



# GRAPPLE

D5.2a Version: 1.0

## Conversion models between GRAPPLE and LMSs

<b>Document Type</b>	Deliverable
<b>Editor(s):</b>	Lucia Oneto
<b>Author(s):</b>	Lucia Oneto, Jose Luis Santos Odriozola, Luca Mazzola, Kai Hoever, Avi Naim, Ekaterina Pechenezhskaya, Dimitri Rambout, Michele Dicerto
<b>Work Package:</b>	WP5
<b>Due Date:</b>	31-01-2009
<b>Version:</b>	1.0
<b>Version Date:</b>	19-03-2009
<b>Total number of pages:</b>	21

**Abstract:** This document describes the conversion models needed for the integration of the GRAPPLE framework with the most representative LMSs.

**Keyword list:** Learning Management System, data model, conversion model, user model

## Summary

This document presents the conversion models able to guarantee the adaptive behaviour of an application in the source system is translated (both ways) into semantically equivalent adaptive behaviour in the target system.

## Authors

Person	Email	Partner code
Lucia Oneto	l.oneto@giuntilabs.com	GILABS
Luca Mazzola	luca.mazzola@lu.unisi.ch	USI
Jose Luis Santos Odriozola	jose.santos@atosresearch.eu	ATOS
Kai Hoever	Kai.Hoever@im-c.de	IMC
Dimitri Rambout	dimitri.rambout@uclouvain.be	UCL
Avi Naim	an10007@cam.ac.uk	UCAM
Ekaterina Pechenezhskaya	e.pechenezhskaya@tue.nl	TUE
Michele Dicerto	m.dicerto@giuntilabs.com	GILABS



# Table of Contents

<b>SUMMARY .....</b>	<b>2</b>
<b>AUTHORS .....</b>	<b>2</b>
<b>TABLE OF CONTENTS .....</b>	<b>3</b>
<b>TABLES AND FIGURES.....</b>	<b>4</b>
<b>LIST OF ACRONYMS AND ABBREVIATIONS .....</b>	<b>4</b>
<b>1 TASK AND DELIVERABLE DESCRIPTION.....</b>	<b>5</b>
<b>2 INTRODUCTION.....</b>	<b>5</b>
<b>3 GRAPPLE SYSTEM OVERVIEW.....</b>	<b>5</b>
<b>4 STANDARDS SPECIFICATIONS.....</b>	<b>6</b>
<b>4.1 IMS Enterprise.....</b>	<b>6</b>
<b>4.2 IMS LIP / IEEE Papi Learner.....</b>	<b>7</b>
<b>4.3 IMS AccessForAll.....</b>	<b>9</b>
<b>5 DATA MODELS INVOLVED IN THE GRAPPLE PROJECT .....</b>	<b>9</b>
<b>5.1 GALE data model.....</b>	<b>9</b>
<b>5.2 Moodle data model .....</b>	<b>10</b>
<b>5.3 Sakai data model.....</b>	<b>11</b>
<b>5.4 Claroline data model .....</b>	<b>14</b>
<b>5.5 learn eXact data model.....</b>	<b>17</b>
<b>5.6 IMC Clix data model.....</b>	<b>18</b>
<b>6 CONVERSION MODELS BETWEEN GRAPPLE AND LMSS .....</b>	<b>20</b>
<b>6.1 Conversion model from Moodle to GRAPPLE ALE.....</b>	<b>20</b>
<b>6.2 Conversion model from Sakai to GRAPPLE ALE.....</b>	<b>20</b>
<b>6.3 Conversion model from Claroline to GRAPPLE ALE.....</b>	<b>20</b>
<b>6.4 Conversion model from CLIX to GRAPPLE ALE.....</b>	<b>20</b>
<b>REFERENCES .....</b>	<b>21</b>

## Tables and Figures

### List of Figures

Figure 1: CLIX data model extraction .....	18
Figure 2: Skills of a learner .....	19

### List of Acronyms and Abbreviations

ALE	Adaptive Learning Environment
GRAPPLE	Generic Responsive Adaptive Personalized Learning Environment
IMS	Instructional Management System
IMS LIP	IMS Learner Information Package
LMS	Learning Management System

## 1 Task and Deliverable Description

### T 5.2 Conversion models and components (GILABS, ATOS, OUNL, USI, UCL, UCAM)

Design and construction of conversion models and components looking for the best integration between the GRAPPLE framework and LMSs, and providing adaptation features described in WP1.

### D5.2a Conversion models between GRAPPLE and LMSs (GILABS, M12)

LMSs like Moodle, Sakai and Claroline will be mapped both ways to GRAPPLE. They will look for a common understanding between pairs in order to provide a bi-directional information model.

### D5.2b Conversion components between GRAPPLE and LMSs (GILABS, M18)

Following up the work in D5.2a, a number of components will be developed to allow for this interoperability. They will concrete the theoretical research starting in D5.1 and focused in D5.2a in a practical outcome.

## 2 Introduction

The GRAPPLE WP5 deals with packaging and learning standards to address the needs for expressing adaptation in learning materials and processes by means of existing specification frameworks. One of the WP5 tasks is to look at a number of widely accepted open source LMSs (like Moodle, Sakai, and Claroline) and commercial LMSs (learn eXact, IMC Clix) to work on an integration with GRAPPLE focused on adaptive features.

This integration will map both parts of the combination (i.e., GRAPPLE-Moodle, GRAPPLE-Sakai) looking for the best understanding between both parts. Furthermore, part of this work will also be focused on the creation of conversion models and components on the basis of specification-specification and specification-LMSs formats, in such a way that the adaptive behaviour of an application in the source system is translated both ways into semantically equivalent adaptive behaviour in the target system.

This document starts with a short introduction to the GRAPPLE system, including a short description of the generic system overview and the type of information that has to be exchanged between the GRAPPLE LMSs and the GRAPPLE ALE (Chapter 3).

Chapter 4 describes the standards actually used by the LMSs and by the GRAPPLE ALE: it provides a clear overview of the packaging and learning standards and specifications involved by the GRAPPLE project.

Chapter 5 includes detailed descriptions of the conversion models identified in the project. Five single conversion models have been identified:

1. Conversion model from Moodle to GRAPPLE ALE
2. Conversion model from Sakai to GRAPPLE ALE
3. Conversion model from Claroline to GRAPPLE ALE
4. Conversion model from learn eXact to GRAPPLE ALE
5. Conversion model from IMC Clix to GRAPPLE ALE.

For any LMS, we give a clear description of the bi-directional information model that satisfies a correct communication between the existing LMSs and the GRAPPLE ALE. The conversion models are based on the standards that are supported within the before mentioned LMSs. The document finishes with the references to related work.

## 3 GRAPPLE System Overview

The main objective of GRAPPLE is to support life-long learning by means of a personalized and adaptive TEL environment (henceforth abbreviated to ALE, for adaptive learning environment) that integrates with major learning management systems (or LMSs) using a service-oriented (web) framework approach. The key to creating a responsive learning environment that motivates, engages and inspires learners, and through this leads to better learning results, is personalization.

Organizations that use learning environments like universities and corporations realize more and more that personalization is a key to creating acceptance of the technology for TEL by the end users. Personalization enables the environment to present the most appropriate, interesting, challenging learning activities, avoids learning resources that only present knowledge the learner already has, and also avoids learning material that is beyond the reach of the learner at the time (but it may present that material later, after the learner has acquired the necessary prerequisite knowledge). In GRAPPLE this personalization is achieved through adaptive learning technology.

It interfaces and is integrated with existing learning management systems. To do this, GRAPPLE is designed to deliver a service-oriented framework that can “plug and play” the adaptation services according to the potential the LMS can offer and make use of. The service oriented integration or interface includes single sign-on, transparent (two-way) transfer of user model data and interaction integration, using adaptive course material within the main learning management system's user interface.

LMSs typically have elaborate services for testing the learner's knowledge. The outcome of such tests should be stored in the user model and used by the adaptive learning component to guide the learner to material to revisit or to new topics to study. A search for other learners with certain knowledge or skills should be possible in the learning management's user base, and communication should be established. It is clear that GRAPPLE needs to produce a “framework of services” or “integration plug-ins”, not a monolithic solution.

This deliverable D5.2 has got the objective to analyse what the LMSs can make available to the GRAPPLE system and that can be used by the GRAPPLE framework.

## 4 Standards Specifications

Nowadays, the correct user modelling for learning adaptive systems is a complex goal which needs to be achieved. The adaptation process shouldn't only consider the learner, as additional actors exist in the system. We need to consider that within the learning process, information on the interactions between the different roles is necessary in order to apply an effective adaptation.

The problem appears when we have to define which kind of information should be included in the user model, and how we should use it. Usually, the LMS (Learning Management Systems) have a domain model that is pragmatic and based on technical data, from which it is hard to produce high conceptual characteristics to define the learner. The user model specifications try to solve this problem existing in the different scenarios produced in the lifelong learning.

In this section, we consider four specifications that correspond to four scenarios not incompatible between them. IMS Enterprise is focused on modelling professional information of the user, IMS LIP and IEEE Papi, are focused on modelling user information in the learning environments, and finally, IMS AccessForAll is a specification focused on providing information about the learner, where the main goal, as its name suggests, is providing access for all. The reasons to consider these specifications are mainly their extended use and the benefits that they provide when used jointly, in the case of IMS specification, and the possibility to map IMS LIP with IEEE Papi [1]. Another advantage of using IMS LIP is the possibility of using existing editors [2].

### 4.1 IMS Enterprise

IMS Enterprise is focused on solving problems regarding the interoperability between the enterprise systems and the LMS. It doesn't provide a system for avoiding issues regarding data integrity, communication and security.

It tries to solve the follow four scenarios: Personal Data Profile Maintenance, Group management, Enrolment management and Final Result Processing.

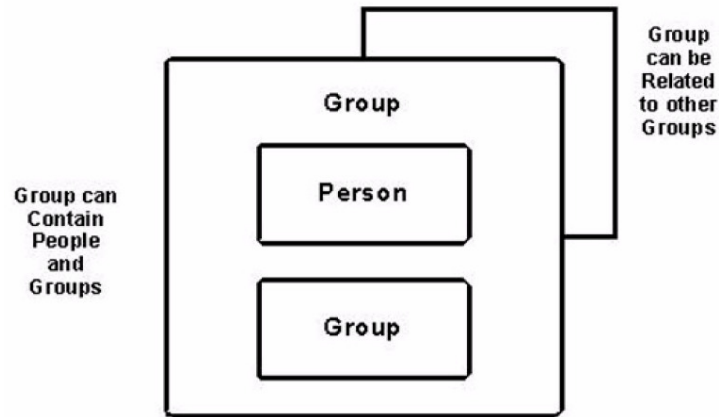


Figure 1: The principal enterprise data objects.

IMS Enterprise is based on three core components:

- A Person which describes the individual inside the system, it haven't the goal to contain all the personal information of the individual.
- A Group which describes the collection of objects related to learning activities or individuals.
- Group membership which defines the member of a Group. The members can be an individual or another Group.

IMS Enterprise is a pragmatic specification. It was designed for solving focused problems between LMS and enterprise systems, but without considering the pedagogical needs of the learner. From a classical point of view of Moodle, Sakai or Claroline, it could be a good specification, but not for adaptive learning environments because they need more elaborated characteristics of the user.

## 4.2 IMS LIP / IEEE Papi Learner

The Learning Information Package is a specification for a standard way of recording information about learners. It was designed in order to allow records about learners and their progress to be transferred between different software applications and institutions. For example, it allows retrieving a record of all the learner's achievements. This means that obtaining information on the students' progress could possibly substitute actual certificates, although for the moment LIP considers including certificates as a characteristic of the learner.

All the information about learners is stored in a XML file; this file uses tags explaining the meaning of each detail of information in the records. This specification addresses the interoperability of internet-based Learner Information systems with other systems that support the internet learning environment.

The specification defines a series of packages that can be used to import and export data from an IMS compliant Learner server which can exchange data with Learning Delivery systems or other Learner Information servers.

The Learning Information server allows the owner of the information to define which part of this information is shared with other systems. The main structures of LIP are based on accessibilities, activities, affiliations, competences, goals, identifications, interests, qualifications, certificates and licences, relations, security keys and transcripts. It's a very flexible structure because, for instance, we can use activity structure modelling very different environments, from informal education to military service. It includes any learning-related activity. Or, for example, the goal structure lets us specify learning goals, career goals and other objectives and aspirations. IMS LIP is also a flexible specification in the way that it considers activities: while, on the one hand, it can consider high conceptual learning models, on the other hand it can also consider simple activities, such as a quiz.

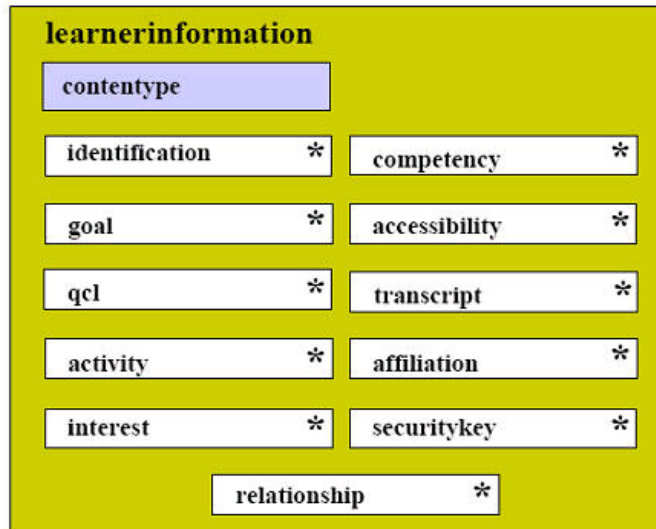


Figure 2: Global Structure of IMS LIP

LIP can be mapped to Learning Design properties, added in Level B [3] and is possible to embed other specifications as IMS Enterprise or IMS AccessForAll.

In the scope of the project, we consider these possibilities of specification sufficient in order to achieve the project objectives, and it is our recommendation. The unique problem that we can identify using IMS LIP is that the interaction logs cannot all be stored in the user model. In our opinion, storing the user logs is a possibility, but it is not the correct way because it is not manageable over time. The interaction logs are part of the domain model and it is not recommended to merge both, as their interpretation is the responsibility of the system that had generated them.

One of the possibilities that IMS LIP offers us is the mapping between itself and IEEE Papi, a converter that has been developed by RWTH Aachen [5].

PAPI distinguishes between personal, relations, security, preference, performance, and portfolio information. The personal category contains information about names, contacts and addresses of a learner. Relations serve as a category for relationships of a specific learner to other persons (e.g. classmate, teachers, teacherof, instructoris, instructorof, belongsto, belongswith). Security aims to provide slots for credentials and access rights. Preference indicates the types of devices and objects which the learner is able to recognize. Performance is for storing information about measured performance of a learner through learning material (i.e. what does a learner knows). Portfolio is for accessing previous experience of a user. Each category can be extended. [6]

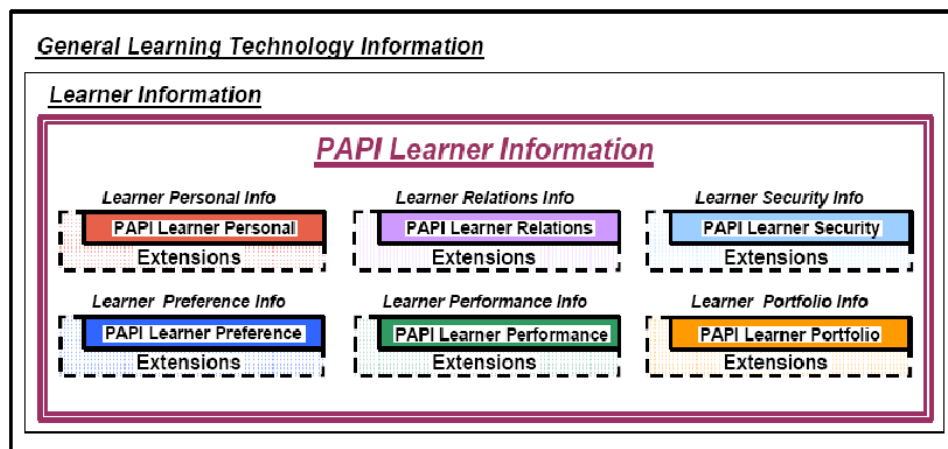


Figure 3: Global Structure of IEEE PAPI

### 4.3 IMS AccessForAll

IMS AccessForAll contains information about the user and how the interaction should be oriented. It does not only provide information about user disabilities, but is also focused on providing information about how the system should act in different environments, for instance, in a very noisy environment.

One of the advantages of this specification is the possibility of using it with other specifications like IEEE LOM, ISO 15836 (Dublin Core Metadata Element Set) or IMS LIP, in this last case, the name specification changes to IMS ACCLIP.

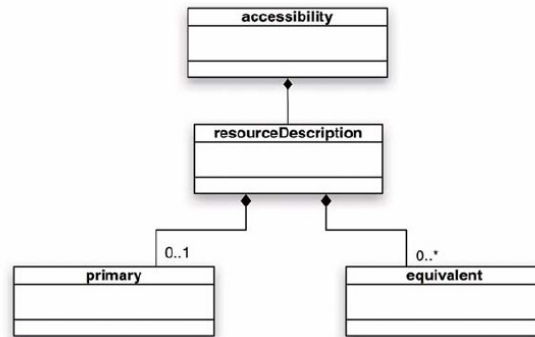


Figure 4: Overall Accessibility Data Model

The AccessForAll Metadata specification describes the resources in two possible categories: the primary resource, it is the default resource, and the equivalent resource that offers an alternative.

The primary resource describes the following:

- Access modality. It describes whether the user requires vision, hearing, touch or text literacy.
- Adaptability. It describes whether the user support personalization and if it amenable to be adapted to display.
- Equivalent. It describes if an alternative exist.

The main advantage that is offered by AccessForAll metadata is for disabled individuals, who are not the target of this project. However GRAPPLE is sensible to the issue of accessibility for disabled individuals and the architecture will take into account the possibility of adding in the future this specification, because an adaptive learning environment will have to consider the adaption not only based in pedagogical criteria but also based on accessibility criteria.

## 5 Data models involved in the GRAPPLE project

### 5.1 GALE data model

The minimal GRAPPLE data model as suggested in D7.2 - Data models and related documentation – first version has been achieved by analyzing the data that the existing LMSs can pass to the GRAPPLE system:

#### ***Attended Courses***

The learner information that consists of the education/training, work and service (military, community, voluntary, etc.) record and products (excluding formal awards). This information may include the descriptions of the courses undertaken and the records of the corresponding evaluation.

- *UserX hasAttended CourseXY*
- *UserX isEnrolled CourseXY*

#### ***Roles***

Roles describe a group of interest to the Learning Management environment. There are many types of groups that may be shared between systems.

- *Teacher*
- *Tutor*
- *Learner*
- *System Administrator*

### ***Personal User data***

The learner information that contains all of the data for a specific individual or organisation. This includes data such as: names, addresses, contact information, demographics and agent.

- *Username*
- *Password*
- *Last Name*
- *First Name*
- *Email*
- *Language*
- *Gender*
- *Date of birth*
- *Street*
- *Town*
- *Postal code*
- *Region*
- *Country*

## **5.2 Moodle data model**

Moodle data model is quite simple, because it is interested only in storing some demographic data and the logs of events. Due to this limitation, we are able to expose a subset of personal data, some information about enrolled course and the user role inside a specific context.

The actual personal data available inside Moodle are:

- Username
- Password (not the plain text, but an hashed format (the MD5 output))
- Last Name
- First Name
- Email
- Language
- Town
- Country.

It is possible to define as much other fields as wanted, but in this case they are not mandatory by default. In this way Gender, Date of birth, Street, Postal code, Region could be collected. Probably, the exposition of

the latter category of data needs some attention due to some problem with privacy issues (for example with gender or year of birth).

About attended courses, Moodle does not have the concept of attended courses, instead it can expose the list of students that are enrolled in a particular course. This is based on the role that the actual user has in the current context. Obviously a step for defining which roles implied the enrolment of a user is needed to export this information. It is clear, from the exposed situation, that a student in a course X could be tutor in a course Y and teacher in another one.

For the competences or skills tracking, Moodle does not have these concepts and don't care of them for its internal functions. Instead, Moodle tracks the log of resource access and the mark with the time used to finish every quiz.

Moodle right now, does not care of special needs (like adaptation for blind or deaf people).

### 5.3 Sakai data model

Personal data in Sakai is stored in the database. Sakai creates all necessary tables in one Data Base during the installation. There are currently 3 databases supported in Sakai (Oracle, MySQL, HSQLDB). By default, all distributions of Sakai are configured to use an in-memory version of HSQL db. But the vast majority of Sakai installations choose to use either MySQL or Oracle in production. This can be configured in sakai.properties file.

Each user of Sakai is identified with a unique Sakai provided (or externally provided) identifier (USERID). This identifier is used by Sakai applications when making associations between their modelled information and a user. Each user of Sakai also has an enterprise id (EID) that is used to connect to external user-oriented information systems, and is also used by the users to authenticate with Sakai (as a login). Personal information, such as names, addresses, pictures, etc, is modelled in a *User Profile*. Contact information, such as an email address or instant messenger profile, is modelled and used by applications to send notifications to end-users. Preferences for items of use to various Sakai applications are modelled and used by the applications to customize their behaviour and appearance for each end-user.

Sakai has a number of end-user and administrative tools to maintain and access the end-user model:

- New User – End-user tool to self-register as a new user in the system.
- Account – End-user tool to modify their enterprise id, password and contact information
- Profile – End-user tool to modify their profile
- Preferences – End-user tool to update their preferences.
- Admin User Editor – administrative control over all User entities.

The User model's API also provides an authentication service, to determine if a given set of user identification "evidence" properly authenticates a known Sakai user, and if so, which one.

The User entity also supports the Sakai Entity Bus, making available all the information about the user as a set of properties named with a known vocabulary. The User entity's properties are arbitrarily extensible, so applications can store additional information about end users.

The primary APIs of this model include:

- UserDirectoryService
- UserDirectoryProvider
- User
- SakaiPerson
- Authentication
- Evidence
- PreferencesService

- Preferences

Below, the structure of the Sakai database's tables where personal data is stored is presented.

The following personal information is stored in SAKAI\_USER table:

- USERID (unique Sakai provided identifier)
- First Name
- Last Name
- Password (not the plain text, but an hashed format (the MD5 output))
- Email
- Type (Role of the user)
  - Instructor
  - Teacher Assistant
  - Student
  - System Administrator

Student's login (enterprise id - EID) is stored in SAKAI\_USER\_ID\_MAP table that relates EID to other tables via USERID.

Student's Profile information is stored in SAKAI\_PERSON\_T table, which is also related to other table via USERID.

The following fields are included into SAKAI\_PERSON\_T table by default:

- USERID (unique Sakai provided identifier)
- First Name
- Last Name
- Affiliation
- Position
- Email
- Picture
- Address information (both office and home) (Postal Code, Postal address, PO. Box, State Province Name, Street Address, Room Number)
- Phone Number (office, home, mobile)
- Fax Number
- Department Number
- Employee Number
- Employee Type
- Preferred language
- Secretary

This student does not need to provide this information during registration. This information is can be provided by the user via Profile Tool using web interface. The administrator of Sakai can configure which fields should be included into the Profile. Usually, this information is automatically derived from institutional stores.

The early versions of Sakai (up to 2.6) had problems in combining Account and Profile information. For example, if the student filled the field Surname while creating account, it was not automatically transferred to the User Profile. Now, it seems that these problems have been solved.

Recently (February, 2009), a new profile tool (Profile2) for Sakai has been developed. It is a complete re-design and re-development of the Profile tool in Sakai. It gives to the student more of a Facebook style interface for editing his/her profile. In addition it provides a comprehensive set of Privacy controls that allow choosing who can see what part(s) of Profile.

Although Sakai stores its information in the database, accessing this information directly is not advisable for a number of following reasons:

- a) Some areas of the database are managed by Hibernate and modifying those areas directly will result in strange behaviour from hibernate as its' in memory cache becomes out of sync with the underlying database.
- b) Other areas of the database are managed with explicit caching which has invalidation schemes. Again if the database is modified without performing cache invalidation, unpredictable side effects may be observed.
- c) Most institutions derive user information and some profile information from institutional stores. This is achieved with Providers, UserDirectoryProviders, GroupProviders and others. Where information is managed by a provider, it will not be in the database, and so any queries to the database will bypass the external data sources and generate incomplete or invalid information sets.

The recommended mechanism of accessing information in Sakai is to use the Service API's that have been created for the purpose. These can be exposed as REST based services or SOAP Web services if required. In Sakai 2.6 the UserDirectoryService and the ProfileService are probably relevant, although other information from CourseManagement or AuthZGroupService may also be of interest.



Figure 5: Sakai APIs.

## 5.4 Claroline data model

The actual personal data available in Claroline are:

- Username
- Password
- Last name
- First name
- Email
- Language

We do not store information regarding gender, date of birth, address... In addition to those data an administrator can also define some user properties specific to his platform.

In Claroline a Course is a kind of workspace providing tools and resources for learning activities. We can provide the information UserX isEnrolled CourseXY concerning the course spaces.

A user cannot attend to a course space in Claroline since this information does not have any meaning in the platform. Instead each Claroline course space contains a Learning Path tool providing user with a set a chain learning modules (quiz, document...) which is the closest to what is called a course in the GRAPPLE terminology. We can provide the information UserX hasAttended LearningPathXY in CourseXYZ for a Learning Path.

Claroline can provide all the Grapple roles requested by the GRAPPLE Data Model. In addition the platform as a more specific role definition depending on the context:

- At the platform level a user can be administrator, course creator or user
- At the course level a user can be course manager, tutor or learner
- At the tool level a user can be tool manager or tool user
- At the group level a user can be manager, tutor or learner

The platform administrator can also define some additional roles and profiles with specific access rights.

At this time Claroline does not manage concepts like competences, skills or knowledge level. But this could be made available to some extent in the future.

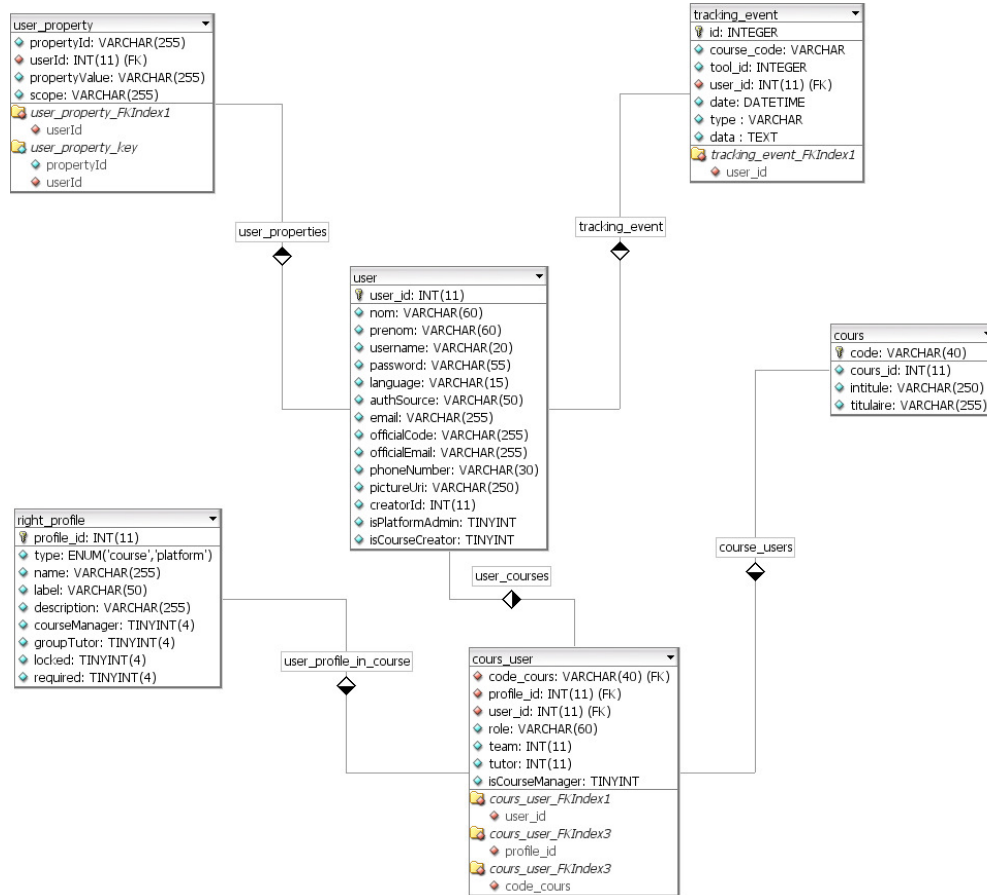


Figure 6 summarizes the user model in Claroline. The platform store some data concerning the access to courses, tools and other resources at the platform level and at the course level (user data, course data, resource locator...).

Claroline also stores more precise information about the quizzes taken by a user. This includes data about the user taking the quiz, the course in which the quiz is located, the score (and scale) of the user, the start time and end time... We do not have any meta data about concepts and skills but some of them could probably be extracted from the title and description of the quiz and the questions it contains.

For the Learning Path tool we can provide a lot of information such as data about the user and the course, progression, completion status, time taken (total or for a given session), score... As for the quiz tool some additional information regarding skills and concepts could be extracted from the title and description of the learning path and the contained modules.

The available information in Claroline is summarized in Figure 7.

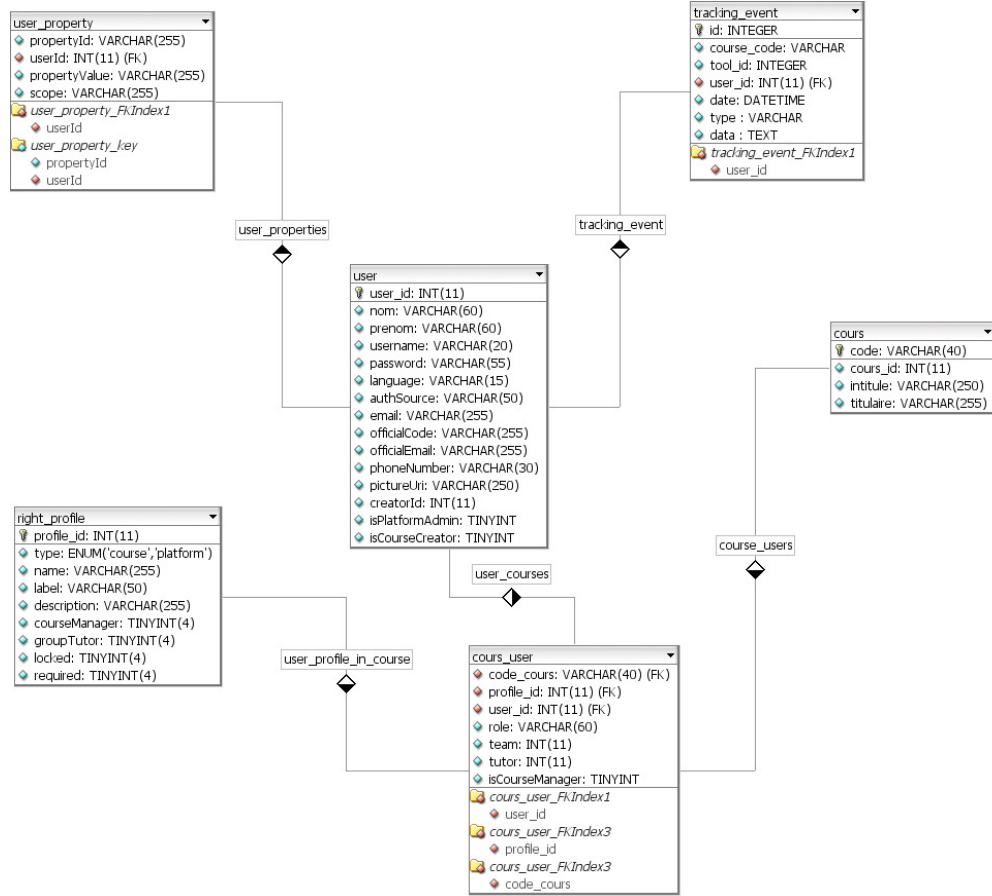


Figure 6: Claroline Data Model.

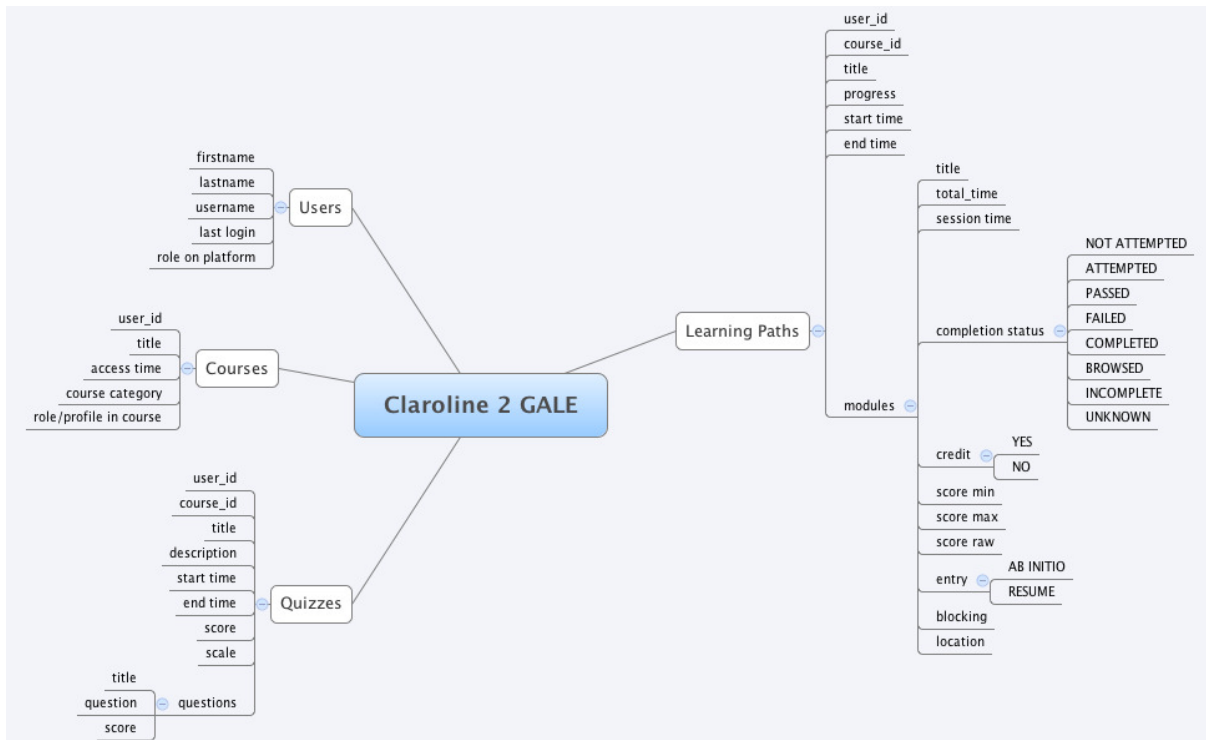


Figure 7: Information available in Claroline.

### 5.5 learn eXact data model

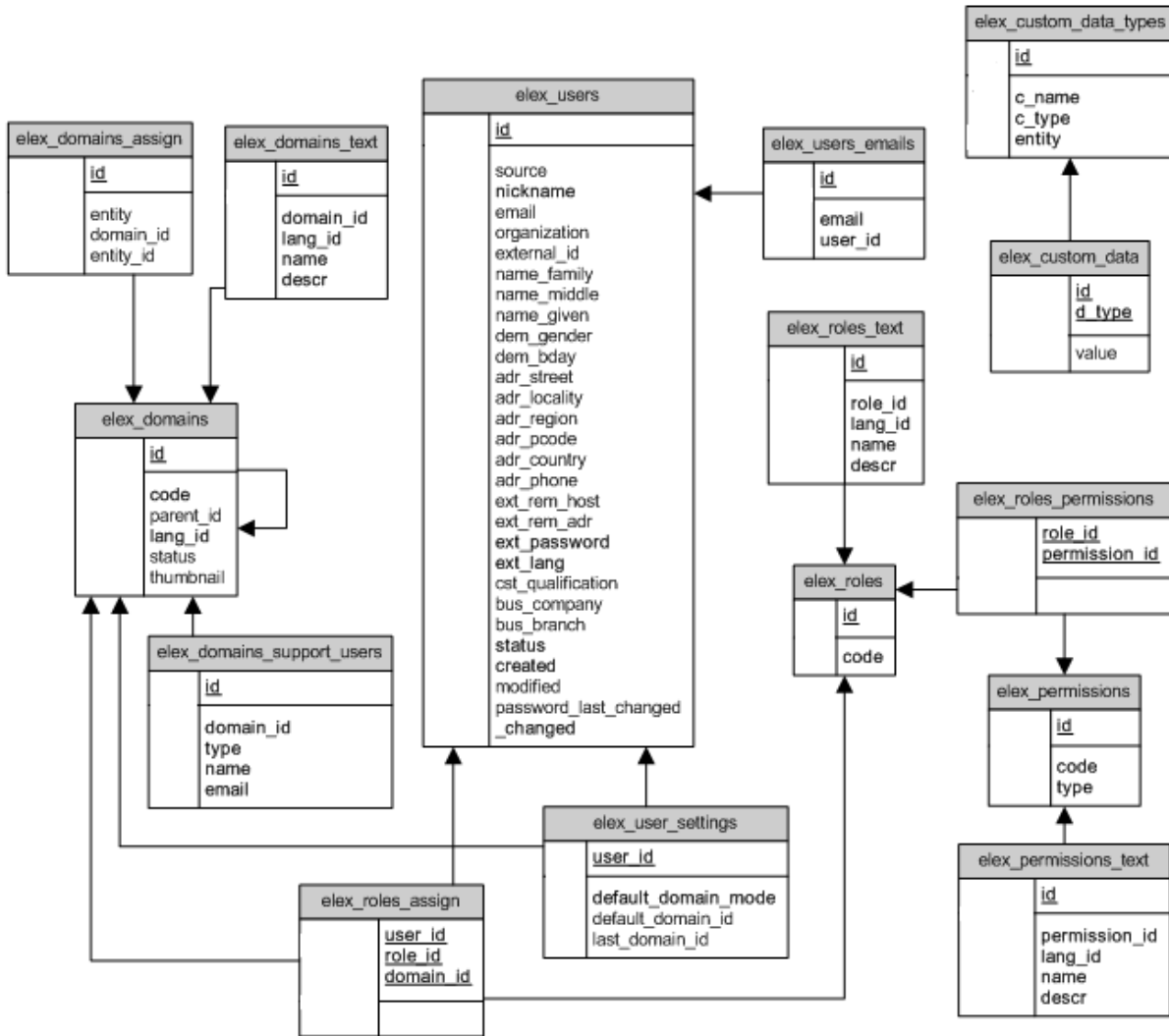


Figure 8: learn eXact data model extraction.

### 5.6 IMC Clix data model

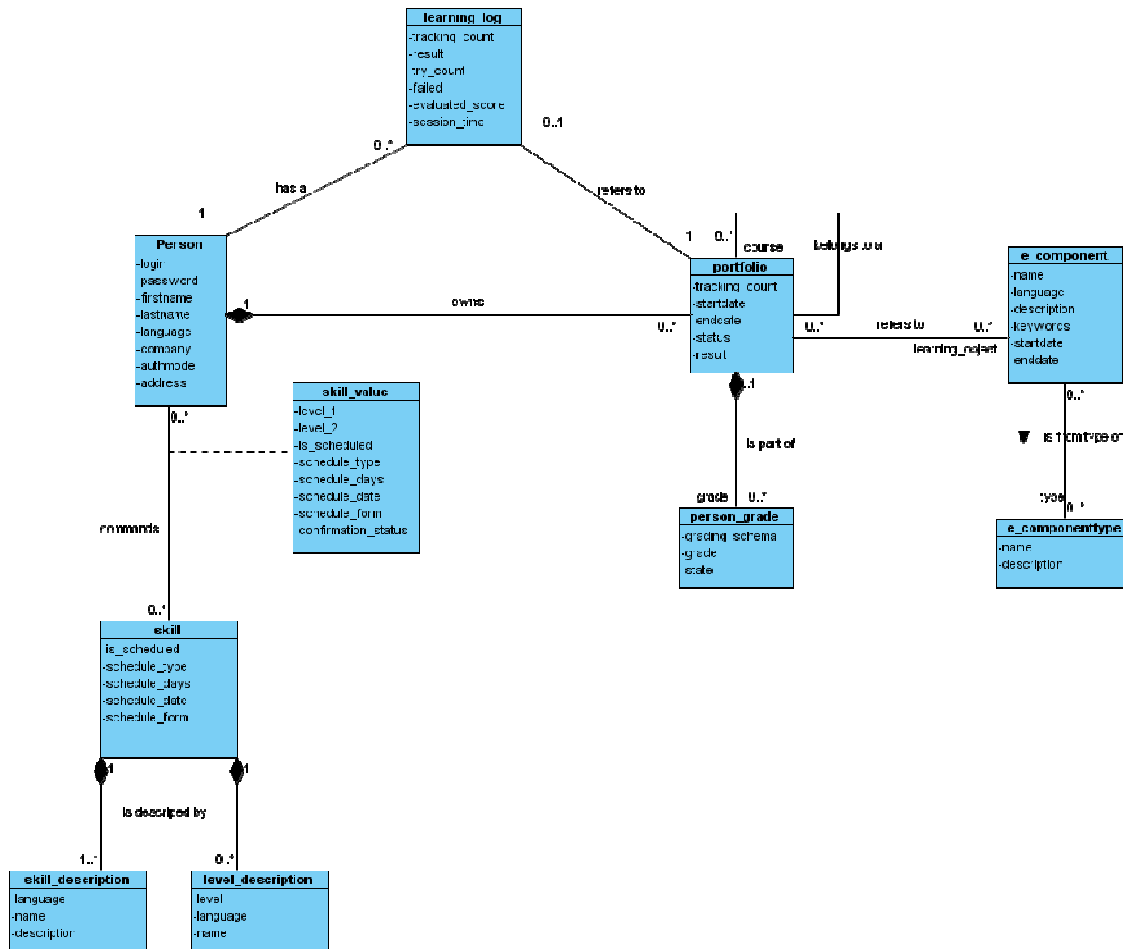


Figure 9: CLIX data model extraction.

In Figure 9 an excerpt of the CLIX data model is provided by an UML class diagram. In the following sections this is explained in more detail.

A user or learner in CLIX is represented by a Person object. In this object several information about the learner are stored. These includes the user’s login name, an encrypted password, the user’s first and last name, the company (or university) she is working for, the user’s address and (preferred) language. Furthermore the standard authentication source (e.g.: LOCAL, LDAP, USER, CSV, SSO) is stored.

Besides a Person there are learning objects in CLIX. Learning objects (e.g. text documents or other media, forums or other communication services, courses, learning communities, etc.) are represented by the e\_component object. A learning object has a name and its language is stored. The type of the learning object can also be defined (is it a text document, a video or something else). Furthermore there is a description which provides information about the learning object itself. Besides the description some keywords can be additionally stored. If the represented learning object is a course, it can also be saved when the course starts and ends.

The status of a learner for a certain learning object is stored in a so called portfolio object. The learning object can be a course or any other learning object (text, video etc.) that may be associated to a course. The portfolio object represents the personal course duration dates and their curriculum components. Besides other data the following data is stored:

- tracking\_count: Number of times the component was called
- startdate: Date when the student started the component

- **enddate:** Date when the student finished the component
- **status:** Status of the student with regard to the component (e.g. Cancelled, Registered, Started, Finished, Passed, Failed)
- **result:** Assessment of the component (achieved points)

If necessary, the grade of a person of an e.g. course or other learning component can be stored in different formats (`person_grade`), for example school grades, university grades or ECTS. The state attribute indicates, if the grade is confirmed by the tutor or not.

Besides the actual learning status stored in the portfolio object, the changing steps leading to the current status are also tracked. This data is represented by the `learning_log` object and has (besides others) the following attributes:

- **tracking\_count:** number of calls of the component by the learner
- **result:** Result of the component for the participant in percent.
- **try\_count:** Number of attempts
- **failed:** Flag indicating whether a component has been locked for exceeding the number of tries.
- **evaluated\_score:** Exact (not percentile) score, as calculated by a test for the last evaluation.
- **session\_time:** Duration of the last session (only for AICC-AUs and SCORM-SCOs)

When a person accessed or passed a course or test, respectively, she acquires a skill or a set of skills. A skill is represented by a skill object. As a learner may lose a skill because of less practicing or necessary repetition, a skill can set to be expired (indicated by the `is_scheduled` attribute). This can be in a certain space of time (`schedule_days`) or on a certain date (`schedule_date`). The attribute `schedule_form` defines how a skill expires (`schedule_type`). A skill can expire completely, or decreased by one level (e.g. expert to intermediate) or decreased to the last accomplished level. The different levels a skill has are described by the `level_description` object. The name of a skill and its description can be stated in different languages (`skill_description` object).

The level of a skill a learner has is stored by the `skill_value` object. Two kinds of levels are distinguished: an entry/requirement level (`level_1`) and a result/target level (`level_2`). For each learner the kind of skill expiration can be adapted if it differs from the standard way stored in the skill object. Furthermore a skill can be confirmed (e.g. by a tutor or superior) if for example the learner accessed the skill by herself.

Figure 10 provides a screenshot of a learner's skills.

Name	Self-assessment	Confirmation status	Confirmed	Information
<input type="checkbox"/> Analytical thinking	□□□□	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Business analysis	□□□□	✓	□□□□ intermediate	ⓘ
<input type="checkbox"/> CLIX back end	□□□□	✓	□□□□ basic	ⓘ
<input type="checkbox"/> CLIX front end	□□□□	✓	□□□□ able to use most functions	ⓘ
<input type="checkbox"/> Communication	□□□□	✓	□□□□ Communicates effectively ...	ⓘ
<input type="checkbox"/> Creativity	□□□□	✓	□□□□ Delivers improvements	ⓘ
<input type="checkbox"/> Data analysis	□□□□	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Database design	□□□□	✓	□□□□	ⓘ
<input type="checkbox"/> Development & training	□□□□	✓	□□□□ intermediate	ⓘ
<input type="checkbox"/> Electronic communication	□□□□	✓	□□□□ expert	ⓘ
<input type="checkbox"/> Internet search techniques	□□□□	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Mathematics	□□□□	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Motivation and coaching	□□□	✓	□□□□ Expert	ⓘ
<input type="checkbox"/> MS Power Point	□□□□	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Negotiating	□□□□	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Operating Systems: Mac OS	□□□□ advanced	✓	□□□□ advanced	ⓘ
<input type="checkbox"/> Oral English	□□□□□□	✓	□□□□□□ business fluent	ⓘ
<input type="checkbox"/> Oral German	□□□□□□	✓	□□□□□□ advanced	ⓘ
<input type="checkbox"/> Programme management	□□□□	✓	□□□□ expert	ⓘ
<input type="checkbox"/> Programming / software development	□□□□	✓	□□□□ expert	ⓘ
<input type="checkbox"/> Project management	□□□□	✓	□□□□ expert	ⓘ
<input type="checkbox"/> Quality assurance	□□□□	✓	□□□□ expert	ⓘ

Figure 10: Skills of a learner.

## 6 Conversion models between GRAPPLE and LMSs

### 6.1 Conversion model from Moodle to GRAPPLE ALE

The choice of using the GALE adaptive engine inside the LMS to offer adaptive features, assures us that the collection of data from adaptive courses is done automatically by GALE, without the need of collecting them inside the LMS. For the collection of other data, such as the personal one or the logs from non-adaptive course, we need to implement a conversion model between the two formats.

As already stated in the paragraph 5.3, the subset of data available inside Moodle that exist could be mapped quite easily in the GRAPPLE ALE format. The problem arises when we need to export data such as the access to a resource or the quiz final mark. In this case could divide in two different classes:

- non-adaptive courses suited to prepare data for exporting process: these are Moodle courses in which author has provided a mapping between resources, quizzes, assignments and other on-line activities to one or more concepts. In this way the semantic interpretation of every student action could be derived from this meta-data structure and information derived could be propagated to the GRAPPLE event bus and collected by the GUMF. We consider that the user model shouldn't be modified by this kind of data in the same way as data collected by GALE course, but that they should be available only for future use inside some adaptive course that are aware of the course and the information itself, without any other effect.
- traditional non-adaptive courses: we consider useless to export data from this kind of courses.

### 6.2 Conversion model from Sakai to GRAPPLE ALE

GRAPPLE ALE automatically collects information about adaptive courses and therefore Sakai does not need to provide any special means of transferring this type of information to GRAPPLE.

Information for non-adaptive courses and personal data can be provided by Sakai using its User API. The User Directory Service enables information such as the student's name and e-mail address to be easily retrieved. The only natural limitation is the extent to which non-adaptive course information is kept in Sakai to begin with. Once a specification of the required information is received from GRAPPLE, Sakai will provide the API for the retrieval of those data.

### 6.3 Conversion model from Claroline to GRAPPLE ALE

Concerning the adaptive courses, no conversion is needed from Claroline to GALE since the collection of data from adaptive courses is done automatically by GALE.

Concerning the non-adaptive course, Claroline can provide converted information from the user model. The data describe in 5.3 can be propagated easily to the GRAPPLE event bus and collected by UMF.

The access to adaptive course contents in Claroline will be made available through a new tool that could be installed by course managers. The platform will only export data from courses in which this tool is made available and not from every traditional course.

### 6.4 Conversion model from CLIX to GRAPPLE ALE

Most of the attributes of the GALE data model are also available in the CLIX data model. CLIX does not store the following personal user data:

- the gender explicitly (but the salutation),
- the data of birth, and
- the region.

If these attributes are really needed, the CLIX data model would have to be extended. At this moment a need for this is not seen.

All other data is also stored in the CLIX data model, hence a conversion is not necessary for this. Nevertheless a common agreement of the attribute names themselves (e.g. course or component) is necessary. Furthermore, the possible values for the single attributes (domain values) need to be converted. This includes for example how to express if a learner has attended a course (0 or 1, yes or no). Furthermore, the names of users and courses need to be converted or predefined.

## References

- [1] M. A. Chatti, R. Klamma, C. Quix, D. Kensche: LM-DTM: An Environment for XML-Based, LIP/PAPI-Compliant Deployment, Transformation and Matching of Learner Models  
*P. Goodyear, D.G. Sampson, D.J.-T. Yang, Kinshuk, T. Okamoto, R. Hartley, and N.-S. Chen (eds.): Proceedings of the 5th International Conference on Advanced Learning Technologies (ICALT 2005), July 5-8, Kaohsiung, Taiwan, 567 – 569*
- [2] LIP Editor <http://www-i5.informatik.rwth-aachen.de/lehrstuhl/staff/chatti/LIPEditor/index.html>
- [3] Burgos, D., Berbegal, N., Griffiths, D., Tattersall, C., & Koper, R. (2005). IMS Learning Design: How the specifications can change the current e-learning landscape (Vol. 2). *Moscow: Magazine of Moscow State University for Economy, Statistics and Computer Science.*
- [4] IMS Learner Information Packaging Information Model Specification  
<http://www.imsglobal.org/profiles/lipinfo01.html>
- [5] LIP2PAPI Converter <http://www-i5.informatik.rwth-aachen.de/lehrstuhl/staff/chatti/LIP2PAPICConverter/index.html>
- [6] Dolog, Gavriolaie, et al.(2003) Integrating Adaptive Hypermedia Techniques and Open RDF-Based Environments. *In Proc. of 12th International World Wide Web Conference*