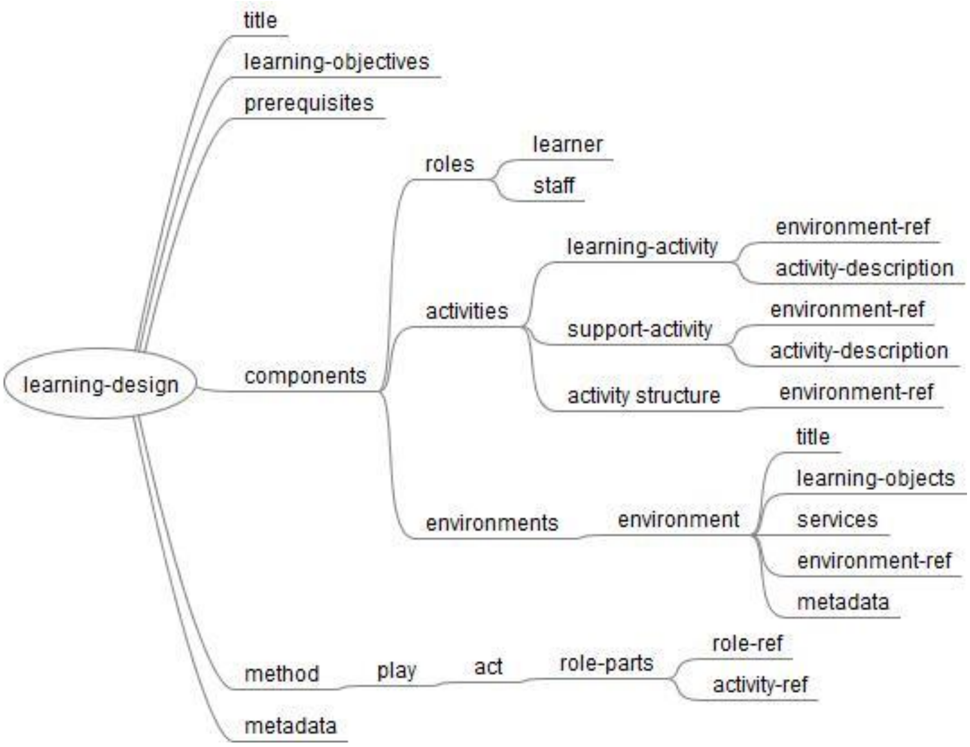

Figure 6: IMS-LD Elements Hierarchy

For the purpose of the mapping from SLOM to IMS-LD and vice-versa, the project will focus mainly on Level A as this forms the core of IMS-LD and realises the bulk of the functionality, and is the minimum requirement for IMS-LD to work.

5.3.2 Transforming of Semantics

Since IMS-LD is represented using the XML-format, so called XSL Transformations (XSL – Extensible Stylesheet Language) can be utilised to transform the SLOM Metadata into IMS-LD and vice-versa. XSLT allow for the automatic transformation of XML into other (XML) formats[12] using the stylesheet as the definition for the transformations. XSLT then uses XPath to search for the defined elements in the XML file and replaces them to form the output file.

In the following a give a brief overview is given of how the transformation between IMS-LD and SLOM can be performed. This is described on a logical level, i.e. leaving the detailed mapping of elements to the actual development of Task 4.4:

[11] A more detailed description can be found here: http://www.reload.ac.uk/ldesign.html, 2004, accessed: 24/07/2013, 16:15 GMT+1.

[12] A good introduction to XSLT is given here: http://www.ibm.com/developerworks/xml/tutorials/x-introxslt/, 2007, accessed: 05/09/2013, 14:15 GMT+1

- Macro-Learning Pathways

  IMS-LD allows essentially for the sequencing of learning activities. If more than one sequence is required (i.e. more than on Learning Pathway is present), `imsld:roles` can be used to distinguish the sequences (each learner can fulfil more than one role). The sequence is realised with the `imsld:activity-structure` (see appendix for further description of elements) element. In addition, sequences can be realised with `imsld:act`, which requires all learners complete an act before the next one can be started.

- Knowledge Domain

  Metadata relating to the Knowledge Domain can be introduced in IMS-LD in the `imsld:learning-objectives` and `imsld:title` elements within each IMS-LD manifest. If this is found unsuitable, the KD information can be included in the `imsld:metadata` element.

- Concept Containers

  As IMS-LD is activity-oriented (versus the content-oriented concept of INTUITEL), the notion of Concept Containers is not easily transferrable to IMS-LD as IMS-LD already structures its content into a sequence of learning activities. However, through `imsld:metadata`, the a association between learning activities and Concept Containers can be achieved.

- Knowledge Objects

  Knowledge Objects can be transformed into the `imsld:learning-object` element which allows for the linking of external resources such as web pages and other material.

- Cognitive Content Model

  For many of the CCM-specific SLOM metadata elements a direct mapping can be realised with the respective elements within IMS-LD including `metadata` and `item` elements.

5.4 Transforming Unstructured Data into SLOM

Although not explicitly in scope of this deliverable, this section provides a specification how even unstructured data (i.e. in form of web-based resources) can be integrated into SLOM. The Consortium expects this approach to contribute to the impact of INTUITEL.

A set of *k* Uniform Resource Identifiers [URI_1, …URI_k], which are supposed to be Knowledge Objects, is called “unstructured data” in the language of INTUITEL. Each of these URIs either

- has an explicit type (meaning MIME-Type) like e.g. “video/mpeg”, or
- points to a file with a MIME-Type that may be inferred from the file name or the content like e.g. “mytitle.mpg”.

In any case, the MIME-Type or the inferred file type determines the Media Type of the INTUITEL KO according to the following table:

MIME-Type:	Media Type:
video/mpg, video/mpeg, …	mtVideoPresentation
audio/mp3, audio/wav, …	mtAudioPresentation
text/svg, image/png, image/jpg, …	mtPhotoPresentation
text/html, text/plain, …	mtTextPresentation

Table 6: MIME-Type and their resulting Media Type in unstructured data transformations

Unstructured data therefore, by definition, does not exhibit any advanced semantics for presentation or learning. It may nevertheless be assumed that a set of URIs for a learning purpose is accompanied by a learning pathway, e.g. the cognitive engineer will have a certain vision of a sequence in which these web pages should be traversed.

The software tool transforming this URI collection into a SLOM representation has to do the following:

- Import them from the WWW either as single media objects or as complete web pages and store them locally in a content directory.
- Create a file imsmanifest.xml, which contains a reference to each of these objects such that they may be played sequentially in a SCORM player. The sequence is given by the sequence of URI in the collection
- Create a directory **intuitel** and copy/create the necessary files. Then, open the INTUITEL Editor to add metadata to these KOs.

6 Adjustments

As the work on INTUITEL progresses, more and more insights are being obtained. This also implies that some of the previous findings need to be refined in order to display the current status of knowledge of the project. This includes that some of the previous documents are updated to contain all relevant information. The following sections provide insights about how the documents have been changed or extended, why this was necessary and how the project benefits from this.

6.1 Pedagogical Ontology

#	Change:	Reason:
01	Restructuring and renaming of macro LP relations to “…like”.	Learning Pathways describe sequences in which LOs should be completed to be didactically reasonable. The initial PO stated that hierarchical relations should be modeled via e.g. a “hasChild” or “hasParent” relation. This however is problematic as not all hierarchies are linear, i.e. the semantics are incorrect. Restructuring macro LP relations and renaming them to state a rather hierarchical-like relation accounts for this issue.
02	Adjustment of micro LP definitions	Originally, micro LPs were planned to result implicitly from certain properties of KOs. This proved to be problematic and has thus been changed to also be realized via object properties. The respective characteristics of micro LP relations have been adjusted accordingly (i.e. now declared as transitive instead of functional).
03	Removing of deprecated/obsolete elements	Some of the elements in the PO are leftovers of now obsolete coherences (e.g. indirect modeling of micro LPs on basis of KO KTs). These elements have been removed to avoid misunderstandings and to keep the ontology lightweight.
04	Completion of micro LPs	Some of the present micro LP relations had yet missing instantiations, ranges or domains. This has been completed.
05	Consistent camel-casing	The naming of elements of the PO uses camel-casing for a better readability. This was yet inconsistent and has been adjusted.
06	Introduced new properties	New properties have been added to add more information to the elements of CMs and CCMs.
07	Correction of typing errors	Some of the elements had minor typing errors that have been adjusted (e.g. “then” has been changed to “than”).
08	Added License	Added a license placeholder in the PO that will be replaced when the project has come to an agreement which one will be used for the PO and similar files.
09	Added more descriptive texts	In accordance to the usage of the Dublin Core Standard, texts in the PO have been adjusted and added.
10	Changed KT-structure	Moved KTs single-choice, multiple-choice and true/false to ticking assignment in order to reflect the correct coherence between them.

11	Singulars	In order to achieve a consistent labeling of elements in the ontology, all names have been adjusted to be singulars (e.g. LearningObject instead of LearningObjects).
12	Restructuring of class hierarchy	Added a new class for Learning Object property types and made MT and KT sub-classes of it to better categorize these classes.

Table 7: Overview of adjustments of the PO

6.2 Data Model

For clarity, the XML Schema files for the validation of messages between the LMS and INTUITEL have been divided to reflect which module initiates the message. Therefore, the XSD from deliverable 1.1 has been split into:

http://www.intuitel.eu/public/intui_DMLPM.xsd

http://www.intuitel.eu/public/intui_DMLMS.xsd

http://www.intuitel.eu/public/intui_DMSLOM.xsd

The ‘intui_DMLPM.xsd’ contains the XML Schema definition for data originating from the Learning Progress Model (LPM). To include this namespace in an XML document for validation, the XML document should have the following root element:

```
<intuilpm:INTUITEL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intuitel.eu/public/intui_DMLPM.xsd
http://www.intuitel.eu/public/intui_DMLPM.xsd"
 xmlns:intuilpm="http://www.intuitel.eu/public/intui_DMLPM.xsd">
</intuilpm:INTUITEL>
```

Codelisting 6: Example root element for a message originating from the LPM

A corresponding test file has been placed in the location:

http://www.intuitel.eu/public/intui_DMLPM_Test.xml

The ‘intui_DMLMS.xsd’ contains the XML Schema definition for data originating from the LMS. To include this namespace in an XML document for validation, the XML document should have the following root element:

```
<intuilms:INTUITEL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intuitel.eu/public/intui_DMLMS.xsd
http://www.intuitel.eu/public/intui_DMLMS.xsd"
 xmlns:intuilms="http://www.intuitel.eu/public/intui_DMLMS.xsd">
</intuilms:INTUITEL>
```

Codelisting 7: Example root element for a message originating from the LMS

A corresponding test file has been placed in the location:

http://www.intuitel.eu/public/intui_DMLMS_Test.xml

The 'intui_DMSLOM.xsd' contains the XML Schema definition for data originating from the SLOM toolset. To include this namespace in an XML document for validation, the XML document should have the following root element:

```
<intuislom:INTUITEL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.intuitel.eu/public/intui_DMSLOM.xsd
http://www.intuitel.eu/public/intui_DMSLOM.xsd"
 xmlns:intuislom="http://www.intuitel.eu/public/intui_DMSLOM.xsd">
</intuislom:INTUITEL>
```

Codelisting 8: Example root element for a message originating from the SLOM toolset

A corresponding test file has been placed in the location:

http://www.intuitel.eu/public/intui_DMSLOM_Test.xml

It is also possible to validate a document containing a mixture of messages origination from the INTUITEL system and from the LMS. In this case one as to use the file:

http://www.intuitel.eu/public/intui_DM.xsd

which imports both of the above. To include this namespace in an XML document for validation, the XML document should have the following root element:

```
<intui:INTUITEL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.intuitel.eu/public/intui_DM.xsd
http://www.intuitel.eu/public/intui_DM.xsd"
    xmlns:intuidm="http://www.intuitel.eu/public/intui_DM.xsd"
    xmlns:intuilms="http://www.intuitel.eu/public/intui_DMLMS.xsd"
    xmlns:intuilpm="http://www.intuitel.eu/public/intui_DMLPM.xsd">
</intui:INTUITEL>
```

Codelisting 9: Example root element for a general message without a specific origin

A corresponding test file has been placed in the location:

http://www.intuitel.eu/public/intui_DM_Test.xml

Please note that these URLs all link to the most recent versions of the respective files. Specific version can be addressed by adding the release year and month the URL (e.g.: http://www.intuitel.eu/public/2013/09/intui_DM.xsd).

7 Appendix

7.1 IMS-LD Level A Element Descriptions

Element	Description	Multiplicity
<imsld:learning-design>	This element specifies the learning design.	undefined
<imsld:learning-objectives>	Learning objectives describe the intended outcome for learners. There are two locations where learning-objectives & prerequisites are specified: - At the level of the learning design (in the root of learning-design) - At the level of learning-activities (within learning-activities). The first ones are a more general description; the second ones are more concrete	0..1
<imsld:prerequisites>	Prerequisites are the entry-requirements for students, such as any pre-knowledge needed	0..1
<imsld:components>	Specifies the building blocks used in the method section.	1
<imsld:method>	The <method> element contains a sequence of elements for the definition of the dynamics of the learning process	1
<imsld:metadata>	Include IMS Meta-Data here, using its namespace	0..*
<imsld:roles>	The <roles> element specifies the roles distinguished in the learning design	1
<imsld:activities>	The <activities> element contains a choice for different activity definitions such as learning activities and support activities. It also provides for the definition of activities structures.	0..1
<imsld:environments>	The <environments> element is a container for the <environment> elements.	0..1
<imsld:learner>	In every learning design there is at least one learner-role. Learners can be 'nested', meaning that a role may be divided in sub roles	1..*
<imsld:information>	The <information> element specifies the set of items pointing to the resource(s) where the information can be found	0..1
<imsld:staff>	Staff members can be 'nested', meaning that a role may be divided in sub roles	1..*
<imsld:learning-activity>	The <learning-activity> element contains a sequence of elements for learning-activity definitions.	1
<imsld:support-activity>	The <support-activity> element contains a sequence of elements to define support-activities	1
<imsld:activity-structure>	An activity-structure groups activities in sequences or selections.	1

	The tree is handled depth first (and not breadth first). If there is a sequencing element it overrules the Attributes: structure-type and number-to-select.	
<imsld:environment-ref>	Refers to an environment in this package	0..*
<imsld:activity-description>	Alias: task. The activity-description is the actual cue given to the user (rendered in the user-interface) to describe the activity to be performed by the user.	1
<imsld:complete-activity>	Contains a choice of elements to specify when an activity is completed. When this element does not occur, the activity is set to 'completed'	0..1
<imsld:user-choice>	This element is used in the completed element of activities and specifies that the user may decide him or her self when the activity is completed	1
<imsld:time-limit>	The time limit specifies that it is completed when a certain amount of time have been passed, relative to the start of the run of the current unit of learning.	1
<imsld:on-completion>	When an activity, act, play or unit-of-learning is completed, the optional actions contained in this element are executed.	0..1
<imsld:feedback-description>	The underlying item elements points to a resource (of type webcontent or imsldcontent), where the feedback description can be found. After completion this text becomes visible.	0..1
<imsld:role-ref>	Role refers to the identifier of the resource of the role.	0..*
<imsld:information>	The <information> element specifies the set of items pointing to the resource(s) where the information can be found.	0..1
<imsld:learning-activity-ref>	Refers to a learning-activity.	1
<imsld:support-activity-ref>	Refers to a support-activity.	1
<imsld:unit-of-learning-href>	This element refers to the resource of a unit-of-learning (uol).	1
<imsld:activity-structure-ref>	Reference to an activity-structure.	1
<imsld:environment>	Contains a sequence of elements to model an environment.	1..*
<imsld:learning-object>	Learning objects are reflected with an included schema (e.g., IMS QTI) or can reference resources through the item elements	1
###other	The ###other construct is provided as an extension mechanism to allow additional elements to be added below <learning-	0..*

	object>. These additional elements would be defined by a modified schema or namespaced extension.	
<imsld:service>	A service is a declaration of a service facility which has to be bound during instantiation of a run of a unit of learning.	1
<imsld:send-mail>	This service is used to send mail to users in roles (with mail address in property for level b/c).	1
<imsld:email-data>	This is used for send-mail purposes (as a service in the environment, or in notifications).	1..*
<role-ref>	Role refers to the identifier of the resource of the role	1
<imsld:conference>	The elements participant, observer, conference-manager, moderator facilitate the setting of the user rights in the conferences.	1
<imsld:participant>	Specifies who the participants are in the conference. Participants can read (listen/see) the information, and can contribute to the conference.	1..*
<imsld:observer>	Specifies who the observers are in the conference. Observers have only reading rights; they may not contribute.	0..*
<imsld:moderator>	Specifies who the moderators are in the conference. Moderators are persons who have the right to control and change the contributions of participants before they are made visible to other participants or observers.	0..*
<imsld:conference-manager>	The conference manager is allowed to create new sub conferences and delete conferences he/she created.	0..1
<imsld:item>	See IMS CP.	1
<imsld:index-search>	Contains a sequence of elements that declare an index and/or search service facility.	1
<imsld:index>	A choice of elements to specify indexing aspects, used to set up a search service.	1
<imsld:index-class>	This element selects the class to make the index on.	1
<imsld:index-element>	This element selects the element to make the index on.	1
<imsld:index-type-of-element>	In this element the type of element to index on is entered. Only one element name per index-type-element occurrence.	1
<imsld:search>	This element specifies how a user can access the indexed entities.	1
<imsld:play>	The play is the root element when interpreting the learning design.	1..*

<imsld:complete-unit-of-learning>	A choice of elements to specify when a unit-of-learning is completed. When this element does not occur, the completed is set to 'unlimited'.	0..1
<imsld:when-play-completed>	This element states that an unit-of-learning is completed when the referenced play(s) is (are) completed.	1..*
<imsld:time-limit>	The time limit specifies that it is completed when a certain amount of time have been passed, relative to the start of the run of the current unit of learning.	1
<imsld:on-completion>	When an activity, act, play or unit-of-learning is completed, the optional actions contained in this element are executed.	0..1
<imsld:feedback-description>	The underlying item elements points to a resource (of type webcontent or imsldcontent), where the feedback description can be found. After completion this text becomes visible.	0..1
<imsld:act>	A play consists of a series of acts and an act consists of a series of role-parts. An act represents a series of concurrent role-parts.	1..*
<imsld:complete-play>	A choice of elements to specify when a play is completed. When this element does not occur, the completed is set to 'unlimited'.	0..1
<imsld:when-last-act-completed>	This element states that a play is completed when the last act is completed.	1
<imsld:role-part>	A play consists of a series of acts and an act consists of a series of role-parts. A role-part relates exactly one role to exactly one type of activity (including the performance of another unit-of-learning and activity-structures).	1..*
<imsld:complete-act>	A choice of elements to specify when an act is completed. When this element does not occur, the completed is set to 'unlimited'.	0..1

Table 8: IMS-LD Level A Elements Description