4

Prototype Implementation

4.1 Back End

Kevin Fuchs

In Section 3.4 we defined the software architecture for a real implementation of the INTUOTEL system. This architecture is just a recommendation given by the INTUITEL consortium. However, as long as you follow the data model and the communication standards given in Section 3.5, you may design an architecture of your own.

Our recommended architecture comprises six logical components that cooperate in the form of a recommendation cycle as illustrated in Figure 4.1. As a seventh component, we recommend a communication layer to act as a proxy mediating the communication between the components. Through this way components are decoupled from each other. Every single component only communicates with the communication manager as if it was communicating with the respective component. This guarantees a high level of interoperability as each component may be replaced or modified at any time without any effect on the other parts of the system.

The system may or may not be implemented in a distributed way with each component being deployed as a single part. Your choice will depend on the question how scalable your system is supposed to be.

The INTUITEL consortium implemented a prototype back end that was partly distributed. The LPM, the Query Builder and the Reasoning Engine were integrated into one physical component but with each sub component having its own REST interface. The Recommendation Rewriter is a component of its own and as recommended above, we implemented a Communication Layer. The SLOM repository was implemented as a simple local directory on the LPM system. In this folder the SLOM data was stored in the form of XML files containing the OWL descriptions of the ontologies. The annotation of course material is done by simply uploading these files into that directory.

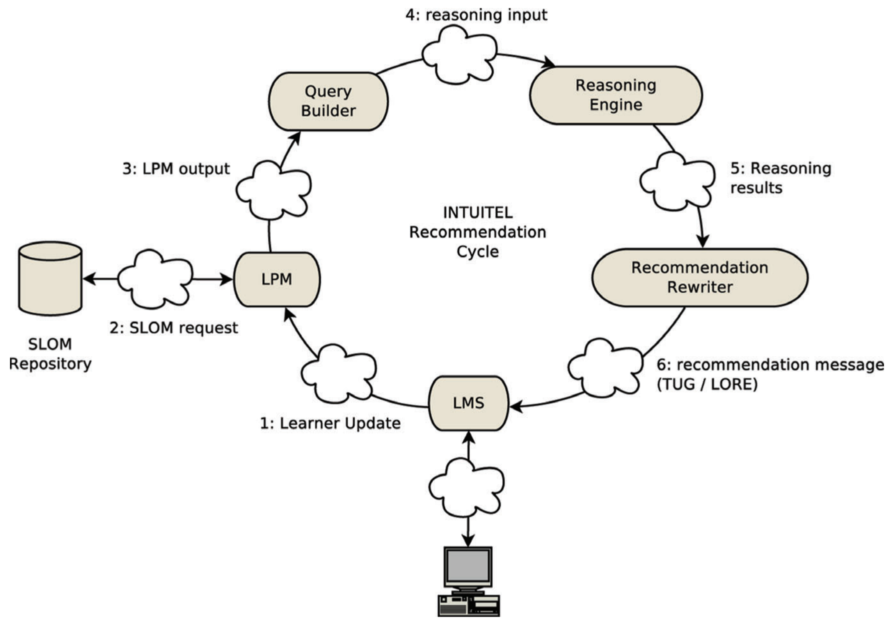


Figure 4.1 INTUITEL recommendation cycle.

For each component only the respective REST end point as shown in Figure 4.1 has to be configured. By using a Communication Layer as a mediator it is sufficient to configure only that Communication Layer as end point for each component. The Communication Layer is the only one that is aware of all other components. Also, the LMS is configured to the Communication Layer end point. This is especially convenient for the development of LMS plugins that then only have to consider that single end point. Its configuration contains routing tables for all message types in order to direct the communication from a specific source to its correct destination. As an example, Table 4.1 lists the routing table for all messages that are needed for the communication with the LMS plugin.

The prototype software can be downloaded from the INTUITEL website[1]. The LPM including the Query Builder and the Reasoning Engine is implemented in Java and can run on any platform. The Recommendation Rewriter is also a Java program. The Communication Layer is written in C# which means that you will need a Windows platform. However, using the Communication is optional although we recommend it. If you do not use the Communication

[1]http://www.intuitel.eu/resources/

Table 4.1 Routing table of the Communication Layer for LMS communication

Source	Message Type	Destination
LPM	LmsProfile request	LMS
LMS	LmsProfile response	LPM
LMS	LearnerUpdate request	LPM
LPM	LearnerUpdate response	LMS
LPM	UseEnv request	LMS
LMS	UseEnv response	LPM
LPM	UsePerf request	LMS
LMS	UsePerf response	LPM
Recommendation Rewriter	LORE request	LMS
LMS	LORE response	Recommendation Rewriter
Recommendation Rewriter	TUG request	LMS
LMS	TUG response	Recommendation Rewriter

Layer you will have to configure the end points for each component as shown in Figure 4.1.

4.2 LMS Plugins

In order to prove the concept of INTUITEL, we now introduce four exemplary implementations of LMS extensions. We provide extensions for Moodle, Ilias, eXact and IMC Learning Suite. These extensions provide the interfaces as defined by the data model in Section 3.5 to make INTUITEL communicate with the respective LMS.

Let us briefly remember the functionality which a LMS extension has to implement. First, a Learner Update message has to be implemented to inform INTUITEL about learner actions. Second, the LMS extension has to provide a USE interface by which INTUITEL can yield performance and environmental data about a learner. Third, the LMS extension must be capable of receiving and displaying TUG and LORE messages. Fourth, the LMS extension has to provide an interface for the INTUITEL Editor.

The definition of these interfaces is exactly the same for any LMS extension. The INTUITEL backend is indifferent to which LMS it connects to as long as the communication standards, which were defined in Section 3.5, are satisfied. Therefore, in the following, we will provide an altering view, focusing on a different aspect of the implementation for each extension. You will realize that the architecture and implementation of the extensions is very different. This is for two reasons: First, the implementation depends on the respective LMS. Second, except for the clearly defined interfaces that are needed for the communication between the extensions and INTUITEL, there are no restrictions with respect to the way of implementation or the technology that is used. The introduced implementations therefore include a variety of technologies like Java, JavaScript, C#, PHP, mySQL etc. Actually this emphasizes INTUITEL's claim to be a technology-independent approach.

4.2.1 Moodle

Elena Verdú, María J. Verdú, Luisa M. Regueras and Juan P. de Castro

Moodle is a pure-HTTP platform developed with accessibility and portability in mind. It has several extension mechanisms that can be used to add code to different stages of the cycle of life of page generation. The INTUITEL extension is implemented using the API for building Blocks. Blocks are a special type of plugins that render a box with contextual content. They are configurable to be embedded besides the main content of many pages. This is the mechanism of choice to implement the communication with the student and to trigger notifications of events to INTUITEL services from the LMS.

With respect to INTUITEL, there is not any communication channel to the student that can be started by the LMS; all information should be pulled by the student's browser. Hence, the Moodle adaptor relies on the persistence of the TUG and LORE messages in the INTUITEL server to function. This approach avoids dealing with messages timing and sequencing and takes advantage of the return channel of Learner Update requests for getting TUG and LORE content.

There is no problem to implement REST service points because every script in the PHP server acts as a potential REST service. Every access to these services is checked against a white-list of allowed remote clients to secure the integration with INTUITEL engines. This list is maintained by an administrator in the global settings of the Moodle platform. All parameters used in SQL queries are filtered using a Moodle API to avoid SQL injection attacks. This behavior fulfills the security requirements of INTUITEL system.

4.2.1.1 Implementation and architecture

Being implemented as a block of Moodle, the source code of INTUITEL interfaces must be placed in the folder "blocks" of Moodle within a folder called "intuitel": [MoodlePath]/blocks/intuitel. Figure 4.2 shows the main components of the architecture of the implementation of the interfaces for Moodle.

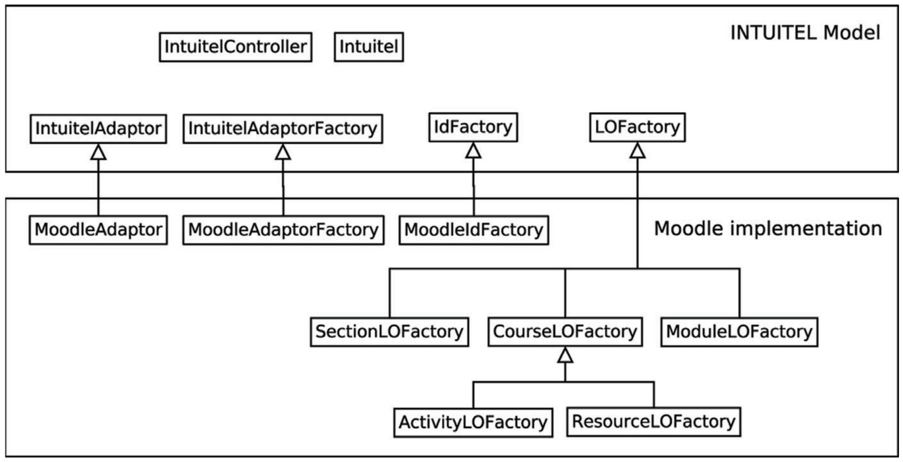


Figure 4.2 Architecture of the Moodle-plugin.

In order to allow the reusability of the model in future implementations or for other LMSs, the classes shown in the frame "Intuitel Model" in Figure 4.2, define the needed data structure, and define, and/or implement, all the operations that allow any LMS to operate with INTUITEL. The classes included in the "Moodle implementation" frame are specific to Moodle, specializing the methods to adapt the data structure and operations to Moodle.

As shown in Figure 4.2, MoodleAdaptor, MoodleAdaptorFactory and MoodleIDFactory are specializations of their respective parent classes needed to perform the defined operations in Moodle. The different descendants of LOFactory are also needed to create the specific LOs of Moodle. Next, some operations of the INTUITEL protocol are going to be detailed. Note that these operations have to be implemented for any LMS extension. Therefore, the following elaborations also apply for the other LMS extensions.

Lmsprofile: Specific capabilities of the Moodle instance are specified in the response to an LMSprofile request. The method to obtain the capabilities is implemented in any descendant of IntuitelAdaptor (i.e. class MoodleAdaptor). It simply returns an array of properties used to compose the response, which include lmsName, husType (fixed as "Moodie") husld (a configuration parameter) and comStyle (fixed as pull communication).

Leamer Update: In order to notify INTUITEL about LO transitions, each interaction of the student triggers an event notified to INTUITEL by means of a LeamerUpdate request that expects in return a TUG and a LORE message piggy-backed in the response (Figure 4.3), By default, only one event is

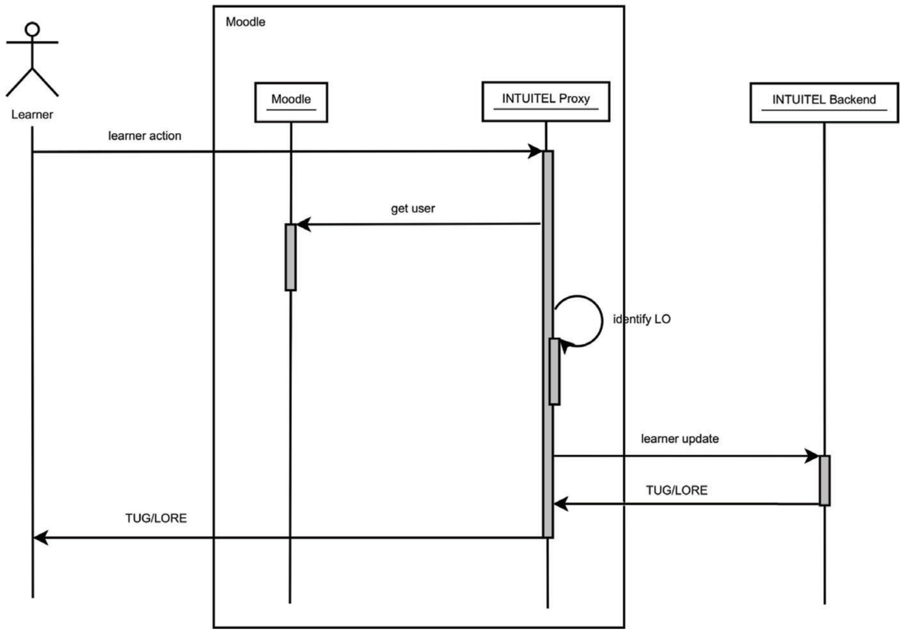


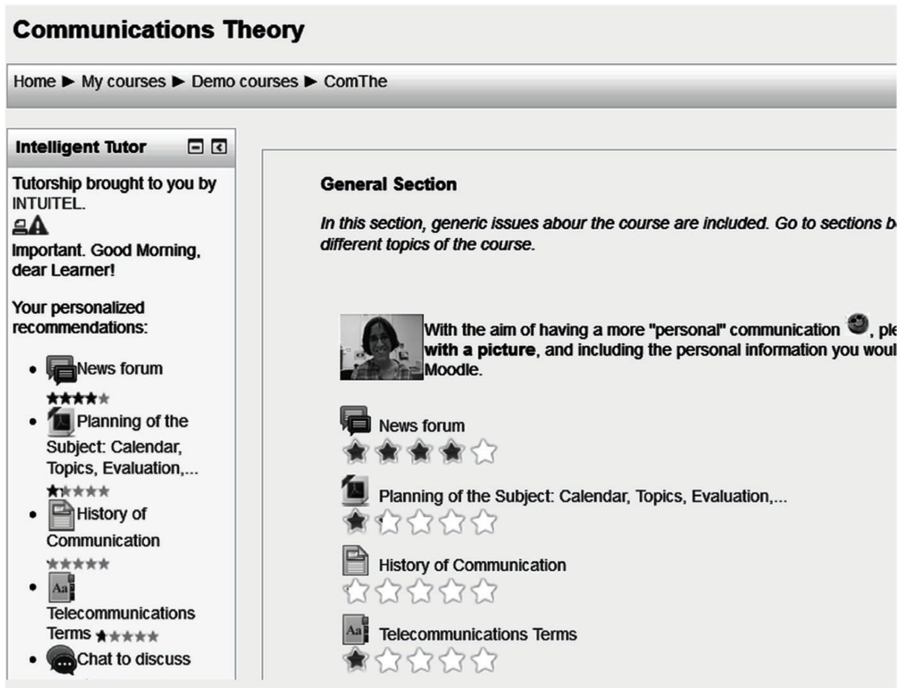
Figure 4.3 Message sequence for learning recommendations in the Moodle-plugin.

reported although the implementation allows also reporting a list of events from the event log, For every user, the system registers the last time when the visited LOs have been informed to INTUITEL.

LO Mapping and Inventory: The INTUITEL editor needs to retrieve the LOs in a course to allow the content creator to create the pedagogical modeL This service returns the structure of a Moodie course including sections and modules.

TUG and LORE: As there is no way to deliver any message to the user in response of a TUG or LORE request from the INTUITEL server, the only answers are: return an “error” response message if the user is unknown; or return a “pause” response message to tell INTUITEL that the user is not aware yet of the message but he is expected to visit a page in the future.

Effective TUG messages and LORE recommendations are received as responses from the INTUITEL server to the Learner Update requests sent by the LMS (Figure 4.3), therefore, they are processed each time the INTUITEL block is loaded in the browser (that is, each time the user accesses a new LO). The IntuitelProxy class is in charge of forwarding both the Learner Update requests from the block in Moodle to the INTUITEL server and the


Communications Theory

Home ▶ My courses ▶ Demo courses ▶ ComThe

Intelligent Tutor

Tutorship brought to you by INTUITEL.

Important. Good Morning, dear Learner!

Your personalized recommendations:

- News forum ★★★★☆
- Planning of the Subject: Calendar, Topics, Evaluation,... ★☆☆☆☆
- History of Communication ☆☆☆☆☆
- Telecommunications Terms ★☆☆☆☆
- Chat to discuss

General Section

In this section, generic issues abour the course are included. Go to sections b different topics of the course.

With the aim of having a more "personal" communication, ple **with a picture**, and including the personal information you woul Moodle.

News forum ★★★★☆

Planning of the Subject: Calendar, Topics, Evaluation,... ★☆☆☆☆

History of Communication ☆☆☆☆☆

Telecommunications Terms ★☆☆☆☆

Figure 4.4 Recommendation messages in the Moodle GUI.

TUG requests when some user answers a TUG message displayed in the block. The implementation displays LORE recommendations within the block content. Besides, it modifies the DOM model of the page adding stars to the LO addressed (Figure 4.4).

USE: The USE Performance request has as response, for each user, a list of scores in each LO contained in every INTUITEL-enabled course the user is enrolled in. There are five LO score types, as defined in the data model (see Section 3.5):

- *Grade*: if the LO is configured as gradable (in Moodle activities can be configured as gradable or not by the teacher) and the student has received a grade, the score item will be included in the response for that LO.
- *Completion*: it will be provided in the response if the teacher has enabled the completion tracking for the activity in Moodle. The completion status will be 0% or 100% as completion tracking in Moodle does not implement partial completion tracking.

- *Accessed*: For all LOs this score type is provided in the response, specifying whether a learner has accessed a LO or not, according to the logs registered by Moodle.
- *SeenPercentage*: For LOs whose media type is video or audio, a 0% is indicated in the response for this score type if the object has not been accessed by the user, and 100% in the opposite case. There is no possibility in Moodle to track a partial percentage.

The response to a UseEnv request includes environmental data of one or more users: name of the user, currently used type of device, current daytime of the learner and the current location of the user.

4.2.1.2 How to configure INTUITEL in a Moodle site

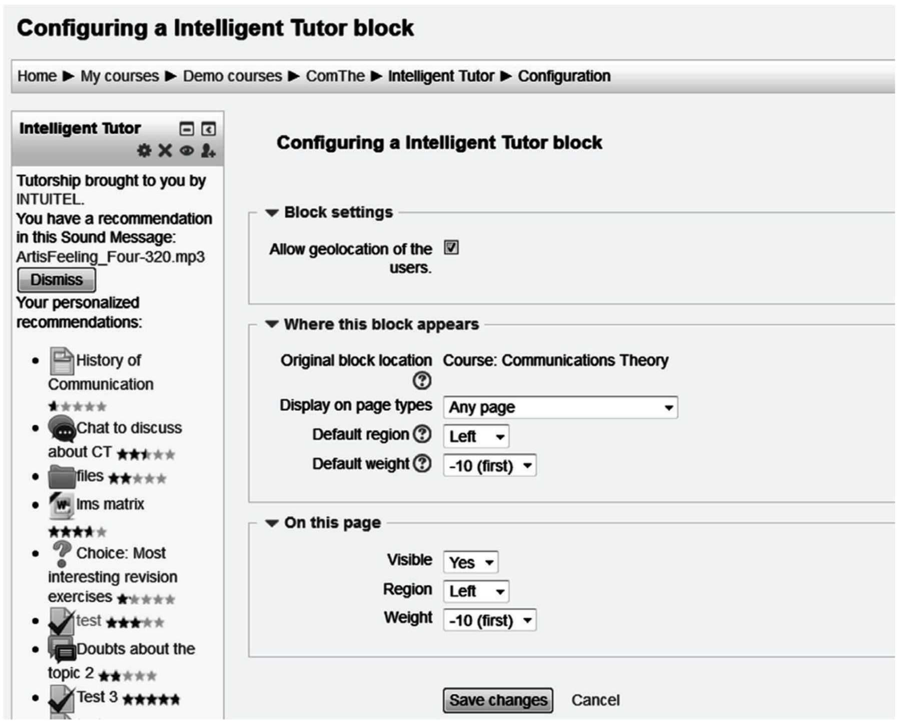
INTUITEL is implemented as a block and therefore it can be installed in a Moodle site as any other plugin of Moodle. Once it is installed, it is important to configure a few different parameters (menu → Administration → Site administration → Plugins → Blocks → Intelligent Tutor):

- *Identification prefix for this Moodle instance*: This string is used as part of identifiers throughout the INTUITEL system. It is used to ensure that the ids are unique and relevant for this platform. The default value is taken from the unique identification of the Moodle installation, but this can be set to another meaningful value or to retain an old prefix in case the server has been moved and need to reuse existing INTUITEL activities and configurations.
- *List of IPs allowed sending INTUITEL events to this LMS*: this is a white-list of the hosts allowed to send messages and requests to this instance. Only IPs of trusted INTUITEL servers should be included here.
- *Service Point IP for using INTUITEL services*: location of the remote INTUITEL servers.

4.2.1.3 How to enable INTUITEL in a Moodle course

INTUITEL in Moodle is presented as a standard Moodle block with name “Intelligent Tutor” that can be added in any course. By default, the users with role “teacher”, “editing teacher”, or “manager” will be able to add the INTUITEL block in the course. With the edition option activated, the user can activate the INTUITEL interfaces for the course by adding the block “Intelligent Tutor” into the course.

Next step is to check the configuration of the block by clicking on the corresponding icon so that the configuration options are displayed (see Figure 4.5).


Configuring a Intelligent Tutor block

Home ▶ My courses ▶ Demo courses ▶ ComThe ▶ Intelligent Tutor ▶ Configuration

Intelligent Tutor

Tutorship brought to you by INTUITEL.
You have a recommendation in this Sound Message: ArtisFeeling_Four-320.mp3

Dismiss

Your personalized recommendations:

- History of Communication ★☆☆☆☆
- Chat to discuss about CT
- files
- lms matrix
- Choice: Most interesting revision exercises
- test
- Doubts about the topic 2
- Test 3 ★★★★★

Configuring a Intelligent Tutor block

Block settings

Allow geolocation of the users.	☑

Where this block appears

Original block location	Course: Communications Theory
Display on page types	Any page
Default region	Left
Default weight	-10 (first)

On this page

Visible	Yes
Region	Left
Weight	-10 (first)

Save changes Cancel

Figure 4.5 Configuration options of the INTUITEL block.

For the options "Display on page types", the option "Any page" must be selected so that the block appears in any page of the course. It is recommended to make the INTUITEL block very visible by situating it in high priority positions so that TUG messages and LORE recommendations are easily noticed by the student. Last, the teacher can select if geolocation of the student will be registered or not.

4.2.2 IMC Learning Suite

Uta Schwertel and Sven Steudter

The INTUITEL services have been integrated into the Learning Management System IMC Learning Suite (ILS)[2]. The IMC Learning Suite is a comprehensive commercial e-learning solution for the planning, implementation and

[2]see https://www.im-c.de/en/learning-technologies/learning-management/ (Last accessed on 15 August 2016).

management of company-wide professional learning and development. The learner interacts with the INTUITEL enhancements (recommendations and tutorial guidance) through an IMC Learning Portal. The Learning Portal offers a lightweight, responsive and easy-to-use front-end that allows learners personalized access to online courses on different end-user devices.

Through the Learning Portal a learner can find and book an online course, and then learn with the course materials (e.g. watch videos, work through Web-based-trainings, read PDFs or other documents, perform multiple-choice tests, discuss with peers about the course content, order certificates etc.). Tutors and administrators use the more powerful IMC Learning Suite administrator view to create courses, supervise courses, administer participants or manage certificates.

For the INTUITEL enhancements we utilized and extended the Universal API of the Learning Suite (4.6). The INTUITEL specific settings (e.g. Service Point IP for using INTUITEL services) and other technical configurations are defined by an administrator in two XML based configuration files. This makes INTUITEL Learning Suite installations easily reproducible on other servers.

An INTUITEL enabled scenario for the IMC Learning Suite requires the following components: an INTUITEL enhanced Learning Portal connected to a Learning Suite, an INTUITEL back-end installation connected to the Learning Suite and Learning Portal, and an integrated online course annotated with INTUITEL specific pedagogical metadata.

In the previous Moodle example we explained in detail the different messsage endpoints (Learner Update, USE, TUG, LORE, etc.). In contrast to the following paragraphs focus on the access to these components from a learner and a tutor/teacher perspective. The overall work flow behind this is similar for all extensions. Depending on the LMS functionality there may be slight differences.

4.2.2.1 Course Creation in IMC learning suite

Defining and creating an INTUITEL enhanced course in the IMC Learning Suite involves the following three main steps:

1. Perform Instructional Design and Content Creation: An expert defines the instructional design of the course and provides the content elements (Knowledge Objects) of the course (texts, videos, audios, WBTs, tests, etc.). The example course “Advanced Java Concepts” for the INTUITEL enhanced Learning Portal has been created by the INTUITEL project partner University of Reading. A graphical overview of the instructional design of the example course can be found in Section 4.4.1 of this book.

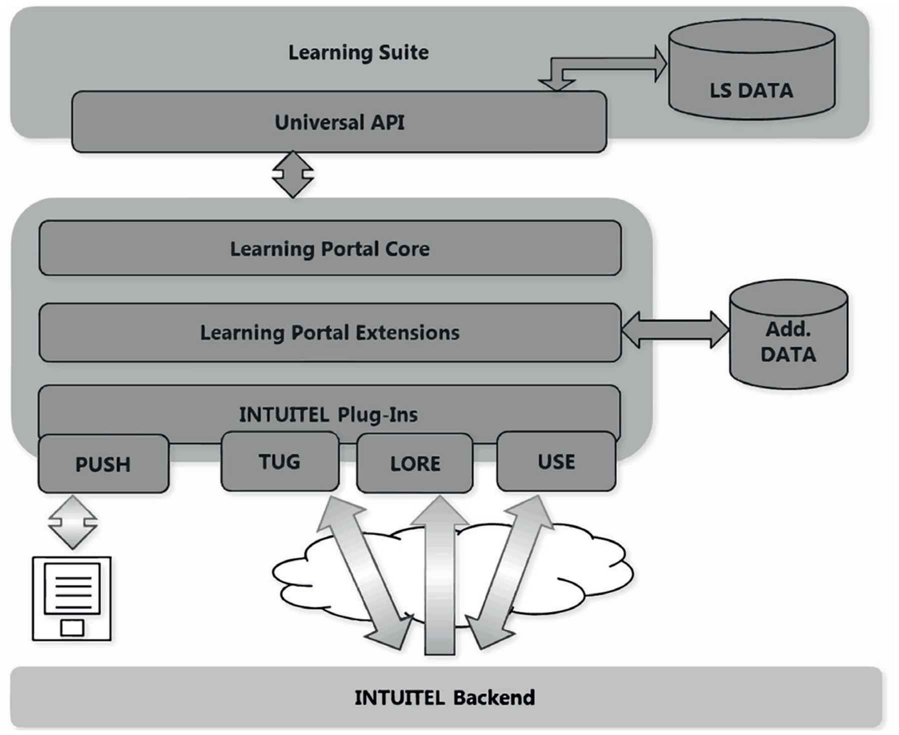


Figure 4.6 Architecture of the INTUITEL enhanced IMC Learning Suite.

2. Create Course in IMC Learning Suite: A course administrator creates the course in the INTUITEL enhanced Learning Suite and uploads the learning objects (Knowledge Objects) for the course. The Learning Suite can be configured as to which meta-data are given within the Learning Suite and which metadata have to be added through the external INTUITEL editor to each learning object. We have chosen to keep the meta-data that have to be added within the Learning Suite sparse and conform to standard non-INTUITEL enhanced courses. Currently available meta-data for each Knowledge Object within the Learning Suite are:

- Name
- Description
- Duration
- Keywords
- Can be used in external applications (yes/no)

- File upload
- Preview image
- Player settings (parameters to configure the internal player)

Note that this set of data is specific for the IMC Learning Suite. Other LMS plugins may provide different data.

3. Annotate Course with INTUITEL Editor: In order to be able of recommending learning steps, INTUITEL needs to have data about the pedagogical design of the course. This data is given outside the Learning Portal and Learning Suite by means of the INTUITEL Editor. This piece of software is an installable program that allows the tutors or teachers to draw Learning Paths by connecting and grouping the Knowledge Objects of the course. The INTUITEL editor is available from the INTUITEL website[3].

To make these annotations, the expert configures the INTUITEL Editor for the respective Learning Management System (with specific configuration data that are given to the expert) and loads the knowledge objects created in the Learning Suite into the Editor. The expert can now define the instructional design in the editor and annotate the objects through the editor.

4.2.2.2 Accessing courses through IMC learning portal

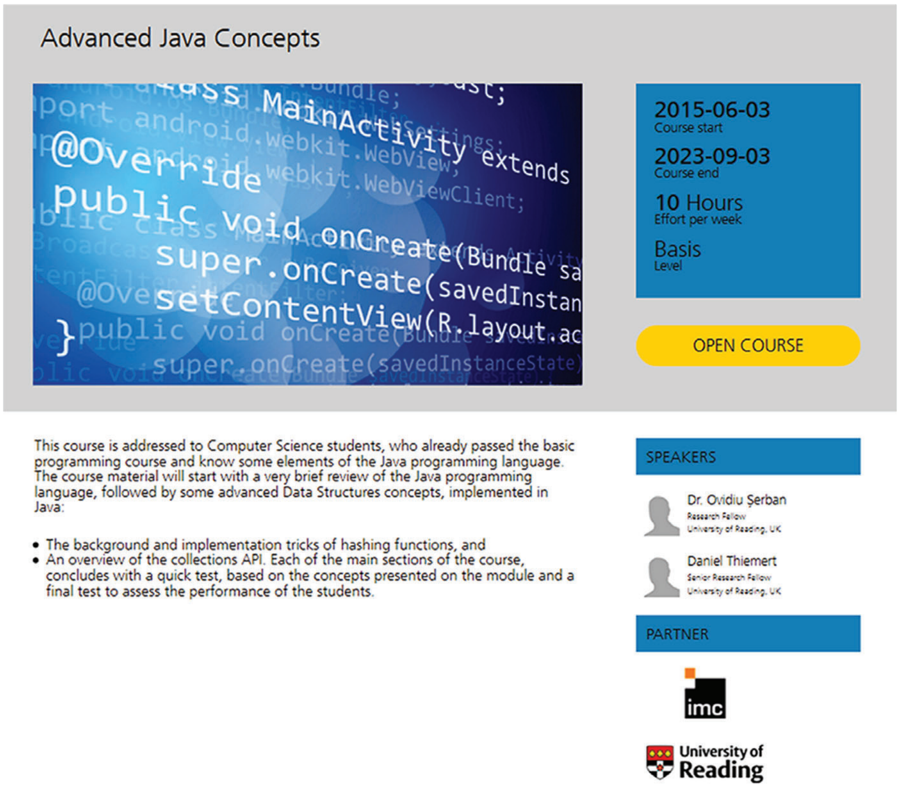
After having finished the above described steps, the course is finally ready for delivery through the IMC Learning Portal. The following paragraphs describe how a learner accesses the INTUITEL enhanced course.

1. Access to INTUITEL enhanced Learning Portal: The learner accesses the landing page of an INTUITEL enhanced Learning Portal. By clicking on “Courses” learners get a list of all available courses in the course catalogue of the system.

2. Sign-up to INTUITEL Learning Portal: The learner has to sign-up to the system to be able to book and use an INTUITEL enhanced course. At this stage it is also possible for the learner to edit his or her user profile data.

3. Book and start the INTUITEL enhanced course: After registering for the course a learner can start the course. The learner will then see the course elements with the main concept containers (CC) and knowledge objects of the course. This means a user can also browse in the course without using the INTUITEL recommendations. Figure 4.7 shows the start page of a course

[3]see http://www.intuitel.de/public-software/ (Last access: 17 November 2016).


Advanced Java Concepts

2015-06-03
Course start

2023-09-03
Course end

10 Hours
Effort per week

Basis
Level

OPEN COURSE

This course is addressed to Computer Science students, who already passed the basic programming course and know some elements of the Java programming language. The course material will start with a very brief review of the Java programming language, followed by some advanced Data Structures concepts, implemented in Java:

- The background and implementation tricks of hashing functions, and
- An overview of the collections API. Each of the main sections of the course, concludes with a quick test, based on the concepts presented on the module and a final test to assess the performance of the students.

SPEAKERS

Dr. Ovidiu Şerban
Research Fellow
University of Reading, UK

Daniel Thiemert
Senior Research Fellow
University of Reading, UK

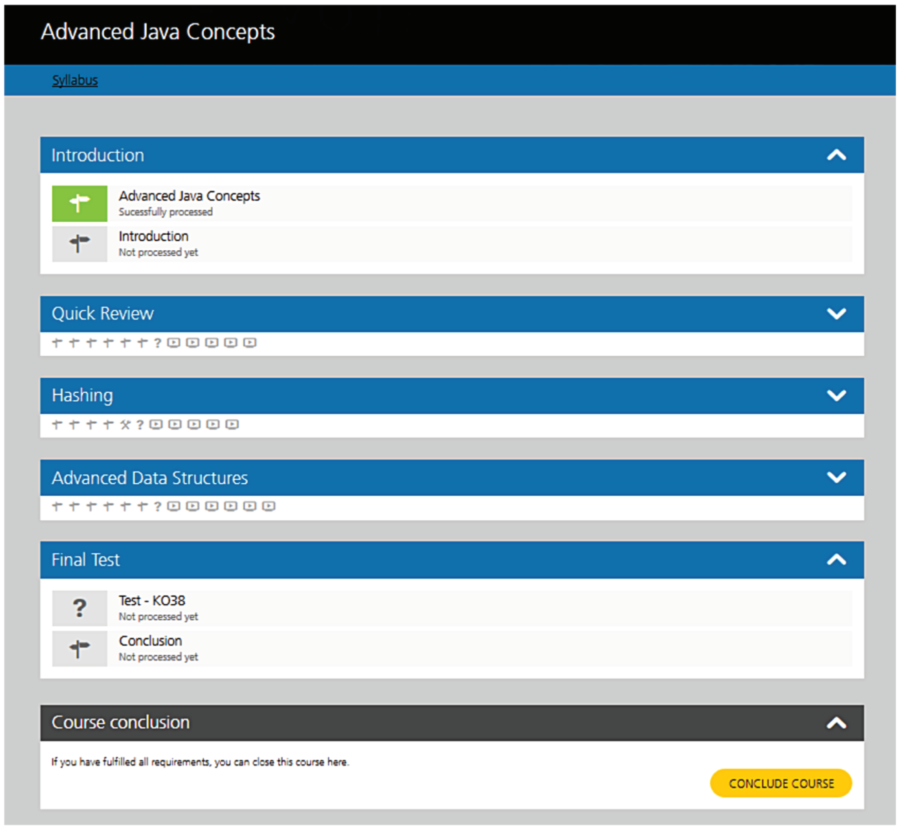
PARTNER

Figure 4.7 Starting Page of a course in IMC Learning Suite and possibility to open course.

whereas Figure 4.8 shows the course curriculum overview as the learner sees it after he or she has booked into the course.

4. Start the course: A learner starts working on the course by clicking for example on the Knowledge Object "Advanced Java Concepts". This will open a detail page where on the right side the learner can work through the material, and on the left side the learner sees boxes where INTUITEL tutorial guidance (Figure 4.9) messages or learning object recommendations (Figure 4.10) will show up.

5. Receiving Learning Recommendations in IMC Learning Portal: As explained in Sections 3.4 and 3.5, learning recommendations are sent to the LMS plugin in the form of TUG and LORE messages. The learner can interact with the messages in the Intelligent Tutor by answering questions or following recommendations shown in the boxes. Depending on the selections, different


Advanced Java Concepts

Syllabus

Introduction

Advanced Java Concepts
Sucessfully processed

Introduction
Not processed yet

Quick Review

Hashing

Advanced Data Structures

Final Test

Test - KO38
Not processed yet

Conclusion
Not processed yet

Course conclusion

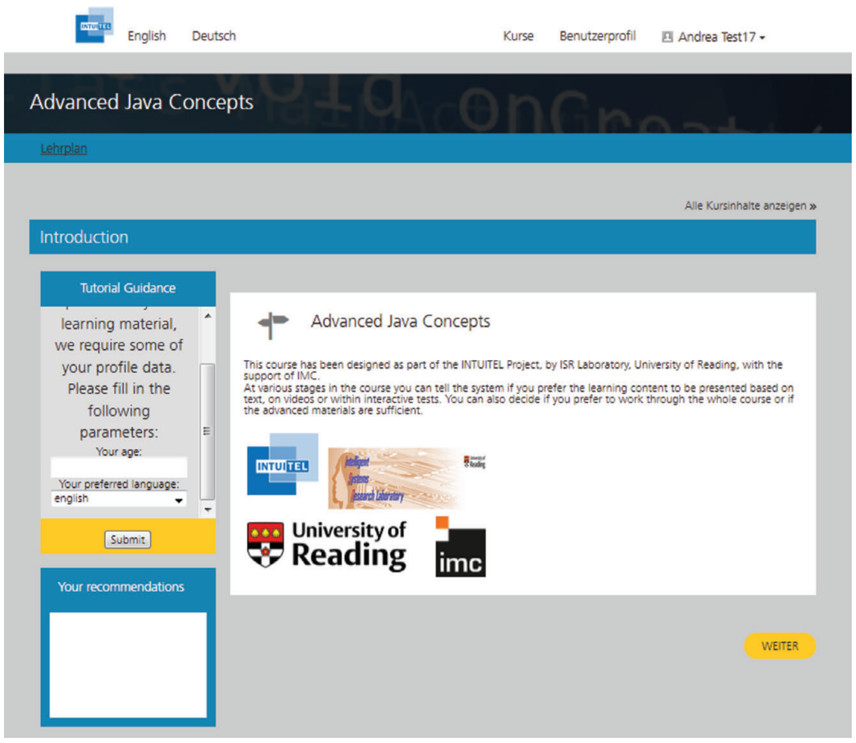
If you have fulfilled all requirements, you can close this course here.

Figure 4.8 Course curriculum in standard view.

recommendations are calculated by the INTUITEL back end. An example run through the INTUITEL example course is described in Section 4.4.1.

Tutorial Guidance (TUG): Through a tutorial guidance message a user can select which Learning Pathway he or she prefers. Other types of tutorial guidance messages ask for other preferences or personal data, e.g. age or preferred language (see Figure 4.9), or simply state that currently no guidance is available or simply state that currently no guidance is available.

Learning Object Recommendations (LORE): In a Learning Object Recommendation message (LORE) the system recommends next relevant learning objects when a user is at a certain state in the learning path. If the system also sends out weightings of the recommendations this is displayed by coloring



learning material, we require some of your profile data. Please fill in the following parameters:

Your age:

Your preferred language: english

Advanced Java Concepts

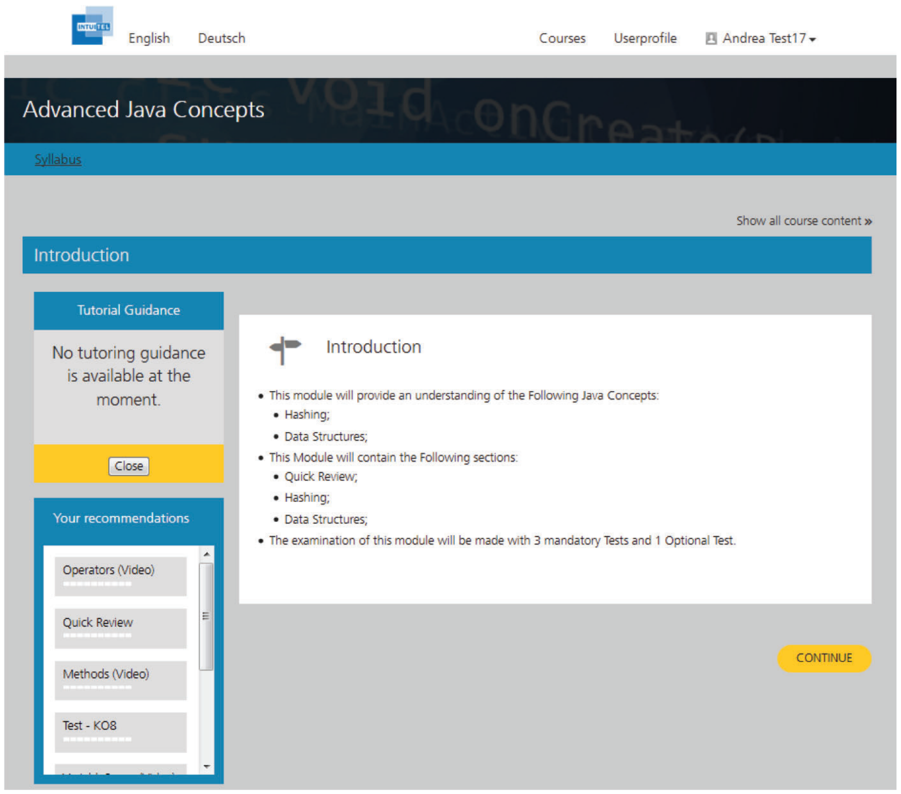
This course has been designed as part of the INTUITEL Project, by ISR Laboratory, University of Reading, with the support of IMC.
At various stages in the course you can tell the system if you prefer the learning content to be presented based on text, on videos or within interactive tests. You can also decide if you prefer to work through the whole course or if the advanced materials are sufficient.

Figure 4.9 Example Tutorial Guidance (TUG) – personal preferences.

the scale chart below the recommendation as illustrated by Figure 4.10 (the width of the chart correlates to the weight of the recommendation).

4.2.2.3 Summary

We have shown how the INTUITEL services have been integrated into the commercial Learning Management System IMC Learning Suite and how the recommendations and tutorial guidance are delivered through the IMC Learning Portal. We have conducted experiments to check if the INTUITEL tutorial guidance improves the user experience and quality of learning. For the IMC Learning Suite these tests have been conducted under the lead of the University of Reading. Users stated e.g. that the INTUITEL enhanced Learning Suite offers great potential for improving Open Online Courses towards a more personalized and motivating user experience. To demonstrate the power of the INTUITEL recommendations more courses with a complex pedagogical modeling will have to be developed and presented to users.



Introduction

- This module will provide an understanding of the Following Java Concepts:
  - Hashing;
  - Data Structures;
- This Module will contain the Following sections:
  - Quick Review;
  - Hashing;
  - Data Structures;
- The examination of this module will be made with 3 mandatory Tests and 1 Optional Test.

Figure 4.10 Example Learning Object Recommendations (LORE).

4.2.3 Ilias

Kevin Fuchs and Peter A. Henning

ILIAS is a free open source Learning Management System. It provides an integrated environment for sharing, structuring and organizing learning content. Moreover, it provides functionality for communication and collaboration among learners and teachers. It is also capable of tracking a user's progress.

To integrate the ILIAS LMS into the INTUITEL system, the LMS had to be extended by a plugin that comprises two components: a REST interface to communicate with INTUITEL and – behind the REST interface – a compontent that retrieves the desired data related to users and learning content from the LMS. The following will briefly describe the implementation of the plugin and how it communicates with the INTUITEL system.

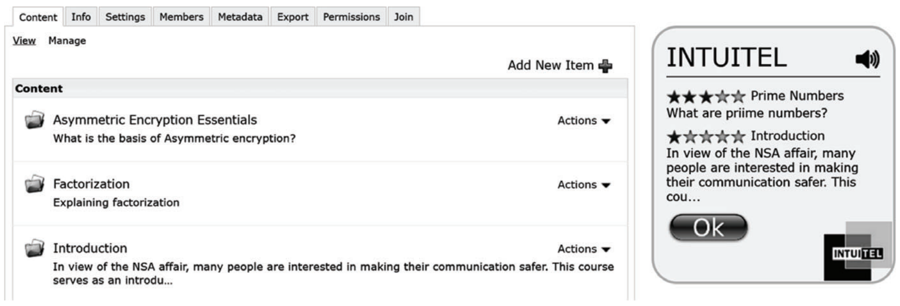
4.2.3.1 Plugin implementation

The plugin is based on a ILIAS-skin that can be activated or deactivated by the user. In his personal settings the user may select an INTUITEL skin instead of the ILIAS default skin. So the user may decide himself if he wants INTUITEL functionality while working with the LMS. In contrast to the IMC Learning Suite, no separate activation of the INTUITEL functionality for a single course is needed. The ILIAS plugin sends Learner Updates for any learning object the learner accesses without caring if the respective course has been annotated. In the case of unannotated learning content, the INTUITEL back end will simply perform a recommendation process with no result. On the one hand this enforces a recommendation process even if it is not needed. On the other hand it makes the plugin more lightweight.

The communication between the plugin and the INTUITEL communication layer runs asynchronously in the background. Hidden from the user, Learner Update messages as described in Section 3.5 are sent to the INTUITEL communication layer. This happens each time a learner enters a learning object or finishes a test.

The Learner Update initiates the respective message sequences (see Section 3.4) to trigger the recommendation process in the INTUITEL back end. The result of this process is sent from INTUITEL to the REST interface of the ILIAS plugin in the form of TUG and LORE messages. The plugin then injects this information into the ILIAS web frontend, displaying a info box to the user. These messages may contain Learning Object recommendations (LORE) in the form of ranked lists of other related learning content or Tutorial Guidance messages (TUG) which may be textual learning recommendations, and interactive forms with which the user is asked some questions in order to refine the recommendation process or let the user choose from multiple recommended learning pathways. Figure 4.11 shows how a LORE message looks like in the web frontend of ILIAS.

In general, ILIAS offers two ways to retrieve data from its database. First there is a SOAP Web Service interface by which data can be read and manipulated. This is most interesting for external applications that want to access the ILIAS database from outside. Second, ILIAS offers a plugin architecture with which new functionality can be easily installed from the ILIAS administration back end. In the case of the INTUITEL system we decided to use the internal plugin mechanism of ILIAS since only this way we could inject the INTUITEL messages into the ILIAS frontend seamlessly. Furthermore, this made it easy to implement the beforehand mentioned skin functionality.



Content

Asymmetric Encryption Essentials
What is the basis of Asymmetric encryption?

Factorization
Explaining factorization

Introduction
In view of the NSA affair, many people are interested in making their communication safer. This course serves as an introdu...

INTUITEL

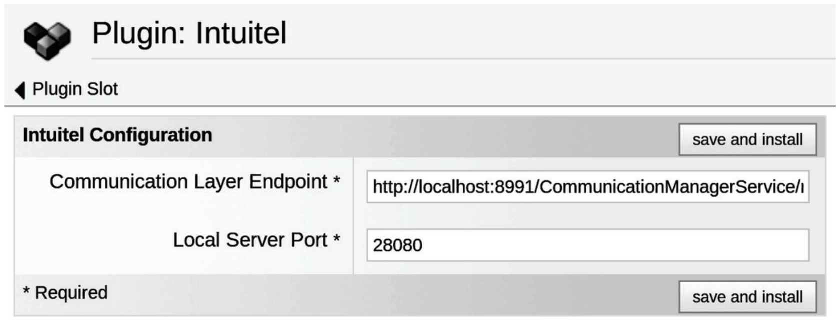
★★★☆☆ Prime Numbers
What are priime numbers?

★☆☆☆☆ Introduction
In view of the NSA affair, many people are interested in making their communication safer. This cou...

Figure 4.11 A LORE message in the ILIAS web frontend (window on the right).

The plugin is managed from the administrations back end and needs only two parameters: the URL to the REST interface of the INTUITEL communication layer and a local port on which the REST interface of the plugin is supposed to run and which is addressed by the communication layer. The plugin, which is implemented in PHP, installs an integration component and the ILIAS skin. The integration component that is written in Java contains a minimally configured standalone tomcat server that is started by the plugin automatically after installation. The tomcat server listens on the port that is specified by the installation parameter "server port" and runs the REST interface for the communication between the plugin and the INTUITEL system. The REST interface consists of the endpoints explained in Section 3.5.

The skin consists of two variants the user can choose from – one for a popup box and one for a box that is embedded into the ILIAS GUI. Both the popup box and the embedded one are created by a JavaScript that accesses the DOM of the HTML structure. The script simply injects the HTML code of the INTUITEL box into the HTML of the ILIAS GUI.



Plugin: Intuitel

Intuitel Configuration

Communication Layer Endpoint *	http://localhost:8991/CommunicationManagerService/ı
Local Server Port *	28080

* Required

Figure 4.12 Configuration of the plugin in the ILIAS administration back end.

4.2.3.2 Learner tracking

In the first instance the tracking of a learner's behavior is performed by a JavaScript function. That function takes the URL of the currently opened ILIAS page that contains the id of the respective learning object presented on that page. By using asynchronous JavaScript calls (Ajax) this id is sent to the REST interface of the integration component waiting asynchronously for a response. Figure 4.13 shows the data flow between browser, integration component and the INTUITEL system.

While the JavaScript function only tracks the object ids the integration component contains the entire logic to communicate with INTUITEL, as well as to collect and process data from the ILIAS database. As a first reaction of being informed about a learner's action the integration component sends a Learner Update message to the communication manager of the INTUITEL back end. INTUITEL then in turn sends a USE request (see Section 3.5) to the integration component to collect user-specific environment and performance data. To generate this data the integration component accesses the ILIAS database.

In the last step, when the integration component has responded to the USE request, the INTUITEL back end starts its recommendation process and sends the result to the integration component in the form of TUG/LORE messages. Having received that TUG/LORE message the integration component satisfies the original REST request of the JavaScript function by responding with the TUG/LORE content. The JavaScript then displays the TUG/LORE content in the INTUITEL info box within the learner's browser. The way the plugin communicates with the INTUITEL system is due to the push scenario as described in Section 3.5. This means, that once a Learner Update has been

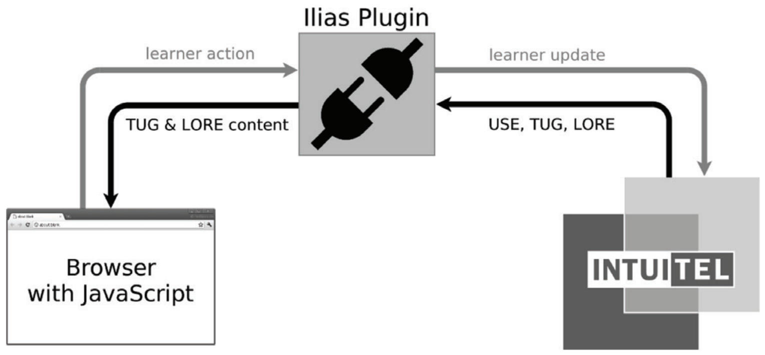


Figure 4.13 Data flow between Browser, integration component and the INTUITEL system.

sent actively by the LMS, then the LMS plugin remains completely passive, only reacting on USETUG, and LORE messages of INTUITEL.

4.2.3.3 Architecture

The inner architecture of the integration component consists of three main components that are illustrated by figure reffig-ILIAS-plugin-architecture and explained in the following.

Communicator The Communicator builds the REST interface with which the INTUITEL communication layer and the JavaScript functions in the user's browser communicate. The Communicator is exchangeable and could also provide any other communication method than REST. being absolutely unaware of the content, sources and destinations of the messages it processes, the only task of the Communicator is the provision of the REST interface as well as passing and receiving messages from and to the layer on the next level which is the Broker described below.

Broker The Broker represents the second layer of the design. Its task on the inbound side is to receive messages from the Communicator and delegate them to the appropriate Core module which can be the TUG, LORE, USE and the Learner Update module, which are described below. This means that–in contrast to the Communicator–the Broker knows about the destinations and sources of messages in order to determine the fitting Core module to which it delegates the message. However, the Broker is indifferent to the contents of messages. On the outbound side the Broker takes messages from the Core components and delegates them to the Communicator.

Core The Core contains the modules that are in charge of handling TUG, LORE, USE messages and Learner Updates. On the outbound side, each of them receives corresponding messages from the Broker. They read the message payloads and take appropriate action. On the outbound side they delegate TUG, LORE, USE and Learner Update messages to the Broker which in turn passes them to the Communicator. The Communicator finally passes the messages either to the INTUITEL Communication Manager or to the user's browser. The Core also contains modules that provide general services for the TUG, LORE, USE and Learner Update modules. Such general services are:

- Marshaling services to transform XML payloads for processing
- Database connectivity to retrieve data from the ILIAS database
- Error handling
- Logging functions

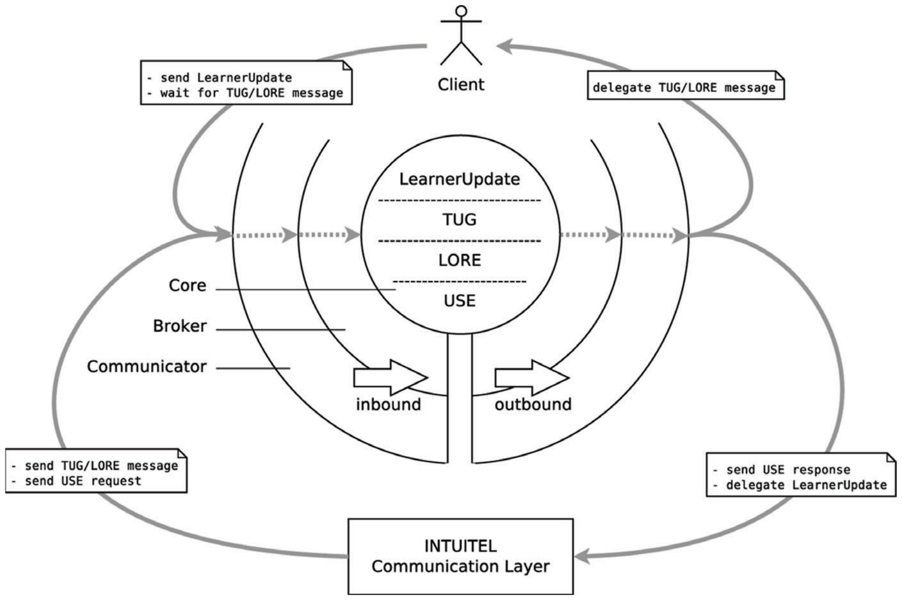


Figure 4.14 Architecture of the ILIAS-plugin.

4.2.3.4 Maintainability

Encapsulating the above described methods in separate services contributes to the ability to replace and modify the according functions without affecting other modules. One could – for example – use other data formatting than XML or migrate to another database. A quite realistic scenario is the occurrence of changes of the ILIAS database scheme, i.e. due to migrations to newer versions. Hence, placing such functionality into separate components contributes to the maintainability and extensibility of the entire system.

4.2.4 eXact learning LCMS

Elisabetta Parodi

eXact learning LCMS[4] is a commercial Learning Content Management System, supporting instantaneous, company-wide collaboration for the creation of standard-compliant, reusable and easily maintainable learning content. Furthermore, the LCMS allows single-source multiple outputs

[4]http://www.exact-learning.com/exact-learning-lcms/

publishing scenarios. The LMS part of it, allowing content delivery, supports course offer and tracking of learners' progress.

In the context of the INTUITEL project, eXact LCMS was extended to interact with the INTUITEL back end in order to send users' data and information about progress in courses and therefore be able to receive INTUITEL related tutoring and guidance messages. Of course, to make a course to be assisted by the INTUITEL tutor, the course needs to be previously annotated, i.e., by the INTUITEL Editor. In the following we present shortly the eXact learning LCMS, how it was extended to exchange communications with the INTUITEL back end and the user experience with the INTUITEL-enhanced interface of the LMS.

4.2.4.1 About eXact learning LCMS

eXact learning LCMS is an industry reference Learning Content Management System that responds to today's varying business pressures, supporting instantaneous, company-wide collaboration for the creation of critical learning content. Developed by eXact learning solutions[5], an Italian SME, eXact learning LCMS maximizes a company's existing content investments, while supporting learning content strategies that improve key business processes. Figure 4.15 shows an overview of the eXact learning LCMS.

Based on XML technology, eXact learning LCMS is designed for the extensive reusability of content chunks. Moreover, it supports SCORM 1.2 & 2004, IMS and Tin Can xAPI standards, allowing single-source multiple output publishing scenarios. eXact learning LCMS allows for rapid content

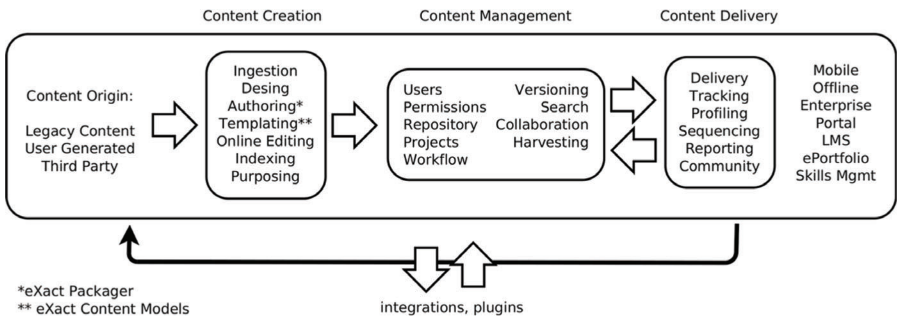


Figure 4.15 Overview of eXact learning LCMS.

[5]http://www.exact-learning.com/company/

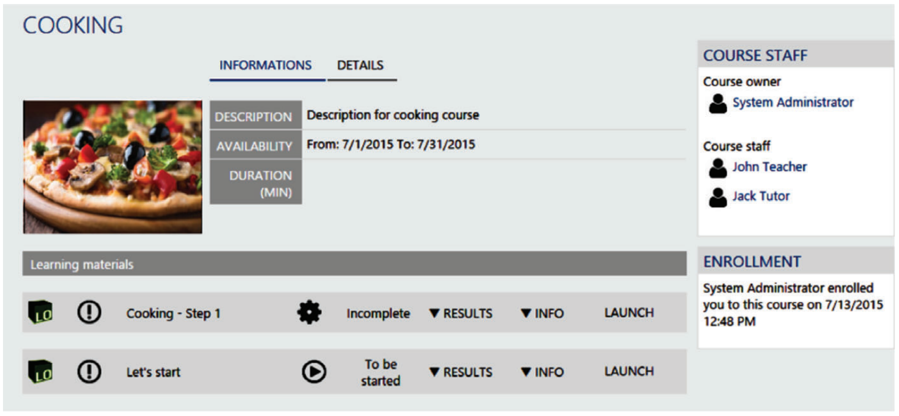
development through the seamless collaboration of subject matter experts and course designers, regardless of their location and specialization. This provides cost-effective, repeatable and consistent content production projects.

The LMS part of the platform allows courses management and delivery. Courses are offered via a catalogue. There are several modalities supported for enrollment, from self-enrollment to teacher's enrollment. Once registered for a course, a learner can access its contents, with an interface similar to the one reported in Figure 4.16.

As from Figure 4.16, information about the whole course is presented on top of the page, while course materials are listed below. For each material, information about its completion is provided by the system (i.e. Incomplete, to be started, etc.). Other information about course staff and enrollment is presented on the right side. The LMS is able to track the learner's interaction with standard-compliant courses. Such information is forwarded to the INTUITEL back end via the developed extension and, together with other information, supports the reasoning for recommendation and guidance.

4.2.4.2 Extension

eXact learning LCMS was put in communication with the INTUITEL back end by means of additional, pluggable code, that we call here "extension". This extension provides on one side back end communication with the INTUITEL services, and on the other side windows added to courses on the front end, supporting the dialogue between the learner and the INTUITEL reasoning. The set-up of the extension is very easy, just creating an IIS virtual directory to the folder containing the code of the web services.



COOKING

INFORMATIONS DETAILS

DESCRIPTION	Description for cooking course
AVAILABILITY	From: 7/1/2015 To: 7/31/2015
DURATION (MIN)	

Learning materials

Cooking - Step 1	Incomplete	▼ RESULTS	▼ INFO	LAUNCH
Let's start	To be started	▼ RESULTS	▼ INFO	LAUNCH

COURSE STAFF

Course owner
System Administrator

Course staff
John Teacher
Jack Tutor

ENROLLMENT

System Administrator enrolled you to this course on 7/13/2015 12:48 PM

Figure 4.16 Sample course interface.

The back end part of the extension is implemented as a set of REST C# web services using Web API Framework (NET Framework 4.5). The choice of such technology is because the eXact LMS is developed in Microsoft C#. The communication is HTTP based and the server side implements a requestre-sponse message system, based on XML format, through the implementation of many specific end points each one of which manages one type of message.

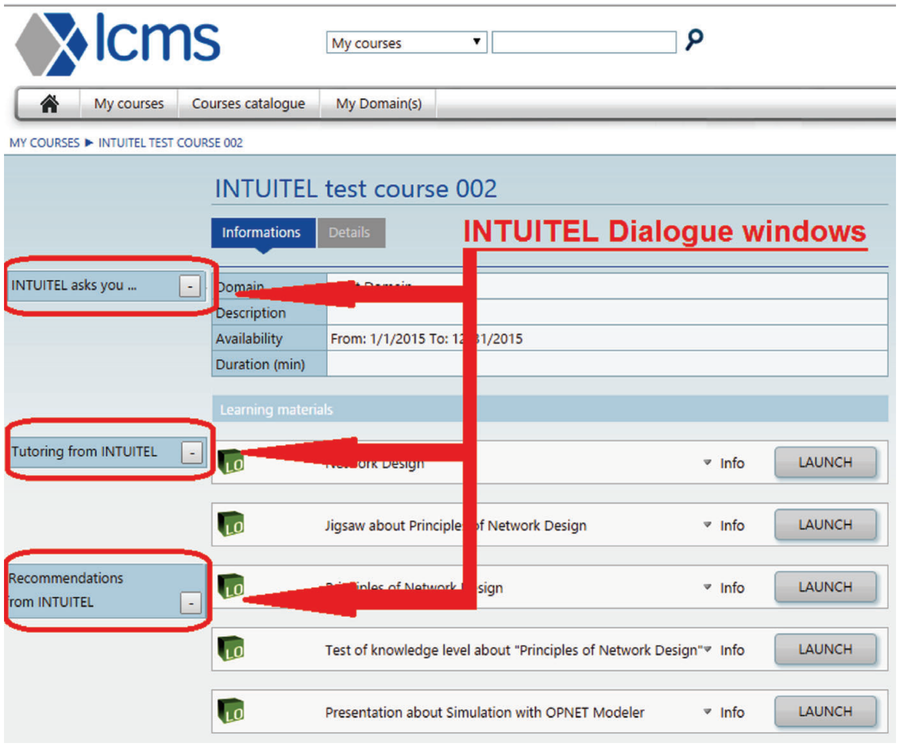
The REST end points and the messages accord to the definitions of Section 3.5. The Web Services are developed in C# as a WEB API project; this can give more flexibility and more compatibility as a standard .Net Web Service. The web services wait for a message from the INTUITEL system and then send an answer back. The Object model has been created using Xsd2Code9 starting from the supplied XSD. Each object (e.g.: Learners, LMSProfile, etc.) implements two specific methods (e.g.: Serialize, Deserialize) used to marshal and transmit the object representation between the communication endpoints. Also the extraction of the data from the eXact LCMS that are requested from INTUITEL is implemented directly in these objects and not in the Web Services, so it possible to use the developed object library also with client applications different from INTUITEL and other communication layers.

For sending Learner Update information we configured the INTUITEL back end for the special communication style "Push with Learner Polling" (see Section 3.5): We implemented the "learners" method assuming that INTUITEL will periodically make calls to the LMS and the LMS returns a list of active users and their activity, meant as courses accessed by each user since previous "learners" call. For the front end, some javascript code manages the windows dealing with the learner-INTUITEL dialogue. The design choice for the user interface was a compromise between the need to allocate part of the screen area for dedicated messages, and the wish to keep the interface neat and essential. Therefore, the adopted solution was to have expandable pop-up windows that can be minimized once the related message has been read.

4.2.4.3 Demonstrator

As previously indicated, our demonstrator contained a course, linked with pedagogical metadata (before being actually delivered to learners) thanks to previous annotation via the INTUITEL Editor. From the learner's point of view, three small frames are added to the default interface, as shown by Figure 4.17.

Entering this course, the first evident novelty respect to non INTUITE-Lenhanced courses is the appearance of three small labels on the left side,



Domain	Domain
Description	
Availability	From: 1/1/2015 To: 12/31/2015
Duration (min)	

Network Design
Jigsaw about Principles of Network Design
Principles of Network Design
Test of knowledge level about "Principles of Network Design"
Presentation about Simulation with OPNET Modeler

Figure 4.17 Sample INTUITEL-assisted course in eXact.

as highlighted in Figure 4.17. These labels can be expanded into small windows, which manage the communication between INTUITEL and the learner. Learners can access and play courses as usual. What is different with the INTUITEL extension is the appearance of the three labels on the left side. These labels can be expanded into small windows managing the communication between the learner and the INTUITEL core. The (expanded) appearance of these windows is presented in Figure 4.18. As illustrated there, INTUITEL-enabled courses in eXact present the following three collapsible windows:

1. A first window presents a few questions to the learner to store information that is not directly retrievable from the basic profile stored by the LMS, such as the daily mood, preferred interaction kind, etc.
2. A second window takes care of the tutoring dialogue, this corresponds to TUG messages. The TUG interface carries out a dialog between the

learner and the INTUITEL system. A TUG Interface must be present in any INTUITEL enhanced system (see Section 3.5 for further description of the TUG interface).

3. The third window provides INTUITEL recommendations about the suggested learning path for the current learner; this corresponds to the LORE interface. The LORE interface makes a recommendation by the INTUITEL system known to the learner. It is, in its simplest implementation, a special case of the TUG interface and therefore present in any INTUITEL enhanced system.

4.3 Compatibility to Existing Learning Formats

Luis de-la-Fuente-Valentin and Daniel Burgos

There are many competing standards for content production and package in the current learning market. For example, SCORM importing and exporting is supported by many Learning Management Systems. Another example is IMS Learning Design, also based on the IMS Content Packaging specification but with a more expressive vocabulary to define learning pathways.

With this technological landscape, with a lot of content already produced in different formats, it is realistic to think of end users as reluctant to shift from their preferred format to SLOM. Furthermore, INTUITEL enabled content requires an edition phase (the enhancement of content with semantic metadata) that may hinder the adoption of the framework.

In order to reduce the INTUITEL adoption gap and make it easy to teachers to migrate from their preferred format to INTUITEL, the framework offers editing capabilities also enhanced with a translation system called the INTUITEL Merger. The Merger is able to translate learning materials from

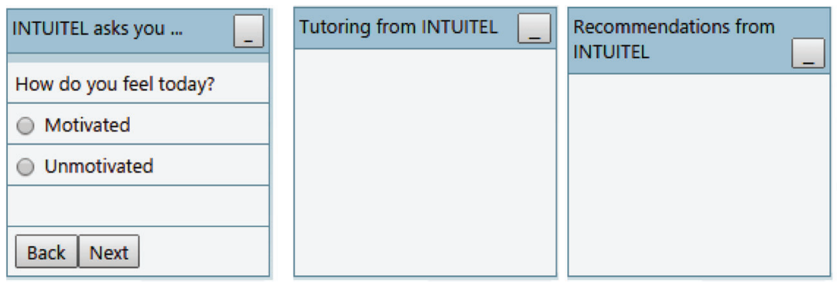


Figure 4.18 Windows for INTUITEL dialogue in eXact LMS.

and to different learning formats, including SLOM. This section offers a technological perspective of the Merger approach and discusses the lessons learned on the implementation of the translation from and to the selected formats, namely IMS LD, Semantic Media Wiki, SCORM and SLOM.

4.3.1 The Transformation Approach

The mantra *write once, run anywhere* was initially proposed by Sun Microsystems as a marketing slogan applied to software source code, specifically the Java virtual machine infrastructure that pursues cross-compatibility among operating systems. This idea of reusing writing efforts conceived for software production has also been adopted in the eLearning field. In other words, as courseware gains more features, it also gains more complexity and, consequently, it is harder to achieve cross-platform compatibility.

Courseware production is a time-consuming, difficult task that requires a great deal of effort from practitioners. Therefore, practitioners may expect their content to be reusable (i.e., may be used more than once) and interoperable (i.e., possible to use on different platforms), but this is not always the case.

This is why interoperability and reusability are always present in any eLearning development. Many standardization efforts have been implemented with different levels of success. A good example is the SCORM framework, whose packaging format has reached the status of a de facto standard in courseware packaging. Many platforms are able to import SCORM packages, while few of them are able to exploit the advanced SCORM features for dynamic content.

New developments must decide whether to be compliant with older formats or to build a completely new paradigm. In the first case, they will inherit the limitations of the older formats. In the second case, new systems will be unable to use a large amount of course content already created for older systems.

Since the SLOM format is an example of this second case, it would be desirable to find a method to SLOM-enhance old course content. The approach taken in the INTUITEL project was to develop a transformation methodology, supported by the INTUITEL Merger as a software transformation tool. In other words, the SLOM format does not inherit restrictions from older content production systems, and provides methods to enable the reuse of older course content.

Many previous works use transformations between formats to achieve course content reusability and interoperability, or even to facilitate new content production paradigms. On the most basic level, the way to translate unstructured learning content into structured learning objects is described [22]. The authors analyze the process of learning object creation and the transformation methods, identifying possible pedagogical issues.

The intrinsic limits of a given framework may also lead to the inclusion of a transformation process in the course life cycle. For example, the inherent complexity of the IMS Learning Design specification hinders the course authoring process, and IMS-LD editors have not achieved the expected ease of use. Certain researchers [6, 51] propose to start the authoring process with well-known techniques used in software production and use transformation tools to generate IMS-LD compliant courses. Barchino et al. [97] follow a similar approach by applying software creation techniques to create SCORM course content. A posteriori analysis of the results reveals an interesting fact: a successful tool and a quality transformation are not enough, since course authoring has pedagogy-specific issues that cannot be addressed by non-pedagogically designed tools. Therefore, it is important that humans complete the process in order to apply the pedagogical touch.

The SCORM format has achieved great success, while it also has some deficiencies. For example, the extensive cataloguing by means of metadata is an obstacle for developers and vendors because the existing content has to be provided with metadata to be SCORM compliant [96]. To increase the interoperability of digital libraries with existing platforms, Arapi et al. [2] developed an architecture that relies on format transformations to interconnect systems.

A review of the literature highlights the relevance of the interoperability and reusability of learning content, and justifies the use of transformation tools to achieve these goals. The following lessons learned have guided the design and implementation of the SLOM transformation tool:

- Transformations may introduce pedagogical problems, as identified in earlier works.
- Well-known tools can be used to improve the authoring experience. Although results are good, pedagogical issues are still present.
- A well-designed transformation tool and methodology enables reusability of successful content in new frameworks.

Different formats for eLearning content production and packaging pursue different goals and have been constructed based on different learning theories. For example, IMS Learning Design claims to support collaborative learning and therefore includes vocabulary to model roles, support activities, forums, groups, etc. However, this is not the case with SLOM, which interprets learning material as something to be individually consumed and takes into account peer activity in order to elaborate recommendations. The same applies to other formats such as SCORM, LOM, IMSCP and AICC. These have been designed on a different basis, and it is therefore not possible to produce a lossless translation between two of these formats.

To overcome this problem, the decision of the INTUITEL team was to support a semiautomatic transformation process, where most of the tasks are automatically carried out by the software, while fine-grain details require human intervention in order to be completed. The complete flow is as follows:

1. The course author identifies a course whose content is adequate for a given knowledge domain, but not written in SLOM format.
2. The course author uses the INTUITEL Merger to generate a SLOM version of the course. This produces a SLOM package.
3. The course author then imports the SLOM package into the INTUITEL Editor and manually completes the transformation.

After this process, the course content is an INTUITEL-enabled course to be used within any INTUITEL-enabled platform. Within this approach, the researchers recognize the impossibility of designing a completely automatic transformation process, while acknowledging the relevance of the details of the course content.

Furthermore, the implemented transformation is bidirectional. That is, SLOM can also be translated into other formats. Such SLOM-to-other-formats transformation follows the same semiautomatic approach, but the other-format-editor required to complete the third step is not provided as part of the INTUITEL framework.

To maximize interoperability and reusability, the INTUITEL team has chosen the best-known eLearning formats on the current market: SCORM and IMS Learning Design. Furthermore, Semantic MediaWiki courses can also be imported and translated into SLOM. Since Semantic MediaWiki (SMW) is not built on a predefined schema nor imposes assumptions regarding the schema used for the description and annotation of learning content, the SLOMto-SMW transformation solution is based on explicit import and mapping declarations that determine how PO and SLOM elements are mapped to Semantic MediaWiki-specific elements and vice versa. These mappings have to be declared on a case-by-case basis since Semantic MediaWiki is per design an ontology-based authoring tool for arbitrary data and domains rather than a prescriptive schema for describing eLearning content such as IMSLD or SCORM. This aspect is fundamental to the SLOM-to-SMW transformation specification. The architectural software approach allows for the inclusion of new formats for translation. The next section briefly describes the implemented transformations.

4.3.2 Implemented Transformations

4.3.2.1 SCORM

The SCORM format stands on the basis of the IMS Content Packaging specification, and therefore uses organization, resource and item as organizational units. The mapping rules take the top-down approach and therefore start by reading the organization elements and hierarchically read inner elements until resources are read.

In the transformation process, each organization found at the SCORM manifest file results in the creation of a CC in the corresponding SLOM file. Each property available in SCORM is transferred as an individual annotation to the CC. Then, items (and related resources) available at the organization are translated to Knowledge Objects in the SLOM file.

Last, sequencing rules are created in the SLOM file. The purpose of INTUITEL regarding a priori sequencing rules is much simpler than in SCORM, because INTUITEL sequencing is calculated at runtime. Thus, the resulting SLOM file has a single Macro Learning Pathway, and each Concept Container has a micro Learning Pathway that contains all the inner Knowledge Objects.

4.3.2.2 IMS Learning Design

The IMS Learning Design also stands on the basis of IMS Content Packaging, and establishes complex sequencing rules that build the structure of the course. The translations read these sequencing rules and map them to the different elements available in the SLOM format.

The procedure follows a bottom-up approach: in the first step, actual learning content is identified (i.e. learning objects inside environments, related to learning activities) and a collection of Knowledge Objects is created. Then, these KOs are wrapped into CC according to the sequencing rules of the IMS-LD course. That is, all the KOs identified in the same act are related to the same CC. Next, the CCs are sequenced into Macro Learning Pathways (MLPs). Finally, the micro Learning Pathways (mLPs) inside the CCs are created. In a typical case, an imported course will have a single MLP and multiple mLPs, one for each CC.

4.3.2.3 Semantic MediaWiki

Semantic MediaWiki (SMW) does not define any new canonical data or description format since the logical model that builds the basis of its knowledge representation formalism is to a large extent based on the Web Ontology Language (OWL). The transformation process is based on the definition of mapping declarations that need to be built for the purpose of a given course, that associate course properties (i.e. Semantic Media Wiki properties) with CCs and KOs.

4.3.2.4 Integration in a common format: INTUITEL merger

The implemented transformations have a very different nature. For example, both SCORM and IMS-LD are based on the IMS Content Packaging specification but, due to their intrinsic characteristics, SCORM transformation is based on XSLT while IMS-LD is built on top of a Java-based procedure. Furthermore, SMW does not offer the concept of a concept package, and the SMW courses are stored in a web server.

Due to all these differences in structure and underlying technologies, a common layer is provided to unify the transformation interface and ease the process. Such a layer, the INTUITEL Merger, is a piece of software able to import from and export to different eLearning formats. The transformation process follows a semiautomatic approach, where fine-grain details require human intervention to be completed.

4.3.2.5 Architectural principles

The purpose of the INTUITEL Merger is twofold. First, its functionality enables the inclusion of already existing course material in the authoring process of INTUITEL-enhanced courses. Second, it enables the exportation of course content in different formats.

As shown in Figure 4.19, the SLOM format plays a central role in all the transformations, that is, whenever an import is executed the result of the importation is an internal representation of the course as SLOM format. This design decision emphasizes the relevance of SLOM as a central part of the INTUITEL Framework. In addition, the structure of the Merger enables non-SLOM transformations. That is, the user can use the Merger to obtain an IMS-LD representation of a Semantic MediaWiki course by just importing and then exporting the course using the proper parameters. With this SLOM-centric procedure, the Merger includes SLOM-to-SLOM importation and exportation, thus enabling the transformation from SLOM files to other formats. According to Figure 4.19, in the first released version of the tool, four formats are

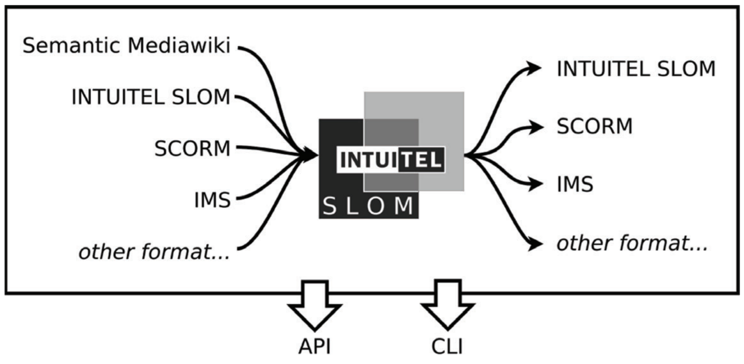


Figure 4.19 Transformations in the INTUITEL merger.

supported (SCORM, IMS-LD, SMW and SLOM). As the implementation of the transformations is plugin-based, the tool is extendable with new supported formats.

4.3.2.6 Merger use cases: Standalone or integrated

The INTUITEL Merger provides an interface for two main use cases: its use as standalone tool and its integration on larger software. In the INTUITEL case, the Merger has been integrated as part of the Editor. Therefore, the user can complete the authoring life cycle. That is, to import a course from an external file, load its SLOM representation into the Editor, manually edit finegrain details or even modify the structure and export to the desired format.

A built-in command line interface enables the use of the INTUITEL Merger as a standalone tool. In this case, the transformation is an end-to-end process that receives an input file (or a URL for SMW imports) and produces an output file in the specified format. Here, the internal SLOM representation is a temporal placeholder that bridges the transformation process, while the end user has no access to such internal representation.

4.4 Sample Courses

To demonstrate the functionality of the INTUITEL prototype, we present three example courses in this section. Note that the creation and structuring of a course is entirely independent from the concrete LMS keeping the learning material. The main task of a course creator is the semantic annotation of the course material on the meta-level according to the Pedagogical Ontology as described in Section 3.1. The concrete LMS content is then linked to the resulting SLOM data (see Section 3.6).

The variety of topics and LMS that we chose for the sample courses especially emphasizes the Plug-and-Play principle by which meta-knowledge can be applied to any learning material of any knowledge domain. Therefore, the following sample courses were all created following the same procedure and we will set a different focus on each of them.

We first start with the course "Advanced Java Concepts" tested with the IMC learning suite. With this course we will focus on the guidance as it is experienced by the user on the screen. The second course – tested with Moodle – is about network design and we will focus on the subdivision of the learning material into Concept Containers and Learning Pathways. The third course was tested with ILIAS and provides an introduction into Albert Einstein's theory of Special Relativity. With this course we will have a closer look at the design of micro and Macro Learning Pathways.

4.4.1 Advanced Java Concepts

Uta Schwertel and Sven Steudter

To test the INTUITEL recommendations with the IMC Learning Suite the INTUITEL partner University of Reading has designed and piloted an online course "Advanced Java Concepts". The course addresses Computer Science students who already passed a basic programming course and know some elements of the Java programming language.

4.4.1.1 Course design

The course starts with a brief review of Java programming language, followed by some advanced Data Structure concepts, implemented in Java: the background and implementation of tricks and hashing functions and an overview of the collections API. Each of the main sections of the course concludes with a quick test, based on the concepts presented in the module and a final test to assess the performance of the students.

The course offers knowledge objects using different media types (video, text, test). For example, the concept "Operators" is explained in text form or alternatively as a video. Furthermore, the course is level oriented and distinguishes between introductory and advanced material and the learner can add preferences regarding the level of detail or the preferred media type. In Figure 4.20 a graphical description of the pedagogical design is given. The elements of the course are summarized in Table 4.2.

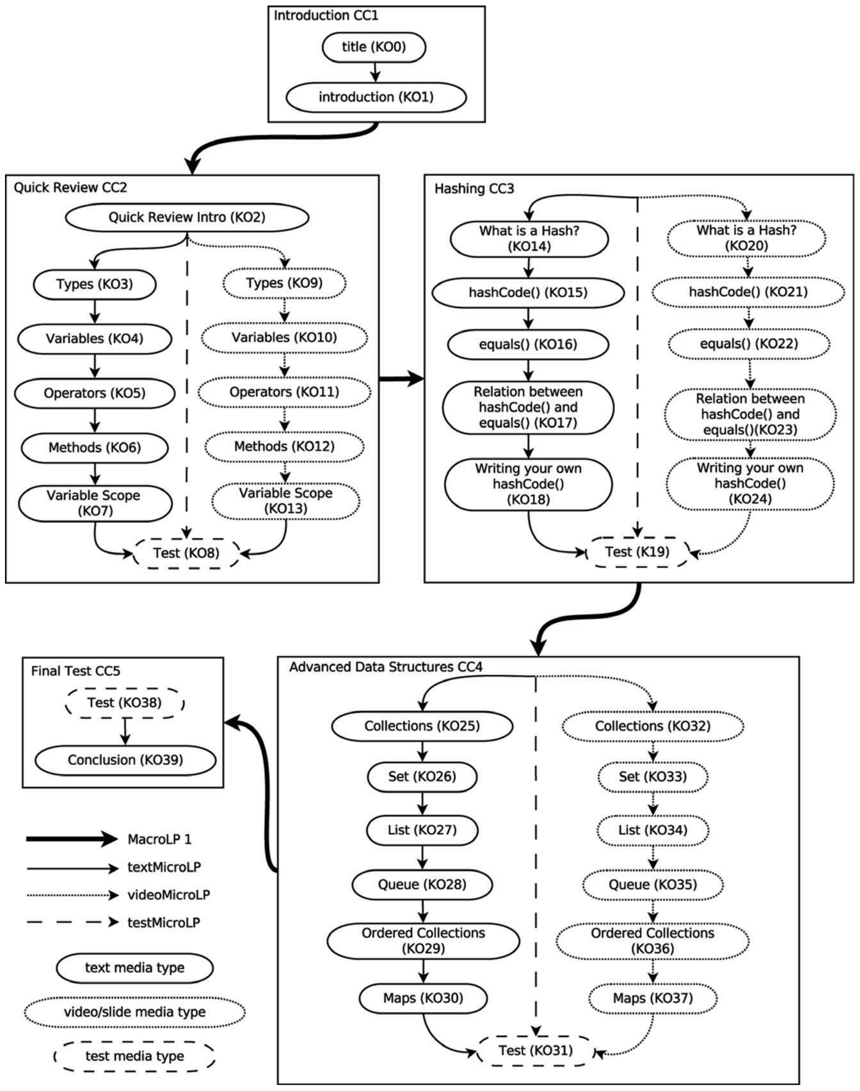


Figure 4.20 Pedagogical design of the course “Advanced Java Concepts”.

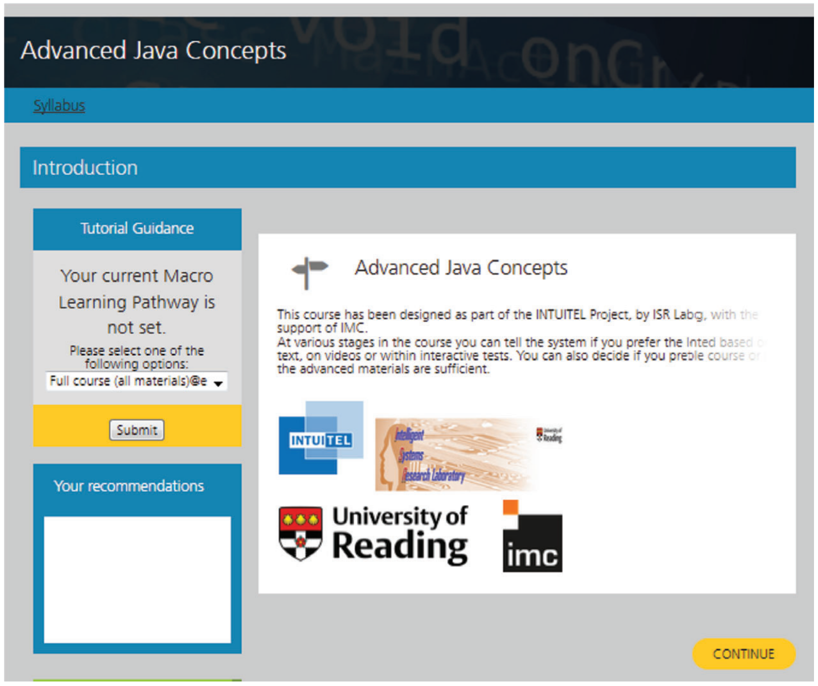
4.4.1.2 Example run through the course

In the following section we describe a brief example run through the course “Advanced Java Concepts”. We assume a newly registered user has opened the “Advanced Java Concepts” Knowledge Object in the course. Through a Tutorial Guidance message triggered by the INTUITEL system the user is then asked for the preferred Macro Learning Pathway, namely, whether he or

Table 4.2 Example elements of course "Advanced Java Concepts"

INTUITEL Structural Concept	Example Elements in Course
Knowledge Domain (KD)	Advanced Java Concepts
Concept Containers (CC)	Introduction (CC1)
	Quick Review (CC2)
	Hashing (CC3)
	Advanced Data Structures (CC4)
	Final Test (CC5)
Knowledge Object (KO) Types	Text, Video, Test
Knowledge Objects (KO)	[KO0] to [KO39] (see Figure 4.20)
MacroLPs	MacroLP1: Full course (all materials)
	MacroLP2: Advanced concepts only
MicroLPs	textMicroLP: I prefer text materials
	videoMicroLP: I prefer videos
	testMicroLP: I prefer tests

she wants to do the full course with all materials or the advanced concepts only. The learner selects in the drop-down list the option "Full course (all materials)", viz. chooses the MacroLP1 (Figure 4.21). In a next step the user



Your current Macro Learning Pathway is not set.

Please select one of the following options:

Full course (all materials)@e

Advanced Java Concepts

This course has been designed as part of the INTUITEL Project, by ISR Labg, with the support of IMC.

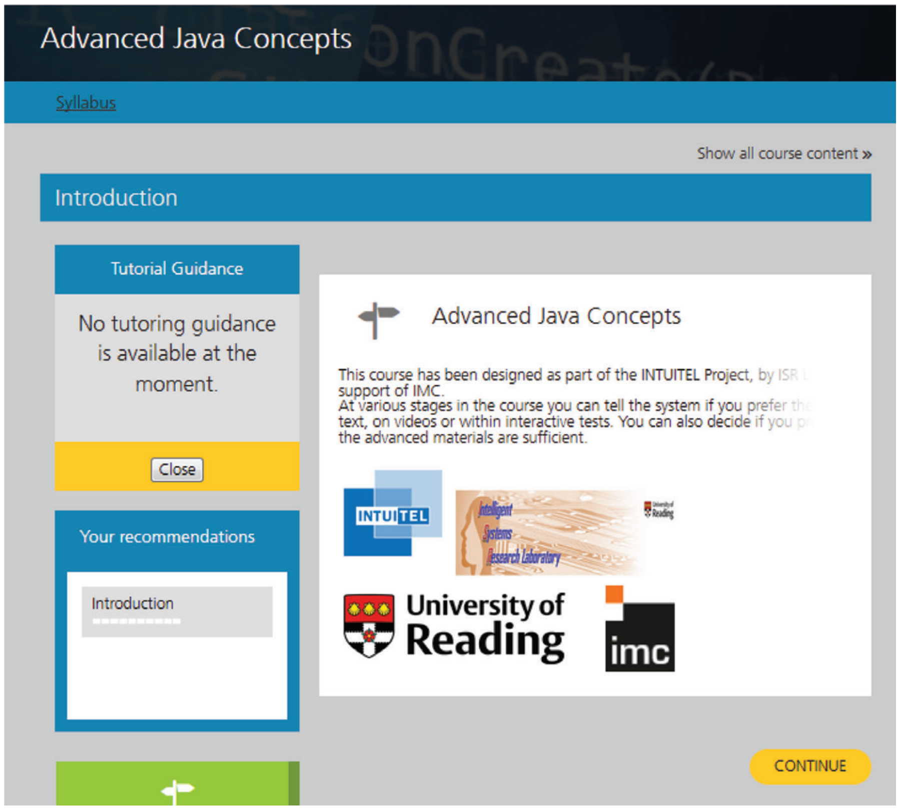
At various stages in the course you can tell the system if you prefer the Inted based on text, on videos or within interactive tests. You can also decide if you preble course or the advanced materials are sufficient.

Figure 4.21 Tutorial Guidance (TUG).

is asked for the preferred MicroLP, viz. whether he or she prefers texts, videos or tests only.

Suppose, the user chooses “I prefer videos”. The modeling of the course does not yet offer alternative Knowledge Objects at this stage which is why the system first recommends the Knowledge Object “Introduction” to the user. This Learning Object Recommendation (LORE) appears in a box where the user can select the recommended learning object (Figure 4.22).

Suppose the user follows the recommendation. This opens the Knowledge Object “Introduction”, which the user can consume, and triggers further Learning Object Recommendations that take into account previous selections of the user regarding the preferred Macro- and Micro Learning Paths, viz. the user is presented preferably videos and is recommended also Learning Objects of Concept Container 2 (CC2: Quick review).



No tutoring guidance is available at the moment.

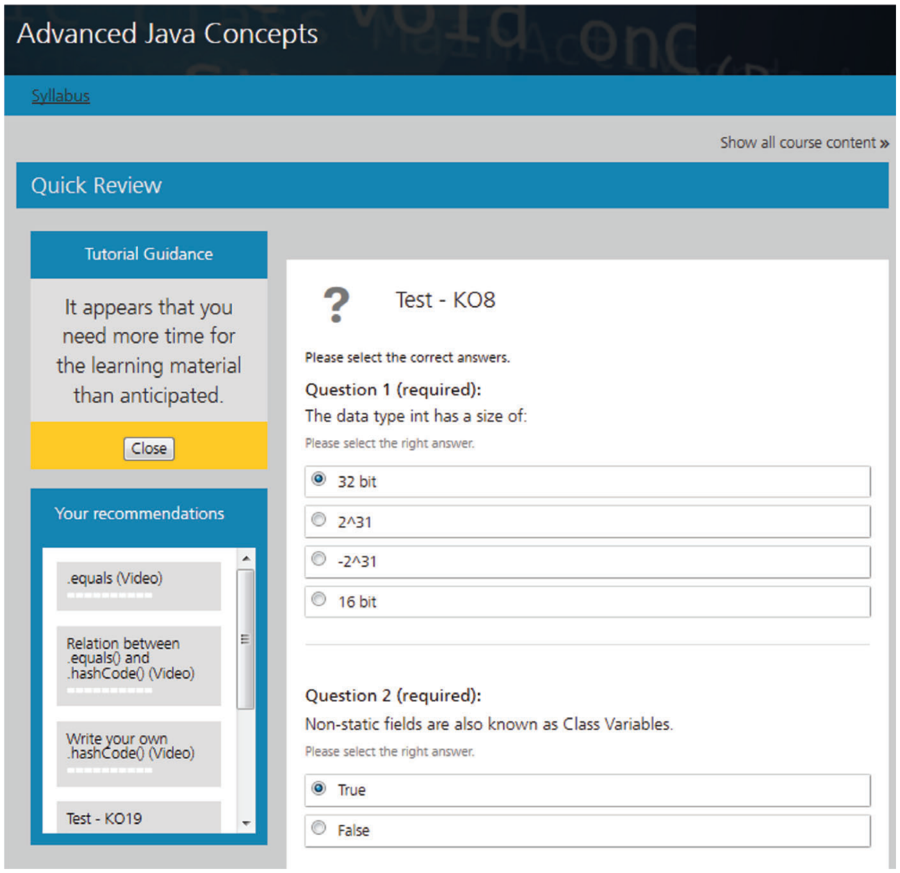
Advanced Java Concepts

This course has been designed as part of the INTUITEL Project, by ISR
support of IMC.
At various stages in the course you can tell the system if you prefer the
text, on videos or within interactive tests. You can also decide if you
the advanced materials are sufficient.

Figure 4.22 Follow first Learning Object Recommendation (LORE).

In our example the user selects the Knowledge Object “Operators (Video)” and can view the video in the right window of the screen. The INTUITEL system calculates further recommendations and displays them in the left box “Your recommendations”. Again, the user is free to choose what to select next, but ideally follows the recommendations provided through the INTUITEL system. The user chooses to do the test (KO8) and is presented the interactive test provided by the IMC Learning Portal (Figure 4.23). The test is evaluated by the Learning Management System and the result (passed/failed) is presented to the user. In the current setting the user can, however, redo the test.

After completing the test the user selects further recommended Learning Objects which now belong to the next Concept Container “Hashing (CC3)”.



It appears that you need more time for the learning material than anticipated.

Test - KO8

Please select the correct answers.

Question 1 (required):
The data type int has a size of:
Please select the right answer.

[✓ 32 bit]
2^31
-2^31
16 bit

Question 2 (required):
Non-static fields are also known as Class Variables.
Please select the right answer.

[✓ True]
False

Figure 4.23 Do a test at the end of a Concept Container.

In the standard view of an online course in the IMC Learning Portal the user can also see which Knowledge Object he or she has already successfully passed. This complements the guidance given by INTUITEL.

4.4.2 Network Design

Elena Verdú, María J. Verdú, Luisa M. Regueras, Juan P. de Castro

The University of Valladolid (UVa) has developed a Cognitive Model for the Knowledge Domain or KD "Network Design". This KD is defined for the course "Laboratory of Design and Configuration of Networks", which is offered at the Telecommunications School of the University of Valladolid. The course is a part of the third year curriculum (out of four) of one of the official degrees given at that School: "Degree on Telecommunications Specific Technologies, mention in Telematics Engineering", and it is supported by Moodle through the Virtual Campus of the UVa.

The course "Laboratory of Design and Configuration of Networks" comprises 15 seminar hours (one one-hour session per week) and 45 laboratory hours (one three-hour session per week) during the 15 week long semester. It does not include any frontal lecture while it is an eminently practical subject, where concepts about communication networks already studied in previous courses are reviewed and applied in order to be able to design and configure communication networks.

Specifically, the course adapted to INTUITEL consists of twenty-four lessons or CCs (Concept Containers) about different aspects related to simulation, network design, IP networking, Wide Area Networks (WAN), Local Area Networks (LAN) and Structured Cabling Systems (SCS). These twenty-four CCs include all the learning objects or KOs (Knowledge Objects) of the Moodle course (both the resources as the activities and communication tools) and are organized through two Macro Learning Pathways as listed in Table 4.3: "Classical Learning Path" and "Alternative Hierarchical Learning

Table 4.3 Macro Learning Pathways for the KD "Network Design"

LP Title	Description
Classical Learning Path	This Learning Path guides you logically through topic clusters as it is usually done in the typical literature of this domain: different types of networks are studied sequentially.
Alternative Hierarchical Learning Path	This Learning Path guides you hierarchically through topic clusters, classifying them by topics and then by type of network.

Path". The first one follows an approach near to "Chronologically from old to new" pathway, while the second one follows a "Hierarchically top down" pathway. The Table 1 shows the defined relations.

Figures 4.24 and 4.25 show the CCs belonging to the KD "Network Design" as well as the relations between them in order to form the sequence of CCs for both macro Learning Pathways: "Classical Learning Path" (Figure 4.24) and "Alternative Hierarchical Learning Path" (Figure 4.25). Gray boxes in Figure 4.24 are used for CCs that belong exclusively to the Classical Learning Path. In Figure 4.25 gray boxes are used for CCs that belong exclusively to the Alternative Hierarchical Learning Path. In both figures white boxes represent CCs that are reused in both macro Learning Pathways (shared CCs).

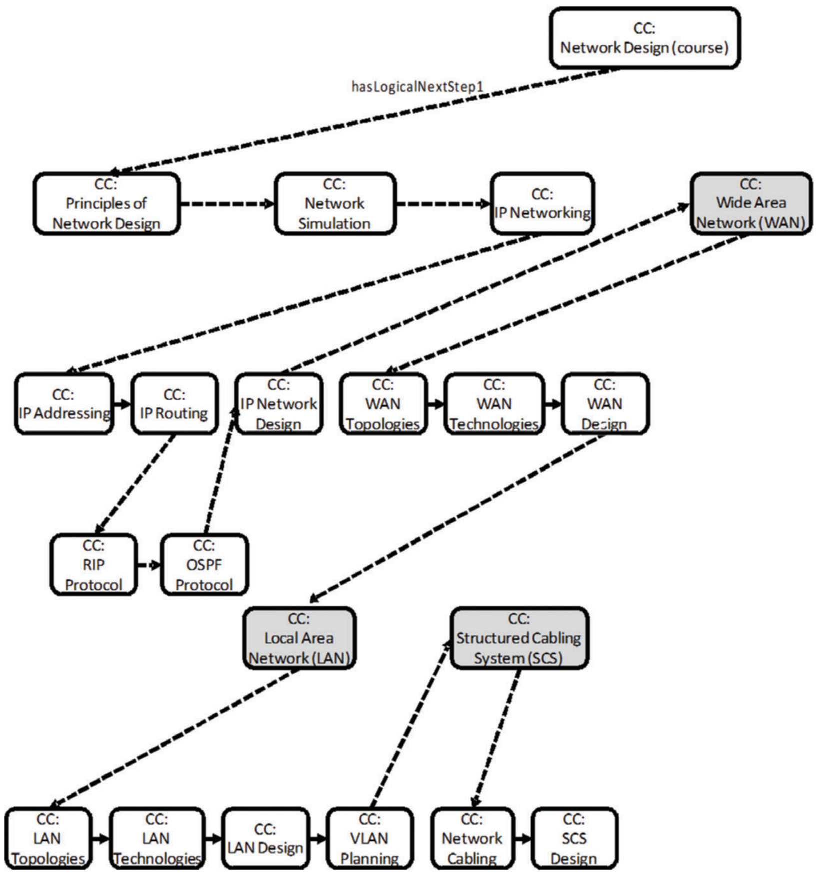


Figure 4.24 Classical Macro Learning Pathway for the KD "Network Design".

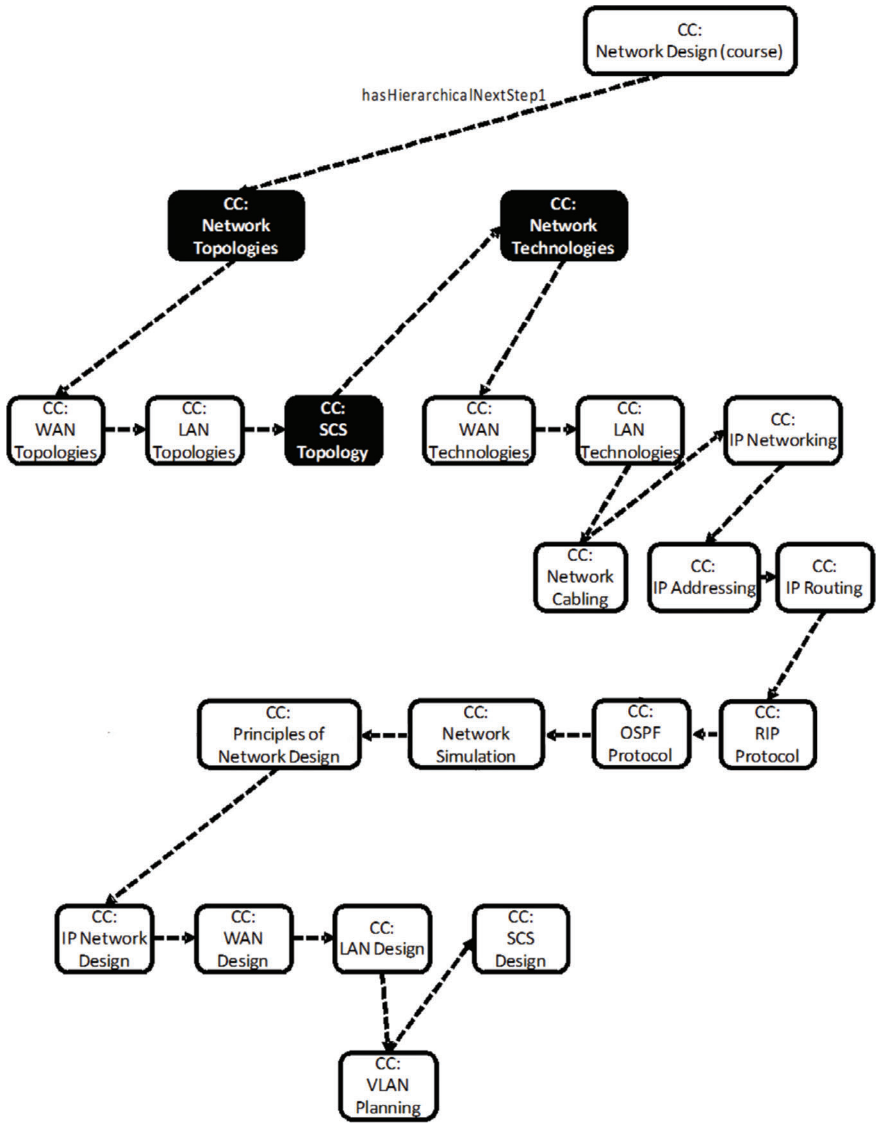


Figure 4.25 Alternative Hierarchical Macro Learning Pathway for the KD "Network Design".

Within each CC, Micro Learning Pathways are used to connect KOs. As a good network design requires examples and practice, the Good-Practice Multi-Stage Learning Pathway has been the most implemented Micro Learning

Pathway, but others have also been used (Simulated Multi-Stage Learning Pathway and Structured Inquiry-Based Learning Pathway). So, for example, the CC "Network Cabling" supports three different Micro Learning Pathways: Good-Practice Multi-Stage Learning Pathway with two different media types (text and video) and Structured Inquiry-Based Learning Pathway (dashed-line arrows and solid-line arrows in Figure 4.26, respectively).

By the dashed-line learning pathway, students receive an explanation of the different types of network cabling through a presentation or an explicative video; while by the solid-line pathway, several questions about the different types of network cabling are presented to students in order to be themselves who learn, study and present the results to their colleagues. Later, in both cases, the students have a tutorial step-to-step (with text or with video) and finally, they must do an exercise about network cabling.

Finally, Table 4.4 shows some examples of the behavior of the INTUITEL intelligent tutoring system depending on the behavior expected by the students

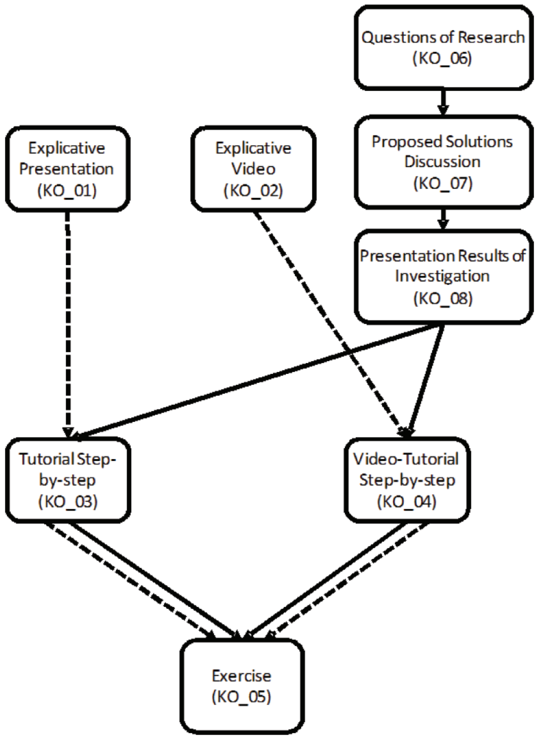

Questions of Research (KO_06)
Explicative Presentation (KO_01)
Explicative Video (KO_02)
Proposed Solutions Discussion (KO_07)
Presentation Results of Investigation (KO_08)
Tutorial Step-by-step (KO_03)
Video-Tutorial Step-by-step (KO_04)
Exercise (KO_05)

Figure 4.26 Micro Learning Pathways in CC "Network Cabling".

Table 4.4 Examples of the INTUITEL behavior for the CC "Network Cabling"

KO	Learner Behavior	INTUITEL Behavior
KO_01	Read the slides during 7 minutes	Recommend KO_03 as next KO
KO_01	Read the slides during 1 minute	Recommend read again KO_01 with a message indicating that it is necessary to read again this KO because 1 minute is an insufficient time.
KO_02	View the video during 5 minutes	Recommend KO_04 as next KO
KO_04	View the video during 7 minutes	Recommend KO_05 as next KO
KO_05	Do the exercise and answer incorrectly the questions proposed	Check if the student prefers text or video: In case the user chooses "Text" recommend KO_03 as next KO with a message indicating that it is necessary to read again the KO_03 in order to review the concepts. In case the user chooses "Video" recommend KO_04 as next KO with a message indicating that it is necessary to view again the KO_04 in order to review the concepts.
KO_05	Do the exercise and answer correctly the questions proposed	Check the Macro Learning Path followed by the student. If this is the "Classical Path" Recommend the first KO of CC "SCS Design" (see Figure 4.24). If it is the "Alternative Hierarchical path" recommend the first KO of CC "IP Networking" (see Figure 4.25).

for the CC "Network Cabling" shown in Figure 4.26. The table uses the short identifiers of the knowledge objects. For their according titles see Figure 4.26.

4.4.3 Special Relativity

Peter A. Henning

The concept of Learning Pathways (LP) and of navigation in the space of learning objects is most easily demonstrated by using a very simple course exhibiting only two Macro and two Micro Learning Pathways (MLP and mLP). For this simple example we have chosen a course on Special Relativity that we ran on an ILIAS platform.

Special Relativity has been developed in 1905 by Albert Einstein, first of all as a mathematical model to explain certain experimental findings in physics. However, as it turned out, the equations of Special Relativity had a very high predictive power they allowed predictions far beyond those simple experiments. Since then, Special Relativity has proven to be a consistent model of space and time under certain (=special) conditions, and is nowadays believed to be the correct mathematical description of nature's reality under these conditions. In the language of natural science, such a mathematical description is called a *theory*. Note, that in many other fields of knowledge the term "theory" is used in a much weaker sense, e.g. as a kind of personal hypothesis that might (or might not) be true. While we will adhere to the stronger meaning of *theory*, nevertheless the wording shall be avoided in the following.

The role of Special Relativity (henceforth abbreviated as SR) as model of space and time has fostered great interest even from outside the physics community as soon as it was published. This is due to the fact, that even the most basic models of philosophy require a model of space and time, as was already known to the ancient Greek philosophers (and, most probably also to thinkers in pre-historic times). Indeed, apart from abstract considerations like those presented by Immanuel Kant about language, thinking and being (and linked to the concept of absolute space and absolute time), even very concrete socio-political considerations require such a model of space and time. How could one have a concept of "neighbor" without a notion of "distance"? How could one have a concept of "causality" without a notion of temporal "before" and "after"?

Obviously, the domain of Special Relativity bears interest in Physics as well as in Non-Physics and these two views therefore will be used as our Macro Learning Pathways in an adaptive e-Learning course. Let us also mention, that these two views on our domain are semantically very different. In this case, our idea of Macro Learning Pathways exhibits a strong similarity to the Semantic Views Model, and to recent semantic clarifications of the role of theories in science [4]. It must be stressed that in most practical applications of adaptive learning systems such a clear distinction of Macro Learning Pathways will hardly be possible. On the other hand, the distinction according to the target group may serve as a blueprint for the application of the INTUITEL technology in an industrial setting, where the learning content is structured according to certain business roles. Within each of these two MLPs, we will present only five Concept Containers (CC).

In the Non-Physics MLP, an introductive CC will be followed by

- CC "nineteenth century", depicting the space-time model and experimental situation at the end of the nineteenth century.
- CC "twentieth century", depicting the findings in the period 1900–1905.
- CC "Annus Mirabilis", describing in detail the wondrous success achieved by Albert Einstein in is miracle year 1905 from a non-physicists view and a conclusive CC.

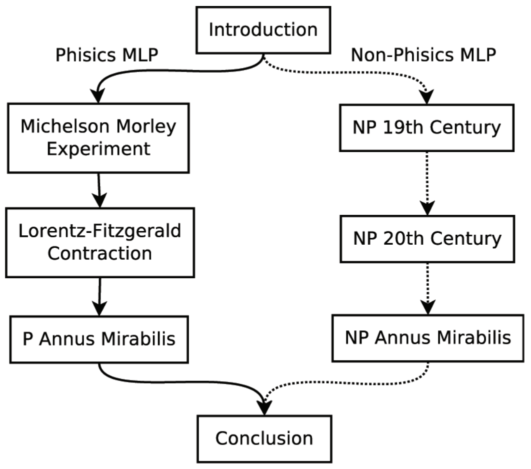


Figure 4.27 Macro Learning Pathways.

In the Physics MLP, the same introductive CC will be followed by

- CC "Michelson-Morley Experiment" hinting towards the non-existence of a ether.
- CC "Lorentz-Fitzgerald Contraction providing a solution to the dilemma that was then taken up by Einstein.
- CC "Annus Mirabilis", describing in detail the wondrous success achieved by Albert Einstein in is miracle year 1905 from a non-physicists view.

Finally, both the Physics MLP and the Non-Physics MLP share a conclusive CC. Figure 4.27 gives an overview of the two MLPs.

As the next step, we will have to specify the two micro learning pathways. The term "micro" was deliberately chosen for this ordering principle, because it is intended to be changed locally in a learners' traversal of the space of learning objects. Note, that this does not mean "instantaneously", as the progress of the current learning object might require to finish it, to finish even a short segment of a given mLP before changing to another. A choice of mLP therefore can be achieved by reasoning on the basis of the current Learner State Ontology including the pedagogical factors in all their variety:

- Learner properties, like age, gender, culture
- Learner behavior
- Learner success
- Dialogue components

- Soft factors like stress and motivation
- Hard factors like bandwidth and environmental information

For our simple example course we will restrict ourselves to a single hard factor determining the mLP: The available bandwidth from the server to the learners' digital interaction device. In particular, we will differentiate only a high-bandwidth and a low-bandwidth mLP. This implies that the media type presented to the learner is chosen according to the bandwidth. However, the INTUITEL reasoning process is not reduced to selection of a proper media, because a proper annotation of the learning objects may still indicate to the learner, that some LO are more adequate than others. This will be outlined below.

The simple bandwidth mLP are depicted in Figure 4.28 for one of our CC. The dashed frames indicate the first and the last element on a mLP. In this CC we have only three Learning Objects:

- A summarizing text, present on both mLP
- A concluding text, present on both mLP
- A virtual experiment, consisting of an interactive video clip and therefore only present on the high bandwidth mLP

Choosing the low bandwidth mLP for the concluding CC therefore only means: Leave out the virtual experiment. Now let us assume, that a learner enters the course with the selection of non-physics MLP and high-bandwidth mLP. This selection may either be done manually or may be determined from information gathered about the learner, like e.g. previous choices, a simple question asked at the beginning of the course and a bandwidth detection component.

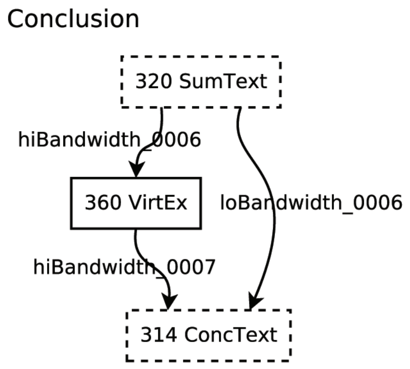


Figure 4.28 mLP for the Concept Container "Conclusion". Dashed frames indicate the first and the last element of a mLP.

On entering the course, the learner will be shown the KO Course Overview usually a short page of meta data, indicating how much time one would need etc. Simultaneously to a configurable sound message, a selection box (see Figures 4.29 and 4.30) will appear at the side of the screen (depending on configuration) that presents to the learner the following three KOs along his current mLP however, not in the ordering they have within the concept container. Instead, the highest recommendation (four stars) is given to the last KO of this CC the KO titled “History 19th century”.

The reason for this is the additional meta data for the course: It attributes a low learner knowledge level to the other two KOs and a high learner knowledge level to the “History” KO. Since the learner also comes with additional meta data, attributing him to be an experienced learner (or, at least one who has visited the course already three times), the lNTUITEL reasoning process determines that it would be more suitable for our particular learner to proceed directly to the most difficult KO in this CC. Note, that nevertheless full flexibility is maintained: The learner may also choose any other KO in the current CC and will then be recommended the remaining KO along his mLP.

Now, what happens if the learner follows the first recommendation, and proceeds to the “History” KO? Of course, he is shown the corresponding KO but at the same time receives already a recommendation for the following KO (see Figures 4.31 and 4.32). These are the KOs of the CC along his current MLP (Non-Physics), in this case only two KOs with equal recommendation level.

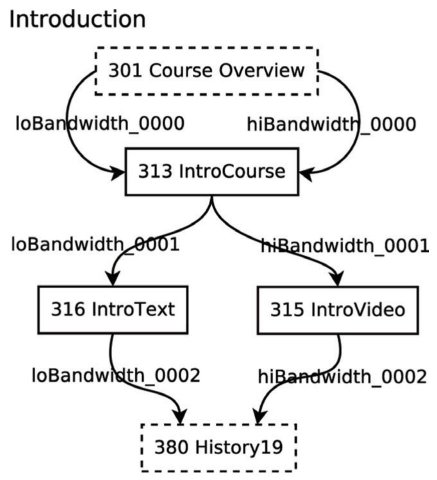
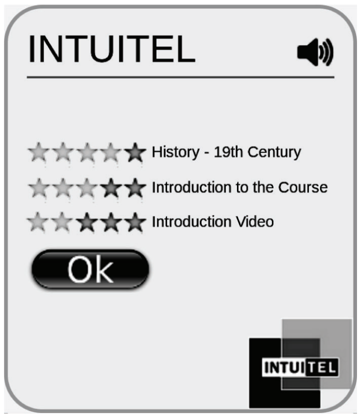


Figure 4.29 mLPS in the Concept Container “Introduction”. Dashed frames indicate the first and the last element of a mLP.


INTUITEL

History - 19th Century

Introduction to the Course

Introduction Video

Figure 4.30 The selection box shown to the learner when entering the Concept Container "Introduction".

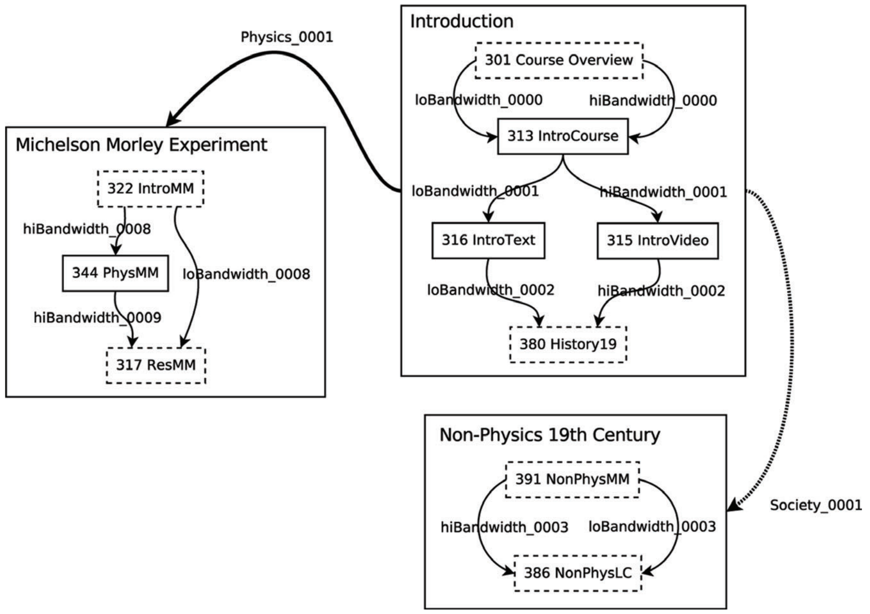
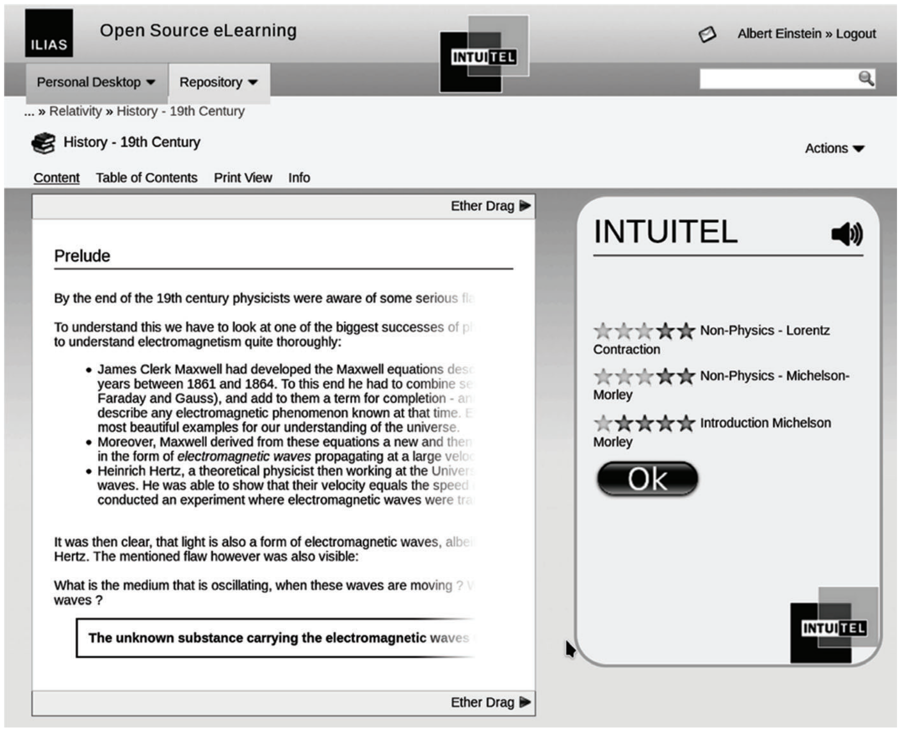


Figure 4.31 Branching of the MLP on exiting the first CC. Dashed frames indicate the first and the last element of a mLP.

However, it is easily possible to set up the INTUITEL system such that from the last KO in the current CC one is also shown the first KO from a CC on another MLP. Consequently, in such a setup the learner will be given the chance to change the Macro Learning Pathway.



Prelude

By the end of the 19th century physicists were aware of some serious fl

To understand this we have to look at one of the biggest successes of p
to understand electromagnetism quite thoroughly:

- James Clerk Maxwell had developed the Maxwell equations desc
  years between 1861 and 1864. To this end he had to combine se
  Faraday and Gauss), and add to them a term for completion - an
  describe any electromagnetic phenomenon known at that time. E
  most beautiful examples for our understanding of the universe.
- Moreover, Maxwell derived from these equations a new and then
  in the form of *electromagnetic waves* propagating at a large veloc
- Heinrich Hertz, a theoretical physicist then working at the Univer
  waves. He was able to show that their velocity equals the speed
  conducted an experiment where electromagnetic waves were tra

It was then clear, that light is also a form of electromagnetic waves, albe
Hertz. The mentioned flaw however was also visible:

What is the medium that is oscillating, when these waves are moving ? W
waves ?

The unknown substance carrying the electromagnetic waves

☆☆☆★★ Non-Physics - Lorentz Contraction

☆☆☆★★ Non-Physics - Michelson-Morley

☆★★★★ Introduction Michelson Morley

Figure 4.32 Selection box shown in parallel to the last KO of the first CC. We present here a screnshot from the ILIAS implementation of INTUITEL, with the “intuitel/integrated” ILIAS skin.

Currently, the INTUITEL system does not allow leaving the current MLP while somewhere inside a CC one always has to proceed to the next branching point. The modular and transparent concept of the INTUITEL software allows changing this behavior with a few lines of Java code but as a result of the INTUITEL project we have understood the pedagogical reasoning process to a great detail and would suggest a different approach: If a learner is motivated to change the MLP while deep inside some CC, the logical structure of the course should be reconsidered, and possibly some more common CCs should be used.

Figure 4.33 depicts the overall structure of the INTUITEL course on Special Relativity. We will not elaborate on the other KO and CC, since in this simple design each MLP has to be followed after it has been chosen the two MLP come together again in the concluding CC.

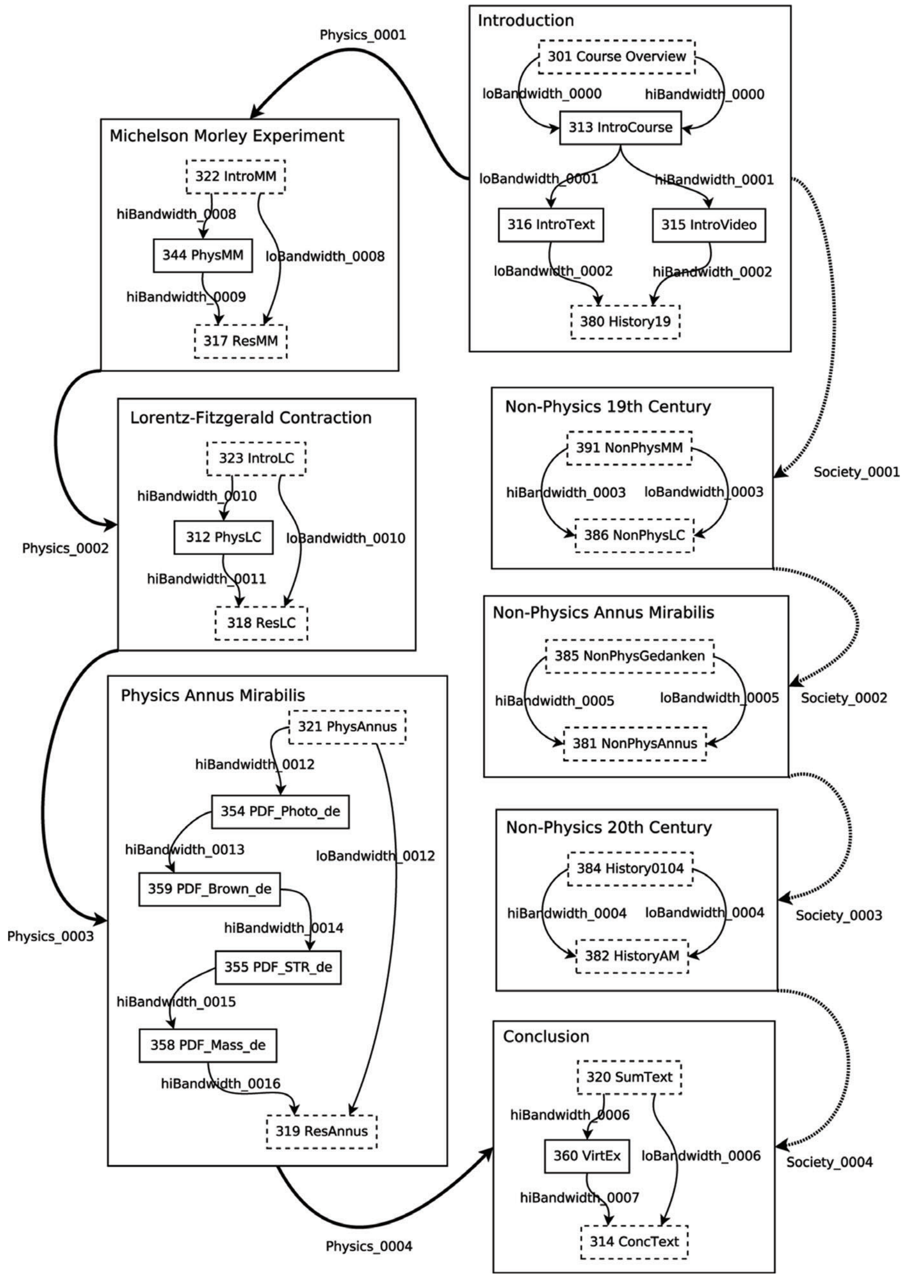


Figure 4.33 Overall structure of the INTUITEL course on Special Relativity. Dashed frames indicate the first and the last element of a mLP.

With only two MLP, two mLP and 26 KOs it is a rather simple example for the potential of INTUITEL. And yet, it makes clear that the main task now lies at the hands of the cognitive engineer: Designing the proper course structure and entering the necessary meta data is the key to successful adaptive learning.

4.5 Evaluation and Testing

Luis de-la-Fuente-Valentin and Daniel Burgos

Frameworks for Technology Enhanced Learning are really difficult to evaluate because of the large amount of factors that may affect the learning scenario. Let's say that a bad design of the user interface hinders a student's interaction with the system. In that case, the student's perception won't be as positive as expected, regardless of how useful the recommendations are. Therefore, the validation has to be very careful with the aspect being tested.

From a technical perspective, the framework development was based on a distributed unit-test approach, in which unit-tests were centralized in a single server and deployed into the different INTUITEL modules as REST requests. This centralized process, daily triggered, guaranteed that all the distributed software modules were compliant with the Communication Layer that mediated all the request-response messages in the framework.

From a pedagogical perspective, the validation was carefully designed to support a set of predefined test cases. The support for these test cases was validated by introducing the concept of "artificial test learners", generated by collecting typical data from institutions of higher education and from the commercial educational interest companies. These test cases are available as an internal working document and are then fed into the INTUITEL enabled prototypes. This section depicts and details the above described validation techniques, focusing on them as a method to guarantee the required reliability prior to the deployment of the framework in a real learning scenario.

Tests on the Communication Layer

The INTUITEL software architecture depicted in 4.4 uses a centralized strategy for messaging in which the Communication Layer receives and delivers every message exchanged in the Recommendation Process. This strategy allows for a distributed software architecture where the different modules may run in different servers as long as they know the location of the Communication Layer. Therefore, the Communication Layer is a key module in the INTUITEL system and must be carefully tested.

A unit-test based testing strategy has been followed. The testing suite was developed during the first months of the project and ensured the homogeneity of the REST messages structure. The testing was executed as follows:

1. For each component being tested, correct and incorrect (designed on purpose) REST messages were sent.
2. The testing suite validated if the obtained response matches the expectations, according to the Communication Layer specifications and the USE/TUG/LORE message definition (see Section 3.5).
3. A report was built and published including the result of every test case.
4. Whenever a mistake was found on a response message, the report was emailed to the developer of the corresponding partner.
5. This process was daily repeated.

It can be noticed that this strategy treated the tested servers as passive subjects, where they only have to offer an endpoint to provide successful answers to the REST messages they receive. However, each server may also send messages in a proactive way. In such case, the Communication Layer was in charge of verifying the correctness of the different REST messages.

As explained in Section 3.5 and provided in Table 3.3, INTUITEL defines eight endpoints for REST messages: lmsprofile, learners, login, mapping, TUG, LORE, USE/performance and USE/environment. For each of those endpoints, the testing suite composed and sent a total of seven different messages:

1. Correct message with valid XML, correct values in HTTP headers and namespaces properly set.
2. A message with valid XML, correct values in HTTP headers but no namespaces defined.
3. A message with valid XML, but incorrect values in HTTP headers.
4. A message with invalid XML (bad element name)
5. A message with invalid XML (bad attribute name)
6. A message with invalid XML (empty attribute value)
7. A non-XML message

The correctness of the response was decided by verifying the HTTP Status Code of the response. A 200 OK status code was expected for messages 1 and 2, while 4XX was expected for the rest of the test cases. The test suite used the following framework:

- REST messages were generated from static XML files and sent via PHP CURL libraries.

- The PHP-Unit library is used to organize the test messages as unit tests and generate XML structured output.
- The PHING Framework is used to generate HTML human readable reports from the XML structured output.
- Doxygen is used to generate source code documentation, so that every report includes the test results and the description of the test cases.
- The test suite scheduled with cron to be executed once a day, and reports were sent to the developers with the UNIX send mail utility.

The testing suite does not verify if the system generates recommendations. It just ensures that all the different modules are alive and compatible with the Communication Layer. The correctness of the provided recommendations are verified by the so called artificial test learners.

Artificial Test Learners: Technical Tests on the INTUITEL Recommendations

The artificial test learners were implemented as software pieces that executed actions similar to those expected from actual learners. Such actions fed the LPM and the INTUITEL Engine so the test cases checked if the produced recommendation was correct according to the INTUITEL expected reasoning. The overall behavior of the testing is depicted in Figure 4.34. Other tasks in the project were devoted to test the LPM and the INTUITEL Engine:

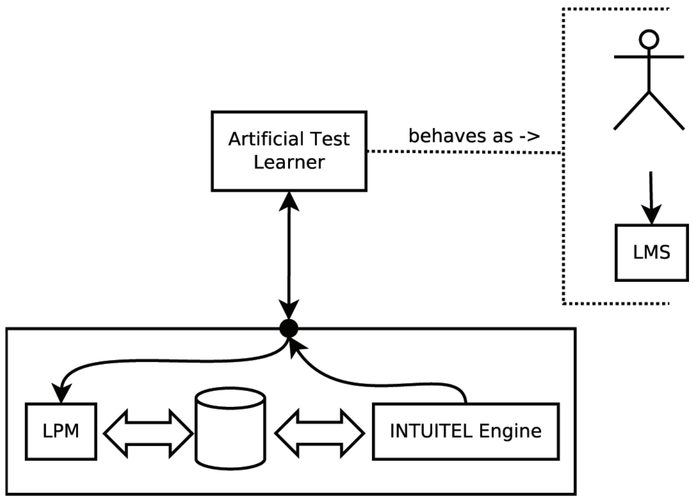


Figure 4.34 Behavior of the testing procedure.

- LPM testing was fed with student actions and examined the selected Learning Pathway. The LPM calculates the learner's current position, and determines the most suitable Learning Pathway based on the history of the learner. The LPM does not produce any recommendation, so LPM testing did not test recommendations.
- INTUITEL Engine testing was fed with LPM output, and the unit testing was devoted to check if the recommendation is the expected one.

Therefore, the two described subsystems do not test the correct behavior of the overall system. Artificial test learners executed the testing from the perspective of the learner: feeds the system with the user actions, and checks if the recommendation (i.e. the communication with the learner) is the expected one.

Changes in the Initial Approach

The initial idea of the artificial test cases (and therefore described in Deliverable 12.1 [24]) was to apply a LMS based approach. That is, the artificial test learner was conceived as an automated browsing (web scrapping) through the course contents, programmed with web scraping techniques such as mechanize, casper or any other similar software library. However, further discussions within the topic revealed the following problems for such approach:

- Testing would be LMS specific. That is, test cases should be re-written for each of the INTUITEL enhanced LMS.
- Timing cannot be efficiently reproduced. That is, if the student takes 2 hours before choosing a KO, then the automated script would take the same 2 hours.
- Problems in the LMS would distract from the actual subject being tested: LPM+Engine.

Therefore, the researchers decided to change the approach and design the artificial test learners as described in this document.

Description of the Test Cases Template

A template guided the development of the test cases used to test the system. This template consists in two main sections: first the description of the initial state, including the values of all Didactic Factors and any other information stored in the database that could be useful to determine the learner state.

Taking this description of the initial state as the starting point, the second part of the template includes the step-by-step students' actions. That is, the messages that the LMS would send to the LPM as a response of the student's actions within the course. More specifically, the template to specify test scenarios is defined by:

- *Initial State Description*: Preliminary data describing the scenario and learner mostly based on the LPM ontology (Didactic Factors and values in Deliverable 3.2 [25]). Didactic Factors can be taken in and out according to the scenario and test requirements. Another important element of the initial description is the database snapshot that describes the student's previous activity. In other words, a list of which learner has accessed which KOs in which order.
- *Student actions*: A sequence of learner's actions. That is, a temporarily ordered sequence of LSO and "reflex" input messages. The test case also included the temporarily ordered sequence of LORE and TUG output messages, so the artificial test learners could check if the received message is the expected one.
- *SLOM metadata*: CM and CCM are needed for reasoning, so SLOM meta data should be included as part of the test case description.

The template for the test cases included the technical information required for the development of such a technical development as the artificial test learners. However, some textual description of the test case is also needed for the proper understanding of the scenario and, if possible, for the reuse of the test cases in other testing tasks.

Data Required to Represent a Test Case

The artificial test learners fed the LPM with the events produced by the learner at the LMS, and verified the result from the INTUITEL Engine. Therefore, the first data required is the piece of information sent from the LMS to the LPM at a specific situation. Second, the LPM will produce an output that will inform the Engine of the most suitable LP and other relevant information, which is the LSO. Third, the Engine will produce an output to be sent back to the LMS, shaped as USE/TUG/LORE messages. Finally, the SLOM data that defines the content itself is required for the testing in order to contextualize the expected output data. In summary, the required pieces of information for the test cases are:

- The messages sent by the LMS: Learner Update and TUG/LORE responses.
- SLOM related to the content needed for reasoning.
- Corresponding snapshots of user database or model needed for reasoning.
- LSO and "reflex" messages produced by the LPM.
- Intuitel Engine responses, shaped as USE/TUG/LORE messages.