



## Deliverable 3.1

### Overall System Design Document

DISSEMINATION LEVEL		
<b>PU</b>	Public	<b>X</b>
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	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.:	D3.1
Document name:	D3.1 Overall System Design Document
Work Package	3
Associated Task	1
Nature <sup>1</sup>	R
Dissemination Level <sup>2</sup>	PU
Version:	1.00
Actual Submission Date:	2013-03-31
Contractual Submission Date	2013-03-31
Editor:	Peter A. Henning, Florian Heberle
Institution:	HSKA
E-mail:	<a href="mailto:peter.henning@intuitel.eu">peter.henning@intuitel.eu</a> , <a href="mailto:florian.heberle@intuitel.eu">florian.heberle@intuitel.eu</a>

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.

The author is solely responsible for its content, it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

<sup>1</sup> R=Report, P=Prototype, D=Demonstrator, O=Other

<sup>2</sup> 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	2012-12-21	Document drafted	Florian Heberle, Peter Henning	HSKA
0.02	2013-01-11	Document revised	Florian Heberle, Peter Henning	HSKA
0.03	2013-02-08	Document revised	Florian Heberle, Peter Henning	HSKA
0.04	2013-02-21	Document revised	Florian Heberle, Peter Henning	HSKA
0.05	2013-02-11	Document revised	Florian Heberle, Peter Henning	HSKA
0.06	2013-03-22	Document revised	Florian Heberle, Peter Henning	HSKA
0.07	2013-03-26	Document updated	Florian Heberle, Peter Henning	HSKA
1.00	2013-03-31	Document finalized	Peter Henning	HSKA

### Distribution List

Date	Issue	Group
2013-01-07	Request for comments	WP03
2013-02-21	Request for comments	All
2013-03-12	Request for comments	All
2013-03-22	Request for comments	All

## List of Contributions

Date	Organization(s)	Person(s)	Contribution
2012-12-03	UNIR, UVA, TIE	Luis de la Fuente, Elena Verdú Pérez, Oscar Garcia Perales, Daniel Burgos	Architectural recommendations
2012-12-10ff	UVIE, HSKA	Christian Swertz, Alexander Schmoelz, Florian Heberle	Emails reg. LMS interfaces and usability
2012-12-14	FZI, HSKA	Jürgen Bock, Catherina Burghart, Florian Heberle, Peter Henning	Meeting reg. implementation of (OWL-)reasoning
2013-01-11	UVA	Elena Verdú Pérez	Overall comments
2013-01-14ff	UNIR, HSKA	Luis de la Fuente, Florian Heberle	Email-discussion reg. INTUITEL Merger and Editor
2013-01-16f	UVIE, HSKA	Christian Swertz, Florian Heberle	Emails reg. learner monitoring
2013-02-08	TIE	Oscar Garcia Perales	Overall comments
2013-02-06ff	UVA, IMC, ELS, TIE, HSKA	Elena Verdú Pérez, Sven Steudter, Elisabetta Parodi, Oscar Garcia Perales, Florian Heberle	Email-discussion reg. LMS interfaces and Back End proxies
2013-02-12ff	HSKA, TIE	Peter Henning, Florian Heberle, Oscar Garcia Perales	Emails reg. Back End proxies and TIE Smart Bridge
2013-02-26	ELS	Elisabetta Parodi	Overall comments
2013-03-01	UVA	Elena Verdú Pérez	Comments reg. connection between Back End and LMS
2013-03-04	TIE	Oscar Garcia Perales	Overall comments
2013-03-06	URE	Daniel Thiemert	Comments esp. reg. natural language unit
2013-03-08ff	TIE, HSKA, UVIE, UNIR, IMC, UVA, IOSB, ELS	Oscar Garcia Perales, Peter Henning, Christian Swertz, Luis de la Fuente, Sven Steudter, Elena Verdú Pérez, Bela-Andreas Bargel, Elisabetta Parodi	Discussion about the INTUITEL communication layer
2013-03-15	UVIE	Christian Swertz	Validation and comments on PO and CM chapter (5)
2013-03-25	URE	Atta Badii	Review
2013-03-31	HSKA	Peter Henning	Finalization

## Table of Contents

<b>1</b>	<b>Preface .....</b>	<b>10</b>
<b>2</b>	<b>Overview .....</b>	<b>12</b>
<b>2.1</b>	<b>Structural overview .....</b>	<b>12</b>
<b>2.2</b>	<b>Data exchange .....</b>	<b>13</b>
<b>2.3</b>	<b>INTUITEL as a reactive system.....</b>	<b>15</b>
<b>3</b>	<b>User roles and their workflows .....</b>	<b>17</b>
<b>3.1</b>	<b>Learner .....</b>	<b>17</b>
<b>3.2</b>	<b>Content creator.....</b>	<b>17</b>
3.2.1	Content creator 1 – Learning content .....	18
3.2.2	Content creator 2 – Cognitive Model .....	18
3.2.3	Content creator 3 – Metadata annotator .....	19
<b>4</b>	<b>Learning Management System integration.....</b>	<b>20</b>
<b>4.1</b>	<b>Detailed overview .....</b>	<b>20</b>
<b>4.2</b>	<b>Connections to other INTUITEL components .....</b>	<b>20</b>
<b>4.3</b>	<b>General services.....</b>	<b>21</b>
<b>4.4</b>	<b>Tutorial Guidance - TUG .....</b>	<b>22</b>
<b>4.5</b>	<b>Learning Object Recommender - LORE .....</b>	<b>23</b>
<b>4.6</b>	<b>User Score Extractor – USE .....</b>	<b>26</b>
<b>4.7</b>	<b>Shared functionalities .....</b>	<b>27</b>
<b>4.8</b>	<b>Privacy issues.....</b>	<b>28</b>
<b>4.9</b>	<b>Enhanced navigation functionality.....</b>	<b>28</b>
<b>4.10</b>	<b>Architectural remarks for the LMS integration .....</b>	<b>29</b>
<b>5</b>	<b>Pedagogical Ontology, Domain Model and Cognitive Model .....</b>	<b>32</b>
<b>5.1</b>	<b>Detailed overview .....</b>	<b>32</b>
<b>5.2</b>	<b>Connections to other INTUITEL components .....</b>	<b>32</b>
<b>5.3</b>	<b>Pedagogical Ontology.....</b>	<b>33</b>
<b>5.4</b>	<b>Domain Model .....</b>	<b>35</b>
<b>5.5</b>	<b>Cognitive Model.....</b>	<b>35</b>

<b>6</b>	<b>Semantic Learning Object Model .....</b>	<b>38</b>
6.1	Detailed overview .....	38
6.2	Connections to other INTUITEL components .....	39
6.3	SLOM specification.....	40
6.4	INTUITEL Editor .....	40
6.5	INTUITEL Merger .....	41
6.6	SLOM repository .....	43
6.7	SLOM metadata repository.....	43
6.8	Extended architectural recommendation for implementation .....	44
<b>7</b>	<b>INTUITEL Back End .....</b>	<b>46</b>
7.1	Detailed overview .....	46
7.2	Connections to other INTUITEL components .....	47
7.3	Learning Progress Model – LPM .....	48
7.3.1	Learning Progress Model .....	48
7.3.2	User database interface.....	50
7.3.3	User database .....	50
7.3.4	Interface to INTUITEL Engine .....	51
7.4	INTUITEL Engine .....	51
7.4.1	Query Builder .....	51
7.4.2	Reasoning Broker .....	52
7.4.3	Reasoning Engine .....	52
7.4.4	Recommendation Rewriter.....	53
7.4.5	Natural language unit .....	54
<b>8</b>	<b>Step by step usage scenarios .....</b>	<b>56</b>
8.1	How to create an INTUITEL-enabled course? .....	56
8.2	Creation of a Cognitive Model with the help of Protégé .....	57
8.3	Creation of a Cognitive Model with the help of an INTUITEL specific tool .....	57
8.4	LMS user learns with an INTUITEL enabled course .....	57
<b>9</b>	<b>Summary and Outlook .....</b>	<b>59</b>
<b>10</b>	<b>FAQ .....</b>	<b>61</b>

## List of figures

Figure 1: Structural overview .....	12
Figure 2: Structural overview of the LMS integration .....	20
Figure 3: Example for LO recommendation integration with the Amazon star-symbols.....	24
Figure 4: Example for LO recommendation integration with a recommendation box.....	25
Figure 5: Integration of a common data storage interface .....	27
Figure 6: INTUITEL integration with direct access to the LMS's DB .....	30
Figure 7: INTUITEL LMS interfaces as direct part of the LMS with an asynchronous message delivery component .....	31
Figure 8: Detailed structural overview of the Pedagogical Ontology and Domain/Cognitive Model...	32
Figure 9: Nesting of pedagogical elements .....	34
Figure 10: Simplified demonstration of a Cognitive Model with the Knowledge Domain "Cooking" ..	37
Figure 11: Detailed structural overview of the SLOM part .....	38
Figure 12: Detailed structural overview of the INTUITEL Back End .....	46

## List of tables

Table 1: Listing of tasks involved in LMS integration .....	20
Table 2: Listing of tasks involved in PO and DM / CM.....	32
Table 3: Listing of tasks involved in SLOM.....	38
Table 4: Listing of tasks involved in the INTUITEL Back End.....	46
Table 5: Summary of INTUITEL modules .....	59

## Glossary

Term	Explanation
DOW	Description of work
C	Content Creator
C[x]	Content creator number x (different roles of content creators)
CC	Concept Container
CM	Cognitive 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



## List of agreed terms and definitions

Term	Meaning
Course	The topmost cognitive container of an INTUITEL enhanced system, equivalent to the Knowledge Domain (KD).
Learner	A learning person using the INTUITEL enhanced system.
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.
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.
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.
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.
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.
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.
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.
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.
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.
CM	A Cognitive Model is the INTUITEL description of a domain of knowledge using the terms and relations as specified in the Pedagogical Ontology.
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.

# 1 Preface

This document is part of INTUITEL work package 03, task 1 and is one of two documents of the deliverable D3.1 “Overall System Design Document”. It is based on the other part of this deliverable, the system design overview diagram, which gives a detailed graphical view of the whole system architecture. Some elements of this illustration will partially or in an abstracted form be integrated into this document to enable a better readability.

The overall system design is based on the original description from the DOW and the results of the other two parallel running work packages 01 and 02. Some of the content produced in these work packages will be incorporated into this document if this would lead to a better comprehensibility. However, for more detailed information, please consult the output of the respective WPs.

Other deliverables connected to this document are:

- D1.1, XML Schema for USE/TUG/LORE
- D2.1, Pedagogical Ontology

The main goal of this specification is to describe and demonstrate “the interplay of the various technical and pedagogical components” [DOW p.17] of INTUITEL. Therefore, the document is structured as follows:

- Firstly, an overview of the system as a whole and the interconnections is given. Also, the main user roles and their typical workflows are described.
- Secondly the individual modules are explained in a higher level of detail. This includes an overview of the respective modules and the interplay between them. If appropriate, some architectural suggestions for the tasks involved are made.
- Thirdly, usage scenarios are outlined. This will give an impression as to how learners and content creators will, in accordance to the system design, create INTUITEL conform learning material and consequentially learn with them. Furthermore, a concluding chapter provides a short summary of the system design and a FAQ section creates additional clarification.

For convenience and for a better understanding of the content, a glossary of INTUITEL typical abbreviations and their meanings has been added after the table of contents. To further avoid misconceptions or wording issues between the different learning management systems, a list of agreed on terms and their definitions has also been added.

Some parts of INTUITEL, especially concerning the Back End, have not yet been specified by the respective WP-teams. Some of these elements are ‘cutting-edge’ research. This means that for some aspects only ideas or rough concepts for their implementation exist so far. This document does nevertheless give an overview about these components and their planned functionalities. The reader should be aware that changes to those modules are likely to occur when the respective WPs and tasks progress. The main aim is to give an overview of their current status from a system design perspective.

As a short overview, the following deliverables will establish further clarification about these issues and provide more detailed insights:

- D3.2, Learning Progress Model, due by June 2013  
A set of rules for pedagogical reasoning in textual as well as processable format, describing how a given learner situation has to be translated into a learning object recommendation and tutorially guiding dialog components. Also contains the proper ontology for the LPM.
- D3.3, LPM Communication Standard, due by September 2013  
A set of connectors and software components which enable the various parts of INTUITEL to communicate among each other.
- D4.1, Specification of SLOM, due by September 2013  
A textual report, description of the new metadata enhanced Semantic Learning Object Model.
- D5.1, INTUITEL Engine, due by September 2014  
A computer program component to perform machine based reasoning on the data of the Learner Progress Model and thereby providing a Learning Object Recommendation, plus its manual as a written report. The report will also contain a manual on how different reasoning engines can be used in INTUITEL.
- D6.1, INTUITEL Merger, due by September 2014  
A computer program component that merges several SLOM input sources into a common representation according to the SLOM container format, plus its manual as a written report.
- D7.1, INTUITEL Editor, due by September 2014  
A computer program component which allows the new semantic annotation of unstructured learning material and the modification of existing annotations, plus its manual as a written report.

Please note that some of the tasks responsible for the creation of these deliverables run longer than the mentioned submission dates. This means that some of these documents will be focused on the architectural and conceptual implementation of the respective components. The actual realization follows in the months afterwards.

## 2 Overview

### 2.1 Structural overview

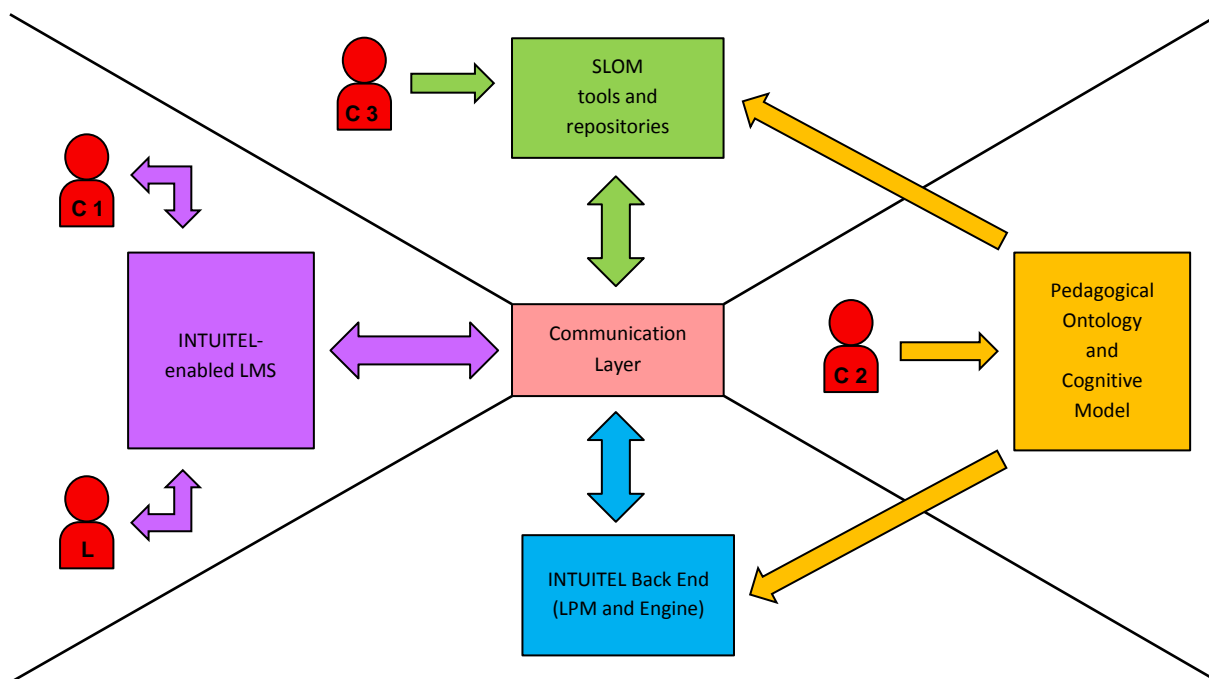


Figure 1: Structural overview

Figure 1 represents the architectural overview of the INTUITEL system and how it interacts with the components and the main actors which make use of it. The actors can be divided in two different groups:

- 1) Learners (L in the diagram)
- 2) Content Creators (C[x] in the diagram)

Due to the complexity of the INTUITEL system and for a better readability and comprehension, Figure 1 shows the main components and not their definitive composition. For an extensive and detailed view of the overall system, see the system design overview diagram.

There are four main components in INTUITEL that are working together to create and present a user specific recommendation of learning objects. For a more detailed explanation, please consult the respective chapters of this document.

- 1) INTUITEL enabled LMS (cf. chapter 4)

The minimal requirement for a LMS to be considered compatible with INTUITEL is to include the TUG interface (see 4.4) and the general services component (see 4.3). This enables the data exchange between the LMS, the learner and the INTUITEL system via the communication layer. In order to be more learner-friendly, the USE (see 4.6) and LORE (see 4.5) interfaces should also be implemented.

These interfaces provide INTUITEL with the functionality to a) initiate a communication with the learner, b) recommend LOs to the individual learners, c) access previous learner knowledge and environmental data and d) monitor learner behaviour and progress.

In general, all INTUITEL dependent changes to a LMS should be as minimal as possible.

## 2) Pedagogical Ontology and Cognitive Model (cf. chapter 5)

This part defines the terms and relationships as well as the domain specific information. The Pedagogical Ontology lists the different pedagogical entities that are needed to describe learning material. The actual representation of a domain of knowledge is the Cognitive Model, which outlines all relevant chunks of information, needed to master that particular domain.

## 3) SLOM tools and repositories (cf. chapter 6)

The Semantic Learning Object Model brings the learning material, the pedagogical models and domain information together. In the first place, SLOM is a specification as to how to pack all these elements into one container. This allows an easy provision, versioning and exchange. The SLOM module also contains tools, providing an efficient and user-friendly GUI to create and edit the SLOM. A file and a metadata repository are specified and created to store this data and to make it available for the recommendation process.

## 4) INTUITEL Back End (cf. chapter 7)

This module consists of a variety of sub-modules that use the available (meta-) data to create learner specific recommendations and forward them to the Learning Management System. The reasoning process mainly relies on two core elements. The first stage is the Learning Progress Model, which manages learning sessions and determines the learner's position in the Cognitive Model. It therefore uses an INTUITEL specific user database, which stores a variety of (historic) information and thus allows a more personalised reasoning. The actual recommendation is created by the INTUITEL Engine afterwards (second stage).

## 2.2 Data exchange

As depicted in Figure 1, INTUITEL contains a component that manages the data exchange related aspects inside the INTUITEL network, the communication layer. The description and specification of this module is not part of this document. Likewise, the different connections between the INTUITEL components and that layer are not included. This is the task of another document which investigates and describes these aspects in detail and will be created in the context of WP 03, task 3 (D3.3).

Since INTUITEL connects multiple different technologically advanced and diverse components, it is crucial to evaluate different possibilities regarding this issue. This ensures that the data exchange in INTUITEL is carried out in an efficient, secure and optimised way. Special emphasis will be put on the importance of it being compatible with the various technological guidelines and possibilities of the different LMSs, as well as on regarding the principal INTUITEL guideline to alter the respective LMSs as little as possible. With this approach, the long-term impact and scalability of INTUITEL is enhanced.

Nevertheless, the following remarks will give a short overview of the connections between the INTUITEL components, in order to convey the underlying ideas and conditions that need to be managed.

All modules described in section 2.1 may, but do not necessarily have to, run on the same machine. Consequently, the connection between them is designed to also work across different network segments (e.g. the Internet).

In a later practical setting, the LMS and INTUITEL will most likely not run on the same physical machine to avoid performance bottlenecks. This is reasonable since INTUITEL will have to process a not yet foreseeable quantity of information per learner at the same time. Depending on the performance of the reasoning process, the required amount of data needed for that and the number of learners to manage, the Back End might require a considerable amount of CPU time and memory.

To speed up the system, it is advisable to use the two physical machines in the same local area network. Depending on the actual usage scenario, it might be necessary to split up the system even further into more independent servers, in order to guarantee a certain level of performance. In the development process, the INTUITEL consortium will therefore identify the components which allow a further segmentation of the INTUITEL system. Furthermore, it must be taken into account that in some use cases the different system components might run on a third-party server farm (e.g. the cloud). Consequently, the data exchange must be carried out in a way that is optimal for nodes connected via a local (secure) network, the Internet or possibly even both. The communication in INTUITEL should thus be realisable with a technology supporting all those usage scenarios.

Since it has not yet conclusively been specified how information will be exchanged and because it might be possible that INTUITEL will support different transmission technologies, the INTUITEL Data Model (D1.1) has been specified in a transmission-neutral way. All information transmitted between INTUITEL and the LMS is contained in XML-elements that can be embedded as payload in different protocols. This caters for a high level of flexibility and allows INTUITEL to be usable with various data transmission technologies.

The Pedagogical Ontology and the Cognitive Models have an exceptional position in these deliberations. Being written in OWL (cf. deliverable 2.1), an easy exchange, publication, storage and access of them is possible. The files will either be stored locally or made available via the Internet. There is no common endpoint in the INTUITEL system managing them. Consequently, they don't need to be included in the INTUITEL network such as, for instance, in the LMS. They could be taken from various sources. The SLOM toolset and the INTUITEL Back End will provide methods to access them in a suitable way (e.g. by using an OWL-framework).

In general, the access and transmission of sensitive user data via unsecure networks is an important issue INTUITEL has to deal with. This must be considered in all usage situations between the LMS, the SLOM toolset and the INTUITEL Back End. Regarding the connection between INTUITEL and the LMS, sensitive user data is, for instance, transmitted when the current learning progress of a learner is sent to the Back End or in the context of SLOM, only the pedagogical personnel of an organisation are allowed to access to the pedagogical metadata.

To assure data safety, INTUITEL differentiates two types of authentication:

1) Authentication between INTUITEL components

For the purpose of the INTUITEL project, it is assumed that the LMS server and the INTUITEL components are connected via a secure communication channel. Depending on the selected technology and server setup, the communication layer ensures this and specifies how this has to be carried out to guarantee a sufficient level of data security.

2) Personal authentication

To achieve a high usability, the SLOM toolset does not require new login data or additional rights management to enable content creators (C3) to access pedagogical metadata. The system authenticates a user by their LMS user credentials. Due to the diverse and complex user right systems of the different LMSs, INTUITEL will entirely rely on the LMS for authentication of individual users and permission control. Therefore, INTUITEL sends an authentication request with the respective user credentials to the LMS for validation and expects a confirmation or refusal (cf. D1.1).

Please note that the INTUITEL system itself is unaware of the correspondence between user IDs and real persons. It is only aware of the name a user has specified in the LMS to make a more personal salutation possible. It has no implications on the functionality of INTUITEL if the user wants to use his/her name or just an alias. Furthermore, even the SLOM toolset itself does not have access to personal information.

## 2.3 INTUITEL as a reactive system

When working with INTUITEL and likewise when developing its components, it should be clear which actions trigger processes in the individual parts of INTUITEL. In this context, it has to be stressed that INTUITEL is a reactive system. This applies to all components and in particular the recommendation creation process. All of the INTUITEL internal processes are caused by an action of a learner, a content creator or the administrator.

The following scenarios lead to a (re-)action of INTUITEL:

- 1) The administrator starts INTUITEL.
- 2) A content creator INTUITEL-enables a course.
- 3) A learner starts a learning session on an INTUITEL-enabled course.
- 4) A learner proceeds to another LO in an INTUITEL-enabled course.

In the first case, the administrator of the respective INTUITEL instance starts all relevant programs on the server. In the context of this process, the system checks the availability of all components and initiates the communication with the LMS. After this procedure, INTUITEL starts reacting to learner LO transition events, meaning the points in time when a learner either accesses an INTUITEL enabled course or requests a new LO in such a course.

However, before this can be carried out, content creators need to activate the services of INTUITEL for their courses by INTUITEL-enabling them. This is done by adding all relevant metadata to the

already available courses from the LMS, by using the INTUITEL Editor. After saving all relevant data in the designated repository, INTUITEL automatically initiates its support for the respective course.

In order to be notified that a learner has started working on an INTUITEL-enabled course, the Back End either uses polling to monitor active learners or it receives an active notification from the LMS. At the very moment a learner enters a respective course, a new learning session gets initiated in the Back End and the tutoring process for the particular course and learner starts.

Each time INTUITEL notices that a learner is navigating between two LOs in an INTUITEL-enabled course, the session data is updated and a new recommendation is created. This process takes place at the exact time such a transition-event is registered. This of course also includes the moment when a learner starts a new learning session (i.e. a LO transition between zero and a specific LO is noticed).

In some cases, it might be necessary to send a message to the LMS at a later point in time. These so called “delayed reactions” are likewise created in the Back End and triggered by some kind of event, but always in response to a previous learner action. Even though this might appear as an independent and active mechanism, these messages are always directly tied to the analysis and monitoring of the current learning process of a learner. This could, for instance, be a timed event that triggers a question regarding the difficulty level of a LO, when the learning time of a LO is exceeded by a considerable amount. Another example would be a question regarding the learner’s preferences, when INTUITEL notices departures from the standard behaviour.

In all cases, INTUITEL is the reactive part of the interaction. INTUITEL will only act when a human-triggered action has previously occurred. This is also in line with the idea of INTUITEL being a teacher or tutor surrogate: Like a lecturer, INTUITEL may draw pedagogical conclusions from the fact that a learner has not responded for some time.



## 3 User roles and their workflows

### 3.1 Learner

The learner (L) is the main beneficiary of the INTUITEL system. By using a set of current environmental and personal meta-information, as well as past learning data, the reasoning module creates recommendations to optimally guide the learner through the learning material. This also requires a certain amount of trust from the learner, since INTUITEL will have to access and store sensitive, personal data.

How much INTUITEL influences the learner's learning process and how it is perceived depends on the individual LMS. INTUITEL does not make onerous requirements on how the recommendations and messages have to be presented to the learner. The LMS integration only has to comply with two directives:

- 1) The presentation has to be subdominant in respect to the whole user interface.
- 2) The learner does not have to respond to a dialogue or follow a recommendation.

A learner remains completely free to follow or ignore recommendations or messages from INTUITEL. If a learner decides to not use the INTUITEL features, the workflow and how a learner works with the learning material remains unchanged. Except for some changes to the LMS GUI, the learner will not notice any differences to the usual learning process. In all scenarios, the learners will occasionally be confronted with the appearance of TUG messages. Depending on the respective INTUITEL integration, this might, for instance, occur as a pop-up or via a dedicated message window. In all cases, the learner does not need to respond to the messages.

In general, the integration of INTUITEL should be as lightweight as possible and the level of adaption and changes to the original LMS GUI should be kept to a minimum. This way, the learners' effort to adapt to the new possibilities should be minimal. The learners can thus work with the LMS as they are used to and comfortably benefit from the INTUITEL features.

### 3.2 Content creator

The content creator (C) in INTUITEL will in some cases be more than one person, since the respective tasks require a diverse set of knowledge and skills:

- 1) Domain knowledge to create and structure learning material
- 2) LMS usage knowledge to correctly create and upload the content
- 3) Ontology (more precisely OWL) knowledge to understand the PO, adapt the content and create the CM
- 4) SLOM toolset usage knowledge to correctly link content and domain knowledge

Naturally, all parts could be fulfilled by one person. However, the number of people possessing all these skills is probably quite low. Therefore, the following sections provide independent role descriptions for these categories. Point one and two are combined in 3.2.1, because they are currently already fulfilled by one person, the lecturer.

In general, the combined workflow consists of three steps:

- 1) Create the learning material in the LMS.
- 2) Create or find a pre-existing Cognitive Model for the respective knowledge domain.
- 3) Use the SLOM toolset (more precisely the INTUITEL Editor) to link the LO with the CM.

If these three steps are performed correctly, the INTUITEL Back End can create personalised recommendations for each user.

It is not necessary to explicitly activate the functionality for the respective course in the LMS. Provided INTUITEL is running and configured correctly, the moment a valid SLOM metadata file is present in the SLOM metadata repository, the Back End automatically uses the information.

### 3.2.1 Content creator 1 – Learning content

This is the role a lecturer usually enacts in eLearning scenarios. A person with knowledge about a certain domain uses the functionality of a LMS to create digital learning material. The workflow and the degree of required IT skills are dependent on the system used.

The usage of INTUITEL does not have any influence on this role, because the annotation of metadata is not incorporated in the respective LMS. This bears some advantages for INTUITEL and lecturers likewise. For example, it is possible to use old content<sup>3</sup> as is for the reasoning process without changing existing courses. Consequently, also no additional data is needed to create or edit valid courses in a LMS.

So, in the case that INTUITEL is not used in a course (e.g. when a lecturer has not yet been introduced to the toolset or the respective Cognitive Model is not yet available) the course creation and administration remains unchanged.

### 3.2.2 Content creator 2 – Cognitive Model

For each INTUITEL enabled course, a Cognitive Model is needed. A person who creates these models does not only need knowledge of the domain, but also knowledge about ontologies. This encompasses OWL as the description language used as well as the use of an OWL-editor such as Protégé or any standard text-editor like the Windows Notepad or Notepad++.

INTUITEL aims to provide a manual or another instructional text document to support users with the task to create Cognitive Models<sup>4</sup>. It will contain guidelines and give hints as to how this process can be performed with minimum effort. This will not be based on pure theoretical assumptions, but be based on the experiences collected in creation example models in WP 02 tasks 2 to 5.

Thus, the content creator (C 2) does need specific knowledge that usual eLearning content creators do not have. This makes it necessary to either create specialised training which these people can participate in, or they need the assistance of ontology specialists. The last point will in many

---

<sup>3</sup> This is true as long as the content is available in a supported format.

<sup>4</sup> The respective document shall be created in context of WP 02.

situations be especially problematic, since ontologies and OWL are often not part of syllabuses of computer science majors.

However, it needs to be stressed that not every organisation or institution which wants to use INTUITEL needs to create its own Cognitive Models. The fact that OWL is easily sharable via the Internet is a huge advantage. In an ideal scenario, just one domain expert needs to create and publish a CM for one knowledge domain. If, for instance, a lecturer in Computer Science from a university creates a detailed and extensive model for an introduction into 'Theoretical Computer Science' and publishes it via the institutions website, other lecturers all over the world can use this for their eLearning courses too.

Nevertheless, the necessity of OWL and pedagogical knowledge is probably the main problem for the basic usability of INTUITEL. However, given the situation that a community of domain experts works together and they share their CMs, this bottleneck gets defused on a noticeable level. Furthermore, the Pedagogical Ontology will be made publicly available via the Internet for all interested parties, giving everybody the opportunity to become acquainted with the INTUITEL technology. This fact and the large scale usage potential of INTUITEL makes it probable that third party tools will be created to assist and ease the respective processes. The development of such a tool from the INTUITEL Consortium itself is not planned, but might be created in context of the SLOM Editor and Merger WPs.

### 3.2.3 Content creator 3 – Metadata annotator

Given that the learning material in the LMS and the respective Cognitive Model are present, the data needs to be linked and made available for the Back End. Therefore, the content creator (C 3) uses the INTUITEL Editor and performs three tasks. First, the user selects the respective course from the LMS and the Editor downloads a listing of the LOs with a specific service (cf. D1.1). Afterwards, the Cognitive Model gets imported from a local file or from the Internet. Please note that these two steps do not necessarily have to be in that order, they might also be interchanged.

The third step is the main and important one for this user role: The user links the LOs from the LMS with the Cognitive Model. INTUITEL can only create a recommendation if an LO is assigned to an element from the CM. Also other metadata like the contained Media Type must be specified.

When all information has been entered, the content creator publishes these annotations in the SLOM metadata repository. From this point on is the respective course available for recommendation creation.

Apart from this essential task, content creators can also use the INTUITEL Merger to combine the domain/pedagogical metadata and the actual LO content in one file. The user therefore needs to use the LMS native export mechanisms to download the targeted learning material from the LMS. The Merger can then import these LOs from the local machine and use the metadata present in the SLOM metadata repository to create a SLOM file. This compact data package can then be shared or stored in the SLOM repository for example for versioning reasons. The content creator can also perform the same task backwards and split up existing SLOM files to import the contained Learning Material into the LMS and the metadata repository.

## 4 Learning Management System integration

Involved task	Task-Leader	Name
1.1	HSKA	USE/TUG/LORE Data Model
1.2	HSKA	USE/TUG/LORE in ILIAS
1.3	UVA	USE/TUG/LORE in Moodle
1.4	IOSB	USE/TUG/LORE in Crayons
1.5	IMC	USE/TUG/LORE in Clix
1.6	ELS	USE/TUG/LORE in eXact and mobile scenarios

Table 1: Listing of tasks involved in LMS integration

### 4.1 Detailed overview

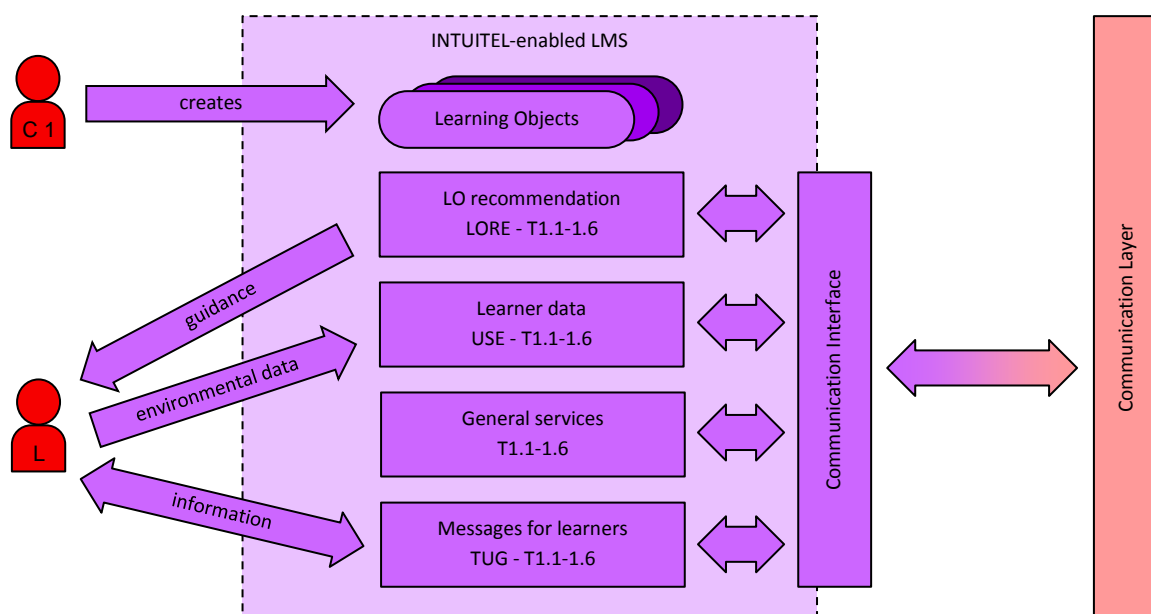


Figure 2: Structural overview of the LMS integration

Please note that USE, TUG, LORE and the general services module need information from the LMS internal data storage. Consequently, they somehow need to access this data. This could but does not necessarily have to be realised in a common LMS module. For simplification and in order to be compatible with different LMS architectures and realisations, this component has not been included in the illustration.

### 4.2 Connections to other INTUITEL components

The INTUITEL LMS extensions are connected to the Back End and the SLOM toolset. The data exchange has to be realised by using an INTUITEL compatible communication technology. To

encapsulate this functionality, it is reasonable to create one common module that manages these transmission related aspects. By providing these services to the USE, TUG, LORE and the general services module, they can focus on their own tasks.

How this is realised in the individual usage scenarios is certainly dependent on the possibilities and guidelines of the respective LMS. It is furthermore up to the LMS integrator to expand the tasks of this module to further encapsulate INTUITEL specific functionalities.

As previously described in section 2.2, the actual message payload is independent of the transmission technology. All exchanged data has to be conformant to an XML Schema which has been created in the context of WP 01 task 1 (deliverable 1.1). So, the LMS integrator could thus decide to create these XML-elements in the respective USE, TUG, LORE and general services modules directly, or to do this in the communication interface. Both scenarios are valid and which one is optimal depends on the individual LMS.

Each LMS integrator has to find the most suitable approach in order to incorporate the INTUITEL communication component into the LMS. Given that the LMS already provides suitable modules for message transmission that can be reused, it is certainly advantageous to adapt the communication interface accordingly. However, in all cases, the INTUITEL guidelines and specifications have to be abided by.

### 4.3 General services

The general service module has to provide basic functionalities that are not directly linked to USE, TUG or LORE:

#### 1) Initialisation

Since the LMSs differ in multiple aspects, the INTUITEL system initially needs to be configured to work properly. Therefore, when started, it asks the connected LMS for its capabilities. This allows that it is easy to adapt the INTUITEL functionality to the possibilities of different LMSs.

#### 2) Learner updates

INTUITEL has to track each individual learner's progress to create recommendations in the right situations and to measure their performance. Therefore, the INTUITEL Back End has to be notified when a learner accesses an INTUITEL-enabled course or when a learner navigates to a new LO in such a course. The INTUITEL Data Model specifies three different mechanisms for this: "Pull", "Push" and "Push with learner polling".

Each time the INTUITEL session management module in the Back End notices such a LO transition, a new reasoning process is started, creating a new recommendation (LORE and in some cases also TUG data) for each particular learner.

Please see the D1.1 document for a detailed description of the different methods and for diagrams outlining the respective communication procedures.

### 3) LO mapping

The INTUITEL recommendation will only rely on LO references and the available metadata. The actual learning material will reside in the LMS. The only situation learning material is extracted from the LMS is in the context of the INTUITEL Merger, when new SLOM files are created for versioning or export purposes. LO references (meaning the LMS specific LO IDs) are initially created by the LMS in the native course creation workflow. There usually is no public interface available to directly access this information from outside the LMS. Therefore, INTUITEL defines its own service to request the needed LO data. The format of these IDs is irrelevant, since it will not be visible to the user and is only used for addressing purposes. This also includes the direct provision of additional metadata for the SLOM Editor, to ease the work for content creators.

### 4) Authentication

In order to use the SLOM toolset, content creators need to authenticate themselves. INTUITEL intentionally does not include its own rights management system. In most LMSs a detailed and sophisticated system for such matters is usually included. This enables the SLOM toolset to use the already present information to authenticate users.

So, to have an advantage of this, the general service module provides a dedicated mechanism to validate user credentials in the LMS. By reusing this LMS functionality, content creators do not need to remember an additional account name and a new password. Furthermore, LMS/INTUITEL administrators can rely on the already available settings and do not need to put extra work into this aspect.

## 4.4 Tutorial Guidance - TUG

The TUG module is responsible for dialogues between the INTUITEL system and the learner. In some situations in the recommendation process or even before that (e.g. on session start) additional data might be needed for the reasoning to create better results. Some of this information might not be present in the LMS and must be requested from the user directly (e.g. “Where did you go to school?”). Other dialogues might give the user a textual recommendation or request feedback.

Furthermore, TUG can be used to emulate USE and LORE. The functionality of both can be achieved without direct integration in the LMS. This is why TUG is mandatory for each LMS, while USE and LORE are only recommended. However, the usability of INTUITEL is much better, if all three interfaces are present. The integration of them should thus not be interpreted as a “nice-to-have”-feature, but as an intermediated step to full integration.

To achieve its goals, the TUG module receives messages from the Back End, which contain the recipient of the message, the message type and the message data. The respective type makes it possible to categorise the incoming message and create a fitting presentation of the message. TUG has to manage that only the indented user receives the respective information or question. This is additionally complicated, since TUG has to regard the current user status. Even though the Back End manages a session list to identify which user is currently active and working on a certain course, it is possible that a user logs out in the timespan between the start of recommendation creation and its

delivery. In that case, the TUG implementation in the LMS returns the designated return value to the Back End as described in deliverable 1.1.

The presentation of the message is managed and designed by each LMS independently, since they use diverse technologies and have different internal structures and design philosophies. It only relies on the LMS integrator, when a message is, for instance, presented as a pop-up-window or gets a reserved space in the overall GUI. However, there are some guidelines that have to be followed:

- 1) Message windows are subdominant with respect to whole user interface.
- 2) Important messages should bear a widget to allow printing (“Print” button).
- 3) Messages requesting user action should bear a widget to submit an answer (“Submit” button).
- 4) Messages should allow closing of the communication (“Close” button).

These points lead to one important statement: TUG messages only provide guidance; they do not enforce a certain response. The whole system goal is to create recommendations. They are neither orders nor instructions that have to be followed. Learners remain completely free in their choice as to which LO they choose and when they proceed. INTUITEL will only assist them in finding the optimal sequence for their learning behaviour.

If the TUG message windows were dominant and enforced some kind of response from the learner, the learning process would noticeably be influenced by INTUITEL. This strong and possibly also negative influence on the LMS usability is explicitly avoided.

## 4.5 Learning Object Recommender - LORE

The LORE responsibility on the LMS side is the reception of recommendations and their presentation to the individual users. The integration should fit to the style of the respective LMS and provide guidance for the users, without distracting them from the actual content. As in the context of TUG, each LMS integrator is free to adapt the presentation to the respective structural conditions and corporate design of the LMS.

A Learning Object in an INTUITEL enabled course can have, but does not necessarily have to have, a recommendation value. A recommendation is not binary, meaning a LO is neither recommended nor recommended. The level of suitability of certain LOs in the current learning status is represented as a number larger than zero and smaller or equal to 100, or mathematically speaking (0,100]. All other LOs not part of the recommendation have a recommendation value of zero.

As the learner progresses in the learning material, the INTUITEL system notices transitions between different LOs via the learner update functionality. Each time such a transition is registered, the INTUITEL Back End starts a new reasoning process for that particular user, resulting in a list of suggested LOs. This list may contain zero, one or multiple LO references. In each case, the previously assigned recommendations in that particular course have to be removed. If the passed recommendation list is not empty, the new values get displayed. If the list is empty, an appropriate message (e.g. “No recommendations available”) should be presented. This could be the case if either the learner finished a course or if additional data is needed. In all situations, some kind of TUG

message could be presented to the user and other recommendations might follow, based on the new information.

Concerning concrete visualisation scenarios, suggestions for implementations are given in the following:

### 1) Recommendation message

In this case, the recommendation is made available to the learner by presenting a special page, using the TUG interface. The message content is composed of a short description and a list of the recommended learning objects. Depending on the capabilities of the LMS, the mentioned LOs could be implemented as links, to enable a better navigation for the user. This method will also be used, when TUG emulates LORE.

### 2) Visualisation I – Icons

Another method, which is less disruptive for the learning process than messages, is the usage of icons in the course overview. Each LO with a rating higher than zero gets assigned one of these symbols to indicate the suitability for the current learning status. Possible icons would, for instance, be stars, as in the Amazon product rating system. The actual used one(s) can freely be chosen by the LMS integrator.

However, the integrator should have in mind that some symbols have different meanings across the world. For example, a “Thumbs up”-icon may be problematic in some cultures. In the European and American culture, this icon usually has a positive meaning. In contrast, in some Islamic and Asian countries it is seen as an offensive gesture. The respective icon(s) should be chosen in accordance with the current and future target groups.

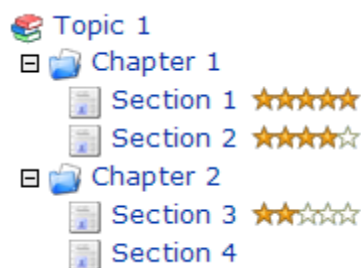


Figure 3: Example for LO recommendation integration with the Amazon star-symbols

In general, this approach promises to be easily perceivable by learners in different learning settings (schoolchildren, students, professional workers, etc.), since it is commonly used in various situations in the Internet (e.g. online shopping). The needed effort of learners to adapt to this visualisation style should thus be lesser than with a pure textual representation.

### 3) Visualisation II – Recommendation box

Many LMSs have one or even multiple sidebars to display different kinds of information to the user (e.g. news, calendar entries or system messages). In ILIAS, for example, the left sidebar contains an overview of the currently selected course in a tree-like style (as, for instance, known



from the Windows Explorer). Such bars offer a good opportunity to add a new container to include the LO recommendations as a list, ordered by the LO rating. The advantage of this approach is that the display is independent of the currently active main view of the learning material. Furthermore, the information is then centralised in one place, which could decrease the implementation efforts and increase the maintainability of the integration.

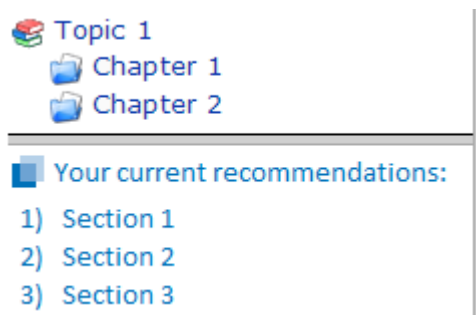


Figure 4: Example for LO recommendation integration with a recommendation box

It would of course also be advantageous to add links to the actual LOs in this overview. A combination of this visualisation style with the icon approach could lead to a better perception of the recommendation value.

#### 4) Rearrangement

The fourth and last of the suggested methods is rather more disruptive for the internal LMS structure. Instead of just attaching the recommendations to the LOs in some way (proposal two) or integrating it as a navigation extension (proposal three), the LORE interface triggers a rearrangement or restructuring of the LO ordering. That way, the original arrangement of the learning material is completely changed. This has the advantage that the recommendation is very easily perceivable and navigable. In the simplest scenario, the learner does not have to choose any of the recommendations. He or she can just proceed through the content by clicking the “next”-button or just by continuing with the element next in line.

However, this approach also has some disadvantages. Firstly, the rearrangement might be problematic to implement in the LMS and could cause undesired side effects. Secondly, the originally intended order of the content creator is lost. Some content creators might be reluctant to accept this behaviour, since they invested some time to structure the content accordingly. Thirdly, the rearrangement requires a quite deep intervention in the LO navigation GUI of the LMS and is thus in conflict with the respective guideline of INTUITEL (cf. section 2.1). Fourthly, some learners might lose track of their current status in respect to the overall learning goals, because of the continuous changes in the LO hierarchy.

Nevertheless, this presentation style could result in the least effort for learners and is thus an interesting presentation option. In the case that INTUITEL produces very good recommendations creating a noticeable profit in the learning performance and outcome, this approach might be the most interesting.

To avoid some of the mentioned obstacles, it could be a clear advantage to implement a switchable view, meaning the user could directly enable or disable the functionality as needed without any inconvenience.

## 4.6 User Score Extractor – USE

In contrast to TUG and LORE, the functionality of USE is in most cases invisible. It fulfils two tasks, both concerned with the query of historical or situational user records:

### 1) Performance data requests

INTUITEL asks the LMS for information about a certain learner concerning his or her personal learning history. This also includes the completion status of learning objects that are in the recent past, meaning also those that have just been completed or aborted.

To create personalised LO recommendations, or more precisely, to determine the learner's current position in the Cognitive Model, this information is essential. Without it, the reasoning process would produce inaccurate LO recommendations when for example started in an already running course. INTUITEL would otherwise not be able to determine if a learner has already completed certain LOs and create outdated recommendations.

### 2) Environmental data requests

INTUITEL will also incorporate the session related environmental aspects of the learner. This is very useful information which can have a huge impact on the selection of LOs. If, for example, a learner is working on a course while being on a train, the recommendation of a LO with a video could be disadvantageous. In this situation, a low bandwidth, low battery or disabled audio are likely factors that make the recommendation of such LOs unreasonable.

This is especially important when considering mobile learning scenarios, which become more and more important. Learning at any time has always been a common scenario in eLearning. However, the recent years also added learning in any place to the list of possibilities. Laptops, smartphones and tablet computers as well as technologies like UMTS and LTE give learners huge freedom in creating their personal learning schedules. The previously present limitations resulting from stationary PCs have been nullified. INTUITEL, as a research project, puts a strong emphasis on these learning scenarios and will be integrated in the mobile eLearning platforms of the consortium members eXact learning solutions and IMC.

The environmental data request also includes information about the learner as a person, which likewise have an impact on reasoning. It could, for instance, be relevant or beneficial to request the learner's age when including statistical models<sup>5</sup> about learning behaviours. Other information like the learner's cultural background, or more precisely the country in which they went to school, have a direct influence on determining a baseline for recommended Learning Pathways.

---

<sup>5</sup> This is up till now only a possibility that has to be evaluated in the respective WPs of the Back End.

In the case that USE is emulated by TUG, all this information can be requested via message dialogues. Instead of querying the learner's level of completion of a certain LO, INTUITEL could send a TUG message asking if the learner has already mastered the respective topic (e.g. "Are you familiar with the theory of 'gravity'?"). This could either be done as yes/no-questions or with a more detailed question, providing better feedback for INTUITEL (e.g. "Please rate your knowledge about the theory of 'gravity' on a scale from one to ten."). However, even though this functionality can be managed by TUG, it is much more user-friendly if USE requests this information in a background task. This approach is much less disruptive and should in many cases also be more accurate.

## 4.7 Shared functionalities

All of the INTUITEL LMS internal modules, meaning USE, TUG and LORE as well as the general services component, share some common functions. An example for this is the need to access and resolve the IDs of users and Learning Objects. This commonly used functionality is essential for INTUITEL. Without it, the connection of LOs and pedagogical metadata in the context of SLOM cannot be performed and consequentially the whole recommendation process is not possible.

It is thus crucial that this service is available and that it works properly. In order to decrease implementation efforts, decrease probability of mapping errors and increase maintainability, this base service should only be implemented once and then made available for the other INTUITEL modules. Creating individual implementations of this functionality could initially speed up the integration process, but also bears the risk of introducing errors in multiple places and consequently creating mismatches between identical IDs. Publishing recommendations for the wrong LOs could be worse than creating none at all.

Furthermore, this task requires access to sensitive, possibly highly frequented data from the LMS database. In order to reduce the load on the LMS, this action should be encapsulated in one module, to cover database related issues, to cache data and to avoid unnecessary redundant queries.

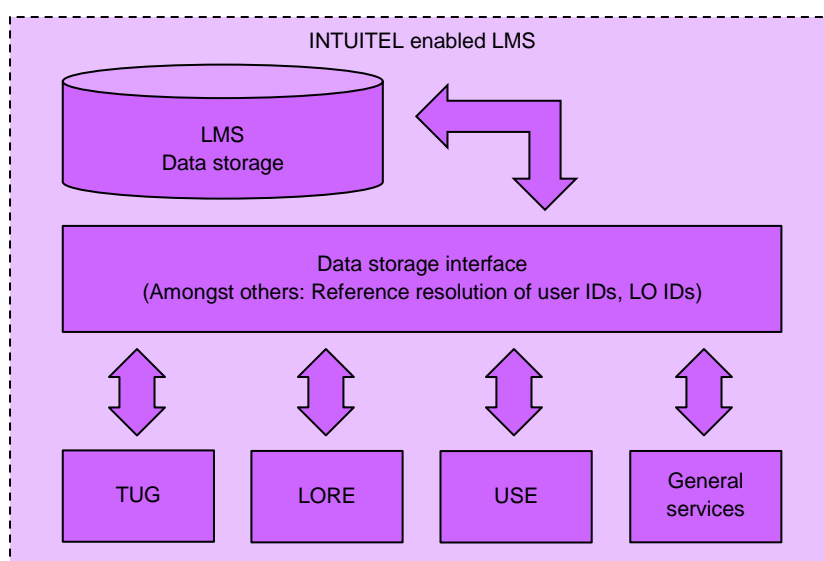


Figure 5: Integration of a common data storage interface

Due to the different LMS architectures and used technologies, this module architecture is rather a suggestion than a requirement. The various conditions present in the respective LMSs could make a specialised or otherwise different implementation more reasonable. Nevertheless, just judged from a general system design perspective, such a module is reasonable.

## 4.8 Privacy issues

INTUITEL will work with and transmit sensitive private user data. The system thus has to guarantee the security for the storage and transmission of this data. Choosing a method or technology for these issues is too early at this stage of the project. At the beginning of the development, the communication between INTUITEL and the LMSs will be executed in plaintext. Nevertheless, cryptographic aspects will be foreseen in the respective work packages, allowing an easy integration in a later stage of the project. The actual implementation of these challenges will be discussed in more detail in the context of the communication layer.

In general, it should also be determined whether the user has to approve the usage of this data in the context of INTUITEL. Each LMS integrator should check if this is already covered by their respective terms of use. If not, a section covering the data requirements of INTUITEL could be added. Another possibility would be a standard INTUITEL disclaimer. The LMS integrators remain free to decide how they want to manage this issue in their respective eLearning platform.

Furthermore is it worth inspecting if the (anonymised) learning metadata of INTUITEL could be used for research purposes and whether the individual statements need to be extended for that case. Since INTUITEL will probably provide very interesting and directly statistically evaluable data about learning behaviour of different kinds of learners, this data should be very valuable for educational purposes. It might also be beneficial to use this data to demonstrate the benefits of INTUITEL itself. The direct inclusion of this scenario is thus worth being considered.

## 4.9 Enhanced navigation functionality

In the original draft, it was planned to offer learners a new navigation view to present them their individual list of INTUITEL recommendations. This would allow them, depending on the actual implementation, to choose between the LMS native navigation method and the new individual guidance approach of INTUITEL.

However, in the Consortium discussions, it has been expressed that this method does not necessarily have to be the only new navigation functionality learners could profit from in an INTUITEL-enabled LMS. Introducing additional methods could result in an even greater usability and learning experience. In order to provide them with a better overview and to strengthen their individual learning skills, several members of the Consortium mentioned that it would be an advantage to include the metadata from the Back End in the LORE user interface.

A feature the INTUITEL partners from the University of Vienna suggested was to introduce additional buttons/links that allow learners to directly navigate to the next KO in accordance with their individual recommendations or in respect to the available metadata (e.g. "Next historical orientation learning object"). This also includes the way back, to enable them to easily repeat a previous step.

eXact learning solutions suggested introducing an overview function that allows the learners to get a graphical representation of the Cognitive Model to better judge their current standing.

Although both proposals are justified from a theoretical/pedagogical perspective, it is not yet possible to judge how technologically possible and learner-friendly they actually are. To determine which metadata elements are valuable for additional navigation features, they initially need to be defined. This is a currently ongoing process in context of WP02 and part of deliverable 2.1. The added value of a Cognitive Model map for learners is likewise hard to predict at the current stage of the project. Although the enablement of a better overview should generally be a good feature, it could be possible that it rather confuses learners. Since it is not yet clear how big a typical Cognitive Model could get, the display of a graph with possibly dozens or hundreds of nodes could be a normal scenario. Grasping a model with these dimensions is a demanding task itself. Furthermore, the disclosure of Cognitive Models could in some cases be necessary.

So, in order to make profound decisions about any additional navigation features, the (intermediate) results of the respective tasks (deliverable 2.1 and the example domains from task 2.2-2.5) need to be awaited. This allows a realistic view on the possibilities and advantages as well as on the problems that might come with these features. If a feature is judged as both advantageous for the learners and realisable from a technical and workload perspective, the arising tasks can be specified and assigned to the responsible partners and WPs.

## 4.10 Architectural remarks for the LMS integration

The INTUITEL LMS integration is intentionally relatively open, since it has to be general enough to support different architectures and application scenarios (websites, apps, etc.) of eLearning platforms. Consequently, the system has been designed to not only be applicable to the five initially chosen LMSs, but also to those platforms that might be willing to benefit from the usage of INTUITEL in the future. With that in mind, INTUITEL aims to be as neutral as possible regarding the different LMS architectures and furthermore does not require specific technologies for its implementation, with the exception of XML.

One striking advantage of this strategy is that this also makes the development of the overall INTUITEL system much easier as it by-passes the vast amount of requirements and regimentations of the different LMSs (e.g. the compatibility with certain versions and types of browsers). This is especially reasonable when taking into account that the identification of (dis-)similarities across different LMSs is an extensive task. Such an evaluation is already difficult if only regarding the five LMSs initially participating in INTUITEL, but even more difficult if including another (open) set of eLearning platforms.

This is also the reason why the INTUITEL integration has not been developed on the basis of a requirement analysis. Even if a representative set of possible input parameters could have been selected, the actual benefit in respect to the applicability and generality would have been questionable. The chosen approach of connecting INTUITEL and the LMS as loosely and openly as possible promised to be the most fitting one, given the present situation and available timeframe.

Not only from this general architectural point of view, but also as seen from the individual LMS integrations, does the clear separation between LMS and INTUITEL promise to be the most efficient one regarding workload, complexity and potential of errors. Although resulting in a relatively high initial workload for planning the LMS integration, the preservation of such a high level of freedom allows LMS integrators to develop the optimal solution for their individual setup.

While pursuing this strategy, the research and discussions of the INTUITEL consortium led to two major statements, which are, as previously outlined, the basis for the LMS integration:

- 1) INTUITEL should not fix certain technologies for the LMS integration.
- 2) INTUITEL should be compatible with different interaction methods (e.g. PUSH vs. PULL).

To account for these points and the chosen approach, the INTUITEL architecture and Data Model have been designed accordingly. With the present specifications, LMS integrators are able to choose the best technology for their distinct use cases.

Take, for instance, the following two cases:

Example 1:

The scenario in Figure 6 outlines an implementation where the data exchange between INTUITEL, the LMS and the individual learners is carried out indirectly. Instead of integrating the INTUITEL endpoint in the LMS code, it is realised as an individual program which is running on the LMS server. The data exchange is managed only by accessing the LMS database. This allows the usage of a different technology for the INTUITEL LMS endpoint and the eLearning platform. The changes to the LMS code are consequently relatively low, since it only has to be extended to display and provide the necessary INTUITEL-specific information.

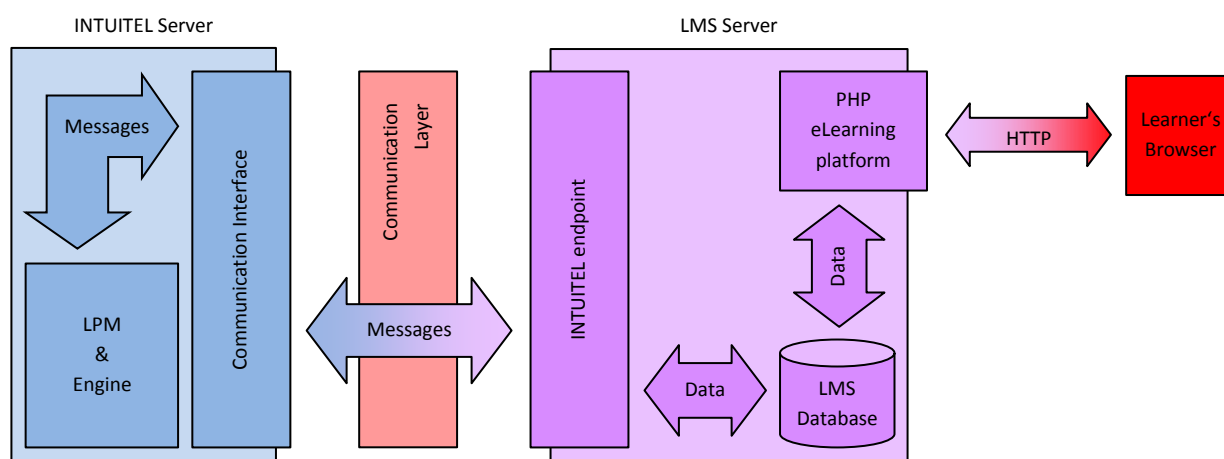


Figure 6: INTUITEL integration with direct access to the LMS's DB

Example 2:

Figure 7 does, in contrast to the previous figure, illustrate a closer collaboration between INTUITEL and the LMS. Instead of being a separate program, the INTUITEL LMS components are implemented as a part of the eLearning platform itself. This allows a more LMS-native integration with a lesser load

on the database, as well as faster processing of the data. It also enables a more direct information exchange between INTUITEL and the learner. Furthermore, if the LMS possesses some kind of plugin mechanism, this implementation scenario should be additionally promising in respect to maintainability and the installation routine.

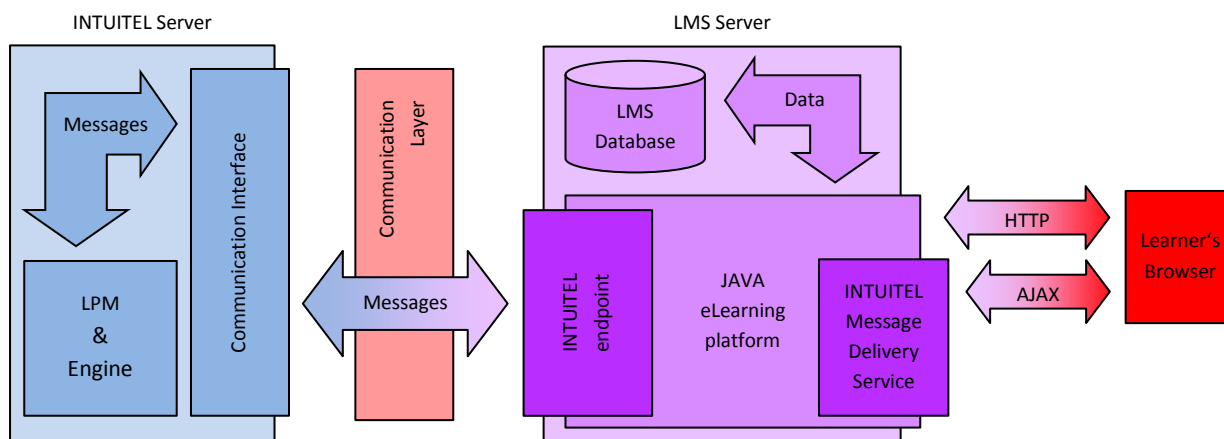


Figure 7: INTUITEL LMS interfaces as direct part of the LMS with an asynchronous message delivery component

In all cases, the LMS integrators should consider different implementation scenarios in order to find the most fitting solution for their individual case. This does, for instance, include reflections about the different technologies, the accessibility of information, interactivity (e.g. push vs. pull) between the learners and the system, as well as maintenance aspects for a future version of the LMS and INTUITEL.

## 5 Pedagogical Ontology, Domain Model and Cognitive Model

Involved task	Task-Leader	Name
2.1	UVIE	Pedagogical Ontology
2.2	UVIE	Cognitive Model for domain A
2.3	UVA	Cognitive Model for domain B
2.4	URE	Cognitive Model for domain C
2.5	IOSB	Cognitive Model for domain D

Table 2: Listing of tasks involved in PO and DM / CM

### 5.1 Detailed overview

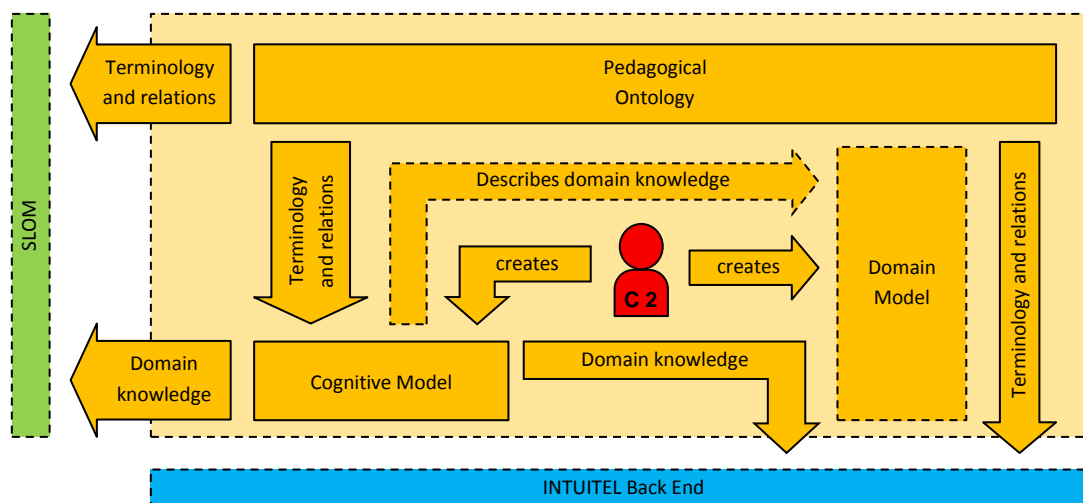


Figure 8: Detailed structural overview of the Pedagogical Ontology and Domain/Cognitive Model

### 5.2 Connections to other INTUITEL components

This part of INTUITEL provides a structured and computer readable representation of pedagogical terms and their relations as well as domain knowledge in a compatible format. It represents the theoretical baseline for the reasoning and metadata annotation. The INTUITEL Back End and the SLOM part rely on the information stored here.

The integration in the overall system is dependent on the utilised technologies. It has not yet been determined which programming languages and frameworks will be used to create the Back End and the SLOM toolset. However, as both the Pedagogical Ontology and the Cognitive Model are written in OWL, it is very likely that some kind of OWL framework will be used to access this information.



Both, the Pedagogical Ontology and the Cognitive Model, do not necessarily have to be stored on the local system. The direct access to these files via the Internet (e.g. when hosted on a website) is also a valid scenario. Depending on which OWL framework is chosen, the connectivity to these data sources could already be encapsulated. The Java OWL framework Apache Jena<sup>6</sup> does, for instance, manage the access to local and external files out of the box. The rest of the system does not even have to know where the file originates from.

### 5.3 Pedagogical Ontology

The Pedagogical Ontology is the foundational output of WP 02. It is the OWL representation of didactical concepts that describe how the learning content has to be classified and linked to be compatible with INTUITEL. The elements defined here are optimised to fit in modern eLearning scenarios and are based on the results of the L3 project<sup>7</sup>. Also, the work of Christian Swertz<sup>8</sup> of INTUITEL partner UVIE and Norbert Meder<sup>9</sup> are important factors for the PO implementation. Furthermore, the prototypical implementations<sup>10,11</sup> of individual learning pathways of the two project partners IOSB and HSKA provide valuable input.

The following element types are part of the Pedagogical Ontology:

- 1) Knowledge Objects (KOs) are container elements to extend learning content with metadata, specifying (amongst others) the Knowledge and Media Type.
- 2) Concept Containers (CCs) are abstract containers to encompass a set of thematically related KOs, representing a (sub-)section in a particular domain of knowledge.
- 3) Knowledge Domain (KD) elements encapsulate a set of CCs to create an outline for a domain of knowledge (e.g. a course).
- 4) Media Type (MT) is the superclass of a set of classes to categorize the presentation format of the content of KOs (e.g. text, audio, video, etc.).
- 5) Knowledge Type (KT) is the superclass of a set of classes to categorize the knowledge described by KO content (e.g. orientation knowledge, explanatory knowledge, etc.).
- 6) A Learning Pathway (LP) is a set of elements to describe the route a learner can take through learning content on different abstraction levels. The PO will include a set of acknowledged base pathways.

---

<sup>6</sup> cf.: <http://jena.apache.org/>

<sup>7</sup> cf.: <http://eacea.ec.europa.eu/llp/>

<sup>8</sup> cf. for instance: Swertz, C.: Web-Didaktik: Eine didaktische Ontologie in der Praxis [Web didactics. A didactical ontology in practical use]; MedienPädagogik 5 (2005)

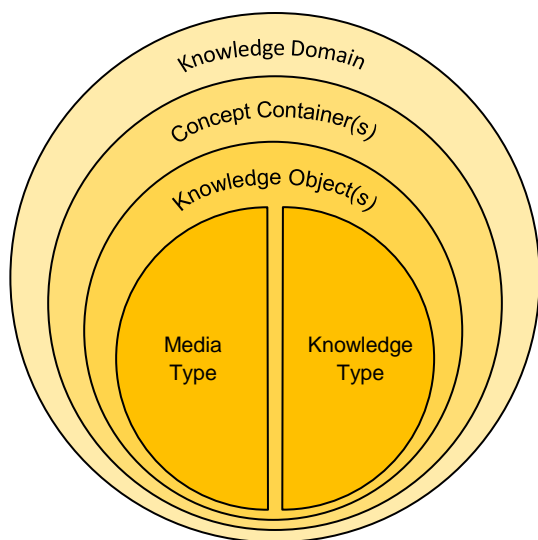
<sup>9</sup> cf. for instance: Meder, N.: Web-Didaktik: Eine neue Didaktik webbasierten, vernetzen Lernens; Bertelsmann; Bielefeld (2006)

<sup>10</sup> cf.: Kenz, T.: Ontologien in wissens- und webbasierten Lernsystemen [Ontologies in knowledge- and web based learning systems]; Master Thesis, supervised by P. A. Henning, HSKA and W. Roller, IOSB (2007)

<sup>11</sup> cf.: Knaus, D.: Konzeption adaptiver Lernpfade in WBT-Modulen [Concepts for adaptive learning pathways in WBT modules]; Thesis, supervised by P. A. Henning, HSKA and W. Roller, IOSB (2006)

These elements also share different relationships between each other. For a complete overview of them and for a detailed description of the elements themselves, please refer the deliverable 2.1.

For a better understanding, the following sketch gives a short overview about the nesting of the different elements:



- One KD references one or more CCs
- One CC references one or more KOs
- One KO has exactly one Media Type
- One KO has exactly one Knowledge Type

Figure 9: Nesting of pedagogical elements

This figure only outlines a flat hierarchy. The different CCs in a KD can also be linked together hierarchically and temporally. Furthermore, CCs can be nested to an unlimited level and thus allow the creation of complex tree or net structures.

Learning Pathways cannot be included in this illustration, since they are only indirectly linked with these elements. There are different types of LPs. Firstly, the macro level pathways, which regulate the transitions between Concept Containers. Secondly, there are the micro level pathways, which define the sequence of Knowledge Objects inside a Concept Container. These micro LPs are split up into two subcategories, the knowledge and the media LPs. The knowledge LPs define the sequence of KOs in accordance to their respective Knowledge Type. The Media Learning Pathway specifies how the KOs should be arranged in respect to their different Media Types.

The Pedagogical Ontology includes a set of base pathways for the reasoning process. The INTUITEL Back End uses the sequences specified in the PO and determines the next best fitting element in accordance with the personal Learning Pathways of each learner. In the beginning, each user is assigned one of the basic pathways. While the learner progresses through the learning content, INTUITEL tracks the selected LOs and their completion states. If any deviations in the sequence of the respective elements on the different LP levels are noticed, the system registers this as a personal preference of the learner. Given that these deviations occur to a certain degree, INTUITEL adapts the individual pathway and creates future recommendations accordingly.

Please note that the adapted pathways of INTUITEL learners are not stored in the Pedagogical Ontology. They remain inside the INTUITEL system to avoid privacy issues.

## 5.4 Domain Model

A Domain Model captures all necessary information concerning one domain of knowledge. It is designed by a domain specialist and defines everything that is necessary to master one particular knowledge domain in certain boundaries (e.g. physics for beginners and not necessarily as an exhaustive description of everything in physics). The Domain Model can be described in any arbitrary (description) language. This means that a pure textual description in shape of a digital or printed document (e.g. a MS Word document) can also be a valid Domain Model.

Its main purpose is to provide a basis for the creation of the Cognitive Model. In the best case, the Domain Model and the Cognitive Model are identical. Generally, content creators should not put too much emphasis on this part of INTUITEL, since it only represents an intermediate step and has mainly been integrated for assistance purposes. In later content creation scenarios, many domain specialists will have some kind of knowledge base that already describes structures and content of the eLearning material they create. Whether this is present as a topic outline, a book, a thesaurus or an OWL ontology is rather unimportant for INTUITEL.

There will be no custom tool assistance to support content creators with the creation of Domain Models or to automate their translation into an INTUITEL compatible format. The amount of different input formats and scenarios is too wide-ranging to provide software based assistance on that level.

## 5.5 Cognitive Model

A Cognitive Model is the INTUITEL standard adaption of a Domain Model. This means that it contains all relevant information chunks to describe a domain of knowledge by using the terminology and relationships of the Pedagogical Ontology. It describes all Concept Containers relevant for that particular knowledge domain. The actual learning content is not part of the Cognitive Model. The KOs themselves are also not part of the Cognitive Model. The links between KOs and the CCs are created in another step by using the INTUITEL Editor.

The content creators are not bound to a tool for the creation of Cognitive Models. The project plan does not include the creation of an own custom tool for their creation. The decision as to whether one will nevertheless be created will be made when WP 06 (INTUITEL Merger) and WP 07 (INTUITEL Editor) come to their first (intermediate) results.

Given that there will not be respective software, the content creator can choose from a variety of different standard tools. Since Cognitive Models essentially are common OWL files, all free and commercial OWL editors can be used, starting from plain text editors, to XML editors or even specialised ontology tools. Until a final statement has been made and a custom editor is eventually available, the INTUITEL collaboration itself uses the free, open-source and platform independent Java tool Protégé<sup>12</sup> from the University of Stanford. It provides a lot of assistance for the creation and editing of, amongst others, OWL based ontologies.

---

<sup>12</sup> <http://protege.stanford.edu/>

In the case that Protégé is used, the workflow for a Cognitive Model creator should be something like the following:

- 1) Create a new OWL ontology with the help of the Protégé wizard.
- 2) Import the Pedagogical Ontology from the web or as a local file.
- 3) Create and describe the Knowledge Domain as central element.
- 4) Create and describe Concept Containers as, for instance, described in the Domain Model.
- 5) Add the relationships between the Concept Containers.
- 6) Save the Cognitive Model locally or publish it on the web.

This is a quite technical workflow and requires special skills in the field of OWL ontology engineering. It is thus probable that only some domain experts will be willing or have the ability to make these efforts. However, in the ideal scenario for INTUITEL, only a few domain experts will have to create these Cognitive Models themselves. Due to the possibility to publish and access these models via the Internet, the less technically sophisticated content creators can use the models which other experts have created. If, for instance, one department of a university provides a CM of their main research domain, other universities, institutions or organisations can use it in their eLearning courses. It would also be a valid scenario if LMS vendors or other commercial organisations offer detailed Cognitive Models for a certain price. This would not only be an interesting business scenario, but also bears powerful marketing advantages. In all these cases, this approach provides a strong possibility to bring the eLearning sector to the next level towards a global learning cloud.

The domain knowledge as provided by the Cognitive Model is one of the main input sources for the Back End and the INTUITEL Editor. It provides the information in a structured, standardised and computer readable way and does thus allow an automated navigation through the content. It outlines a certain subject as seen from a more neutral, topical level. With the INTUITEL Editor, these rather abstract elements are linked with the actual learning material<sup>13</sup> to create a semantically rich scaffolding of the content, which is afterwards used in the reasoning process.

The Cognitive Model makes it possible to composite a certain domain of knowledge on a higher level than the learning material itself. Teachers or lecturers can structure the content in two different ways. Firstly, they can create the learning material as appears reasonable to them in the context of the syllabus of the respective course. On the second level, the content can be outlined in a more global scale and include additional information, not directly present in the learning material. A content creator could, for example, create the eLearning course in such a way that the content gradually becomes more difficult. Abstracted from this, the CM then also stores chronological and hierarchical coherences between the different sub-topics. This offers teachers a whole new level for preparing courses.

The CCs and the relations they share in a CM can also be used to create knowledge maps on different thematic levels. If the CCs in such ontologies get represented as nodes and their relations as the edges between them, a Cognitive Model provides the data basis for a graph describing that particular

---

<sup>13</sup> To be exact: Not the content, only their references are linked.

Knowledge Domain. This method of interpreting an eLearning course provides interesting possibilities for the reasoning in INTUITEL and also the visualisation of that topic. It could, for example, be used to automatically determine the central parts in that domain. A variety of different procedures or algorithms could be applied, which can be taken from general graph theory or adapted from its well-established practical applications like route planning. It also provides the possibility to automatically create graphical maps to enable a learner to get a quick overview about a certain topic or course. Just these few examples already show the potential which such a combination of domain and pedagogical knowledge makes possible. INTUITEL can thus draw on a large technological potential and implement various innovative features.

The following Cognitive Model representation is based on the sample domain “Cooking”, which is being developed as a working example in the context of WP 02. It serves for demonstration purposes to show how CCs are defined, how they are interconnected with which relationships and what the overall structure should ideally look like. Please note that the following figure is just a small example and that the shape of a Cognitive Model is very dependent on its domain.

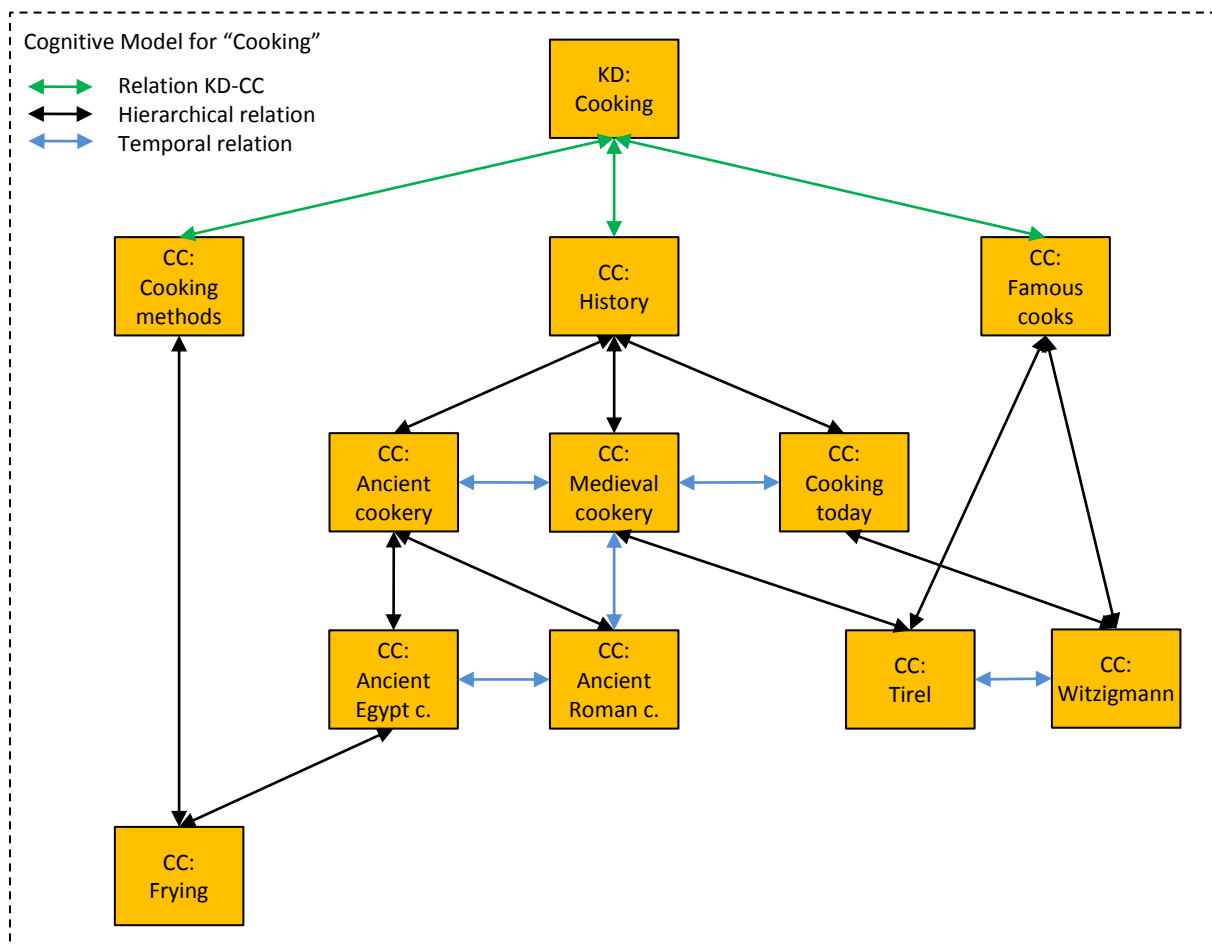


Figure 10: Simplified demonstration of a Cognitive Model with the Knowledge Domain “Cooking”

## 6 Semantic Learning Object Model

Involved task	Task-Leader	Name
4.1	HSKA	Specification of SLOM
4.2	FZI	SLOM for Semantic Media Wiki
4.3	TIE	SLOM for SCORM
4.4	URE	SLOM for IMS-LD
6.1	UNIR	Software development (of the INTUITEL Merger)
6.2	UVA	Prototype Testing (of the INTUITEL Merger)
7.1	TIE	Software Development (of the INTUITEL Editor)
7.2	ELS	Prototype Testing (of the INTUITEL Editor)

Table 3: Listing of tasks involved in SLOM

### 6.1 Detailed overview

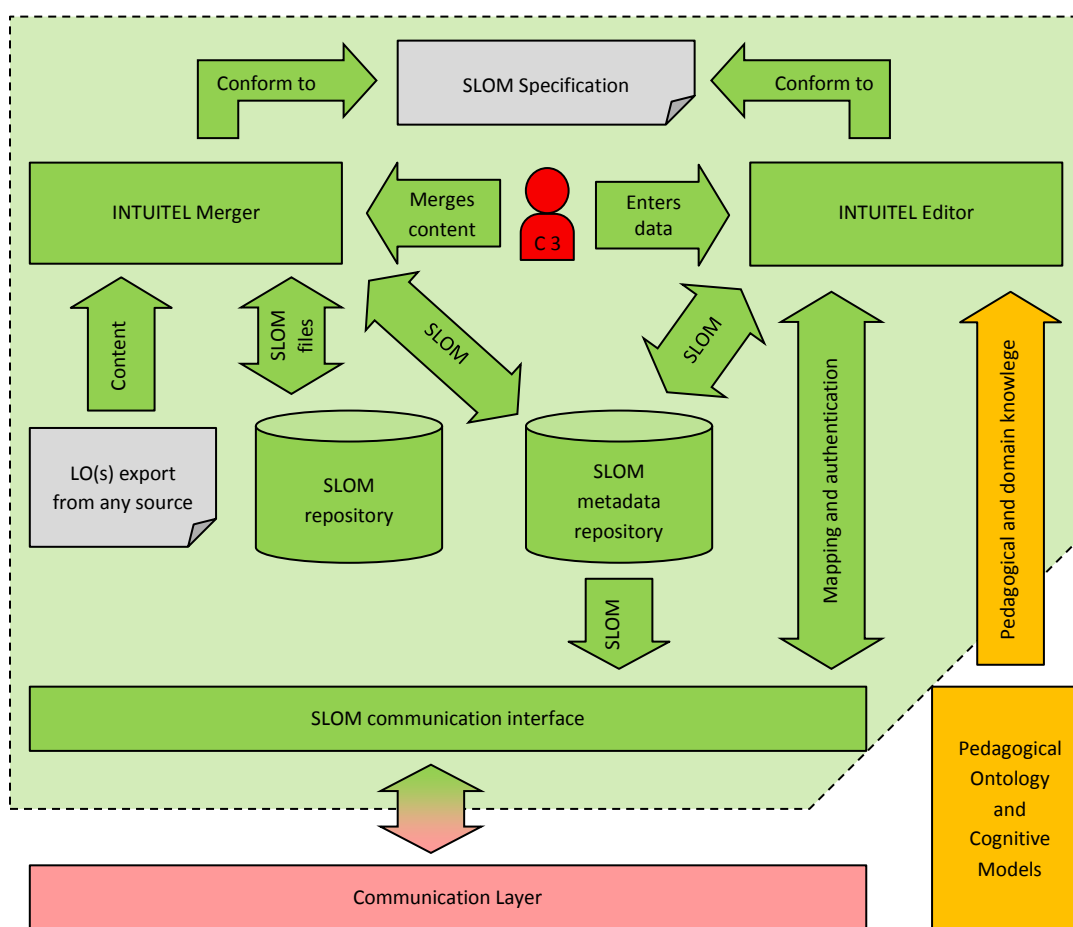


Figure 11: Detailed structural overview of the SLOM part

## 6.2 Connections to other INTUITEL components

The SLOM part is connected with the LMS integration, the Pedagogical Ontology, the respective Cognitive Models and the INTUITEL Back End. It fulfils the task to bring the content and pedagogical aspects together to provide the information for the reasoning process.

The INTUITEL Engine therefore uses a service to request the LO IDs from the LMS. The Editor itself is unaware of the actual learning material. It only accesses the IDs and some of the already available metadata (e.g. LO name) to link them with the domain knowledge, which is present via the integration of the respective Cognitive Model.

The Pedagogical Ontology provides the basic terminology and the relationships. The Editor uses them and the SLOM specification to assign metadata to the LO IDs as submitted by the LMS. In this process, the Editor is used to “transform” LOs from the LMS to KOs in the context of INTUITEL. The metadata, which is entered in this step, only affects KOs. For a complete integration, they also need to be connected to the CCs from the Cognitive Model.

Both files, the PO and the CM, can either locally or externally (via the web) be made available for the Editor. The usage of an OWL framework to mask connectivity and locality aspects from the rest of the Editor should, as judged from an architectural perspective, be an advantage.

The Merger, in contrast to the Editor, does not need a direct connection to any part of INTUITEL, except to the two repositories (descriptions below). This is the case, because the Merger is an optional tool in the workflow of creating INTUITEL-enabled courses. It uses LO exports in a supported format from any source (not necessarily from the used LMS) and imports them directly. The actual KO assignment is then done afterwards with the INTUITEL Editor.

The data generated in this part of INTUITEL is saved in two different repositories. The first one is for whole SLOM files (SLOM repository) and the second one is only for the pure metadata (SLOM metadata repository). While the file repository purpose is concerned with the storage, versioning and sharing of whole files, the metadata repository only takes care of the semantic aspect of SLOM and provides the information for the Back End. How this connection will be handled has not yet been decided, since a respective technology has not yet been determined by the responsible partners. A statement cannot yet be made, because the associated tasks do not start until month 13.

The SLOM communication interface enables the SLOM toolset to communicate with the other parts of INTUITEL. It encapsulates the transmission related aspects. Furthermore, it also provides access to the SLOM metadata repository. This is an essential task, because the Back End needs the therein contained information to create recommendations.

Whether the PO and CM access is managed via the INTUITEL Editor itself or via the communication manager has not yet been decided. Depending on the later implementation it might be advantageous to directly include this functionality in the Editor, because some OWL-frameworks already include such functionalities. Since the PO and CM are OWL files, a decision about integrating their access directly or indirectly is less critical and can be made in the context of the respective Editor and Merger work packages.

## 6.3 SLOM specification

The SLOM specification is a document which defines the format and content of SLOM files. It combines the actual learning material and encompasses it with the respective domain information in one file. It is thus a container format for LOs as well as the cognitive and pedagogical data as specified by the CM and PO. Its main goal is to enable the storage of all data necessary for INTUITEL in a system compatible and comfortable way.

SLOM files consist of two main parts that are, due to the different usage scenarios in the INTUITEL content creation workflow, treated independently:

- 1) The metadata part
- 2) The LO inclusion part

Point one should to a large extent rely on the previous work and results from WP 02. It will be conforming to the structures defined in WP 02 and consequently rather resemble an actual implementation of these elements than introduce new ones. The technologies used for the metadata description will thus be compatible to the OWL files from the PO and CM. The metadata part will also be usable as a standalone file, as it contains the information stored in the metadata repository. The decision as to whether XML, OWL, Turtle or any other file format will be used is, amongst others, subject of WP 04 task 1 (starting in month 7).

The second aspect will probably have more drawbacks in its implementation. The research conducted so far has shown that an export of identical LOs from different LMSs can lead to non-identical results, although the same standard (e.g. SCORM) is used. This makes the basic idea of exchanging LOs between different LMSs questionable, because information and content might be lost in the process. This issue must further be inspected in the context of the SLOM specification and the INTUITEL Merger. Consequently, it is still vague as to how and to what degree this aspect will be incorporated into the SLOM container format.

INTUITEL will include other eLearning formats that are currently present on many LMSs in its research activities for the SLOM specification. This does in particular involve the IMS Learning Design (LD)<sup>14</sup> as the state of the art method for the inclusion of semantic metadata into learning material. However, INTUITEL will have a critical look on all aspects, as well as on the known shortcomings. Also, the ongoing discussions in the eLearning community and especially the considerations published on the LETSI website<sup>15</sup> will be taken into account.

## 6.4 INTUITEL Editor

The INTUITEL Editor is the main contact point for content creators in respect to the SLOM part of INTUITEL. It is the essential tool in the workflow for INTUITEL-enabling courses and allows the connection between the learning materials from the LMS with the CCs from the Cognitive Model. The content creator can/must furthermore add other metadata to provide enough information for the

---

<sup>14</sup> cf.: <http://www.imsglobal.org/learningdesign>

<sup>15</sup> cf.: <http://www.letsj.org>



reasoning process. In this step, content creators “transform” the LO(-ID)s as taken from the LMS to KOs as used in the context of the INTUITEL Back End.

According to the DOW, the Editor is an independent software module that allows addition and modification of semantic information to the learning material. It is used to describe the learning content with the terminology and relations of the Pedagogical Ontology and thereby uses domain specific knowledge of local or external Cognitive Models. It fulfils this task by relying on the SLOM specification, which describes how this has to be done in order to be compatible with the SLOM metadata repository and the INTUITEL Back End.

The Editor is solely responsible for generating the metadata of the SLOM file format. It does not modify the actual learning material in the LMS. The Editor itself is unaware of the actual content and only knows the respective LO IDs and the available metadata. All aspects of SLOM that are involved with combining learning content with INTUITEL specific data is managed by the INTUITEL Merger. This allows the implementation of a lightweight solution that can focus on this important task.

A normal content creator only comes in touch with INTUITEL itself by using the Editor. It bridges the gap between eLearning courses and the domain/pedagogical knowledge. This makes it very important for the success of the project as a whole. If content creators are reluctant to use the INTUITEL Editor, due to, for example, a too complex GUI, the LOs will not be semantically annotated and no content will be produced for the recommendation process. Due to that fact, special emphasis must be put on the usability and quality of the INTUITEL Editor. Even less technically experienced teachers must be able to use it as comfortably and easily as possible.

One factor that accounts for that challenge is that the Editor does not require the creation of a new user account. Via a dedicated service, the Editor exchanges the credentials of a certain user and checks whether he or she is authorised to access and edit a certain course. The content creator can thus log in with the same username and password they already use for the LMS. This also includes advantages for INTUITEL, since the rights management of a LMS is usually quite extensive and complex. This way, INTUITEL will not have to manage this challenging aspect itself. The INTUITEL toolset and the respective administrators from its later usage organisations/institutions can completely rely on the already implemented and approved right system of the LMS.

The basic workflow for INTUITEL-enabling a course should be as follows:

- 1) Open the INTUITEL Editor and enter LMS credentials.
- 2) If authenticated, choose the course from the set of courses available for editing.
- 3) Select the fitting Cognitive Model for the selected course.
- 4) Assign the INTUITEL-specific metadata.
- 5) Save the data to the INTUITEL metadata repository.

For a more detailed description for this process, please see section 8.1.

## 6.5 INTUITEL Merger

Not all content creators will come in touch with the INTUITEL Merger as in contrast to the Editor. It is likewise specified as a standalone software module and responsible for combining learning material

and metadata in one file, which is the second aspect of SLOM. This also involves the splitting of these files to make them usable by the LMS and the INTUITEL Back End again.

It comes with a GUI that also enables less technically skilled content creators to perform this task in a comfortable way. This includes masking all INTUITEL or format specific processes and aspects as well and reasonably as possible. Like the Editor, the Merger does not require the creation of INTUITEL specific user accounts. It checks via a dedicated service if a user is allowed to access and edit certain files. The credentials are thereby transmitted to the LMS, which responds whether the access is granted or not.

The Merger also affords the possibility to save its output in a special file repository and that way allows an easy, centralised and secure storage of SLOM files. Thus, content creators and administrators do not have to invest additional time in this issue. Also, in the case of splitting the files, a functionality to store the metadata in the respective metadata repository and a way to export the pure LOs to the local machine will be provided.

Furthermore, the complexity of the Merger is on a certain level reduced, as it does not have to access and extract the LOs from the respective LMS itself. It is irrelevant whether the merged LOs come from the organisation's own LMS or from any other source. The Merger uses LO exports as provided by any native LMS or other editor export as long as they are available in a supported format.

As already mentioned in the SLOM specification section, this creates some problems. It is foreseeable that there will be difficulties in respect to the contained information of these LO exports. Even though most LMSs support a variety of standards, they often interpret the data differently and enrich it with other information or include it in an LMS specific container. So, even if the contained LOs are identical, the export will create some sort of deviations and it must be investigated as to what degree these variations occur and affect the imports. As judged from the Merger's perspective, especially the technical aspects like varying file formats or codecs are important.

The FP 6 funded EU project LUISA<sup>16</sup> (Learning content management system Using Innovative Semantic web services Architecture), which greatly improved the interoperability of learning resources, could provide valuable input regarding these difficulties arising from the inclusion of LOs across different LMSs.

Technically, the SLOM Merger will probably rely on standards for addressing elements inside of XML documents, namely XLink<sup>17</sup> and XPointer<sup>18</sup>. Being developed by the World Wide Web Consortium (W3C), this provides a reliable basis for the interoperability of learning content and metadata.

---

<sup>16</sup> cf.: [http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ\\_RCN=8506252](http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_RCN=8506252)

<sup>17</sup> cf.: <http://www.w3.org/TR/xlink/>

<sup>18</sup> cf.: <http://www.w3.org/TR/xptr/>

## 6.6 SLOM repository

The SLOM repository is used to store SLOM files, meaning the actual learning material and the INTUITEL metadata in one file, in a structured and secure way with the support of some kind of versioning system.

The actual shape and specifications of the SLOM repository are not yet foreseeable. This will rely on a variety of factors that have not yet been investigated in detail.

- 1) SLOM as file format is not yet specified.
- 2) The differences in the content formats have not yet been investigated, which also affects the specification of SLOM itself.
- 3) The respective technologies of the modules in the SLOM part of INTUITEL have not yet been determined.

Due to these reasons, it is not yet possible to make explicit comments on this part of the SLOM toolset package.

## 6.7 SLOM metadata repository

The SLOM metadata repository has to deal with the same issues as the file repository. It is likewise not yet determinable how this will be implemented and which specifications it has to fulfil.

However, there are two differences:

- 1) It does not have to work with complete SLOM files, only with its metadata part. This will probably be XML or a similar format and eventually enable an easier processing.
- 2) It will need some kind of access mechanism to allow the Back End to access the information securely.

The metadata repository represents the endpoint which provides the recommendation process with the essential LO (meta-) data it needs. The Back End requests specific course information from it, when a learner begins to work on a course. If the data is available, the reasoning is started and LOs get recommended. Otherwise, the Back End rates the respective course as not INTUITEL-enabled and does not create any LO suggestions for the user.

It is thus also a crucial aspect that the transmission of data is possible at a sufficient speed to not unnecessarily slow down this process. When the responsible partners accordingly make architectural/technological decisions, the avoidance of such bottlenecks should be kept in mind.

Regarding the later implementation of the SLOM metadata repository, the project SRRN<sup>19</sup> (Semantic Registry and Repository Network) and TripCom<sup>20</sup> could provide valuable input. SRRN already provides a set of key functionalities such as data distribution, fault tolerance and semantic retrieval

---

<sup>19</sup> cf.: (INTUITEL partner TIE) Abels, C., Chepegin, V., Campbell, S.: Semantic Interoperability for Technology-Enhances Learning Platforms; Proc. Ninth IEEE International Conference on Advanced Learning Technologies (2009)

<sup>20</sup> cf.: <http://www.tripcom.org/>

functionalities. INTUITEL will thereby also have a careful look on known deficiencies as, for instance, data security and the support of large data volumes.

## 6.8 Extended architectural recommendation for implementation

The DOW specifies the SLOM Editor and Merger as individual software tools. This allows the partners who work on the respective WPs to implement it as it appears most reasonable for INTUITEL purposes. The decision which technologies are used and how the applications will be structured will be agreed by those WP participants. However, it seems advisable to stick, if possible, to one common scheme and technology in INTUITEL as a whole and the SLOM toolset in particular.

In order to determine the optimal programming language and architecture, the fundamental usage setting should be inspected. There are certain parameters which have a direct influence on the implementation of the Merger and Editor in respect to their later users. As seen from the bottom-line, the individual content creators minimally have enough technical skills to operate the respective LMS and create the learning material. In context of INTUITEL, the collaboration assumes that an INTUITEL enabled LMS is provided with some sort of web interface. This deduces that a web browser and a connection to the Internet/Intranet are available and the content creators are capable of operating this setting properly. Providing the graphical user interface of the Engine and the Merger in the same way should result in the least effort for the user. Content creators then only need to get used to the new UI and its functions and not used to new technologies.

Following these reflections, the SLOM part should be based on a technology that allows the creation of such web based applications with a HTML and CSS Front End (e.g. J2EE). The integration of an additional framework for the dynamic aspects (e.g. jQuery) also appears to be reasonable.

Using another technology (e.g. standalone Java, C#, etc.) which requires the installation of the software on the client computers might introduce unnecessary obstacles:

- 1) The user needs to install the software on his or her local machine.
  - It is not guaranteed that the user has the knowledge to do that.
  - It is not guaranteed that the user has the rights on the computer to perform this installation. Especially in business settings, organisation policies could prohibit that.
  - Eventual dependencies of INTUITEL itself need to be managed and might require further adaptations on the local machine (frameworks etc.).
  - Organisations using INTUITEL would have to provide support for this installation process.
- 2) A decentralised toolset leads to multiple management problems.
  - Content creators might use different software versions of the Engine or Merger and possibly also of their respective frameworks. Either INTUITEL would have to support multiple versions at the same time or a centralised update process needs to be specified to provide users with (automatic) update functionalities.
  - Individual software packages with possibly a variety of different versions might complicate the maintenance and error detection of the system.

- Not all technologies are likewise compatible with different OSs and their respective versions (e.g. C# is only fully compatible on Windows systems).<sup>21</sup>
- The computers of content creators might require a higher level of hardware performance, than in a lightweight browser environment.
- Proxy servers and firewalls lead to further implementation and configuration efforts as well as possible connectivity problems.

A web based solution does (at least to a certain degree) not have these disadvantages, but also has multiple benefits:

- 1) Easy separation of functionality and style via HTML and CSS.
  - INTUITEL can delay design related aspects to a later stage of the project.
  - Changes to the GUI are easy to implement.
  - If required, different institutions can adapt the GUI to their personal corporate identity.
- 2) Editor and Merger could run on the same machine as the Back End.
  - Direct interaction between modules without delay accountable to network connectivity.
  - No authentication/security issues concerning the access to the metadata repository.
  - Only one statement about minimal / optimal hardware requirements needed.
  - Only initial installation efforts when setting up the system.
- 3) Integration of a multitude of different devices.
  - It is irrelevant if the content creator uses a PC, Mac, any form of tablet computer, etc., as long as it has a suitable web browser.
  - Since the Editor and Merger are not dependent on local files, content creators can work with different devices without lacking data. (E.g.: PC at work, tablet when mobile, Mac at home)

Furthermore, the implementation of Engine and Merger as one software program with only one Front End also suggests directly including the SLOM communication interface. This should result in an overall lesser workload concerning implementation and maintenance of the whole SLOM package.

This is just a short list and the enumerated points are partially common advantages all web based software products share. Nevertheless, the reasons mentioned above do, as seen from an architectural and usability perspective, advise an implementation of the INTUITEL Editor and Merger as web-based software.

---

<sup>21</sup> This is also the case if the Engine and Merger are installed on a central server. Nevertheless should this be much less problematic, since it only affects one instead of a multitude of (different) computers.

## 7 INTUITEL Back End

Involved task	Task-Leader	Name
3.2	IOSB	Specification of the Learning Progress Model
3.3	TIE	Connection to USE/TUG/LORE interface
3.4	UNIR / IOSB	Interface to INTUITEL Engine
5.2	URE	Connectors to ontologies, SLOM and User Model
5.3	FZI	Query builder and reasoning broker
5.4	IOSB	Implementation of a reasoning engine
5.5	URE	Recommendation Rewriter

Table 4: Listing of tasks involved in the INTUITEL Back End

### 7.1 Detailed overview

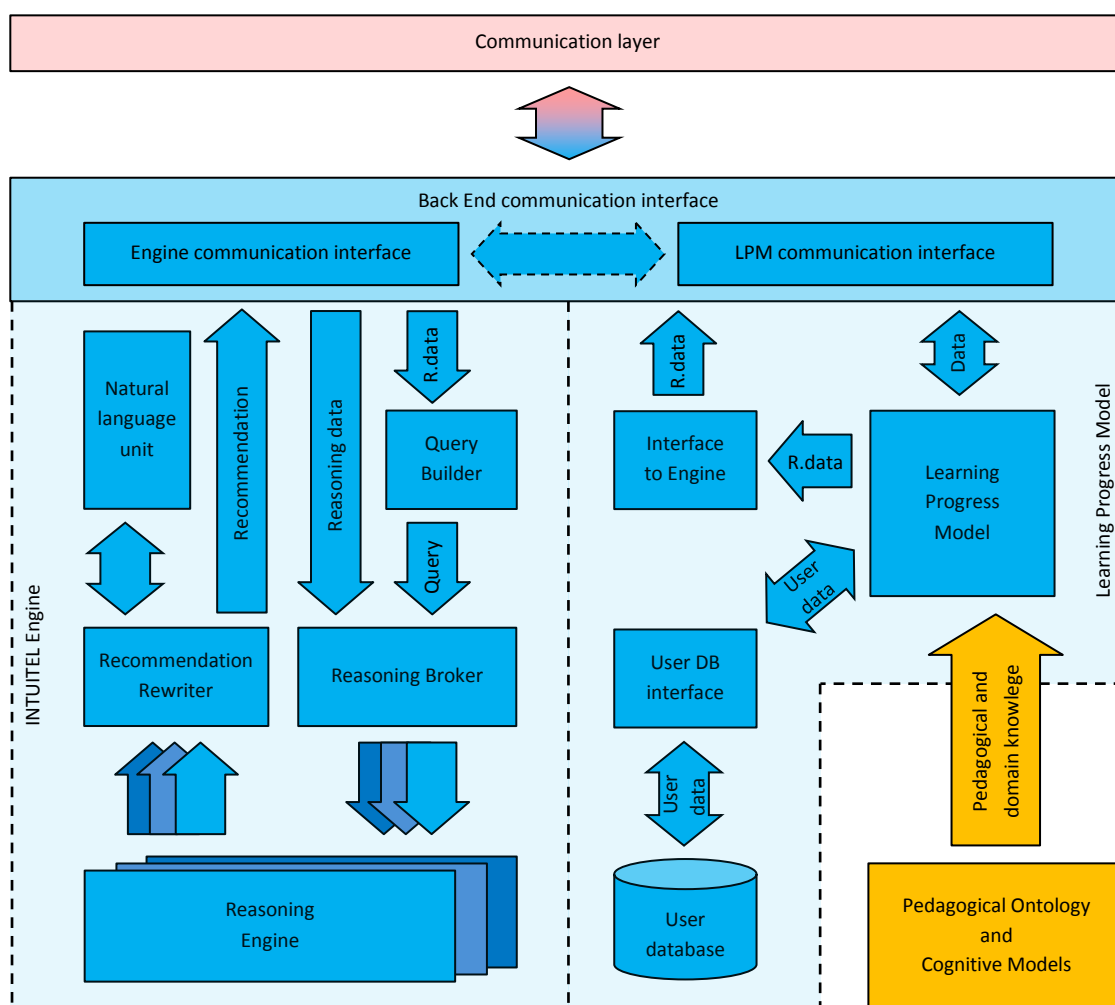


Figure 12: Detailed structural overview of the INTUITEL Back End

The structure of the Back End is more complex than the previously described parts of INTUITEL, because this is the place where the actual added value of INTUITEL is created for the learning process. The main components are the communication interface, the Learning Progress Model and the INTUITEL Engine.

The communication interface provides a comfortable access mechanism to the other INTUITEL components. The Learning Progress Model manages session related aspects and provides the input for the recommendation creation process. The INTUITEL Engine creates the actual recommendations for the individual learners.

Please note that this is not necessarily the final architecture of the INTUITEL Back End. Depending on the results of the individual WPs, the numbers of learners to manage and which communication technology gets chosen, some components could further be split up or even moved to individual servers.

## 7.2 Connections to other INTUITEL components

The Back End is connected to all other parts of INTUITEL, because the reasoning process needs information from all of them. The message exchange is managed on one common module, the communication interface, which contains a set of modules encapsulating all functionalities relevant for data transmission. The idea behind this is to mask the technology related aspects, in order to enable the LPM and the Engine to focus on their core tasks:

- 1) Creating and sending messages to learners - TUG
- 2) Creating and sending LO recommendations to learners - LORE
- 3) Integrating learner scores and environmental information in recommendations – USE

These are the three core tasks of the recommendation process. They are essential for INTUITEL to work properly. Therefore, the LMS has three dedicated modules to provide and receive the respective data. However, as noted in the LMS integration chapter and as described in more detail in the INTUITEL Data Model specification (D1.1), LORE and USE are not mandatory. They are categorised as modules that should be included in each INTUITEL-enabled LMS, but do not necessarily have to be. In the case that they are not present, INTUITEL is notified in the initialisation process. In order to enable an easier implementation of the Engine and the LPM, the communication interface includes an automatic emulation of LORE and USE via TUG. Thus, the LPM and INTUITEL Engine themselves are not aware of the implementation level, enabling a clearer internal architecture.

There are three more important aspects regarding the interconnectivity of the Back End with other INTUITEL components:

- 4) Management and Monitoring – General services

This component is like TUG always present in every LMS. It provides the basic services that are not directly related to USE, TUG and LORE. This does, for instance, include the learner update message exchange. This is the core service which triggers the recommendation creation process in the Back End.

#### 5) Requesting SLOM metadata

The communication interface also allows the comfortable and secure access of INTUITEL course metadata, which is stored in the SLOM metadata repository. This is essential information, providing INTUITEL with the data to link personal with course specific data, i.e. the foundational LO data for the recommendation creation.

#### 6) Requesting Cognitive Model data and the Pedagogical Ontology

If a direct access of the Pedagogical Ontology or the Cognitive Model is necessary for the Back End is not yet sure. In general, this data is relevant for the reasoning process. However, depending on the SLOM specification, which has not yet been created, it might be possible that all necessary data items are already available via the SLOM metadata repository. In that case, a direct connection would be redundant. In the other case, it must be decided whether the LPM accesses the data directly or by using a special component in the communication interface. A decision about that will be made in the context of the LPM work package.

Another basic aspect of the Back End communication is that it might be split into multiple parts, depending on the development decision that will be made in the communication layer, the LPM and the Engine. As already depicted in the overview diagram (Figure 12), the LPM and the Engine are only indirectly connected. It is foreseen that LPM and Engine could, but do not necessarily have to run on the same machine. This guarantees that INTUITEL can be utilised in very different usage scenarios, meaning a possibly very large number of parallel active learners. Additionally, if the INTUITEL partners come to the conclusion that it is advantageous or necessary to further split up the Back End to guarantee a certain level of performance, this will be investigated in detail. The communication interface and the respective components would then have to be adjusted accordingly.

## 7.3 Learning Progress Model – LPM

### 7.3.1 Learning Progress Model

The Learning Progress Model is the central element in the whole INTUITEL system. Not only the package itself, also its main component has this name, because they effectively fulfil the same tasks. The other components contained in the module helps to support the LPM accomplishing these tasks.

The main goal of the LPM is provide the Engine with all the necessary data for the reasoning process. It thus represents a preliminary stage of the Engine.

In order to work properly, the LPM has to accomplish a multitude of different tasks. This is quite complex, because the data preparation process touches many areas that have to be treated individually. The following points give an impression of the tasks of the LPM:

#### 1) Session management

The LPM has to keep track of the learners, which are currently logged in and work on an INTUITEL-enabled course. This is necessary in order to trigger the reasoning process, when a new recommendation is needed. This is done by actively requesting or receiving learner update



messages from the LMS. The session management component then processes this data and updates the respective INTUITEL-session states. This gets additionally complicated when learners work on multiple courses in one session at the same time.

Furthermore, the session management regards the environmental aspects related to the individual sessions, as they have a huge impact on the LO recommendations (e.g. concerning Media Types and available bandwidth or battery status).

## 2) User data storage and integration

In order to integrate the data of past learning sessions, the LPM needs to store all relevant information and access it when needed. It thereby has to decide which data is currently interesting in relation to the reasoning process. This is necessary to be as efficient as possible while not lowering the quality of the recommendation.

INTUITEL will include the results of other research and development activities concerning the aspect of user modelling. One important project which will be investigated is the Generic User Modelling Ontology (GUMO)<sup>22</sup>. It is the current state of theoretical knowledge on user modelling and has been practically implemented in eLearning in the GRAPPLE<sup>23</sup> project (i.e. in the Grapple Core User Modelling Ontology – GCUMO). This will allow INTUITEL to nourish from the achievements of other (e-learning) projects.

## 3) Generation and management of personal Learning Pathways

The Pedagogical Ontology defines a set of different Learning Pathways on different abstraction levels. Each user gets initially assigned one of these LPs per level. When the learner works on an INTUITEL enabled course, the system tracks their LO selections and can determine when and to which degree deviations from the suggested LPs occur. If this exceeds a certain threshold, the LPM creates a personal LP for that particular user, reflecting the learner's preferences. This data is then stored/managed by the LPM and integrated in the LO recommendation process in the INTUITEL Engine.

## 4) Determining the position of the learner in the Cognitive Model

One key task of the LPM is to determine the current standing of the learner in regard to the Knowledge Domain, specified via the Cognitive Model. By using the multiple (abstract) levels with which INTUITEL describes the learning material and with the collected learner data, the LPM can identify the position of the learner in that model. This is very valuable information for the reasoning process, because, due to the strongly interwoven connections between CCs and KOs, the decision-making does not necessarily have to be linear.

## 5) Provision of input for the INTUITEL Engine

This is the crucial task for the recommendation creation. It is carried out for each individual learner, when a LO transition is noticed by the session management. In this step, the LPM takes

---

<sup>22</sup> cf.: Heckmann, D.: Ubiquitous User Modelling; Akademische Verlagsgesellschaft; Berlin (2006)

<sup>23</sup> cf.: GRAPPLE project website, <http://www.grapple-project.org>

all available input data and transforms it into a format that is manageable by the INTUITEL Engine. How the reasoning is carried out afterwards and which results are created is not directly relevant for this step. Its main purpose is to identify and collect all relevant input parameters, use them to select methods (e.g. a set of transformation rules) and to bring the data into a shape that is usable by the reasoning modules in the INTUITEL Engine.

Please note that these reflections and the task-/module overview are not yet complete. The LPM as a WP is running for one and a half years and is touching multiple different parts of INTUITEL that are likewise not yet specified. The sheer number of different tasks and the still undefined methods that will be developed here suggest that the LPM will be split up in a multitude of sub-modules. How the internal architecture of the LPM will actually look is part of the decision-making process of task 3.2.

### 7.3.2 User database interface

The user database interface is a LPM internal component that masks the transmission and technology related issues of the user database. Since the LPM and consequently also the decisions about database related issues are still at an early stage, an access/DB technology has not yet been determined. In all cases, it will act as an interface and enable a unified access to the data without the need to know the table structures. By using this software module, the other LPM components can comfortably rely on the exposed services. The selected database type and its mechanisms are then isolated from the rest of the system.

This approach makes it possible to change and update the respective database schema without the need to change the rest of the package. It is furthermore possible to support different database technologies within INTUITEL<sup>24</sup>. Given that INTUITEL runs on the same machine as the respective LMS, the same database type could be used, in order to avoid software conflicts and additional efforts and costs for the operator.

### 7.3.3 User database

Not all data and especially not the INTUITEL specific data is generally available in the LMS. In order to provide the INTUITEL Back End with a complete dataset of the learner's learning history, this data must be stored as long as the learner is part of the eLearning platform. Consequently, to assure that all past learning sessions and conclusions about a learner are available, a user database is needed to provide centralised storage for these datasets<sup>25</sup>. This is important for the preservation of the collected and deduced knowledge about learners, but also for the creation of new deductions. Without the permanent and historically consistent availability, the individual learner habits and preferences cannot be integrated into the reasoning process in the INTUITEL Engine.

---

<sup>24</sup> It is currently not planned to directly support more than one DB-technology, but since INTUITEL is an open-source project, respective changes and addons could be developed afterwards.

<sup>25</sup> INTUITEL will store the information as long as necessary to guarantee an optimal quality for recommendation creation. Methods to compress the data in order to also guarantee a certain performance level might be developed in that context.

Four different tables have been outlined so far<sup>26</sup>:

1) User model:

Encapsulates the information related to general and personal user data (e.g. gender and age).

2) User sessions:

Stores the (historical) session related aspects (e.g. login times, session times, environmental information).

3) User pathways:

Registers the Learning Pathway related information, for example recommendations and the actually chosen KOs of the learner.

4) User learning history:

Registers the learning history per learner, covering finished KDs, CCs, KOs, completion statuses and achieved scores, as well as other related aspects.

### 7.3.4 Interface to INTUITEL Engine

The 'Interface to INTUITEL Engine' connects the Learning Progress Model with the Engine. Since those two modules are developed in two different, only partially overlapping work packages, this interface introduces an intermediate layer to enable a fluent and transparent exchange of data.

This is reasonable, because the internal architectures of the LPM and the Engine are still upcoming tasks. It is consequentially not yet possible to estimate which data will be exchanged between them and which format it will have. Due to this, the creation of a plan as to how those parts could be interwoven would most probably not be successful at this stage of the project.

Nevertheless, it is certain that some kind of data will be exchanged between those two modules, which makes it necessary to incorporate this fact in the overall system design. The providence of the rather vague 'Interface to INTUITEL Engine' accounts for that requirement, while providing enough freedom to include upcoming results and decisions.

It is also possible that this component might be integrated into the communication interface.

## 7.4 INTUITEL Engine

### 7.4.1 Query Builder

The Query Builder is the first point of contact after the relevant data is handed over by the LPM. It takes the provided pre-processed data and uses it to create a query for the reasoning process, which will afterwards be used by the Reasoning Broker and the Reasoning Engine.

---

<sup>26</sup> Since this is a currently ongoing research issue, the enumerated tables are not final and might be changed, omitted or complemented.

The query fulfils two aims. It firstly defines the target outcome of a reasoning process and secondly makes it possible for the Broker to assign/select an optimal Engine for the recommendation. It therefore structures and optimises the query accordingly to be compatible with the following modules.

### 7.4.2 Reasoning Broker

The Reasoning Broker is the last module before the actual reasoning starts. It is based on a pre-existing system, called HERAKLES<sup>27</sup>, of the INTUITEL partner FZI, which has sophisticated experiences with the selection of optimal OWL reasoners<sup>28,29</sup>. The main task of the Broker is to take the previously created queries and data as the basis to determine the best Reasoning Engine for the respective jobs. This is necessary because INTUITEL will not exclusively use just one (type of) Engine.

The INTUITEL project team will provide one basic Reasoning Engine for general purposes. This should be sufficient for standard eLearning purposes and produce good recommendations. Other perhaps pre-existing or especially created open-source or commercial Engines will be usable too. Which Engines are suitable for INTUITEL purposes will be specified in the context of WP 5. They can be used to cover specialised recommendation needs.

The Reasoning Engines are started and stopped by the Reasoning Broker as needed by the current usage situation. Consequently, the Broker does not only manage the task to determine the most suitable Engines, but also keeps track of which ones are running and creates new instances when needed. This also includes the administration of waiting lists, or queues, for the respective jobs.

This enables INTUITEL to profit of current state of the art multi-core systems and provides recommendations for a large number of learners in a short time.

Technically, the Reasoning Broker interacts with the different Reasoning Engines (at least in the case of pure OWL reasoning) by using OWLlink<sup>30,31</sup>, which is a protocol for the communication with OWL-Reasoners. With the support of HTTP messages and an available JAVA API, OWLlink provides a good foundation for INTUITEL.

### 7.4.3 Reasoning Engine

The Reasoning Engine is the software module in INTUITEL that actually creates the recommendation (priority value for a specific LO and user) and possibly also TUG-data, based on the beforehand created query and provided data<sup>32</sup>. In a practical usage scenario, not just one Reasoning Engine

---

<sup>27</sup> cf.: [http://sourceforge.net/apps/mediawiki/herakles/index.php?title=Main\\_Page](http://sourceforge.net/apps/mediawiki/herakles/index.php?title=Main_Page)

<sup>28</sup> cf.: Bock, J. et al.: A Reasoning Broker Framework for OWL; Proceedings of the 6<sup>th</sup> International Workshop on OWL: Experiences and Directions (OWLED 2009)

<sup>29</sup> cf.: Bock, J. et al.: Automatic Reasoner Selection using Machine Learning; Proceedings of the 2<sup>nd</sup> International Conference on Web Intelligence, Mining and Semantics (WIMS 2012)

<sup>30</sup> cf.: <http://www.owl.link.org/>

<sup>31</sup> cf.: Liebig, T. et al.: OWLlink; Semantic Web – Interoperability, Usability, Applicability, pages 23-32 (2011)

<sup>32</sup> This data will contain the relevant information (e.g. CC and KO relations) from the LMS and SLOM metadata repository in a suitable, XML-based format, which is needed for recommendation creation.

operates alone. A set of multiple Engine instances, probably optimised for different usage and application situations, run parallel.

Out of the box, INTUITEL provides one basic Engine that covers general usage scenarios. In order to also cover specialised and optimised organisation or domain requirements, it is possible to import other third party Engines. These add-on Engines can either be open-source or commercial. This opens up new marketing possibilities for LMS vendors and new sales markets for Engine creators.

A Reasoning Engine itself only has to perform the actual recommendation creation. The management of waiting lists or queues for “reasoning jobs” is carried out by the INTUITEL Reasoning Broker. The results are afterwards processed and edited by the Recommendation Rewriter, to make them compatible with the INTUITEL system. This makes it possible for an Engine to focus on its actual task and to (at least to a certain degree) ignore the INTUITEL specific conditions. However, some kind of specification of the basic requirements, tasks and interfaces, especially in terms of the provided data input, will be created in WP 05 (cf. D5.1). When an Engine conforms to that specification, it will be usable in INTUITEL.

How the actual reasoning will be performed cannot yet be determined, because this relies on multiple factors that have so far not been specified and discussed in detail.

The multiple factors which have an impact on the reasoning process will be collected and analysed in WP 05 and in the tasks leading to it. This requires thorough research concerning the pedagogic and computer science aspects of this process. Some of the questions that will be investigated in these tasks will for instance be:

- What kind of data will be handed over to the Engine?
- In which format will the data be available?
- What will a query look like?
- Which (pedagogical) factors have an influence on a recommendation?
- How to combine different aspects (environmental, personal and pedagogical)?

A more accurate specification of the reasoning process inside an Engine will follow when these and other questions have been answered. However, it is yet foreseeable that pure OWL reasoning might not be sufficient. Other methods that are being discussed by the Consortium are, for instance, State-Machines and Rule-Based-Systems.

A detailed specification of an INTUITEL compatible Engine as well as its surrounding components will consequentially not be created until the respective tasks 1 and 3 in WP 05 come to a conclusion. The respective documents will, according to the current schedule, not be available until month 13.

#### **7.4.4 Recommendation Rewriter**

The Recommendation Rewriter manages the task of transforming a Reasoning Engine output into a usable and transmittable format for the TUG and LORE components. It is not yet clear, how the different Reasoning Engines will carry out the queries and what their output will look like. It is thus possible that the results may have different formats and structures. In order to ease the creation of Recommendation Engines and to give developers more freedom, the Rewriter interprets the output

and uses the respective proxies to send the recommendations to the user. The Engines will thus not have to completely adapt to that aspect of the INTUITEL system.

This however only covers the simplest case when a recommendation for LOs is created. In some cases, the Engine could decide to request further feedback from the user to generate better results. The Rewriter then creates a textual (or possibly audio/video) message for the learner and sends it to the LMS via the TUG interface. The learner's feedback is then integrated in the reasoning process.

The Rewriter is also responsible for forwarding the generated recommendations to the LPM. This information is then stored in the user database and afterwards used to calculate the learner's deviations from the suggested Learning Pathway.

#### **7.4.5 Natural language unit**

This software unit helps the Recommendation Rewriter to create TUG messages using natural language. A direct interaction with the learner using human-like phrases increases the usability of INTUITEL within the LMS. The learner will probably be more willing to accept recommendations if the tutorial guidance addresses him or her as a human would do.

The exact functionalities and algorithms of this unit have not yet been determined. The University of Reading (INTUITEL partner no. 3 – URE) has already developed and worked with such systems and provides the knowledge and technology to implement this module. Aspects of dialogue management will be utilised for this unit. A number of options are available as solutions for meaningful interaction between the user and the system. These include frame-based, plan-based and hybrid approaches (a combination of frame- and plan-based approaches).

Frame-based approaches provide frames with slots that need to be filled in by the user through interaction to take the processing to a conclusion, i.e. action taking. Each frame is typically limited to a single topic or interaction and may be linked to predecessor or successor frames.

Plan-based approaches aim to identify the intention of the user through meta-data as well as the unfolding conversation. These are complex systems that require domain knowledge (e.g. through Ontologies to understand underlying concepts) and extensive reasoning.

Hybrid systems use plans with a limited set of frames, rearranging frames to overcome limitations of both systems. For INTUITEL, these appear to be the most suitable ones as they give a more human like interaction without the computation requirements, and can be extended dynamically by adding domain knowledge as well as suitable frame sets.

The unit needs to maintain for each user a session of the communication, for each user id, based on the input received at the start of the learning session, the feedback/responses from the user as well as the messages passed on to the user (e.g. coming from the recommendation rewriter). This is to ensure that the communication/interaction plan is maintained and knowledge-ful and appropriate interaction is taking place. This will also allow reference to previous conversation artefacts even though they might not be strictly necessary for the communication – this allows for a more human-like communication.

Natural Language Processing will be necessary to retrieve the meaning of the user input. Responses by the system are determined through the frames which can be dynamically restructured and re-aligned based on the plan and/or changes therein. Responses as per the frames are assigned with semantics to allow for various forms of output without repeating the exact same message.

In the first instance, it is planned to use English as the interaction language. However, previous work by URE has utilised English as a pivotal language to achieve multi-lingual capabilities. This means English is used for internal processing which is automatically translated from and into the target language using APIs such as Google Translate.

Thus the Natural Language Unit aims to create a reduced conversational agent.

## 8 Step by step usage scenarios

The following sections outline exemplary usage scenarios for the planned functionality of INTUITEL. These step by step instructions serve as guidelines for the implementation of the system itself as well as a foundation for internal discussions. If the later workflows are equal or similar is dependent on the decisions of the interface implementation of the SLOM toolset and the LMSs as well as the technical conditions the individual software modules have to manage.

The following descriptions assume that the LMS and the INTUITEL system are properly configured and running. How to install, start up and configure INTUITEL correctly will be specified in due time when actual working software is available.

### 8.1 How to create an INTUITEL-enabled course?

A) Create a common eLearning course (Proceed with B if course is already available)

-- Start of the common LMS specific workflow of content creation

- 1) Open the LMS GUI.
- 2) Log into the respective LMS with personal LMS credentials.
- 3) Open the target directory in the LMS file/course/repository-view.
- 4) Create a new course with the LMS native editing tools.
- 5) Create/Import actual learning content.
- 6) Repeat the previous step until course content is complete.
- 7) Enable the course in the LMS (this step can be postponed until course is INTUITEL-enabled).

--- End of common eLearning course creation

B) INTUITEL-enable the course

--- Start of the INTUITEL specific content creation workflow

- 1) Open the INTUITEL Editor.
- 2) Login with personal LMS credentials.
- 3) Start LMS course mapping request.
- 4) Select course and open the course to be INTUITEL-enabled.
- 5) Import Cognitive Model (local or external).
- 6) Define KOs by tagging the respective LO elements from the LMS import.
- 7) Define Media Types of the KOs.
- 8) Define Knowledge Types of the KOs.
- 9) Add further metadata to KOs (List of metadata dependent on PO and SLOM specification).
- 10) Link KOs with fitting CCs from the CM.
- 11) Save/Publish the assignments as SLOM package in the SLOM metadata repository.

--- End; Recommendations can be created; Course is now INTUITEL-enabled



Step 3 and 4 might be managed in one step in the INTUITEL Editor. The implementation of a wizard for the creation of a new INTUITEL-enabled course appears to be reasonable. Since both datasets (the LO mapping and Cognitive Model) are must-have elements, the creation of a new 'project'<sup>33</sup> should not be possible without them.

Step 5 requires the availability of a fitting Cognitive Model for the knowledge domain. The creation of such a model is a more complicated endeavour and is described in a step by step description. INTUITEL assumes that not all Content Creators will (have to) create their own CM. The sale or free distribution of CMs for various domains is intended and expected after INTUITEL reaches a critical level of market penetration. Up till then, the CM availability is a critical success factor of INTUITEL.

## 8.2 Creation of a Cognitive Model with the help of Protégé

A description cannot be provided until WP02 (especially the Pedagogical Ontology) reaches a certain completion level.

## 8.3 Creation of a Cognitive Model with the help of an INTUITEL specific tool

A description cannot be provided until the development of the tool reaches a certain level. Currently, it is not yet determined if such a tool will even be part of the project.

## 8.4 LMS user learns with an INTUITEL enabled course

The actual usage scenario for learners is, depending on the integration in the LMS, only minimally different. Because INTUITEL follows the guidelines to change as few as possible in the LMS and its GUI, the learning workflow is only altered at just one step – the selection of the next LO.

-- Start of the common eLearning learning workflow

- 1) Learner opens the LMS.
- 2) Learner logs in with personal LMS credentials.
- 3) Learner selects and opens an INTUITEL-enabled course.

--- INTUITEL starts monitoring the learner's navigation

- 4) (Optional: Learner reads (and answers) INTUITEL welcome message (request of session data).)
- 5) Learner selects next LO...
  - a. from LMS-native LO navigation.
  - b. from INTUITEL recommendation view.
- 6) (Optional: If TUG-message is displayed, learner answers or ignores INTUITEL message.)
- 7) Learner works on LO.
- 8) Learner finishes or aborts work on LO.
- 9) Learner proceeds...

---

<sup>33</sup> Or whatever name the Consortium agrees on – The compound of information needed to create a valid SLOM metadata file.

- a. with the next LO (go to Step 5)).
- b. by logging off.

- - - End of learner's LMS and INTUITEL session

## 9 Summary and Outlook

To conclude the specification of the overall system design, the following table provides a short summarizing overview about the INTUITEL packages, their modules and their functionalities, as described in this document.

Package:	Description:	Functionalities:
LMS Integration	<p>The LMS Integration is the part in INTUITEL which is concerned with the enhancement of the LMSs. This includes two to four modules that are needed or should be implemented in a LMS to be considered as (fully) compatible with INTUITEL:</p> <ol style="list-style-type: none"> <li>1) TUG – Tutorial Guidance (mandatory)</li> <li>2) General services (mandatory)</li> <li>3) LORE – Learning Object Recommender</li> <li>4) USE – User Score Extraction</li> </ol>	<ul style="list-style-type: none"> <li>- GUI extension to present recommendations (LORE)</li> <li>- GUI extension to present TUG-messages as well as forwarding learner responses to the Back End</li> <li>- Provision of the data necessary for INTUITEL-enabling courses: LO mapping and user authentication</li> <li>- Provision of the learner information necessary for the reasoning data: The USE interface</li> </ul>
Pedagogical Ontology and Cognitive Models	<p>The Pedagogical Ontology provides the basic specification of how to describe learning material to be usable in the context of INTUITEL. A Cognitive Model is the description of domain knowledge which is compatible to the Pedagogical Ontology.</p>	<ul style="list-style-type: none"> <li>- Definition of the basic elements (Knowledge Objects, Concept Containers ...) and their relationships</li> <li>- Definition of Learning Pathways (LPs)</li> </ul>
Semantic Learning Object Model	<p>SLOM is format specification that defines how to enhance learning material with INTUITEL metadata. SLOM as a package in INTUITEL also includes the INTUITEL Editor and Merger as well as the SLOM repository and the SLOM metadata repository which are used to create and store SLOM (metadata) files.</p>	<ul style="list-style-type: none"> <li>- Specification of how to store INTUITEL metadata alone and in combination with the respective learning material</li> <li>- A software tool to enter SLOM metadata (INTUITEL Editor)</li> <li>- A software tool to combine SLOM metadata and learning material in one file (INTUITEL Merger)</li> <li>- Repositories to store SLOM (metadata) files</li> </ul>
INTUITEL Back End	<p>The INTUITEL Back End contains two big components. These are the Learning Progress Model (LPM), which is used to manage and prepare reasoning data, and the INTUITEL Engine, which is used to create recommendations.</p>	<ul style="list-style-type: none"> <li>- Management of learning sessions</li> <li>- Storage of learner data for the reasoning process</li> <li>- Preparation of reasoning data</li> <li>- Creation of reasoning queries</li> <li>- Management of reasoning engines</li> <li>- Creation of recommendations</li> <li>- Creation of messages for learners</li> </ul>

Table 5: Summary of INTUITEL modules

Regarding the upcoming work of the INTUITEL Consortium, there are three main goals to be fulfilled in the next months that are particularly important for the project. They are all due by the next three or six months and fill the gaps between the current project status and the fundamental functionalities of INTUITEL.

1) Learning Progress Model (D3.2) – due by June 2013

The LPM is the central element in the whole INTUITEL system architecture. It collects all relevant information from the different INTUITEL components and manages the data storage as well as its analysis in order to create the reasoning data for the INTUITEL Engine. The upcoming research of the INTUITEL Consortium will provide insights on how this can be carried out. This means that a prototypical system will be specified which takes the vast amount of learner and pedagogical input parameters and processes them in a semantically rich way.

2) LPM Communication Standard (D3.3) – due by September 2013

The work on three of the four main packages in INTUITEL has already started. Deliverable 3.3 is the essential element which connects them all and enables the individual components to exchange data. It is thus a key element for the implementation of LMS integration, the SLOM toolset and the INTUITEL Back End.

3) Specification of SLOM (D4.1) – due by September 2013

The SLOM specification provides one half of the information needed to create recommendations in the Back End. In order to ensure that the LPM and the INTUITEL Engine have all the information they need about the domain knowledge, the SLOM specification needs to define how the SLOM metadata is structured.

## 10 FAQ

### 1) Which elements are parts of a Cognitive Model?

A Cognitive Model contains the description of a Knowledge Domain and all the Concept Containers that describe the topics or so to say the cognitive elements of which it consists. It also describes the different (hierarchical and temporal) relationships between the Concept Containers and the Knowledge Domain.

### 2) What is the difference between the INTUITEL Editor and the INTUITEL Merger?

The difference is their usage scenarios. The Editor is part of the INTUITEL-enablement workflow of courses and is used to add metadata to the individual LOs from a course. The Merger is, in contrast to that, an optional part of this workflow. Its task is to take the previously created SLOM metadata and the course export package from the LMS and combine them into a single container file. This also includes the way back, meaning the extraction of the contained data. So, the Mergers goal is to allow a comfortable method to store (different versions) of INTUITEL-enabled courses.

### 3) Will there be a tool for the creation of Cognitive Models?

No, a custom tool for that purpose is currently not planned. However, the members of WP 06 and WP 07 might decide to also incorporate that aspect into their toolset. A decision about that issue will be made when the respective WPs produce their first (intermediate) results.

### 4) Will INTUITEL provide cross-platform modules for different LMSs?

The INTUITEL partners are currently evaluating if some software components could be bundled across multiple LMSs. This would speed up the development and ease the implementation in other upcoming LMSs. It is not yet foreseeable when a reliable statement about this issue can be made.

### 5) Does the LMS have to know how Knowledge Domains and Concept Containers are structured?

The LMS will not have to be concerned about these Cognitive Model related issues. The respective data will be stored, managed by and stay inside INTUITEL.

### 6) Which types of video and audio formats will be used in context of TUG?

This an open question. Since INTUITEL is built for web based scenarios, a fitting data will be chosen at the appropriate time. At this stage of the project, this is of minor importance, since the respective mechanism only links audio and video files.

### 7) Will INTUITEL create messages or recommendations for learners in their absence?

INTUITEL is a purely reactive system. This means that for each of the INTUITEL actions, a user must have performed some kind of action in advance (e.g. opening a course in the LMS). This should happen immediately afterwards, but could also be performed after a certain timespan. Although a learner might have logged out in the meantime, it has initially been caused by the

learner. So, yes, INTUITEL could create messages or recommendations in the learner's absence, but only in a session based context. Currently logged out learners are not taken into consideration by INTUITEL.

8) What is the difference between Learning Objects, Knowledge Objects and Concept Containers?

Learning Objects are in INTUITEL the general term for learning material in context of a LMS. Knowledge Objects are semantically enriched Learning Objects without the actual learning material. The Concept Containers cluster different Knowledge Objects into a thematically fitting way. They are of a rather abstract nature and not part of the LMS.

9) Will INTUITEL recommend only Knowledge Objects, or could it also recommend Concept Containers?

INTUITEL will only recommend Knowledge Objects, since Concept Containers are only known to the INTUITEL system and are not part of a course.

10) When a learner is working on a certain Concept Container, could INTUITEL recommend Knowledge Objects belonging to another Concept Container?

Yes, but whether this is reasonable depends on the individual scenario. It is influenced by the particular learning history of the learner, the Cognitive Model of the respective course and the session related environmental information.

11) Will there be examples for Cognitive Models?

Yes, INTUITEL will produce Cognitive Models for four different domains in the context of WP 02. They will be available by the end of December 2013. Those aim to represent an actual eLearning course and should be usable in practical scenarios. Furthermore, INTUITEL aims to produce a more lightweight example of the Knowledge Domain "Cooking", which should be available earlier.

12) What kind of technologies can be used for integrating INTUITEL into as LMS?

There are no regimentations as to which technologies should be used. Each LMS integrator can choose freely, as long as they are conformant to the Data Model (D1.1) and the respective guidelines.

13) How long are messages and recommendations valid?

In general, messages and recommendations are only valid in the session they have been created in, since the environmental aspects could have changed in between. However, it is up for discussion as to whether (at least) some of them will be reused across different sessions. The validity aspect of messages and recommendations will be evaluated in the context of the WP 03 (LPM) and 05 (INTUITEL Engine).