

GRAPPLE

D2.2b Version: 1.0

Tool for user event detection

Document Type	Deliverable
Editor(s):	
Author(s):	Fabian Abel, Dominikus Heckmann, Eelco Herder, Jan Hidders, Geert-Jan Houben, Daniel Krause, Erwin Leonardi, Kees van der Sluijs
Reviewer(s):	
Work Package:	WP2
Due Date:	July 31, 2010
Version:	1.0
Version Date:	November 7, 2010
Total number of pages:	29

Abstract: In this deliverable the second version of the event detection and contextualisation components that are part of the Grapple User Modelling Framework are presented. The first version was presented in Deliverable D2.2a. This follow-up deliverable particularly reports on the further developments of the framework: the conversion of IMS-LIP data into Grapple Statements and the contextualisation of user data using the Mypes service.

Keyword list: Event Detection, User Context, Aggregation, User Modeling, Mashups, User Pipes, User Observations, Logging, Tools, Mypes

Summary

This deliverable builds upon D2.2a, in which the GUMF approach for user event detection and contextualization was introduced. User events are detected and sent to GUMF by its clients (LMSs, GALE, further event detection tools). Contextualisation components gather information from other sources, which is used for enriching the data to make it suitable as an input for user modelling. This deliverable focuses on two important aspects of the contextualisation process. First, the framework was extended with functionality for syndicating events from LMSs in IMS LIP format. IMS LIP statements are mapped to one or more Grapple Statements, which allow for the reasoning processes necessary for reasoning and contextualisation. Next functionality was added for the aggregation of user profiles and activity data from the social Web by means of the Mypes component. The Mypes aggregation functionality has been evaluated with respect to accuracy and performance. Experimental results show the benefits of aggregated profile and activity data for enhancing profiles that typically exist in single systems, such as LMSs.

Authors

Person	Email	Partner code
Fabian Abel	abel@l3s.de	LUH
Dominikus Heckmann	heckmann@dfki.de	DFKI
Eelco Herder	herder@l3s.de	LUH
Jan Hidders	a.j.h.hidders@tudelft.nl	TUD
Geert-Jan Houben	g.j.p.m.houben@tudelft.nl	TUD
Daniel Krause	krause@l3s.de	LUH
Erwin Leonardi	e.leonardi@tudelft.nl	TUD
Kees van der Sluijs	k.a.m.sluijs@tue.nl	TUE

Table of Contents

SUMMARY	2
AUTHORS	2
TABLE OF CONTENTS	3
TABLES AND FIGURES.....	4
LIST OF ACRONYMS AND ABBREVIATIONS	4
1 INTRODUCTION	5
1.1 Task and Deliverable Description	6
1.2 Outline of this Deliverable	6
2 RELATED WORK	6
3 EVENT DETECTION AND CONTEXTUALISATION FRAMEWORK	7
3.1 Modelling Events and Context	7
3.2 Event Detection and Contextualisation within GUMF	8
3.3 Summary of other Event Detection Tools	9
4 SYNDICATING USER EVENTS FROM GRAPPLE ENVIRONMENT	10
4.1 Listening for user events coming from LMSs	10
4.2 IMS-LIP Vocabulary Concepts.....	11
4.3 Conversion from IMS-LIP to Grapple statements.....	13
4.3.1 Student Enrollment	13
4.3.2 Role change.....	14
4.3.3 Tests/quizzes.....	15
4.4 Synopsis.....	17
5 CONTEXTUALISING USER EVENTS WITH DATA FROM THE SOCIAL WEB	17
5.1 Profile Aggregation and Enrichment with Mypes.....	17
5.2 Format of Aggregated Profiles	19
5.3 Accuracy and Performance of GUMF	21
5.4 Benefits of Profile and Activity Aggregation	23
5.4.1 Benefits of Profile Aggregation	23
5.4.2 Benefits of Activity Aggregation.....	25

5.4.3 Summary of Findings..... 26

6 CONCLUSIONS..... 26

REFERENCES 27

Tables and Figures

List of Figures

Figure 1 - GUMF Architecture 9

Figure 2 - Example Setup: plug-ins P1, P2, and P3 are used for contextualising the data that is available in dataspace D1. Therefore, the plug-ins might access other services, dataspaces, and the Web. 9

Figure 3 - Listening for IMS-LIP-formatted user events 11

Figure 4 - Linkage, Aggregation, Alignment and Enrichment of Public User Profiles with GUMF 18

Figure 5 - Aggregated profile visualised in Mypes. 20

Figure 6 - Visualization of tag-based profile (filtered by location) on a Google map..... 21

Figure 7 - Precision of semantic enrichment with WordNet categories..... 22

Figure 8 - Average time (in milliseconds on a logarithmic scale) required for obtaining tag-based and traditional profiles and the corresponding standard deviation. 22

Figure 9 - Completing service profiles with aggregated profile data 24

Figure 10 - Entropy and self-information of service-specific profiles in comparison to the aggregated Mypes profiles. 25

List of Tables

Table 1 - Grapple Core (gc:) constructs for the formal representation of (user) observations..... 8

Table 2 - IMS-LIP RDF vocabulary..... 13

List of Acronyms and Abbreviations

ALE	Adaptive Learning Environment
FOAF	Friend of a Friend
GRAPPLE	Generic Responsive Adaptive Personalised Learning Environment
Grapple Statement	Specific statements about a user → a user profile is a set of GRAPPLE statements
GUMF	Grapple User Modelling Framework
GUMO	General User Model Ontology
LMS	Learning Management System
LOM	Learning Object Metadata
OWL	Ontology Web Language
RDF	Resource Description Framework
SCORM	Sharable Content Object Reference Model

SUMO	Suggested Upper Merged Ontology
TEL	Technology-Enhanced Learning
UM	User Model (or sometimes “User Modelling”) a set of Grapple Statements
UMF	User Model Framework
UserML	User Model Markup Language
UserQL	User Model Query Language
WP	Work Package
WPL	WP Leader

1 Introduction

User modelling starts with the observation of users, i.e. collecting data about users. In traditional human-computer interaction, this data is event-based: Whenever a user performs some *activity* (e.g., a mouse click) then – from the system’s perspective – this is interpreted as an *event* performed by the user. To interpret a user event a system needs to know in which (application) context the event happened. For example, the knowledge about a mouse click is rather useless if one does not know in which application and for what purpose the click was done. Hence, event detection requires some sort of *contextualisation* of the events to provide a useful and meaningful basis for user modelling.

User events can be modelled at different levels of granularity. A series of low-level events such as clicks and keyboard inputs can be interpreted as an event as well. For example, if a user bookmarks a Web page then she possibly first enters name, description and tags to describe the page before she finally saves the bookmark. Modelling user events is heavily application-specific. In Grapple, where students are using learning management systems (LMS), user events are mainly related to e-learning, e.g. participating in a course, answering a question in a test, finishing a test, etc. However, different learning management systems still have the opportunity to model similar types of events in a different way: while one LMS might model the completion of a test by simply summarising the percentage of correctly answered questions, another LMS might rate the performance of the student in the test by assigning a grade from A to F.

The goal of the Grapple User Modelling Framework (GUMF) is to provide flexible event detection functionality, which allows for contextualisation and semantic enrichment of user events resulting in a meaningful collection of user data that can be exploited for user modelling. Events of interest to GUMF are not only those that are performed in a LMS but also those events that are performed in neighbouring applications. For example, for modelling the interests of a student it might be interesting and worthwhile to detect – in addition to courses the student participates in – the kind of bookmarks the student created in her bookmarking tool. Event data fusion, which includes the functionality to map and align data originating from different sources, is therewith an important feature to be provided by GUMF.

In Deliverable 2.2a the approach to and the initial framework for user event detection and contextualisation was introduced. Events are modelled as Grapple (Observation) Statements. Clients can report statements via the Java Client API or the Grapple Event Bus. Further statements can be derived from the Desktop Logger Tools and other Web 2.0 sources. These further statements are used for contextualising and enhancing available data. It also showed how data can be mapped and synchronised using SILK and Yahoo Pipes.

This deliverable concentrates on how Grapple-specific user events are syndicated. In the integration WP7 it was decided that the Learning Management Systems will use IMS-LIP for exchanging learning events. A mapping tool has been developed that transforms XML-formatted IMS-LIP data into Grapple Statements, which allow for reasoning about the data (more details on reasoning can be found in deliverable D2.3).

A common problem in LMSs is that the user population and the number of user activities is too small for reliably reasoning about the data. Moreover, user profile information is often limited, as most LMSs do not require students to fill out complete profiles. Over the past few years, LMSs have embraced functionality for communication and collaboration and several learning institutions report that they currently use Web 2.0 tools [35]. A service called Mypes, has been implemented that makes use of Social Web aggregation to enable end-users to directly benefit from GUMF’s profile aggregation and contextualisation.

1.1 Task and Deliverable Description

The task to be performed by WP2 in the second and third period of the Grapple project aims at the development of an event detection and contextualisation framework. The event detection components, which are part of the Grapple user modelling framework, contextualise users' events (e.g. selecting a link, providing textual input, or other interactions with standard user interface elements). The contextualisation of events happens by

- a) embedding the event within the semantic descriptions of the source page or object and its user interface elements
- b) detecting the current navigation or interaction pattern to which the event belongs. A navigation pattern may consist of a set / sequence of pages which belong together, e.g. are navigated within a certain period of time or are thematically related or have frequently been used together by this user or by others, etc.

Navigation patterns will be identified by mining (a) the individual navigation paths, and (b) the collaborative navigation behaviour. All captured information about a user event will be stored in an ontology-based format.

1.2 Outline of this Deliverable

Section 2 summarises related work, which is discussed in more detail in D2.2a. From the GUMF perspective, event detection is a data aggregation and data fusion problem. User event data from different sources has to be mashed up and contextualised to be usable for the user modelling components of GUMF.

Section 3 summarises the Event Detection and Contextualisation framework and explains the structure of Grapple Observation Statements.

Section 4 deals with the syndication of user events from the Grapple Environment. An important aspect of the syndication process is the conversion of IMS-LIP data to Grapple Statements. After introducing the IMS-LIP vocabulary the conversion process is explained in detail.

Contextualisation of user events, with data from the Social Web, is described in Section 5. After introducing the Mypes aggregation service, an initial evaluation of the benefits of profile aggregation, accuracy of semantic enrichment and runtime analysis is presented.

The deliverable concludes with a summary of the work.

2 Related Work

The event detection and contextualisation framework that is embedded into the Grapple user modelling framework (GUMF) is highly inspired by the principles of mashups and focuses on the aggregation, intermixture and alignment of event data originating from different applications. This section briefly summarises related work on mashups and intertwining data from different sources, tools that support event detection and contextualisation and approaches for aligning event data that adhere to different schemata. For more details on related work please refer to Deliverable 2.2a.

Connecting data from different sources and services is in line with today's Web 2.0 trend of creating mashups of various applications [36]. Support for the development of interoperable services is provided by initiatives such as the data-portability project¹, standardisation of APIs (e.g. OpenSocial) and authentication and authorisation protocols (e.g. OpenID, OAuth), as well as by (Semantic) Web standards such as RDF, RSS and specific Microformats². It also gets easier to connect distributed user profiles including social connections due to the increasing take-up of standards like FOAF [3], SIOC³, or GUMO [4]. Recently, several mashup editors for building new mashups applications have become available. Examples are Yahoo! Pipes [29], IBM Damia [30], Microsoft Popfly [6], and Intel Mash Maker [32].

Conversion approaches allow for flexible user modelling [37]. GLUE [27] is a semi-automatic tool for generating ontology mapping using machine learning technique. It focuses on finding a one-to-one mapping between concepts of two given ontologies. In other words, for each concept in one ontology, it finds the *most similar* corresponding concept in another ontology. It consists of three main modules: *Distribution Estimator*,

¹ <http://www.dataportability.org/>

² <http://microformats.org/>

³ <http://rdfs.org/sioc/spec/>

Similarity Estimator, and *Relaxation Labeller*. The Silk framework [16] is a tool for finding relationships between entities within Linked Data sources.

Solutions for user identification form the basis for personalisation across application boundaries [38]. Google's Social Graph API⁴ enables application developers to obtain the social connections of an individual user across different services. Generic user modelling servers such as CUMULATE [39] or PersonIs [40] as well as frameworks for mashing up profile information [41] appear that facilitate handling of aggregated user data.

3 Event Detection and Contextualisation Framework

The event detection and contextualisation framework is embedded into the Grapple User Modelling Framework (GUMF). Deliverable 2.2a introduces the framework in detail. This section summarises the most important characteristics of the approach to event detection and contextualisation. The event and context model (see Section 3.1) are recapitulated and report on the architecture of the framework (see Section 3.2). A short summary of the event detection and contextualisation applications introduced in Deliverable 2.2a is also presented.

3.1 Modelling Events and Context

Deliverable 2.1, proposes a model that allows for capturing events and context by means of special Grapple statements, where the general structure of an observation can be phrased as follows.

ID of observation, reporter, time of reporting: Who – did what – when – on something.

The core user event is given by “who did what” and is enriched with some basic context information such as the time (i) when the event was observed and (ii) when the observation was reported⁵, a reference to the object, on which the user performed some activity (“on something”) and a pointer to the application that reported the event (“reporter”). An observation statement can be serialised with (special) Grapple statement constructs. Table 1 lists these constructs and explains their meaning.

construct	represents	type	range	description
gc:Observation	The entire observation.	Class	n.a.	A gc:Observation is a specialisation of a Grapple statement (gc:Statement). Mandatory basic properties: gc:ID, gc:creator, gc:created, gc:event Optional basic properties: gc:subject, gc:when, gc:event, gc:object
gc:ID	ID of observation	Property	(as in gc:Statement)	cf. gc:ID of Grapple statements
gc:creator	reporter	Property	String (URI)	The reporter of the event is put into the gc:creator property of a Grapple statement. Ideally, the reporter is specified via a URI that allows GUMF to unambiguously identify the creator.
gc:created	time of reporting	Property	Literal (xsd:dateTime)	The time the event was reported should be encoded according to the W3C-DTF profile [18].
gc:subject	who	Property	String (URI)	In general, the component “who” is represented by the gc:subject construct. Ideally, it is referenced by a URI. If the actor of an observation can be characterised more precisely, then

⁴ <http://socialgraph.apis.google.com>

⁵ Note that observations might be reported not at the same time when the observation is done, e.g. observations can be reported within a batch operation.

				specialisations of gc:subject can be used, e.g. if a user is the actor of an observation then gc:user can be applied.
gc:event	did what	Property	String (URI)	The “did what” component is represented by gc:event, which is a specialisation of the gc:predicate, i.e. the predicate of a Grapple statement. It refers to a property that describes the user activity. Usually, the semantics of the property should be defined in a (domain) ontology. The value of gc:event can, for example, be http://www.grapple-project.org/nop.owl#performed , which means that the user performed some NOP event.
gc:object	on something	Property	String	The value might be a URI that points to the object on which the activity was performed and which might also describe the observation in more detail.
gc:when	when	Property	Literal (xsd:dateTime)	The time the event was observed. It should be encoded according to the W3C-DTF profile [18].
				18

Table 1 - Grapple Core (gc:) constructs for the formal representation of (user) observations.

3.2 Event Detection and Contextualisation within GUMF

Figure 1 shows the GUMF architecture as presented in Deliverable 6.1b. Events are detected by client applications (GALE, LMS, general Web applications, event detection tools – see next subsection). Via the Client API (Grapple Event Bus, Java API or SOAP) the events are submitted to GUMF.

The event storage and contextualisation components are integrated into GUMF by means of dataspace, which logically bundle user data (see Deliverable 2.1), and plug-ins, which allow for modifying/enriching the behaviour of dataspace. A dataspace has only one owner, but administrators can decide to share dataspace – for example in order to exchange learner progress between LMSs.

Reasoning plug-ins make use of Grapple Derivation Rules (GDR), which are described in more detail in D2.3. Reasoning may be limited to simple scale conversion, but GDR also provides various options for contextualisation (e.g., localisation) and semantic enhancement (for example by linking keywords to WordNet or Wikipedia). A typical setup is displayed in Figure 2.

The dataspace approach allows for flexible event detection and contextualisation: the administrator of a dataspace can activate or deactivate plug-ins according to the requirements of the client applications that are allowed to access the dataspace (cf. Figure 6 in Deliverable 2.2a). The plug-in principle gives developers of plug-ins as well as the actual clients of GUMF a lot of freedom to implement and adjust the contextualisation of event data so that GUMF, in general, provides a generic solution for event detection and contextualisation.

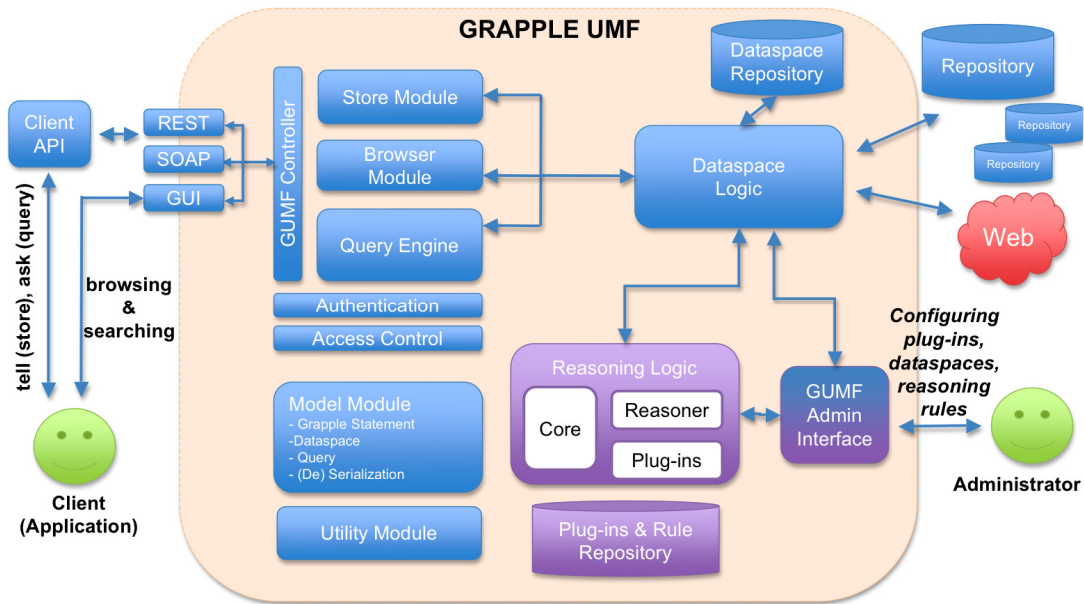


Figure 1 - GUMF Architecture

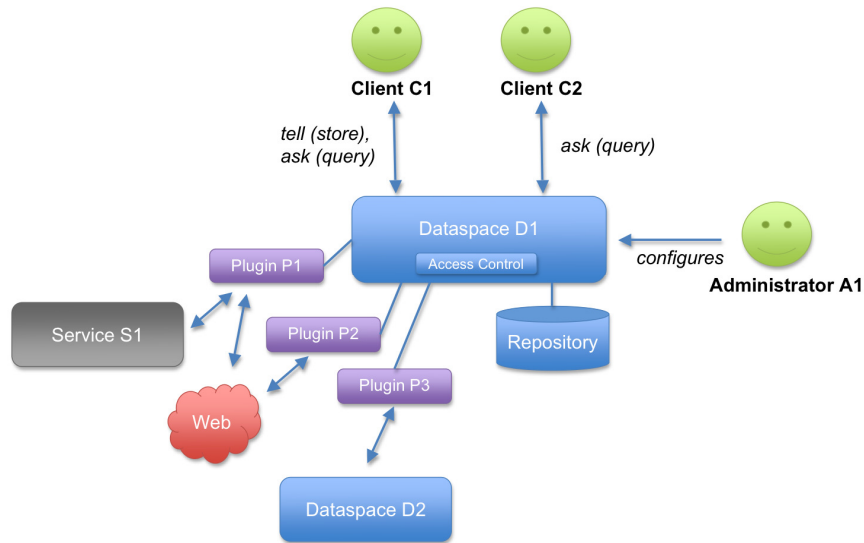


Figure 2 - Example Setup: plug-ins P1, P2, and P3 are used for contextualising the data that is available in dataspace D1. Therefore, the plug-ins might access other services, dataspace, and the Web.

3.3 Summary of other Event Detection Tools

As mentioned earlier, client applications are responsible for detecting user events and reporting these events to GUMF. These client applications include GALE and other Learning Management Systems, but may be any kind of other (Web) application.

In order to be able to obtain more (detailed) user event data – to complement or enrich the data gathered by LMSs or other client applications –, more event detection strategies have been developed and which already have been described in Section 5 of Deliverable 2.2a. Among these tools are the following.

1. **User observation reported via Desktop Logging Tools:** A so-called user observation hub has been implemented by means of a GUMF plug-in that receives user activity data as sent by the

DragonTalk⁶ browser plug-in. Such event data (e.g., information about bookmarks, Web sites visited by the user) is transformed into Grapple statements using the NOP ontology⁷ as domain vocabulary. Finally, the Grapple statements are reported to the corresponding GUMF dataspace.

2. **Mashing up user data with SILK:** A tool has been implemented that aggregates data from different sources and moreover aligns possibly heterogeneous vocabularies by using the SILK framework [25], a framework that facilitates the linkage of heterogeneous, distributed data. A showcase was implemented aggregating social networking data from Orkut⁸ and Facebook. Experiments with different similarity measures have been performed to map between the two different vocabularies used by the two services and to deploy an interface that generates so-called "SILK descriptions" that are applied to enforce vocabulary mappings.
3. **User Pipes:** Yahoo pipes [29] and Semantic Web pipes [14] have been applied to connect and interweave distributed user data. These tools have been exploited to combine user activity streams with other data streams coming from social bookmarking services such as Delicious to deduce new user profile information such as user interests.

A fourth event detection tool will be presented in Chapter five of this deliverable: the Mypes service. Mypes gathers information on a learner's profile data and activities from the social Web. As will be shown, this data is useful for complementing and enriching the (often limited) learner (activity) data that is typically available in LMSs.

4 Syndicating User Events from GRAPPLE environment

Many user activities that happen within the Grapple environment are performed in the Learning Management Systems. For example, users access courses or perform tests and quizzes in the LMSs. The LMSs report these user activities to GUMF via the Grapple Event Bus.

Information such as student enrolments or the result a student achieved in some test/quiz is formatted in IMS-LIP⁹. The decision to use IMS-LIP is reported in D7.2b and is mainly motivated by the fact that the LMS partners are familiar with this standard. However, due to a more complex syntax, IMS-LIP statements cannot be directly stored as Grapple Observation Statements. Additionally, due to the expressiveness of IMS-LIP and the variety of statements, IMS-LIP statements are not suitable as input for reasoning plug-ins.

In order to facilitate the exchange of user events between LMSs, a mapping tool has been developed that transforms the XML-formatted IMS-LIP data into Grapple statements and then defines an RDF vocabulary that models the IMS-LIP concepts¹⁰ (Namespace: <http://www.grapple-project.org/ims-lip/>, Namespace abbreviation: *ims*).

The next section gives an overview on how GUMF listens for user events coming from LMSs. Section 4.2 gives an overview of the relevant IMS-LIP concepts and how they are taken into account in Grapple Observation Statements. Section 4.3 provides a number of examples on how IMS-LIP statements are transformed into Grapple Statements.

4.1 Listening for user events coming from LMSs

Figure 3 depicts the GEB listener service that monitors the Grapple Event Bus for user events that occur in the different LMSs. Whenever a new student registers, joins/leaves a course, finishes a test or performs some other learning activity in an LMS, the LMS sends an IMS-LIP-formatted message to the GEB containing the user ID as well as a description of the learning activity performed by the student. GEB notifies the GEB-Listener, which can be interpreted as a type of IMS-LIP activity sensor, and forwards the

⁶ <http://dragontalk.opendfki.de/>

⁷ RDF Schema of NOP ontology:
<https://usercontext.opendfki.de/repos/trunk/UserObservationHub/src/de/dfki/km/usercontext/userobservationhub/vocabulary/nop.rdfs>

⁸ <http://www.orkut.com/>

⁹ <http://www.imsglobal.org/profiles/>

¹⁰ <http://pcwin530.win.tue.nl:8080/grapple-umf/rdf/ims-lip-rdf.owl>

corresponding event to the listener. GUMF expects user data to be formatted as Grapple statements, i.e. the listener has to transform the IMS-LIP data into Grapple statements.

A single IMS-LIP data message might produce several Grapple statements, for example, if the IMS-LIP message specifies that a set of roles have been assigned to the student in the context of a particular course the mapping tool will generate a Grapple statement for each of these roles. Finally, the GEB listener sends the Grapple statements that represent the IMS-LIP message to GUMF using GUMF's client API (see Figure 1).

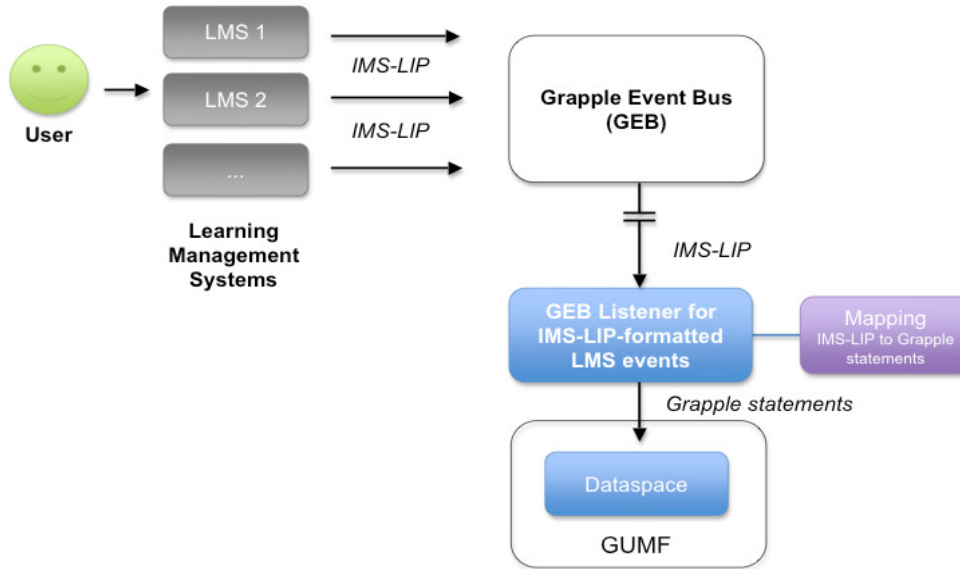


Figure 3 - Listening for IMS-LIP-formatted user events

4.2 IMS-LIP Vocabulary Concepts

The main concepts of the RDF-based IMS-LIP vocabulary are listed in Table 2 (a short summary of this vocabulary is also presented in Deliverable 7.2c). The usage column explains how the vocabulary concept is applied. There is differentiation between

- Properties that will be used as value of *gc:predicate* (*value of gc:predicate*). These are the properties that IMS LIP uses for representing an *event* (e.g. accessed) or some *information* about a user (e.g. name, city)
- Properties that provide additional information about a certain event or a statement about a user (for example a course Id). These properties are *metadata* that can simply be attached to a Grapple statement, i.e. domain of the property is *gc:Statement* while the range is usually some string (*gc:Statement metadata*).

property (UM variable)	Description	usage
accessed	User accessed a course (or some other resource): accessed	value of gc:predicate
attends	User attends a course (the value of this property should be the ID of the course the user attends)	value of gc:predicate
changedRoleTo	User changed role (property): Property that describes the role of the student in context of a specific context.	value of gc:predicate
completedTest	User completed/finished a test or quiz	value of gc:predicate
numberOfAttempts	Number of attempts a user required for/within a test or quiz.	gc:Statement metadata or value of gc:predicate
courseId	ID of a course	gc:Statement metadata

courseTitle	name/title of a course	gc:Statement metadata
descriptionLong	Long description (<description><long> long description </long></description>))	gc:Statement metadata
descriptionShort	Short description (<description><short> short description </short></description>)	gc:Statement metadata
addedLearningActivity	user added a learning activity	value of gc:predicate
changedLearningActivity	user changed a learning activity	value of gc:predicate
removedLearningActivity	a learning activity was removed for the user (e.g., by the user)	value of gc:predicate
activityId	ID of a learning activity	gc:Statement metadata
activityTitle	title/name of a learning activity. This property is a sub-property of ims:title.	gc:Statement metadata
courseId	ID of a course	gc:Statement metadata
courseTitle	title/name of a course. This property is a sub-property of ims:title.	gc:Statement metadata
title	The title of something (e.g., the title of a course). This property extends dc:title.	gc:Statement metadata or value of gc:predicate
topic	The topic of something (e.g., the topic of a course).	gc:Statement metadata or value of gc:predicate
studentEnrollment	This property states that a student enrolled a course, e.g. the gc:subject refers to the student/user and the gc:object refers to the course.	value of gc:predicate
name	the (full) name of a user	value of gc:predicate
firstName	the first/given name of a user	value of gc:predicate
lastName	the last name of a user	value of gc:predicate
email	the email of a user	value of gc:predicate
phone	the phone of a user	value of gc:predicate
gender	the gender of a user	value of gc:predicate
birthDate	the date of birth of a user	value of gc:predicate
placeOfBirth	the place where the user was born	value of gc:predicate
ipAddress	the IP address of a user	value of gc:predicate
language	the preferred language of a user	value of gc:predicate
streetName	a postal address (of the user): street name	value of gc:predicate
streetNumber	a postal address (of the user): street number	value of gc:predicate
region	a postal address (of the user): region	value of gc:predicate
city	a postal address (of the user): city	value of gc:predicate
country	a postal address (of the user): country	value of gc:predicate
postcode	a postal address (of the user): postcode	value of gc:predicate
pobox	a postal address (of the user): PO Box	value of gc:predicate
organization	the organisation a user is associated with	value of gc:predicate

Table 2 - IMS-LIP RDF vocabulary

4.3 Conversion from IMS-LIP to Grapple statements

There is not just one generic way to convert IMS-LIP statements to Grapple statements. On the one hand, this is due to the expressiveness of IMS-LIP, on the other hand this is due to the specific way in which LMSs may interpret the IMS-LIP specification. Therefore, conversions are defined manually for the most common usages.

The subsequent sections list examples that illustrate how the conversion between IMS-LIP data – sent by the LMS – and Grapple statements – that will be stored in GUMF – is done (cf. also Deliverable 7.2c). More examples are available via the GUMF help pages¹¹.

4.3.1 Student Enrollment

IMS-LIP data that describes that a student/user enrolled to a course is transformed into a single Grapple statement using the `ims:studentEnrollment`.

IMS-LIP:

```
<?xml version="1.0" encoding="UTF-8"?>
<statement>
<origin>
<learnerinformation xml:lang="en" xmlns="http://www.imsglobal.org/xsd/imslip_v1p0">
  <comment xml:lang="en">Student enrollment</comment>
  <securitykey>
    <keyfields>
      <fieldlabel>
        <typename>
          <tyvalue xml:lang="en">User Id</tyvalue>
        </typename>
      </fieldlabel>
      <fielddata>aclixlearn</fielddata>
    </keyfields>
  </securitykey>
  <activity>
    <contenttype>
      <referential>
        <sourcedid>
          <source>LMS-CLIX-ID</source>
          <id>117359</id>
        </sourcedid>
      </referential>
    </contenttype>
  </activity>
  <identification>
    <ext_identification>Learner</ext_identification>
  </identification>
</learnerinformation>
</origin>
</statement>
```

Corresponding Grapple statement:

```
<gc:Statement rdf:about="http://www.grapple-project.org/umf/10505">
  <gc:subject>aclixlearn</gc:subject>
  <gc:predicate
    rdf:resource="http://www.grapple-project.org/ims-lip/studentEnrollment"/>
  <gc:object rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
    117359
  </gc:object>
  <gc:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
    2010-03-10T16:23:04.243+01:00</gc:created>
  <gc:creator>CLIX</gc:creator>
</gc:Statement>
```

Important parts of the corresponding Grapple statement:

- `gc:subject`: the user who was enrolled for some course
- `gc:predicate`: `studentEnrollment` denotes that the user enrolled for a course
- `gc:object`: the course ID the id is only taken from the `sourcedid` not the `source`)

¹¹

<http://pcwin530.win.tue.nl:8080/grapple-umf/help/ims-lip-rdf.html>

- ge:context / gc:spatial: if available then the ge:context and gc:spatial attributes describe in which context the user enrolled - usually the URL of the LMS instance (in the above example no context was available in the IMS-LIP data)

4.3.2 Role change

The user's role is changed: ims:changedRoleTo.

IMS-LIP:

```
<?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">
      <comment xml:lang="en">Role change</comment>
      <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>student3</fielddata>
        </keyfields>
      </securitykey>
      <identification>
        <comment>Role</comment>
        <ext_identification>Learner</ext_identification>
      </identification>
      <identification>
        <comment>Role</comment>
        <ext_identification>User</ext_identification>
      </identification>
    </learnerinformation>
  </origin>
</statement>
```

The IMS-LIP is transformed into two Grapple statements (one statement for each role):

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:gc="http://www.grapple-project.org/grapple-core/"
  xmlns:ge="http://www.grapple-project.org/grapple-extended/"
  xmlns:ims="http://www.grapple-project.org/ims-lip/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <gc:Statement rdf:about="http://www.grapple-project.org/grapple-core/id-not-set-yet-0">
    <gc:subject>student3</gc:subject>
    <gc:predicate
      rdf:resource="http://www.grapple-project.org/ims-lip/changedRoleTo"/>
    <gc:object>Learner</gc:object>
    <gc:creator>ELEX</gc:creator>
    <gc:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2010-03-11T08:12:50.100+01:00
    </gc:created>
    <gc:spatial rdf:resource="http://localhost:3916/www/">
    <ge:context rdf:resource="http://localhost:3916/www/">
  </gc:Statement>

  <gc:Statement rdf:about="http://www.grapple-project.org/grapple-core/id-not-set-yet-1">
    <gc:subject>student3</gc:subject>
    <gc:predicate
      rdf:resource="http://www.grapple-project.org/ims-lip/changedRoleTo"/>
    <gc:object>User</gc:object>
    <gc:creator>ELEX</gc:creator>
    <gc:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2010-03-11T08:12:50.147+01:00
    </gc:created>
    <gc:spatial rdf:resource="http://localhost:3916/www/">
    <ge:context rdf:resource="http://localhost:3916/www/">
  </gc:Statement>
```

</rdf:Description>

Important parts of the corresponding Grapple statements:

- gc:subject: the user for whom the role changes in some context
- gc:predicate: changedRoleTo denotes that the user changed his role to...
- gc:object: the actual role of the user
- ge:context / gc:spatial: describes in which context the user has the given role (here: <http://localhost:3916/www/>)

4.3.3 Tests/quizzes

A student/user completed a test/quiz: *ims:completedTest*.

IMS-LIP:

```
<?xml version="1.0" encoding="UTF-8"?>
<learnerinformation xml:lang="en" xmlns="http://www.imslobal.org/xsd/imslip_v1p0">
  <comment xml:lang="en">Tests/quizzes</comment>
  <securitykey>
    <keyfields>
      <fieldlabel>
        <typename>
          <tyvalue xml:lang="en">UserName</tyvalue>
        </typename>
      </fieldlabel>
      <fielddata>116004</fielddata>
    </keyfields>
  </securitykey>
  <activity>
    <contenttype>
      <referential>
        <sourcedid>
          <source>LMS-CLIX-ID</source>
          <id>117359</id>
        </sourcedid>
      </referential>
    </contenttype>
    <evaluation>
      <date>
        <typename>
          <tysource sourcetype="list">
            AccessDate,CreationDate,StartDate,StopDate,BirthDate</tysource>
          <tyvalue xml:lang="en">StartDate</tyvalue>
        </typename>
        <datetime>2009-10-14T10:07:47.774+02:00</datetime>
      </date>
      <date>
        <typename>
          <tysource sourcetype="list">
            AccessDate,CreationDate,StartDate,StopDate,BirthDate</tysource>
          <tyvalue xml:lang="en">StopDate</tyvalue>
        </typename>
        <datetime>2009-10-14T10:07:55.430+02:00</datetime>
      </date>
      <noofattempts>8</noofattempts>
      <result>
        <score>
          <fieldlabel>
            <typename>
              <tyvalue xml:lang="en">Total</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>30.0</fielddata>
        </score>
        <interpretscore>
          <fieldlabel>
            <typename>
              <tyvalue xml:lang="en">MinScore</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>0</fielddata>
        </interpretscore>
        <interpretscore>
          <fieldlabel>
            <typename>
              <tyvalue xml:lang="en">MaxScore</tyvalue>
            </typename>
          </fieldlabel>
          <fielddata>30.0</fielddata>
        </interpretscore>
        <interpretscore>
          <fieldlabel>
```

```

        <typename>
          <tyvalue xml:lang="en">Threshold</tyvalue>
        </typename>
      </fieldlabel>
      <fielddata>60.0%</fielddata>
    </interpretscore>
  </result>
  <description>
    <short xml:lang="en">We want to learn something about the Force</short>
  </description>
</evaluation>
<description>
  <short xml:lang="en">Final test for becoming a Jedi Knight.</short>
  <long xml:lang="en">star_wars jedi_knight the_force</long>
</description>
</activity>
</learnerinformation>

```

Corresponding Grapple statement:

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:gc="http://www.grapple-project.org/grapple-core/"
  xmlns:ge="http://www.grapple-project.org/grapple-extended/"
  xmlns:ims="http://www.grapple-project.org/ims-lip/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <gc:Statement rdf:about="http://www.grapple-project.org/grapple-core/id-not-set-yet-0">
    <gc:subject rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
      116004
    </gc:subject>
    <gc:predicate
      rdf:resource="http://www.grapple-project.org/ims-lip/completedTest"/>
    <gc:object rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
      117359
    </gc:object>
    <gc:level rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
      30.0
    </gc:level>
    <ims:numberOfAttempts rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
      8
    </ims:numberOfAttempts>
    <gc:creator>CLIX</gc:creator>
    <gc:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2010-03-15T10:48:31.726+01:00
    </gc:created>
    <ge:levelRange>0-30.0</ge:levelRange>
    <ge:levelRangeMin rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
      0
    </ge:levelRangeMin>
    <ge:levelRangeMax rdf:datatype="http://www.w3.org/2001/XMLSchema#double">
      30.0
    </ge:levelRangeMax>
    <ge:levelRangeThreshold>60.0%</ge:levelRangeThreshold>
    <ims:title>We want to learn something about the Force</ims:title>
    <ge:temporalStart rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2009-10-14T10:07:47.774+02:00
    </ge:temporalStart>
    <ge:temporalEnd rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">
      2009-10-14T10:07:55.430+02:00
    </ge:temporalEnd>
  </gc:Statement>

```

Important parts of the corresponding Grapple statement:

- gc:subject: the user who finished/completed the test or quiz
- gc:predicate: completedTest denotes that the user completed a test of quiz
- gc:object: the ID of the test or some ID that enables people to deduce which test/quiz the user completed (only the id from the sourcedid is taken, not the source)
- gc:level: the score the student/user obtained in the test or quiz
- ims:numberOfAttempts: the number of attempts the user required in/for the test or quiz
- ge:levelRange: the range of the level (gc:level) as string
- ge:levelRangeMin: the minimum value the level (gc:level) could have
- ge:levelRangeMax: the maximum value the level (gc:level) could have
- ge:levelRangeThreshold: a threshold level (gc:level), e.g. the level that is required to pass the test or quiz successfully
- ge:temporalStart and ge:temporalEnd: when the test or quiz was started/finished

- ims:title: the title of the test/quiz

4.4 Synopsis

The IMS-LIP user event data is important for the user modelling and user profile reasoning provided by GUMF. For example, events that report about the results a student achieved in a quiz or test can be used to adjust the student's knowledge level regarding the concepts that are related the quiz/test. Therefore, the ID of the test is essential as it is required to obtain the concepts related to the quiz/test. User events about joining and leaving a course allow GUMF to deduce how the interests of a student change. Deliverable 2.3 details the corresponding user profile reasoning strategies.

5 Contextualising User Events with Data from the Social Web

A common problem in LMSs is that the user population and the number of user activities is too small for reliably reasoning about the data. Moreover, user profile information is often limited, as most LMSs do not require students to fill out complete profiles: hence, LMSs are often not sufficiently aware of the learner's interests, background knowledge and contacts with peer learners.

Over the past few years, LMSs have embraced functionality for communication and collaboration (groupwork, networking, portfolio); 50% of learning institutions report that they currently use Web 2.0 tools: social networking (74%), Wiki (68%), blog (66%) and chat (47%) (Herder and Marenzi, 2010). This development makes it more likely that activities in the social Web (forum messages, blogs, chats, tweets, messages to friends) are relevant sources for inferring learner interests, background knowledge and contacts.

This chapter presents a GUMF client application for profile aggregation and enrichment making use of Social Web data. The next section provides a high-level overview of this application, which is called Mypes. Section 5.2 explains the format of aggregated user profiles in more detail. Section 5.3 presents an evaluation of the accuracy and performance of Mypes. Section 5.4 discusses the benefits of aggregating data from different applications.

5.1 Profile Aggregation and Enrichment with Mypes

Mypes is a GUMF client application that allows the aggregation of form-based as well as tag-based profiles. Mypes features include linkage, alignment, and enrichment of distributed user profile data. Mypes supports the task of gathering information about users for user adaptive systems and aims to provide a uniform interface to public profile data distributed on the Social Web. Such an interface is valuable for casual users, who would like an overview of their distributed profile data, as well as systems that require information about their users. Learning Management Systems can exploit the data gathered by Mypes as (complementary) user data.

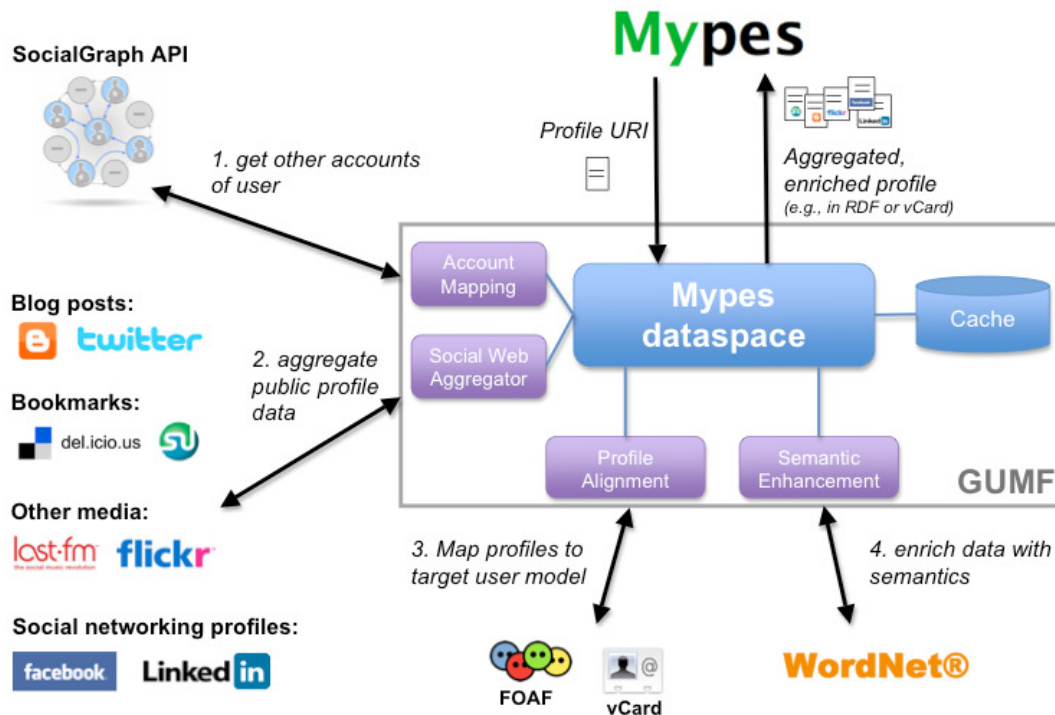


Figure 4 - Linkage, Aggregation, Alignment and Enrichment of Public User Profiles with GUMF

The Grapple User Modelling Framework aims to provide a uniform interface to user data that might be distributed on the Web. To feature access to distributed user data and to align and enhance the data, GUMF and the corresponding GUMF components depicted in Figure 4 respectively perform the following steps.

1. Account Mapping. A user's URI of an online account will give the account mapping plug-in the possibility to gather other online accounts of the same user by exploiting the Google Social Graph API¹², which provides such mappings for all users who linked their accounts via their Google profile, for example:

```
"http://www.google.com/profiles/fabian.abel": "claimed_nodes": [
  "http://delicious.com/fabianabel",
  "http://fabianabel.stumbleupon.com",
  "http://www.last.fm/user/fabianabel/", ...]
```

For those users where no mappings can be obtained via API, it is possible to provide appropriate mappings by hand. The account mapping module finally provides a list of online accounts that are associated with a particular user.

2. Social Web Aggregator. For the URIs associated to the user, the aggregator module gathers profile data from the corresponding services. In particular, traditional profile information (e.g., name, homepage, location, etc.), tag-based profiles (tagging activities), and posts (e.g., bookmark postings, blog posts, picture uploads) are harvested from nine different services as depicted in the table below.

¹² <http://code.google.com/apis/socialgraph/>

traditional profile attributes	Facebook	LinkedIn	Twitter	Blogspot	Flickr	Delicious	Stumble Upon	Last.fm	Google
nickname	x	x	x	x	x	x	x	x	x
first name	x	x							
last name	x	x							
full name	x	x	x		x				x
profile photo	x		x		x				x
about		x							x
email (hash)	x				x				
homepage	x	x	x						x
blog/feed			x	x	x	x	x	x	
location		x	x		x				x
locale settings	x								
interests		x							
education		x							
affiliations	x	x							
industry		x							
tag-based profile					x	x	x	x	
posts			x	x	x	x	x		

3. Profile Alignment. The profiles gathered from the different services are aligned with GUMF's uniform user model by means of hand-crafted rules, i.e. the user data is modelled by means of Grapple statements (see Deliverable 2.1) using FOAF or vCard as domain-specific vocabulary for the actual user attributes (e.g., name, homepage, etc.). For example, given *full name* of a user's Google or Flickr profile, GUMF creates a statement as follows.

```
@prefix gc: <http://grapple-project.org/grapple-core/> .
<http://grapple-project.org/gumf/statement-2010-05-19-bob-name>
  rdf:type      gc:Statement;
  gc:user       <http://bob.myopenid.com>;
  gc:predicate  <http://xmlns.com/foaf/0.1/name>;
  gc:object     "Bob Mayer";
  gc:created    "2010-05-19T16:23:04.243+01:00" .
```

4. Semantic Enrichment. Tag-based profiles are enriched and clustered by means of WordNet¹³ categories so that GUMF client applications can, for example, access particular parts of a tag-based profile such as facets related to *locations* or *people*.

The four plug-ins can be plugged into dataspace. They have been applied to the Mypes dataspace which forms the basis for the Mypes service.

5.2 Format of Aggregated Profiles

The core task of Mypes is the aggregation of data on user profiles and activities. This data is made available via the Mypes dataspace in GUMF (see Figure 4), as well as via a lightweight RESTful interface (in FOAF and vCard format). Learning Management Systems can subscribe to this dataspace and exploit any available data on their learners for adaptation purposes.


The end-user interface of Mypes exploits its dataspace in GUMF in order to allow users to view and inspect their distributed profile data. This way the completeness of their profiles can be analysed in order to see what kind of data about them is publicly available and what information can be extracted from this data. This section discusses an example aggregated profile.

¹³

<http://wordnet.princeton.edu/>

Mypes profile information

URI: <http://out.l3s.uni-hannover.de:8080/mypes/user/116033/>

attribute	value
full name	Fabian Abel Fabian Abel
picture	
username	Fobse
website	http://delicious.com/fabianabel http://fabianabel.stumbleupon.com http://www.flickr.com/photos/fabianabel/ http://www.flickr.com/photos/fabianabel/ http://www.google.com/profiles/109472200289645138170 http://www.last.fm/user/fabianabel/
blog/feed	http://feeds.delicious.com/rss/fabianabel http://rss.stumbleupon.com/user/fabianabel/favorites http://api.flickr.com/services/feeds/photos_public.gne?id=84707785@N00 (=en-us&format=rss ; http://api.flickr.com/services/feeds/photos_public.gne?id=84707785@N00 (=en-us&format=rss ;
profile page	http://delicious.com/fabianabel http://fabianabel.stumbleupon.com http://www.flickr.com/people/fabianabel/ http://www.flickr.com/people/fabianabel/ http://www.google.com/profiles/109472200289645138170 http://www.last.fm/user/fabianabel/



Export: FOAF  | vCard 

Figure 5 - Aggregated profile visualised in Mypes.

Mypes user data consists of two different parts: traditional user *profile* information and user *activity* data. Figure 5 shows an example of aggregated user profile data, which consists of the profile attributes gathered from the various services, including the full name, user name, web site, location, affiliation, links and pictures. When accessing <http://mypes.groupme.org/mypes/user/116033/rdf> the FOAF profile in RDF/XML syntax is returned. Mypes exports all available values for a profile attribute: if a user specifies her name differently at the different services then all these different values are provided.

GUMF also connects the *tagging activities* users perform in different systems. As the *semantic enrichment* plug-in (see Figure 4) extends tag assignments with meta-information to which the WordNet category corresponding tag belongs to, it is possible to filter the aggregated tag cloud of a user according to WordNet categories. For example, Figure 6 shows the aggregated tag cloud filtered so that only tags related to locations are displayed. For this kind of tag cloud, Mypes provides an alternative visualisation: tags related to locations are mapped to country codes (using the *GeoNames* Web service¹⁴), which are sent to Google's visualisation API to draw a geographical intensity map that highlights those countries that are frequently (possibly indirectly) referenced by tags in the profile. Thanks to the *profile alignment* plug-in (see Figure 4), Mypes also features RDF export for these (specific facets of) tag-based profiles using the Tag Ontology¹⁵ and SCOT¹⁶ vocabulary.

¹⁴ <http://www.geonames.org/>

¹⁵ <http://www.holygoat.co.uk/projects/tags/>

¹⁶ <http://scot-project.org/scot/>

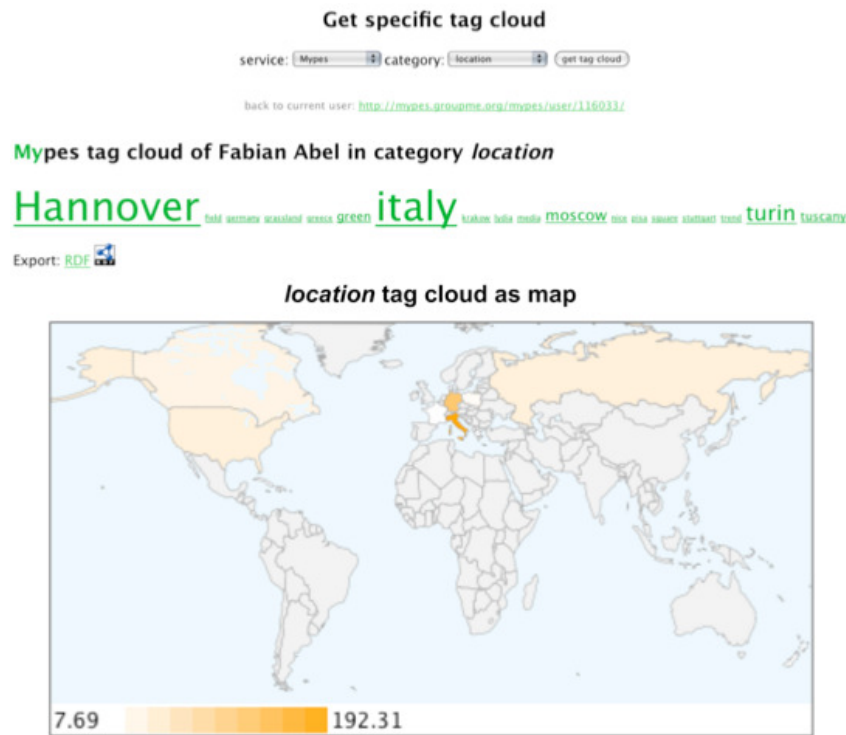


Figure 6 - Visualization of tag-based profile (filtered by location) on a Google map

5.3 Accuracy and Performance of GUMF

The accuracy of GUMF's profile aggregation functionality depends on the accuracy of the single plug-ins:

- The precision of the *account mapping* is influenced by the users who link their different online accounts in their Google profile. It is possible that users claim that some online account belongs to them even if it does belong to another user (see *My Links* at Google Profile editing page¹⁷). However, for the 53 tested users who linked the nine services that are currently covered by GUMF's Social Web aggregator (more details in the next section), this did not happen.
- It is assumed that the accuracy of the *Social Web Aggregator* is always 100% because it could only drop below 100% if a service provider would deliver profile information that does not belong to the account for which GUMF is requesting information.
- The *profile alignment* of traditional profiles also does not affect the accuracy in a negative way as it is based on hand-crafted rules that map service-specific attributes to a uniform user model.
- The *semantic enrichment* component is intended to add additional value to the aggregated profiles: tag-based profiles are enriched with metadata that specifies to which WordNet category a tag belongs to. Such metadata might be wrong.

Hence, the accuracy of the semantic enrichment is analysed in more detail.

Thirty users were randomly selected and all tag-based GUMF profiles were inspected and marked whether the attached metadata – i.e. the WordNet category assigned to a tag – is correct. On average, the tag-based profiles contained 159.4 tags. Figure 7 lists the precision of the semantic enrichment: the number of *correct* WordNet category assignments divided by the *overall* number of WordNet category assignments.

The overall precision of the semantic enrichment is 73.1%. However, the quality varies heavily with the particular WordNet category. For example, regarding tags related to *artifacts* (e.g., bike) or *communication* (e.g., hypertext, web) the accuracy is the best with 90.5% and 88.2% respectively. By contrast, the 33.1% precision for tags related to *persons* (e.g., me, george) is rather poor.

¹⁷

<http://www.google.com/profiles/me/editprofile?edit=s>

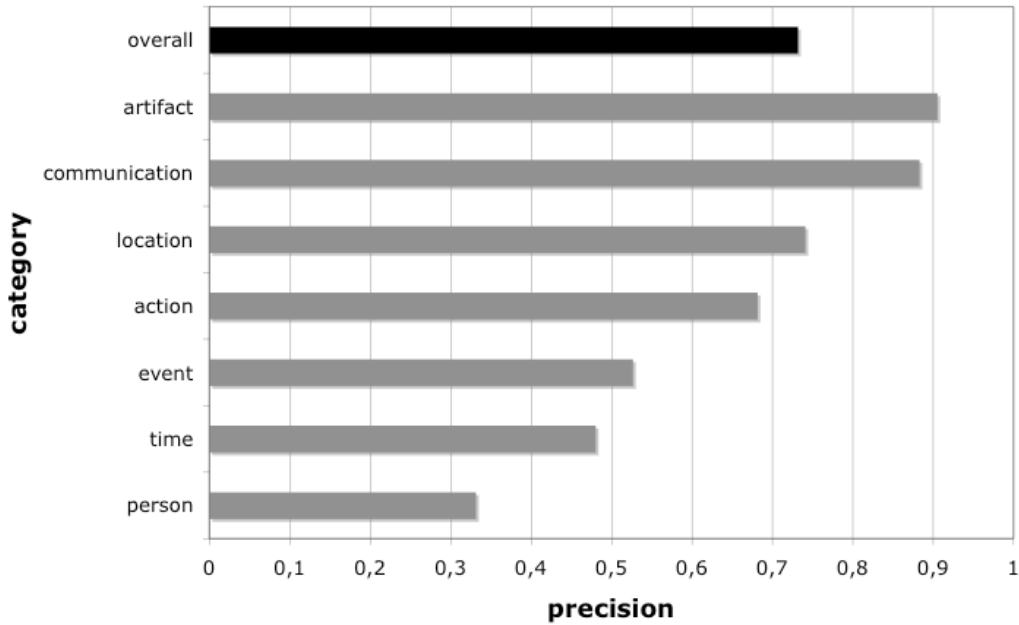


Figure 7 - Precision of semantic enrichment with WordNet categories.

In summary, it has been discovered that the accuracy of GUMF's Social Web aggregation and contextualisation features depend on single components. Account mapping, profile aggregation and profile alignment are based on hand-crafted rules and therefore do not influence the accuracy in a negative way. The semantic enrichment, which automatically attaches semantics to the tag-based profiles, produces a high precision of 73.1%.

Runtime Analysis

Using the test of the 30 randomly selected users from the previous section, the runtime behaviour of Mypes has been measured. Figure 10 summarises the results of this preliminary evaluation.

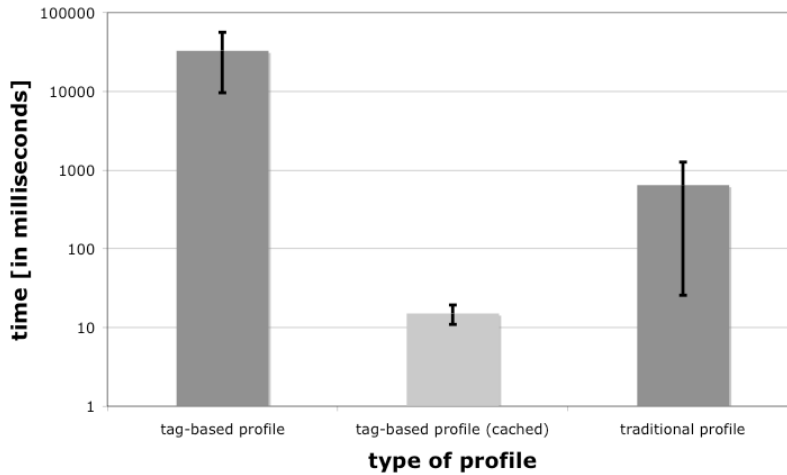


Figure 8 - Average time (in milliseconds on a logarithmic scale) required for obtaining tag-based and traditional profiles and the corresponding standard deviation.

The aggregation of traditional profiles took, on average, 645 milliseconds and is clearly much faster than gathering the tag-based profiles which took, on average, 32830 milliseconds. The huge difference can be explained by the high number of tagging activities: GUMF considered, on average, 526.3 tagging activities (= tag assignments) to construct the tag-based profiles which required calling the service APIs multiple times to

obtain the required data. GUMF caches tag-based profiles (cf. Figure 4) which improves the performance significantly as depicted in Figure 8.

In summary, GUMF aggregates traditional profiles very fast (less than one second) while the aggregation of tag-based profiles works slowly. GUMF therefore provides caching functionality.

5.4 Benefits of Profile and Activity Aggregation

As mentioned at the beginning of this chapter, the goal of profile and activity aggregation is to provide LMSs and other applications with more complete user profile information and user activity that takes place externally from the LMS.

User profile information can be used by LMSs to tailor courses to the background of the learners. This may be done based on simple demographics, such as age, location, language. Other relevant (static) user profile information may include the learner's interests, background knowledge and affiliations. It is a known problem that users fill out their user profiles only to a certain extent. With aggregate external user profile information, LMSs can make these profiles more complete and provide more accurate adaptation.

Similarly, e-learning systems are just one of the many systems that learners visit in the context of their learning activities. This implies that only part of these activities can be captured by the LMS; other relevant activities, such as Web search, online discussions and resource tagging, can therefore not be taken into account. By aggregating the relevant external user activities, LMSs have a more complete picture of the learners' activities and therewith a more substantial base for reasoning about learner knowledge, interests and goals.

To evaluate the GUMF's profile aggregation and alignment functionality, the public profiles of 116.032 distinct users have been analysed via Google's profile search¹⁸. Thereby we obtained

- 338 users that specified a *traditional profile* at Facebook, LinkedIn, Twitter, Flickr, and Google profiles
- 139 users have a *tag-based profile* at Flickr, StumbleUpon, Delicious, and Last.fm
- 53 users have an account at all services mentioned before.

The analysis presented in this section focuses on two main questions:

1. Are aggregated profiles indeed more complete and correct than the user profiles of each single service – in other words, does aggregation bring additional learner profile information?
2. Do aggregated tag-based profiles reveal more aspects of a user than the profiles in each single service – in other words, does aggregation bring more information about the learner's activities and interests?

5.4.1 Benefits of Profile Aggregation

The advantage of GUMF and Mypes is that it makes the distributed profile traces available to end-users and application developers. For example, application developers do not need to aggregate and align profile data from different services, but can just access GUMF via its client API as described in Figure 1.

Such aggregation would be needless if a service already exists that makes profiles available which are more valuable than Mypes profiles. Hence, aggregated Mypes profiles really embody more detailed information than the service-specific profiles.

¹⁸

<http://www.google.com/profiles?q=query>

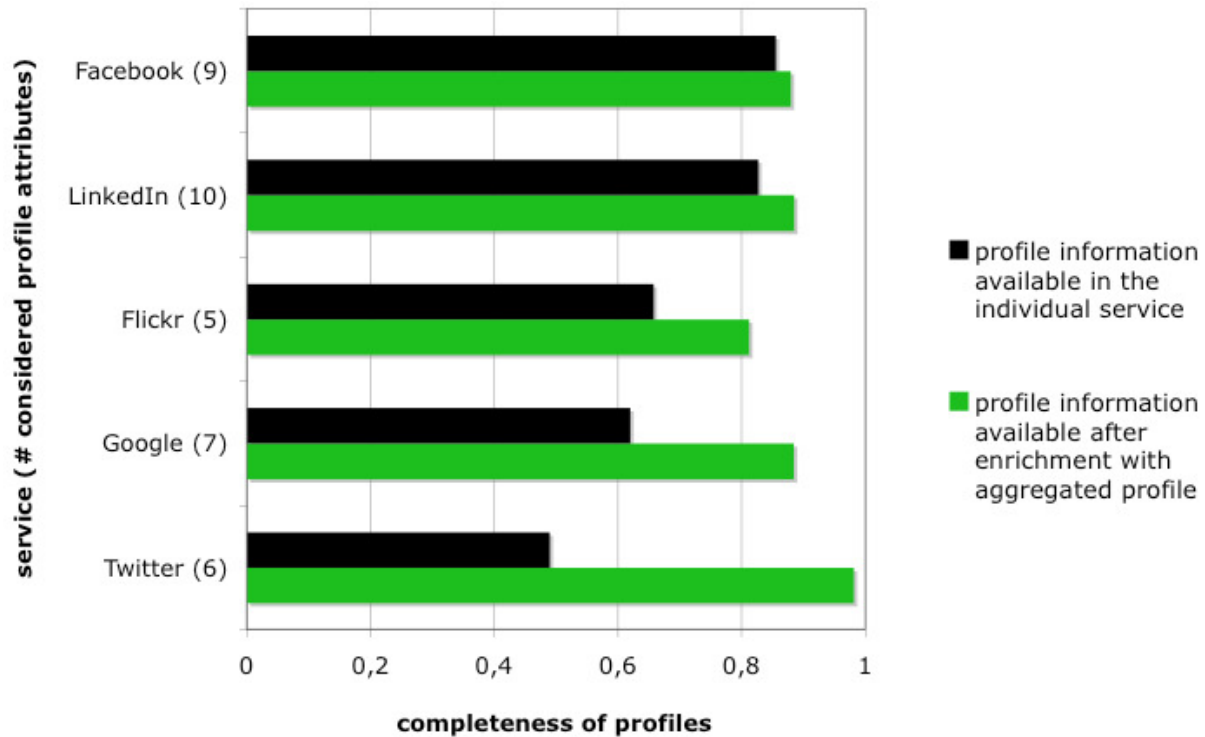


Figure 9 - Completing service profiles with aggregated profile data

Figure 9 depicts the completeness of the 338 traditional profiles and shows that the completeness of the profiles varies from service to service. The public profiles available in social networking sites Facebook and LinkedIn are filled more accurately than the Twitter, Flickr, or Google profiles, which might be explained by the intention of the different services.

The aggregated profiles reveal more facets (17 distinct attributes) about the users than the public profiles available in each separate service. On average, the completeness of the aggregated profile is 83.3%: more than 14 attributes are filled with meaningful values. As a comparison, this is 7.6 for Facebook, 8.2 for LinkedIn and 3.3 for Flickr. This proves that aggregated profiles reveal significantly more information about the users than the public profiles of the single services.

Profile aggregation enables completion of the profiles available at the specific services. For example, by enriching the incomplete Twitter profiles with information gathered from the other services, the completeness increases to more than 98% (see Figure 9): profile fields that are often left blank, such as location and homepage, can be obtained from the social networking sites. Moreover, even the rather complete Facebook and LinkedIn profiles can benefit from profile aggregation: LinkedIn profiles can, on average, be improved by 7%, even though LinkedIn provides three attributes – interests, education and industry – that are not in the public profiles of the other services.

5.4.2 Benefits of Activity Aggregation

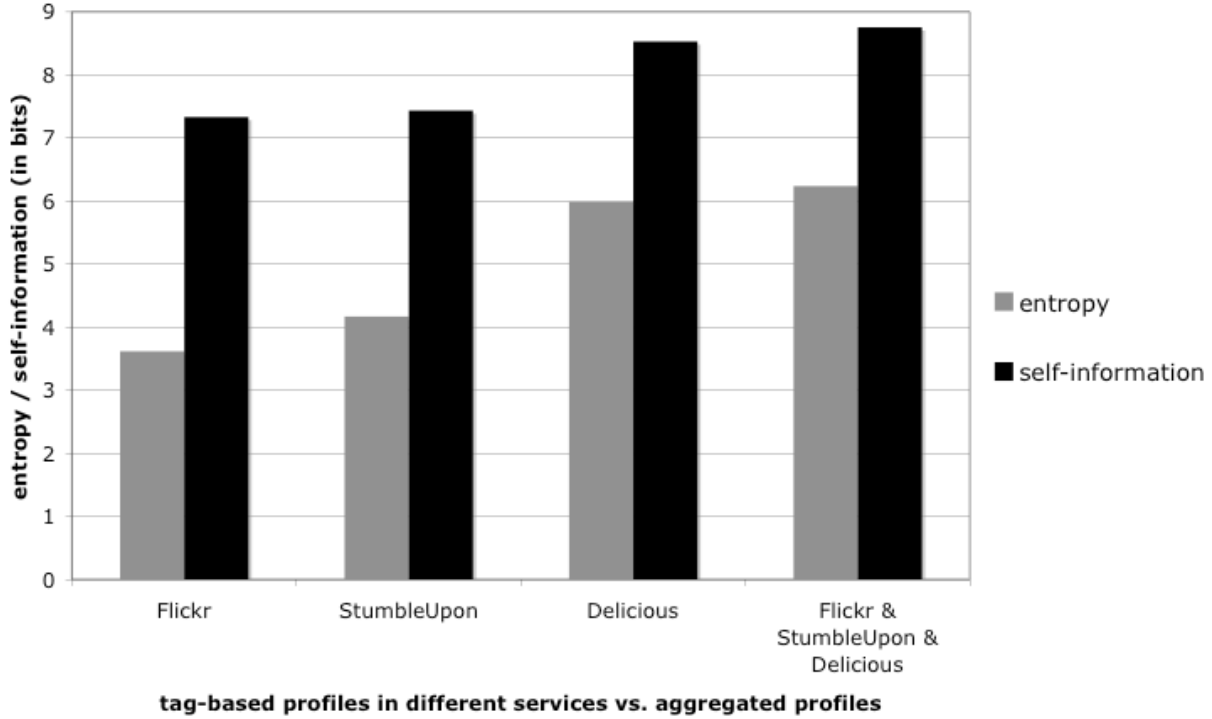


Figure 10 - Entropy and self-information of service-specific profiles in comparison to the aggregated Mypes profiles.

To investigate whether the aggregated tag-based profiles also disclose more information than the tag-based profiles of the individual services, the entropy and self-information of the profiles has been measured. Figure 10 compares – for the 139 users with accounts at services providing tag-based profiles (see table above) – the average entropy and self-information of the tag-based profiles obtained from the different services with the aggregated profile. The entropy of a tag-based profile T , which contains of a set of tags t , is computed as follows.

$$entropy(T) = \sum_{t \in T} p(t) \cdot self-information(t)$$

In the equation above, $p(t)$ denotes the probability that the tag t was utilised by the corresponding user. Self-information is the logarithm of $p(t)$ multiplied by -1 (using base 2 for the computation of the logarithm allows for measuring self-information as well as entropy in bits):

$$self-information(t) = -log_2(p(t))$$

To clarify the meaning of entropy and self-information in context of the tag-based user profiles, the applied metrics to example profiles that belong to a specific user are:

profile	tag (frequency)	self-information	entropy
flickr-bob	hannover (8) italy (8)	1	1
stumble-bob	research (8) semantic web (4)	1.08	0.92
delicious-bob	semantic web (10) social web (5) hannover (3) user modeling (3)	2.19	1.8
mypes-bob (aggregated)	semantic web (14) hannover (11) italy (8) research (8) social web (5) user modeling (3)	2.75	2.44

The self-information and entropy of the example profiles listed above depend on the number of tags that appear in the profiles and the corresponding usage frequencies as well. Bob's tag-based profiles in Flickr (*flickr-bob*) and StumbleUpon (*stumble-bob*) both contain two distinct tags. However, the self-information of the StumbleUpon profile is higher than the self-information of the Flickr profile as tags appear with different probabilities (e.g., $p(\text{research}) = 8/12$ and $p(\text{semantic web}) = 4/12$) instead of being uniformly distributed (e.g., $p(\text{hannover}) = 8/16$ and $p(\text{italy}) = 8/16$). In contrast, entropy is higher for those tag-based profiles having a rather uniform distribution and implying a higher level of randomness. The aggregation of the three example profiles (*mypes-bob*) reveals the highest self-information and entropy.

Figure 10 summarises self-information (= average of the mean self-information of the users' tag-based profiles) and entropy of the real-world profiles belonging to the 139 users considered. Among the service-specific profiles, the tag-based profiles in Delicious, which also have the largest size (on average 165.83 distinct tags), bear the highest entropy and average self-information. By aggregating the tag-based profiles, self-information increases clearly by 19.5% and 17.7% with respect to the Flickr and StumbleUpon profiles respectively. The tag-based profiles in Delicious can benefit from the profile aggregation as the self-information would increase by 2.7% (from 8.53 bit to 8.76 bit) which is also considerably higher, considering that self-information is measured in bits (e.g., with 8.53 bits one could describe 370 states while 8.76 bits allow for decoding of 434 states). The aggregated tag-based profiles therefore reveal more valuable new information about individual users than focusing just on information from single services.

5.4.3 Summary of Findings

Mypes extends the Grapple User Modelling Framework with functionality for the aggregation of user profiles and user events from various social Web systems. This external data is meant to complement the – often limited – data available in LMSs and to provide means for contextualising these LMS events.

The benefits of profile aggregation have been evaluated by comparing a number of sites from the social Web with respect to the completeness of user profiles and the variety of tags in tag-based profiles, and how these aspects would be improved when this data would be combined in an aggregated profile. The results show that in all cases the aggregated profiles provide more complete and more diverse aspects of the users.

The relevance of these findings with respect to LMSs and practical implications has been shown with various use-cases in Deliverable D2.2a. The experimental results have been obtained with data from the social Web and there are many reasons for believing that the conclusions apply to learning management systems as well. In contrast with social networking sites such as Facebook, a user/learner profile is not essential for starting to interact with a LMS; therefore it is likely that learner profiles will be rather incomplete – similar to the Flickr and Twitter profiles analysed in this study. LMSs are typically used for a shorter period or to a less comprehensive extent than popular sites from the Social Web. If even the activity profiles of these popular sites are considerably enhanced, this should also be the case for learner profiles.

6 Conclusions

This deliverable presents the design as well as the development of the event detection and contextualisation framework that is directly embedded into GUMF. Event detection and contextualisation in GUMF is based on dataspace. GUMF client applications (LMSs, GALE, other event detection tools) can add user event data (observations about users) to a dataspace, modelled as Grapple statements. Dataspace can be enhanced by plug-ins that contextualise, complement or enhance the basic event data, by enriching the data with information gathered from other sources.

In order to allow for reasoning and contextualisation, the data provided by the LMSs needs to be stored in the Grapple Statement format, which follows the basic format 'who did what when on something'. IMS-LIP – the format adopted by the LMSs – allows for more complex statements. Our IMS-LIP parser converts this data into one or more Grapple Statements.

Refining and extending the approach of Social Web aggregation using SILK and Pipes, as described in D2.2a, the Mypes service has been implemented and evaluated, which allows public profile data to be available to LMSs that require additional information about their users. Experimental results show that aggregated profile data as well as tag data is more complete than data from a single application – this is particularly true for learning management systems that are typically used only for a limited period by a relatively low number of users.

Additional user profile information may be used by LMSs for better understanding the demographics, locations, affiliations and local settings of their users – which can be used for adapting course content to the

user's language, location and age. In a similar way, examples can be selected or tailored based upon the learner's tag-based interest profiles.

The quality and amount of user events will heavily influence the quality of user modelling in GUMF. The more valuable information GUMF has about the users/learners, the better it can deduce their interests, preferences, etcetera. This deliverable describes several opportunities for doing so – opportunities that now have to be exploited by the LMSs (in particular GALE) in actual courses.

References

1. Abel, F., De Coi, J. L., Henze, N., Kösling, A., Krause, D., and Olmedilla, D.: Enabling Advanced and Context-Dependent Access Control in RDF Stores. In Proceedings of 6th International Semantic Web Conference (ISWC), Busan, Korea, pages 1-14, Springer (2007)
2. Brickley, D., and Miller, L.: FOAF Vocabulary Specification 0.91 – OpenID Edition. FOAF project, November 2007. Available at: <http://xmlns.com/foaf/spec/>
3. Broekstra, J. and Kampman, A.: An RDF Query and Transformation Language. In Staab, S. and Stuckenschmidt, H. (editors), *Semantic Web and Peer-to-Peer*, pages: 23-39, Springer (2006)
4. Carmagnola, F. and Cena, F.: User identification for cross-system personalisation. In *Journal of Information Sciences*, 179(1-2): pages 16-32, Elsevier Science Inc. New York, NY, USA. (2009)
5. Chernov, S., Demartini, G., Herder, E., Kopycki, M., and Nejd W.: Evaluating Personal Information Management Using an Activity Logs Enriched Desktop Dataset. In Proceedings of 3rd Personal Information Management Workshop (PIM 2008), Florence, Italy (2008)
6. Damerau, F.: A Technique for Computer Detection and Correction of Spelling Errors. In *Communications of the ACM*, ACM, vol. 7, no. 3, 171-176. (1964)
7. Dragontalk. <http://dragontalk.opendfki.de>
8. DCMI Usage Board: DCMI Metadata Terms. DCMI Recommendation, Dublin Core Metadata Initiative, January 2008. Available at: <http://dublincore.org/documents/2008/01/14/dcmi-terms/>
9. Google Social Graph API. <http://code.google.com/apis/socialgraph/>
10. Jameson, A.: Adaptive Interfaces and Agents. In J. A. Jacko, A. Sears (Eds) *Human-Computer Interaction Handbook*, pp 305–330. Mahwah, NJ: Erlbaum. (2003)
11. Joachims, T.: Optimizing Search Engines Using Clickthrough Data, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM. (2002)
12. OpenID.net: OpenID Authentication 2.0 - Final. December 5, 2007. http://openid.net/specs/openid-authentication-2_0.html
13. Pedersen, T., Patwardhan, S. and Michelizzi, J.: WordNet Similarity - Measuring the Relatedness of Concepts. In the Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, San Jose, California, USA. (2004)
14. Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., and Morbidoni C.: Rapid prototyping of semantic mash-ups through semantic web pipes. In Proceedings of 18th International World Wide Web Conference, Madrid, Spain, pages 581-590, ACM. (2009)
15. Prud'hommeaux, E., and Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation January 2008. Available at: <http://www.w3.org/TR/rdf-sparql-query/>
16. Schwarz, S.: A Context Model for Personal Knowledge Management Applications. In *Modeling and Retrieval of Context*, Second International Workshop, MRC 2005, Edinburgh, UK, July 31 - August 1. (2005)
17. Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G.: Silk - A Link Discovery Framework for the Web of Data. 2nd Linked Data on the Web Workshop (LDOW2009) at WWW2009, Madrid, Spain. (2009)
18. Wolf, M., Wicksteed, C.: Date and Time Formats. W3C Note. Available at: <http://www.w3.org/TR/NOTE-datetime>
19. Wu, Z. and Palmer, M.: Verb semantics and lexical selection. In 32nd Annual Meeting of the Association for Computational Linguistics, 133–138. (1994)

20. Yudelson, M., Brusilovsky, P., and Zadorozhny, V.: A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In C. Conati, K. F. McCoy, and G. Paliouras, editors, *User Modeling*, volume 4511 of *Lecture Notes in Computer Science*, pages 27-36. Springer (2007)
21. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. *Int. J. Digit. Libr.* 1 (1997) 108–121
22. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): *Theoretical Aspects of Computer Software*. *Lecture Notes in Computer Science*, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438
23. van Leeuwen, J. (ed.): *Computer Science Today. Recent Trends and Developments*. *Lecture Notes in Computer Science*, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
24. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
25. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk – A Link Discovery Framework for the Web of Data. In the *Proceedings of the 2nd Workshop about Linked Data on the Web (LDOW2009)*, workshop at WWW 2009, Madrid, Spain, April 2009.
26. Phuoc, D. L., Polleres, A., Hauswirth, M., Tummarello, G., Morbidoni, C.: Rapid prototyping of semantic mash-ups through semantic web pipes. In the *Proceedings of the 18th International Conference on World Wide Web (WWW 2009)*, Madrid, Spain, April, 2009.
27. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to Map between Ontologies on the Semantic Web. In *VLDB Journal, Special Issue on the Semantic Web*, 2003.
28. Wikipedia - Mashup (web application hybrid).
[http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))
29. Yahoo! Pipes. <http://pipes.yahoo.com/pipes/>
30. Simmen, D.E., Altinel M., Markl V., Padmanabhan S., Singh A.: Damia: data mashups for intranet applications. In the *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, Vancouver, BC, Canada, June, 2008.
31. Microsoft Popfly. <http://www.popfly.com/>
32. Intel Mash Maker. <http://mashmaker.intel.com/web/>
33. Wang, G., Yang, S., Han, Y.: Mashroom: end-user mashup programming using nested tables. In the *Proceedings of the 18th International Conference on World Wide Web (WWW 2009)*, Madrid, Spain, April, 2009.
34. Kongdenfha, W., Benatallah, B., Vayssière, J., Saint-Paul, R., Casati, F.: Rapid development of spreadsheet-based web mashups. In the *Proceedings of the 18th International Conference on World Wide Web (WWW 2009)*, Madrid, Spain, April, 2009.
35. Herder, E. and Marenzi, I. RTST Trend Report: lead theme Connecting Learners. Deliverable D1.2 of the Stellar Network of Excellence. March, 2010.
36. Zang, N., Rosson, M.B., Nasser, V.: Mashups: Who? What? Why? In: *Proceedings of Conference on Human Factors in Computing Systems on Human factors in computing systems (CHI '08)*, New York, NY, USA, ACM (2008) 3171-3176
37. Aroyo, L., Dolog, P., Houben, G., Kravcik, M., Naeve, A., Nilsson, M., Wild, F.: Interoperability in personalized adaptive learning. *J. Educational Technology & Society* 9 (2) (2006) 4-18
38. Carmagnola, F., Cena, F.: User identification for cross-system personalisation. *Inf. Sci.* 179(1-2) (2009) 16-32
39. Yudelson, M., Brusilovsky, P., Zadorozhny, V.: A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In: *11th International Conference on User Modeling*. (2007) 27-36
40. Assad, M., Carmichael, D., Kay, J., Kummerfeld, B.: PersonisAD: Distributed, active, scrutable model framework for context-aware services. (2007) 55-72

41. Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G.J., Krause, D., Leonardi, E., van der Sluijs, K.: A framework for flexible user profile mashups. In: Int. Workshop on Adaptation and Personalization for Web 2.0 at UMAP '09, CEUR Workshop Proceedings (2009) 1-10