

# GRAPPLE

D5.2b Version: 2.0

## Conversion components between GRAPPLE and LMSs

<b>Document Type</b>	Deliverable
<b>Editor(s):</b>	Lucia Oneto
<b>Author(s):</b>	Lucia Oneto (Gilabs), Paolo Tenerini (Gilabs), Luca Mazzola (USI), Simon Gaeremynck (UCAM), Patrick Pekczynski (IMC), Frederic Minne (UCL)
<b>Reviewer(s):</b>	Patrick Pekczynski (IMC), Frederic Kleinermann (VUB)
<b>Work Package:</b>	WP5
<b>Due Date:</b>	M18
<b>Version:</b>	2.0
<b>Version Date:</b>	26-01-2010
<b>Total number of pages:</b>	27

**Abstract:** This document contains the description of the GRAPPLE Conversion Components introduced into the GRAPPLE system to allow interoperability among the existing LMSs and the GRAPPLE framework.

**Keyword list:** Conversion, learning information, LMS, standard

## Summary

This document presents an update of D5.2a and D7.2a for a set of harmonized data models to be exchanged within the GRAPPLE system, able to guarantee interoperability within the operational infrastructure and to harmonize the data flows between the participating systems.

The great challenge of data modelling for sharing knowledge between LMS and ALE is completely under progress and not straight forward at all, since several projects and tools on the topic are still under development. We need to do pioneer work which however promises to come up with a better solution for integration and interoperability. The task of this work package is to carefully select and collect data models for GRAPPLE.

## Authors

Person	Email	Partner code
Simon Gaeremynck	sg555@cam.ac.uk	UCAM
Luca Mazzola	luca.mazzola@lu.unisi.ch	USI
Frederic Minne	frederic.minne@uclouvain.be	UCL
Lucia Oneto	l.oneto@giuntilabs.com	GILABS
Patrick Pekczynski	Patrick.Pekczynski@im-c.de	IMC
Paolo Tenerini	p.tenerini@giuntilabs.com	GILABS

## Table of Contents

<b>AUTHORS</b> .....	<b>2</b>
<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>TABLES AND FIGURES</b> .....	<b>3</b>
<b>LIST OF ACRONYMS AND ABBREVIATIONS</b> .....	<b>4</b>
<b>1 INTRODUCTION</b> .....	<b>5</b>
1.1 <b>Task and Deliverable Description</b> .....	<b>6</b>
<b>2 SPECIFICATION OF THE GRAPPLE CONVERSION COMPONENT (GCC)</b> .....	<b>6</b>
2.1 <b>Access to a course</b> .....	<b>7</b>
2.2 <b>Tests/quizzes</b> .....	<b>8</b>
2.3 <b>Registration</b> .....	<b>10</b>
2.4 <b>Student enrolment</b> .....	<b>12</b>
2.5 <b>User login</b> .....	<b>13</b>
2.6 <b>Role change</b> .....	<b>14</b>

- 2.7 Learning activity change..... 15**
- 2.8 Learning activity addition ..... 16**
- 2.9 Learning activity removal ..... 17**
- 2.10 Access to a learning activity..... 18**
  
- 3 INSTALLATION PROCEDURES..... 19**
- 3.1 How to install GCC in Claroline..... 19**
  - 3.1.1 Claroline Release Version ..... 19
  - 3.1.2 Installation..... 19
  - 3.1.3 Installation..... 20
- 3.2 How to install GCC in CLIX..... 20**
  - 3.2.1 CLIX Release Version ..... 20
  - 3.2.2 Installation..... 20
  - 3.2.3 Availability..... 21
  - 3.2.4 Open Issues..... 22
- 3.3 How to install GCC in ELEX..... 22**
  - 3.3.1 ELEX Release Version ..... 22
  - 3.3.2 Installation..... 22
  - 3.3.3 Availability..... 25
- 3.4 How to install GCC in Moodle..... 26**
  - 3.4.1 Moodle Release version ..... 26
  - 3.4.2 Installation..... 26
  - 3.4.3 Availability..... 26
- 3.5 How to install GCC in Sakai..... 26**
  - 3.5.1 Sakai Release version & General Sakai setup:..... 26
  - 3.5.2 Installation..... 26
  - 3.5.3 Availability..... 27
  
- 4 OPEN ISSUES ..... 27**

## Tables and Figures

### List of Figures

- Figure 1: The role of the GCC in the GRAPPLE framework. .... 5
- Figure 2: IMS LIP structure. .... 6

## List of Acronyms and Abbreviations

ALE	Adaptive Learning Environment
GCC	GRAPPLE Conversion Component
GRAPPLE	Generic Responsive Adaptive Personalized Learning Environment
GUMF	GRAPPLE User Modelling Framework
IMS	Instructional Management System
IMS LIP	IMS Learner Information Package
LMS	Learning Management System

# 1 Introduction

The GRAPPLE project will enable the cooperation of Adaptive Learning Environments (ALE) and Learning Management Systems (LMS). In order to reach this challenging task, suitable data models for interoperability are needed.

In the GRAPPLE project, WP5 is focused on 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 analyze a number of widely accepted open source LMSs (like Moodle, Sakai, and Claroline) and commercial LMSs (ELEX - learn exact<sup>1</sup>, IMC CLIX<sup>2</sup>) to work on an integration with the GRAPPLE framework.

In the first year of the project, it was possible to achieve a complete analysis of the standards available and of the data the above mentioned LMSs can provide to the GRAPPLE framework. Outcome of this analysis is D5.2a - Conversion models between GRAPPLE and LMSs [1].

This deliverable describes the status of the implementation work done to build a GRAPPLE Conversion Component (GCC) for any LMS. The set of all the GCCs produced in the GRAPPLE project compose D5.2b.

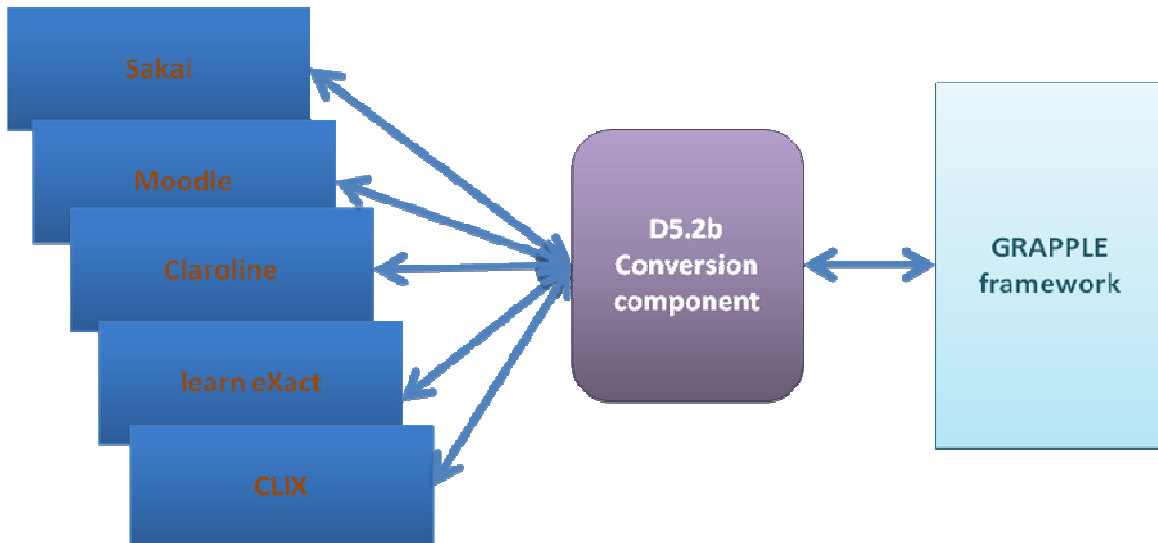


Figure 1: The role of the GCC in the GRAPPLE framework.

As described in [1], any GCC specific to a given LMS, is in charge to convert a set of LMS data to the IMS LIP 1.0.1 standard [4] and then manageable by the GRAPPLE framework.

<sup>1</sup> [http://www.giuntilabs.com/learn\\_eXact\\_Enterprise/index.php](http://www.giuntilabs.com/learn_eXact_Enterprise/index.php)

<sup>2</sup> <http://www.im-c.de/en/we-make-e-learning-easy-learning-management-system-rapid-authoring-consulting-and-content-services/>

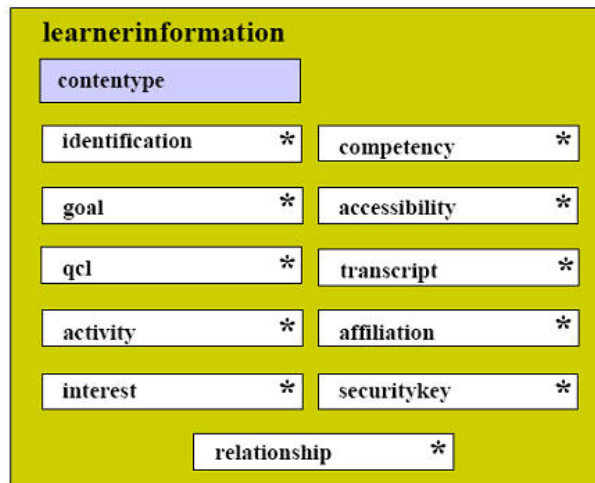


Figure 2: IMS LIP structure.

## 1.1 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 Specification of the GRAPPLE Conversion Component (GCC)

The LMSs are a useful source of information for the GRAPPLE system: they have got relevant information about the user. Then GUMF is the natural destination of all the user details necessary for the course adaptation GALE is in charge of.

The following user events have been identified. They provide user information to GRAPPLE (more specifically to GUMF):

1. *Access to a course*: it identifies the moment when the user access a course
2. *Tests/quizzes*: the user has terminated a test
3. *Registration*: the user logs the first time in the LMS integrated with GRAPPLE
4. *Student enrolment*: the student is enrolled in a new course
5. *User login*: the user logs in the LMS integrated with GRAPPLE
6. *Role change*: the user has changed his role
7. *Learning activity change*: a particular learning activity has been changed
8. *Learning activity addition*: a particular learning activity has been added
9. *Learning activity removal*: a particular learning activity has been removed
10. *Access to a learning activity*: it identifies the moment when the user access a learning activity.

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.*

A list of user parameters has been prepared for each event.

The user parameters are available to the GRAPPLE system only if they are made public (see first scenario in [2]).

The LMS is in charge to provide the user parameters to the GRAPPLE system: for the most of the LMSs it is not possible to provide the data in real time. Then it is necessary to set up a batch system able to send periodically the user parameters related to the events mentioned above.

During the implementation work, the definition of the events have been discussed and evaluated more times and the events and related parameters were redefined as described below.

All the parameters related to any event have been mapped to the IMS LIP standard and encapsulated into GrappleStatements [3] with the following syntax:

**<statement>**

**<origin>**

<learnerinformation>

IMS LIP SPECIFICATION

</learnerinformation>

**</origin>**

**</statement>**

## 2.1 Access to a course

The user accesses a particular course: the user can be anyone with permission rights to access a course. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contentype – 13.3 – sourcedid (13.3.1.1)
- url: contentype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contentype – 13.3 – sourceid (13.3.1.1)
- Date of access: activity - 13.6 - date ( 13.6.1 - typename, 13.6.2 - datetime)
- Course topic: activity - 6.12 - description (13.5.2 - long)
- Course title: activity - 6.12 - description (13.5.1 - short)
- Course date of creation: activity - 13.6 - date ( 13.6.1 - typename, 13.6.2 - datetime).

The following XML code is an example sample for the implemented event:

```
<?xml version="1.0" encoding="UTF-8"?>
<statement>
```

```

<origin>
<learnerinformation
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
http://www.imsglobal.org/xsd/imslip_v1p0.xsd"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0"
  <contenttype>
    <referential>
      <sourcedid>
        <source>ELEX</source>
        <id>http://localhost:3916/www/</id>
      </sourcedid>
    </referential>
  </contenttype>
  <securitykey>
    <keyfields>
      <fieldlabel>
        <typename>
          <tyvalue>User Id</tyvalue>
        </typename>
      </fieldlabel>
      <fielddata>student</fielddata>
    </keyfields>
  </securitykey>
  <activity>
    <contenttype>
      <referential>
        <sourcedid>
          <source>GrappleDR_0</source>
          <id>3</id>
        </sourcedid>
      </referential>
    </contenttype>
    <date>
      <typename>
        <tysource sourcetype="list">AccessDate, CreationDate, StartDate, StopDate, BirthDate</tysource>
        <tyvalue>AccessDate</tyvalue>
      </typename>
      <datetime>2009-12-14T17:46:49</datetime>
    </date>
    <date>
      <typename>
        <tysource sourcetype="list">AccessDate, CreationDate, StartDate, StopDate, BirthDate</tysource>
        <tyvalue>CreationDate</tyvalue>
      </typename>
      <datetime>2009-11-26T15:11:46</datetime>
    </date>
    <description>
      <short>GrappleDR</short>
      <long>Tests quizzes about Grapple project.</long>
    </description>
  </activity>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.2 Tests/quizzes

The user (learner) completes a test or quiz. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourcedid (13.3.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contenttype – 13.3 – sourceid (13.3.1.1)
- Topic: activity - 6.12 - description (13.5.2 - long)
- Goal: activity - 6.11.12 - evaluation description



- Score: activity - 6.11.11.2 - score
- Scale description: activity -6.11.11.1 - interpretscore
- Timestamp – start (last): activity - 13.6 - date ( 13.6.1 - typename, 13.6.2 - datetime)
- Timestamp – end: activity - 13.6 - date ( 13.6.1 - typename, 13.6.2 - datetime)
- Attempts: activity - 6.11.9 – noofattempts.

The following XML code is a sample of the implemented event:

```
<?xml version="1.0" encoding="UTF-8" ?>
<statement>
  <origin>
    <learnerinformation xmlns="http://www.imsglobal.org/xsd/imslip_v1p0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0 http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <fieldlabel>
          <typename>
            <tyvalue>User Id</tyvalue>
          </typename>
        </fieldlabel>
        <fielddata>student</fielddata>
      </keyfields>
    </securitykey>
    <activity>
      <contenttype>
        <referential>
          <sourcedid>
            <source>http://localhost:3916/www</source>
            <id>3</id>
          </sourcedid>
        </referential>
      </contenttype>
      <activity>
        <contenttype>
          <referential>
            <sourcedid>
              <source>http://localhost:3916/www</source>
              <id>5</id>
            </sourcedid>
          </referential>
        </contenttype>
        <evaluation>
          <date>
            <typename>
              <tyvalue>StartDate</tyvalue>
            </typename>
            <datetime>2009-12-04T17:37:52</datetime>
          </date>
          <date>
            <typename>
              <tyvalue>StopDate</tyvalue>
            </typename>
            <datetime>2009-12-04T17:38:26</datetime>
          </date>
          <noofattempts>112</noofattempts>
          <result>
            <score>
              <fieldlabel>
                <typename>
                  <tyvalue>Total</tyvalue>
                </typename>
              </fieldlabel>
            </score>
          </result>
        </evaluation>
      </activity>
    </contenttype>
  </learnerinformation>

```

```

        <fielddata>45</fielddata>
    </score>
    <interpretscore>
        <fieldlabel>
            <typename>
                <tyvalue>MinScore</tyvalue>
            </typename>
        </fieldlabel>
        <fielddata>0</fielddata>
    </interpretscore>
    <interpretscore>
        <fieldlabel>
            <typename>
                <tyvalue>MaxScore</tyvalue>
            </typename>
        </fieldlabel>
        <fielddata>45</fielddata>
    </interpretscore>
</result>
<description>
    <short>Grapple Tests/Quizzes</short>
    <long>Test SCORM course</long>
</description>
</evaluation>
</activity>
</activity>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented, except MinScore for Moodle: MinScore for quizzes is not available, the value is set to 0 by default.

## 2.3 Registration

The user (learner) logs the first time in the LMS integrated with GRAPPLE: all the personal details are passed to the GRAPPLE system (GUMF is the final receiver of this data). The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contentype – 13.3 – sourcedid (13.3.1.1)
- url: contentype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Last Name: identification - 2.4 -name (2.4.4 - partname, 2.4.4.2 text)
- First Name: identification - 2.4 -name (2.4.4 - partname, 2.4.4.2 text)
- Organisation: identification - 2.4 -name (2.4.4 - partname, 2.4.4.2 text)
- Email: identification - 2.6 contactinfo (2.6.8 - email)
- Language: accessibility - 3.3 - language
- Gender: identification - 2.7 - demographics (2.7.5 - gender)
- Date of birth: identification - 2.7 - demographics (2.7.6 - date)
- Street: identification -2.5 - address (...)
- Town: identification -2.5 - address (...)
- Postal code: identification -2.5 - address (...)
- Region: identification -2.5 - address (...)
- Country: identification -2.5 - address (...)
- IP address: identification - 2.9 extension (14.13 ext\_identification)
- Role: identification - 2.9 extension (14.13 ext\_identification).

Role describes 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.*

Note: since many Registration attributes are can be made public and available by the LMS administrator (see First scenario in D7.1b), they are not mandatory. In any case the structure of the event does not change.

The following XML code is a sample of the implemented event:

```
<?xml version="1.0" encoding="UTF-8" ?>
<statement>
  <origin>
    <learnerinformation xmlns="http://www.imslobal.org/xsd/imsliip_v1p0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imslobal.org/xsd/imsliip_v1p0
      http://www.imslobal.org/xsd/imsliip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>
              <tyvalue>User Id</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>student</fielddata>
        </keyfields>
      </securitykey>
      <identification>
        <name>
          <partname>
            <typename>
              <tysource sourcetype="imsdefault">Last Name, First Name, Organisation</tysource>
              <tyvalue>Last Name</tyvalue>
            </typename>
            <text>StudentLastName</text>
          </partname>
          <partname>
            <typename>
              <tysource sourcetype="imsdefault">Last Name, First Name, Organisation</tysource>
              <tyvalue>First Name</tyvalue>
            </typename>
            <text>StudentFirstName</text>
          </partname>
          <partname>
            <typename>
              <tysource sourcetype="imsdefault">Last Name, First Name, Organisation</tysource>
              <tyvalue>Organisation</tyvalue>
            </typename>
            <text />
          </partname>
        </name>
        <contactinfo>
          <email>student@giuntilabs.com</email>
        </contactinfo>
        <demographics>
          <gender gender="M" />
        </demographics>
      </identification>
    </learnerinformation>
  </origin>
</statement>
```

```

        <date>
          <typename>
            <tysource sourcetype="list">AccessDate, CreationDate, StartDate, StopDate,
BirthDate</tysource>
            <tyvalue>BirthDate</tyvalue>
          </typename>
          <datetime>1990-01-01T00:00:00</datetime>
        </date>
      </demographics>
      <address>
        <street>
          <streetname />
        </street>
        <city />
        <region />
        <country />
        <postcode />
      </address>
    </identification>
    <accessibility>
      <language>
        <typename>
          <tysource
sourcetype="imsdefault">English,Dutch,French,Spanish,German,Italian</tysource>
          <tyvalue>English</tyvalue>
        </typename>
      </language>
    </accessibility>
    <identification>
      <comment>IP address</comment>
      <ext_identification>127.0.0.1</ext_identification>
    </identification>
    <identification>
      <comment>Role</comment>
      <ext_identification>Learner</ext_identification>
    </identification>
  </learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented. For Moodle this event is propagated only when a new user is created, and NOT when it logs in the first time.

## 2.4 Student enrolment

The user (learner) is enrolled to a course. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contentype – 13.3 – sourcedid (13.3.1.1)
- url: contentype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contentype – 13.3 – sourceid (13.3.1.1)
- Role: identification - 2.9 extension (14.13 ext\_identification).

The following XML code is a sample of the implemented event:

```

<statement>
  <origin>
    <learnerinformation
      xmlns="http://www.imsglobal.org/xsd/imslip_v1p0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0
http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contentype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>

```

```

</contenttype>
<securitykey>
  <keyfields>
    <fieldlabel>
      <typename>
        <tyvalue>User Id</tyvalue>
      </typename>
    </fieldlabel>
    <fielddata>student</fielddata>
  </keyfields>
</securitykey>
<activity>
  <contenttype>
    <referential>
      <sourcedid>
        <source>http://grapple.giuntlabs.com/elexGrapple/</source>
        <id>2</id>
      </sourcedid>
    </referential>
  </contenttype>
</activity>
<identification>
  <comment>Role</comment>
  <ext_identification>Learner</ext_identification>
</identification>
<identification>
  <comment>Role</comment>
  <ext_identification>User</ext_identification>
</identification>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.5 User login

The user (learner) logs in the LMS integrated with GRAPPLE. This event does not correspond to the Registration event. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourcedid (13.3.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contenttype – 13.3 – sourceid (13.3.1.1)
- IP address: identification - 2.9 extension (14.13 ext\_identification).

The following XML code is a sample of the implemented event:

```

<statement>
  <origin>
    <learnerinformation
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0 http://www.imsglobal.org/xsd/imslip_v1p0.xsd"
      xmlns="http://www.imsglobal.org/xsd/imslip_v1p0"
    >
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>
              <tyvalue>User Id</tyvalue>
            </typename>
          </fieldlabel>

```

```

    <fielddata>student</fielddata>
  </keyfields>
</securitykey>
<activity>
  <contenttype>
    <referential>
      <sourcedid>
        <source>http://localhost:3916/www</source>
        <id>2</id>
      </sourcedid>
    </referential>
  </contenttype>
</activity>
<activity>
  <contenttype>
    <referential>
      <sourcedid>
        <source>http://localhost:3916/www</source>
        <id>3</id>
      </sourcedid>
    </referential>
  </contenttype>
</activity>
<identification>
  <comment>IP address</comment>
  <ext_identification>127.0.0.1</ext_identification>
</identification>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.6 Role change

One of the user's roles has changed. This event does not correspond to the Registration. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourcedid (13.3.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Role: identification - 2.9 extension (14.13 ext\_identification).

The following XML code is a sample of the implemented event:

```

<statement>
  <origin>
    <learnerinformation
      xmlns="http://www.imsglobal.org/xsd/imslip_v1p0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0
        http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>
              <tyvalue>User Id</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>student</fielddata>
        </keyfields>
      </securitykey>
    </learnerinformation>
  </origin>
</statement>

```

```

<identification>
  <comment>Role</comment>
  <ext_identification>Learner</ext_identification>
</identification>
<identification>
  <comment>Role</comment>
  <ext_identification>User</ext_identification>
</identification>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.7 Learning activity change

A learning activity has been changed. Learning activity is part of a course (a course can be composed by one or more learning activities). The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourcedid (13.3.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contenttype – 13.3 – sourceid (13.3.1.1)
- Activity id: activity - activity - 6.13 - contenttype – 13.3 – sourceid (13.3.1.1)
- Status: activity - activity - 6.13 – status – 13.8 - typename (13.4). The Status identifies this event as **change**.

The following XML code is a sample of the implemented event:

```

<statement>
  <origin>
    <learnerinformation xmlns="http://www.imsglobal.org/xsd/imslip_v1p0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0
        http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>
              <tyvalue>User Id</tyvalue>
            </typename>
          </fieldlabel>
          <felddata>student</felddata>
        </keyfields>
      </securitykey>
      <activity>
        <contenttype>
          <referential>
            <sourcedid>
              <source>GrappleDR</source>
              <id>3</id>
            </sourcedid>
          </referential>
        </contenttype>
      </activity>
      <contenttype>
        <referential>

```

```

        <sourceid>
        <source>French material</source>
        <id>4</id>
    </sourceid>
</referential>
</contenttype>
<status>
    <typename>
        <tysource sourcetype="list">Changed, Added, Removed</tysource>
        <tyvalue>Changed</tyvalue>
    </typename>
</status>
</activity>
</activity>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.8 Learning activity addition

A learning activity has been added. Learning activity is part of a course or the course itself. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourceid (13.3.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contenttype – 13.3 – sourceid (13.3.1.1)
- Activity id: activity - activity - 6.13 - contenttype – 13.3 – sourceid (13.3.1.1)
- Status: activity - activity - 6.13 – status – 13.8 - typename (13.4) ). The Status identifies this event as **addition**.

The following XML code is a sample of the implemented event:

```

<statement>
  <origin>
    <learnerinformation
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0
        http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourceid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourceid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>
              <tyvalue>User Id</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>student</fielddata>
        </keyfields>
      </securitykey>
    </activity>
    <contenttype>
      <referential>
        <sourceid>
          <source>GrappleDR</source>
          <id>3</id>
        </sourceid>
      </referential>
    </contenttype>
  </origin>
</statement>

```



```

    </sourcedid>
  </referential>
</contenttype>
<activity>
  <contenttype>
    <referential>
      <sourcedid>
        <source>French material</source>
        <id>4</id>
      </sourcedid>
    </referential>
  </contenttype>
  <status>
    <typename>
      <tysource sourcetype="list">Changed, Added, Removed</tysource>
      <tyvalue>Added</tyvalue>
    </typename>
  </status>
</activity>
</activity>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.9 Learning activity removal

A learning activity has been removed. Learning activity is part of a course. The following user variables are identified and mapped to IMS LIP 1.0.1 (the numbers below indicate the single table and row in the IMS LIP specification) [4]:

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourcedid (13.3.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contenttype – 13.3 – sourceid (13.3.1.1)
- Activity id: activity - activity -6.13 - contenttype – 13.3 – sourceid (13.3.1.1)
- Status: activity - activity - 6.13 – status – 13.8 - typename (13.4). The Status identifies this event as **removal**.

Claroline, Moodle, Sakai, Clix and learn eXact provide all the above parameters.

```

<statement>
  <origin>
    <learnerinformation
      xmlns="http://www.imsglobal.org/xsd/imslip_v1p0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0
        http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>
              <tyvalue>User Id</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>student</fielddata>
        </keyfields>
      </securitykey>
    </activity>

```

```

<contenttype>
  <referential>
    <sourcedid>
      <source>GrappleDR</source>
      <id>3</id>
    </sourcedid>
  </referential>
</contenttype>
<activity>
  <contenttype>
    <referential>
      <sourcedid>
        <source>Italian material</source>
        <id>4</id>
      </sourcedid>
    </referential>
  </contenttype>
  <status>
    <typename>
      <tysource sourcetype="list">Changed, Added, Removed</tysource>
      <tyvalue>Removed</tyvalue>
    </typename>
  </status>
</activity>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

## 2.10 Access to a learning activity

This event identifies the moment when the user access a learning activity (the numbers below indicate the single table and row in the IMS LIP specification) [4].

- LMS type (CLIX, Claroline, Moodle, ELEX, Sakai): contenttype – 13.3 – sourcedid (13.3.1.1.1)
- url: contenttype – 13.3 – id (13.3.1.2)
- User id: security key - 11.4 - keyfields (11.4.1 - fieldlabel, 11.4.2 - fielddata)
- Course id: activity - contenttype – 13.3 – sourceid (13.3.1.1)
- Activity id: activity - activity -6.13 - contenttype – 13.3 – sourceid (13.3.1.1)
- Date of access: activity - activity -6.13 – date - 6.4 – (13.6.1 – typename, 13.6.2 – datetime)
- Activity topic: activity - activity -6.13 – description – 6.12 (13.5.2 – long)
- Activity title: activity - activity -6.13 – description – 6.12 (13.5.1 – short).

The following XML code is a sample of the implemented event:

```

<statement>
  <origin>
    <learnerinformation
      xmlns="http://www.imsglobal.org/xsd/imslip_v1p0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imslip_v1p0
      http://www.imsglobal.org/xsd/imslip_v1p0.xsd">
      <contenttype>
        <referential>
          <sourcedid>
            <source>ELEX </source>
            <id> http://localhost:3916/www </id>
          </sourcedid>
        </referential>
      </contenttype>
      <securitykey>
        <keyfields>
          <fieldlabel>
            <typename>

```

```

        <tyvalue>User Id</tyvalue>
    </typename>
</fieldlabel>
<fielddata>student</fielddata>
</keyfields>
</securitykey>
<activity>
    <contenttype>
        <referential>
            <sourcedid>
                <source>GrappleDR</source>
                <id>3</id>
            </sourcedid>
        </referential>
    </contenttype>
    <activity>
        <contenttype>
            <referential>
                <sourcedid>
                    <source>Grapple Tests/Quizzes</source>
                    <id>4</id>
                </sourcedid>
            </referential>
        </contenttype>
        <date>
            <typename>
                <tysource sourceType="list">AccessDate, CreationDate, StartDate, StopDate,
BirthDate</tysource>
                <tyvalue>AccessDate</tyvalue>
            </typename>
            <datetime>2009-12-03T16:20:34</datetime>
        </date>
        <description>
            <short>Test SCORM course</short>
            <long>This learning activity assesses your knowledge</long>
        </description>
    </activity>
</activity>
</learnerinformation>
</origin>
</statement>

```

For any LMS all the fields have been implemented.

### 3 Installation procedures

This chapter describes the installation procedures of the GRAPPLE Conversion component (GCC) for any LMS involved in the project.

#### 3.1 How to install GCC in Claroline

##### 3.1.1 Claroline Release Version

The GCC integration requires the latest development version of Claroline available from the project subversion trunk. This version will be the next stable version of Claroline and will be released as Claroline 1.10 in September 2010.

##### 3.1.2 Installation

This integration also requires two additional modules for Claroline:

1. The first one is named "GRPLAD" and allows the Claroline kernel to send data over GEB.
2. The second one named "GRAPPLE" is a fork from the Claroline Learning Path tool with additional features needed to send data to GRAPPLE.

To install GCC at this time in Claroline you need to:

1. Get the latest development version of Claroline from sourceforge : <http://claroline.svn.sourceforge.net/viewvc/claroline/trunk.tar.gz?view=tar>
2. Untar the archive and install it by following the instructions in INSTALL.txt
3. Create a module folder at the root of Claroline (at the same level as the platform and courses folders)
4. Get the latest version of the GRPPLAD module : <http://claroline-pack.svn.sourceforge.net/viewvc/claroline-pack/trunk/modules/GRPPLAP.tar.gz?view=tar>
5. Untar the archive and copy the GRPPLAD folder to the module directory created at step 3
6. Go to the Claroline module administration and install the module
7. Get the latest version of the GRAPPLE module : <http://claroline-pack.svn.sourceforge.net/viewvc/claroline-pack/trunk/modules/GRAPPLE.tar.gz?view=tar>
8. Untar the archive and copy the GRAPPLE folder to the module directory created at step 3 9. Go to the Claroline module administration and install the module.

In the future we will provide two ready to install modules and the installation process will be easier.

### 3.1.3 Installation

No IPRs restrict the usage of the GCC module for Claroline.

## 3.2 How to install GCC in CLIX

As CLIX is Java-based, the GCC for CLIX is written in Java and uses a Castor-xml binding in order to be able to map valid IMS-LIP xml files to Java objects. From those Java objects CLIX can dynamically create user-related events and send them to the Grapple Event Bus (GEB).

However, the developed GRAPPLE Conversion Component (GCC) for CLIX will only work if a CLIX installation is used that also implements the connection layer between the CLIX-system and the GCC for CLIX. This connection layer is responsible for tracking the CLIX user events and fires the corresponding user events as defined in this deliverable via the developed GCC for CLIX library.

### 3.2.1 CLIX Release Version

Release Version will be CLIX 9.0 (further integration into succeeding version may be possible like CLIX 10.0).

### 3.2.2 Installation

As the GCC for CLIX has recently been ported to be a "Maven" (<http://maven.apache.org/>) project, the installation process is as follows:

#### 3.2.2.1 Installation from source code

- Download the source code from the URL indicated in section 3.2.3
- Install apache maven (<http://maven.apache.org/>)
- Go to the downloaded directory (e.g. "cd gcc")
- type "mvn install"
- Use the created java-library from the target directory (e.g. gcc/target/gcc-\$version\_numer\$.jar to include it in your java project.

#### 3.2.2.2 Installation using the binaries

- Download the java library files (jar-files) from the URL indicated in 3.2.3

- Include the jar-files into your java project.

### 3.2.2.3 Installing the GCC jar-file

The resulting java archive file (jar-file) has to be included in the CLIX web application library path.

### 3.2.2.4 Configuration

Before the modified CLIX instance can be started one needs to modify a CLIX configuration file, `clix.conf` such that it contains the following section:

```
[GRAPPLE]
createFiles=true
sendEvents=true
sendName=setUMData
gebEndpoint=http://dyn070.win.tue.nl:8080/GrappleEventBus/gebListenerService
eventEndpoint=http://dyn070.win.tue.nl:8080/GrappleEventBus/eventGEBListenerService
listenerName=http://www.learning-demo.eu/grapplix/eventEventListenerService?wsdl
methodList=setUMData,getCoursesResponse,method_test,method_test2
```

---

Table 1: Adapting the CLIX configuration file for GCC

Here one has the possibility to configure several parameters:

- **createFiles**  
The parameter “createFiles” denotes, whether for each event a respective XML IMS LIP file shall be created or not
- **sendEvents**  
The parameter “sendEvents” denotes, whether each event shall be sent to the GRAPPLE Event Bus or not (this allows debugging of the events)
- **sendName:**  
The parameter “sendName” contains the name of the event as which the user event shall be sent to the GEB.
- **gebEndpoint**  
This parameter specifies the URL at which the Grapple Event Bus is reachable. This is necessary for the LMS Event Listener to be registered
- **eventEndpoint**  
This parameter specifies the URL at which the GEB event listener is reachable and hence specifies what URL CLIX shall route the user events to
- **listenerName**  
This parameter specifies the name of the CLIX event listener which needs to be a URL that can be reached from the running GEB instance CLIX wants to register at
- **methodList**  
Finally, the parameter “methodList” allows the administrator to define several event names the CLIX event listener shall listen to.

## 3.2.3 Availability

The IPRs make the GCC available. As the java library does not reveal any CLIX-specific parts the GCC for CLIX is made freely available for the project partners but does not yet contain any licensing information.

The latest version of the source code can be downloaded from

- [https://svn.win.tue.nl/repos/GRAPPLE/LMSconnectors/imc\\_clix/gcc/](https://svn.win.tue.nl/repos/GRAPPLE/LMSconnectors/imc_clix/gcc/)

The latest precompiled version (jar-file) can be downloaded from:

- [https://svn.win.tue.nl/repos/GRAPPLE/LMSconnectors/imc\\_clix/gcc/src/main/resources/releases/](https://svn.win.tue.nl/repos/GRAPPLE/LMSconnectors/imc_clix/gcc/src/main/resources/releases/).

### 3.2.4 Open Issues

The underlying “castor”-XML-binding library is not yet fully capable of converting the <ELEMENT>-entity of the IMS-LIP XML scheme. Hence this mapping is not part of the conversion component.

## 3.3 How to install GCC in ELEX

### 3.3.1 ELEX Release Version

The GCC module can be installed only on eLex v. 2009 1 3617 150. An installation of eLex is available at the following url: <http://grapple.giuntilabs.com/elexGrapple/LoginSA.aspx>.

Credentials: username *user1* and password *user1*.

### 3.3.2 Installation

The GCC module for GRAPPLE is composed by GiuntiLabs.eLex.Grapple.dll, available in SVN on the Eindhoven server (<https://svn.win.tue.nl/repos/GRAPPLE/LMSconnectors/elex/gcc/>).

Usually the eLex folder contains the www folder and the ws folder. The GCC installation must to be done in the www folder, this implies the following steps:

- Update of the Web.config file
  - Insert the eLex http address as **value** in the **key** “AppBaseUrl” (e.g.: <http://grapple.giuntilabs.com/elexGrapple/>).

- In the section **httpModules** insert

```
<!-- GRAPPLE -->
<add name="UserLogin" type="GiuntiLabs.eLex.Grapple.DataModel.UserLogin,
GiuntiLabs.eLex.Grapple"/>
<!--END GRAPPLE -->
```

That feature ensures, after the first login in eLex, to load just one instance of the class able to get the event TestQuizzes.

- Insert the section

```
<!-- GRAPPLE -->
<objectFactorySettings>
<businessObjects>
  <add key="GiuntiLabs.eLex.Core.Business.Durp.UserMgr"
value="GiuntiLabs.eLex.Grapple.DataModel.Registration, GiuntiLabs.eLex.Grapple"/>
  <add key="GiuntiLabs.eLex.Core.Business.Durp.AuthenticationMgr"
value="GiuntiLabs.eLex.Grapple.DataModel.UserLogin, GiuntiLabs.eLex.Grapple"/>
  <add key="GiuntiLabs.eLex.Core.Business.PlannerMgr"
value="GiuntiLabs.eLex.Grapple.DataModel.Enrollment, GiuntiLabs.eLex.Grapple"/>
  <add key="GiuntiLabs.eLex.Core.Business.Durp.SecurityMgr"
value="GiuntiLabs.eLex.Grapple.DataModel.RoleChange, GiuntiLabs.eLex.Grapple"/>
</businessObjects>
</objectFactorySettings>
<!-- END GRAPPLE -->
```

That feature ensures to get the events **Access to a course, TestQuizzes, Registration, Student enrolment, User login, Role change, Learning activity addition, Learning activity removal, Access to a learning activity**. eLex is not able to get the event **Learning activity change**.

- Update of the **Planner/MyCourse.aspx** file
  - Insert the script able to get/send in eLex the event **Access to a course**

```
<script runat="server">
  private string AppBaseUrl()
  {
```

```

string url = null;
string eLexVirtualPath = HttpRuntime.AppDomainAppVirtualPath;

Configuration webConfiguration =
System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration(eLexVirtualPath);

if (webConfiguration.AppSettings.Settings.Count > 0)
{
    KeyValueConfigurationElement appBaseUrl =
webConfiguration.AppSettings.Settings["AppBaseUrl"];
    if (appBaseUrl != null)
    {
        url = appBaseUrl.Value;
    }
}
return url;
}

protected override void OnInit(EventArgs e)
{
    base.OnInit(e);
    try
    {
        GiuntiLabs.eLex.Common.Data.ContextData currentContext =
GiuntiLabs.eLex.Web.Code.WebContext.Current;

        GiuntiLabs.eLex.Core.Business.PlannerMgr plannerMgr =
GiuntiLabs.eLex.Core.Business.Factory.GetPlannerMgr(currentContext);

        string courseId = Request.QueryString["_courseId"];
        GiuntiLabs.eLex.Common.Data.Planner.DSCourse dsCourse =
plannerMgr.GetCourse(Int32.Parse(courseId),
GiuntiLabs.eLex.Common.Data.LearnerList.WithoutUserData,
GiuntiLabs.eLex.Common.Data.StaffList.WithoutUserData, false);

        if (dsCourse.CourseTable.Count > 0)
        {
            int version = 0;

            for (int i = 0; i < dsCourse.CourseTable.Count; i++)
            {
                GiuntiLabs.eLex.Common.Data.Planner.DSCourse.Course course =
dsCourse.CourseTable[i];
                if (course.CourseType ==
(short)GiuntiLabs.eLex.Common.Data.CourseType.DRMaterial)
                {
                    GiuntiLabs.eLex.Common.Data.DR.MaterialData materialData =
plannerMgr.GetCourseMaterial(Int32.Parse(courseId), out version);
                    GiuntiLabs.eLex.Common.Data.DR.DSMaterials.MaterialRowDataTable
materialRowDataTable = materialData.MaterialRowTable;
                    for (int j = 0; j < materialRowDataTable.Count; j++)
                    {
                        GiuntiLabs.eLex.Common.Data.DR.DSMaterials.MaterialRow
materialRow = materialRowDataTable[j];
                        if (materialRow.Type.Equals("scorm"))
                        {
                            GiuntiLabs.eLex.Grapple.DataModel.CourseAccessIMSLIP
courseAccessIMSLIP = new GiuntiLabs.eLex.Grapple.DataModel.CourseAccessIMSLIP();
                            courseAccessIMSLIP.Id = course.Id.ToString();
                            courseAccessIMSLIP.Code = course.Code;
                            courseAccessIMSLIP.Url = AppBaseUrl();
                            courseAccessIMSLIP.Title = course.BestName;
                            courseAccessIMSLIP.Topic = course.Description;
                            courseAccessIMSLIP.DateAccess =
System.DateTime.Now.ToString("s");
                            string dateTimeCourseCreated =
course["created"].ToString();
                            System.DateTime dateTime =
DateTime.Parse(dateTimeCourseCreated);
                            courseAccessIMSLIP.DateCreation = dateTime.ToString("s");

                            GiuntiLabs.eLex.Grapple.DataModel.CourseAccessThread
courseAccessThread =
new GiuntiLabs.eLex.Grapple.DataModel.CourseAccessThread(currentContext.UserNickname,
courseAccessIMSLIP);
                            System.Threading.Thread
threadStartEventGEBListenerService =
new System.Threading.Thread(new
System.Threading.ThreadStart(courseAccessThread.SetUMData));
                            threadStartEventGEBListenerService.Start();

```







## 3.4 How to install GCC in Moodle

### 3.4.1 Moodle Release version

The GCC and the GExporter (the module that sent the logs to the GEB) were developed using the version 1.9.5 of the Moodle platform. It should be possible to use every version in the 1.9 series (now the latest stable release is the 1.9.7).

### 3.4.2 Installation

Because the GCC work directly with the Moodle database, we do not provide, right now, a repository to download the GCC, but it is available on the SVN at Eindhoven (<https://svn.win.tue.nl/repos/GRAPPLE/LMSconnectors/Moodle/>).

We are going to add an administration interface to the exported data in the next months, probably implementing a new block for the Moodle infrastructure.

### 3.4.3 Availability

No IPRs restrict the usage of the GCC module for Moodle.

## 3.5 How to install GCC in Sakai

### 3.5.1 Sakai Release version & General Sakai setup:

<http://confluence.sakaiproject.org/display/BOOT/Development+Environment+Setup+Walkthrough>

### 3.5.2 Installation

#### Grapple tool requirements:

All libraries are pulled in automatically. The only required steps are the following:

Edit your sakai.properties files under tomcat/sakai/sakai.properties and append the following:

Modify the URL's so that they point to the correct Grapple Event Bus.

#### GRAPPLE Configuration

The URL where we can register our listener on the Grapple Event Bus.

grapple.gcb.registerListenerURL = <http://localhost:9000/GrappleEventBus/gcbListenerService?wsdl>

The URL where we can send events to.

grapple.gcb.sendEventURL = <http://localhost:9000/GrappleEventBus/eventGEBListenerService?wsdl>

Also, make sure your serverUrl property (usually the second one from the top) is correctly filled in.

To deploy this tool to Sakai (2.5 and 2.6) just do mvn clean install sakai:deploy in the grapple directory.

It will then copy the generated warfile to the tomcat/webapps directory and (re-)load itself.

When the GRAPPLE tool gets loaded up it will attempt to register the Sakai listener with the GEB.

When you start up tomcat there are a lot of webapps that need to load. This makes it impossible to know when tomcat has finished starting up. That is why there is a timer running that will attempt to register it every 10 seconds. The log statements should say if the registering was successful.

You can also always check the GEB to see if the listener was registered.

By default, this tool will send a request on each event it recognizes.

### **3.5.3 Availability**

No IPRs restrict the usage of the GCC module for Sakai.

## **4 Open issues**

In order to have the GCC tested properly, it is necessary to have the GRAPPLE framework in a more advanced status.

## **References**

1. Oneto L., et all.: D5.2a - Conversion models between GRAPPLE and LMSs
2. Oneto L., et all.: D7.1b – Updated specification of the operational infrastructure
3. Oneto L., et all: D7.2a – Data models and interoperability
4. <http://www.imslobal.org/profiles/index.html>