



Specifications, models and standards for personalisation features and inquiry learning apps

Adrian Holzer

► To cite this version:

Adrian Holzer. Specifications, models and standards for personalisation features and inquiry learning apps. 2013. hal-00983715

HAL Id: hal-00983715

<https://telearn.hal.science/hal-00983715>

Preprint submitted on 13 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Go-Lab

Global Online Science Labs for Inquiry Learning at School

*Collaborative Project in European Union's Seventh Framework Programme
Grant Agreement no. 317601*



Deliverable 5.1

Specifications, models and standards for personalisation features and inquiry learning apps

Editor	Adrian Holzer (EPFL)
Date	26 th July, 2013
Dissemination Level	Public
Status	Final



©2013, Go-Lab consortium

The Go-Lab Consortium

Beneficiary Number	Beneficiary Name	Beneficiary short name	Country
1	University Twente	UT	The Netherlands
2	Ellinogermaniki Agogi Scholi Panagea Savva AE	EA	Greece
3	École Polytechnique Fédérale de Lausanne	EPFL	Switzerland
4	EUN Partnership AISBL	EUN	Belgium
5	IMC AG	IMC	Germany
6	Reseau Menon E.E.I.G.	MENON	Belgium
7	Universidad Nacional de Educación a Distancia	UNED	Spain
8	University of Leicester	ULEIC	United Kingdom
9	University of Cyprus	UCY	Cyprus
10	Universität Duisburg-Essen	UDE	Germany
11	Centre for Research and Technology Hellas	CERTH	Greece
12	Universidad de la Iglesia de Deusto	UDEUSTO	Spain
13	Fachhochschule Kärnten - Gemeinnützige Privatstiftung	CUAS	Austria
14	Tartu Ülikool	UTE	Estonia
15	European Organization for Nuclear Research	CERN	Switzerland
16	European Space Agency	ESA	France
17	University of Glamorgan	UoG	United Kingdom
18	Institute of Accelerating Systems and Applications	IASA	Greece
19	Núcleo Interactivo de Astronomia	NUCLIO	Portugal

Contributors

Name	Institution
Sten Govaerts, Adrian Holzer, Evgeny Bogdanov, Lu Yao	EPFL
Lars Bollen, Jakob Sikken, Anjo Anjewierden	UT
Panagiotis Zervas, Alexandros Trichos	CERTH
Sven Manske, Tobias Hecking	UDE

Legal Notices

The information in this document is subject to change without notice. The Members of the Go-Lab Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the Go-Lab Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material. The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

Executive Summary

Personalisation of the Go-Lab front end solutions is critical to improve usability, findability and user experience. In this deliverable, we analyse the requirements for personalisation through use cases that detail how teachers can be assisted during their activities by these functionalities. Afterwards, we provide comprehensive surveys on existing standards and specifications that enable personalisation. We first discuss the data needed to enable personalisation.

Labs, inquiry learning spaces (ILS), apps and learning resources will include rich metadata on top of their content that can be used for effective filtering and recommendation. For apps, Go-Lab will follow the OpenSocial metadata specification and the ROLE Ontology. For resources, ILS and labs, metadata specifications are still under discussion. Then, we present specifications for personalisation in Go-Lab. More specifically, personalisation in Go-Lab will be centered around *internationalisation* and *recommendation*. For internationalisation, Go-Lab will support the personalisation of languages at ILS creation time. For recommendation we propose to use a hybrid recommender system using collaborative-filtering and a multi-relational graph used in Graasp. We also propose to investigate federated and time-sensitive recommender systems. Finally, we present specifications for inquiry-learning apps, which will be based on OpenSocial standards for communication and data storage & retrieval. The apps will target both desktop and mobile devices. Discussions on software libraries that enable the use of apps on both desktop and mobile clients are still on-going.

This deliverable prepares the specifications, models, and standards for personalisation which are the foundation for the specification of the Go-Lab Portal (D5.2).

Table of Contents

1	Introduction	8
2	Requirement analysis	10
2.1	Use case 1 – Personalisation in ILS	10
2.1.1	Teacher activities	10
2.2	Use case 2 – Personalisation in apps	12
2.2.1	User interface	12
2.2.2	Recommendation	12
2.2.3	Scaffolding	12
3	General metadata for personalisation	13
3.1	User profiles and preferences	13
3.1.1	State of the art	13
3.1.2	User profile proposal for Go-Lab	14
3.2	Metadata specifications	14
3.2.1	State of the art	15
3.2.2	Metadata proposal for Go-Lab	17
4	Specifications for personalisation	19
4.1	Internationalisation	19
4.1.1	The state of the art in internationalisation	20
4.1.2	Language detection mechanisms	22
4.1.3	Internationalisation proposal for Go-Lab	22
4.2	Personalised recommendation	23
4.2.1	Use cases	23
4.2.2	Recommendation state of the art	23
4.2.3	Recommendation proposal for Go-Lab	24
5	Inquiry learning apps	25
5.1	Examples of inquiry learning apps	25
5.1.1	Concept mapping tool	25
5.1.2	Hypothesis creation tool	26
5.1.3	Experiment design tool	26
5.1.4	Reporting tool	26
5.2	Specifications for inquiry learning apps	26
5.2.1	OpenSocial specification	27
5.2.2	Space-to-app communication	27
5.2.3	App-to-app communication	27
5.2.4	Data storage/retrieval	27
5.3	Mobile devices support	28
5.3.1	Responsive Web Design	28
5.3.2	Unified user interface frameworks state of the art	28

5.3.3 System Testing	29
6 Conclusion	30
References	31

1 Introduction

Personalisation is an important component of the Go-Lab front end solutions as it can potentially improve the usability of the portal, the ILS, and the apps, as well as the findability of entities and thus contribute to the satisfaction of the overall user experience. In the DoW, we described the personalisation task (T5.2) in Go-Lab as follows: *“Personalisation features developed in this task and integrated in the Go-Lab portal include adaptation to the disciplines, class level, scaffolding requirements, histories, dynamic contexts and inquiry learning process stages, as well as spoken language (localisation). Each space that is dedicated to a given inquiry learning activity and is set by a teacher, as well as the learners exploiting it, will benefit from such personalisation features. Domain knowledge and user preferences will be taken into account. Additional services offered as integrated apps will enable learner-initiated personalisation as well as system initiated recommendation.”* In summary, through personalisation of Go-Lab entities, users will be able to customize their online environment. Contrarily to software adaptation, personalisation does not so much take decisions for users as it provides relevant options that the user can decide to follow.

Users. In Go-Lab, users will mainly be students, teachers. Students are the main target for the usage of the Go-Lab portal and the inquiry learning (IL) activities. However, as far as personalisation is concerned, students will have a limited control over their environment. Therefore, this deliverable focuses on teachers. Note that if ongoing participatory design activities (WP3 deliverables) indicate that teachers feel the need for more personalisation focused on students directly, we will provide such mechanisms in the next version of this deliverable (D5.5). Enhancing the user experience of teachers is critical in order to bring the portal to the classroom. Therefore, we plan to assist these stakeholders through personalisation from the time they post or discover inquiry learning spaces (ILS) on the Go-Lab portal, to the time they customize a learning space to their needs and also when they evaluate the result of their inquiry learning sessions.

Entities. There are several entities in the Go-Lab portal, such as *labs*, *ILS*, *apps*, *resources*, and *people*. Most of these are affected by personalisation. The central entities of the Go-Lab portal are the *labs*. In this context, labs refer to online labs as described in the DoW (exposed as smart devices and accessed through lab client apps), i.e. real remote labs, simulated virtual labs or measured data coming from real instruments. *Apps* are OpenSocial Web applications that can run in the Go-Lab portal. For example, such apps include controls for the aquarium lab, a video stream from the Faulkes telescope, a concept map, a calculator, or a chat client. Apps, can be part of an ILS, which is the space used by teachers to support a specific learning activity. For instance, the *Conservation of Momentum* learning space supports teachers in a learning activity that involves the Hypatia lab. These spaces can be created, shared and repurposed by teachers. A space will typically be populated with lab client apps, scaffolding apps, resources designed for a specific activity in a specific discipline for a

specific student level. This is the type of objects the teachers will search for in the Go-Lab Portal, which integrates a repository for ILS. Examples of *resources* are text files, videos, bookmarks, or presentations. *People* represent portal users that can have personal profiles. Through personalisation, users can change these entities, through language selection on an app, or by adding or removing some resources to a space for example.

Roadmap. Deliverable 5.1 is the first deliverable of WP5 and other upcoming deliverables will further build on this one, for instance D5.2 will elaborate on the Go-Lab portal. Deliverable 5.1 is structured as follows. Chapter 2 presents personalisation requirements for teachers through detailed use cases. These use cases show when and how personalisation can assist the stakeholders in their activities. Teachers need assistance during ILS discovery, creation and analysis. Then Chapter 3 details the data that will be used to provide personalisation services, namely user profiles and preferences, and entity metadata. In Go-lab no specific profile will be created, we will rely on simple log in for teachers and anonymous log in for students through secret URLs. For metadata, Go-Lab apps will conform to OpenSocial metadata specifications and to the ROLE Ontology. Metadata specifications for other entities are still under discussion (see G4.2). In Chapter 4, we present the Open social for personalisation in Go-Lab. We focus on internationalisation and recommendation. For internationalisation, teachers will be able to select the language of the ILS at creation time. Recommendation will be handled by a hybrid recommender system with possible time-sensitive relationship and on a collaborative filtering mechanism. Furthermore, recommendation should be based on federated resource repositories. In Chapter 5, we present examples of inquiry learning apps relevant to Go-Lab with the specifications. Finally, Chapter 6 wraps up with a conclusion and points to the future deliverables following this document.

2 Requirement analysis

Hereafter we present the two use cases for personalisation in Go-Lab namely personalisation in ILS and personalisation in apps. These use cases have been designed during brainstorming sessions at the technical meeting in Leuven on April 9, 2013.

2.1 Use case 1 – Personalisation in ILS

In ILS, personalisation will mainly aim at supporting teachers in their activities. Students will have limited control over the learning spaces and personalisation for them will be achieved through apps (e.g. through scaffolding and recommendation). Figure 1 gives an overview of the main Go-Lab use cases. More elaborate use cases and requirements for the Go-Lab portal are described in [14] and will be discussed in D5.2.

2.1.1 Teacher activities

There are three main activities performed by teachers in Go-Lab, i.e. searching for (or discovering) ILS, creating them and monitoring their usage. Personalisation supports all these activities.

ILS discovery. Teachers will typically first visit the Go-Lab portal, where ILS can be searched and accessed. The language of the portal is inferred from the teacher's profile or the preferences of the browser, but the teacher can switch the language of the portal to her preferred language using a toggle at the top of the page. When the teacher navigates to the portal, she will see a search field and a list of existing ILS and a list of default recommended ones. For instance, when the user logs in, personalised ILS recommendation can be based on her past usage of ILS and on the usage of other teachers with a similar profile. When a teacher searches for ILS, she can use several different filters to refine the search. For example she can filter the IL domain, the type of experiment, the age range, or the language. Furthermore, a recommended list of people (i.e. teachers) will also be displayed in order for the teacher to get assistance for a specific ILS.

ILS creation. When a teacher selects an ILS in the lab portal, she can either use it as it is for her class, or start customizing it. ILS can be customized by adding resources such as videos, documents, apps (such as scaffolding) or people. In order to make it easier for teachers to find relevant resources, Go-Lab provides a resource search field with filters and resource recommendation, based on the selected ILS.

IL monitoring. During the IL session, the teacher can access a dashboard that allows to monitor the work of each student to recognize e.g. which students need help. Once a session has been closed, the dashboard can be used to analyse the previous session e.g. to assess the work of each student. This dashboard consists of several monitoring apps

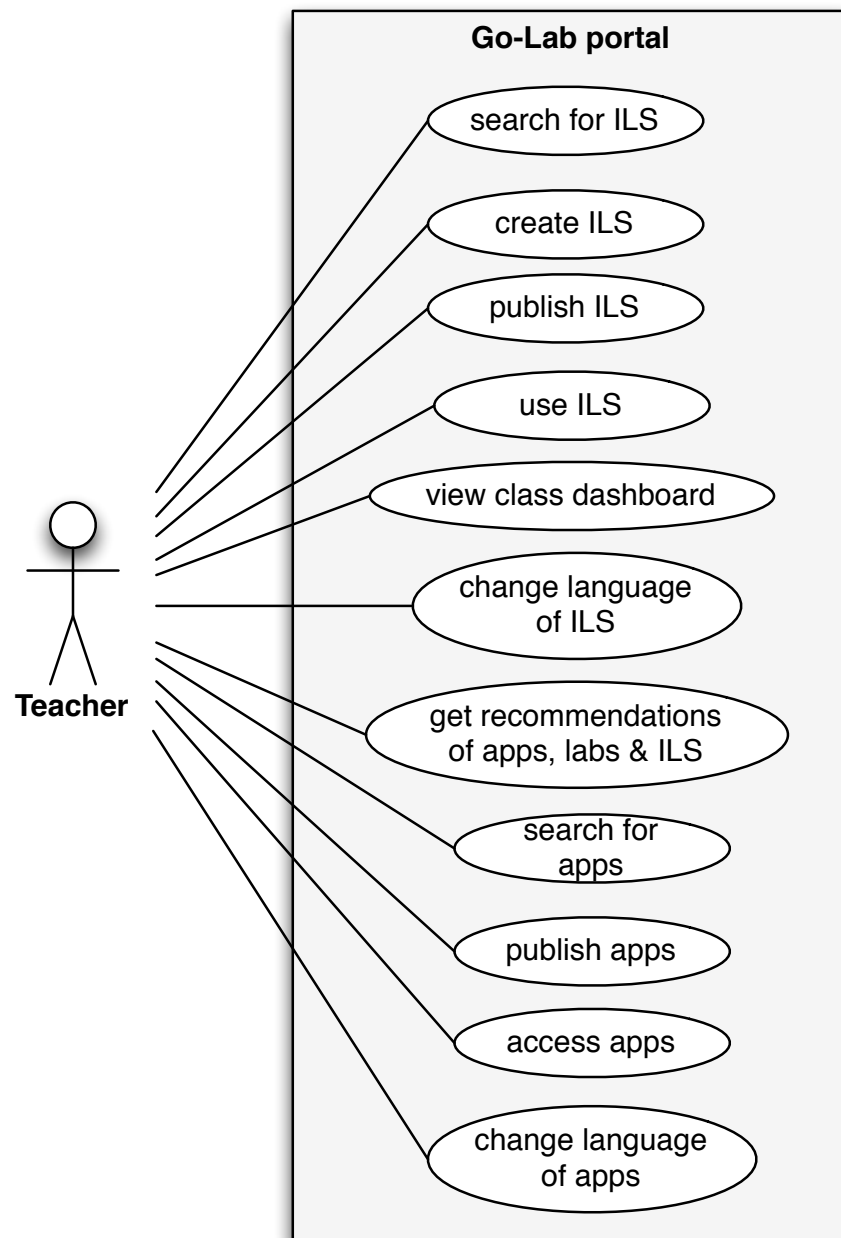


Figure 1: Go-lab use case diagram.

and can be customized, i.e. apps can be searched and added. A list of recommended apps will help the teacher make her selection. The utility of monitoring apps depends on the educational level of the class as well as on the specific task in the IL activity. Thus the ILS metadata can be used to recommend apps that fits the information needs of the teacher and the purpose of the ILS.

2.2 Use case 2 – Personalisation in apps

Apps will benefit from personalisation in different ways. For instance, their user interface (e.g. the user interface language can be adapted to the learner's mother tongue), their content can be personalised (e.g. through recommendation) and scaffolding apps can be personalised to assist students. Since apps are used by all Go-Lab users, the personalisation in apps will be available to all users. Hence, we will not differentiate among users like in Use case 1.

2.2.1 User interface

The language of the user interface of the lab workspaces and the apps can be changed to match the user's mother tongue or preferred language. For instance, the user's language preference can be determined based on settings in his user profile or automatic language detection in the browser. The language of the user interface can be further optimised for students. An elementary school student will typically be comfortable with less advanced words than a final year high school student, while students with previous knowledge of a topic will typically be acquainted with the terminology related to the topic.

2.2.2 Recommendation

Users can discover new relevant content through the use of recommendation. The content can be for instance: learning resources (e.g. texts, slides or video), other apps, people, labs or spaces. The wide range of content enables the application of recommendation in various domains. In particular, a student working on an ILS about buoyancy could be using a recommendation app that contains some videos and slides about gravity and fluid dynamics. On the other hand, a teacher who is preparing the same ILS could use the same app to discover related documents or apps to include in the ILS for the students. If the teacher would have some technical problems or would like to discuss how to apply this ILS in her course, she could use a people recommender apps that recommends other teachers or technical experts that have experience using the ILS. More detailed use cases on recommendation are described in section 4.2.

2.2.3 Scaffolding

Scaffolding (potentially with the help of learning analytics) can benefit greatly from personalisation in apps, targeting solely learners. As such, scaffolds can for instance adapt the content of an app for learners based on the actions of this particular learner. This deliverable will not deal with scaffolding and learning analytics. For a discussion on these themes, we refer to D4.2.

3 General metadata for personalisation

The metadata used for personalisation in Go-Lab will be mainly retrieved from *user profiles* and from *metatada* contained in apps, spaces and resources. Hereafter we discuss these metadata specifications.

3.1 User profiles and preferences

Clearly to achieve personalisation, we need to know as much as possible about the user. A user profile could be used to store such data including personal data of the user, social data, competencies and skills. Various specifications are available to store user information. In this section, we will first detail a few that are useful for an educational context and then discuss our proposed solutions for Go-Lab.

3.1.1 State of the art

Hereafter, we review various user profile, competence and preference specifications.

Competence frameworks. Competences are used to define “a set of knowledge, skills and attitudes that an individual possesses or needs to acquire, in order to perform an activity within a specific context” [1, 24, 10]. A competence framework consists of a number of competences, which can be applied to a broad number of roles within an organisation or a sector. Various frameworks that describe teacher competences exist: e.g. the UNESCO ICT Competency Framework for Teachers [29], the eLearning Competence Framework for Teachers and Trainers [11], Australian Competence Framework for Teachers [21], the French Competence Framework - Computing and Internet Certificate (C2i) for Teachers [4], eTQF Teacher ICT Competence Framework [13]. Other competence frameworks targeting students are available, e.g. IEEE Reusable Competency Definitions (IEEE RCD) [23] and Integrating Learning Outcomes and Competences (InLOC).¹

FOAF. The Friend of a Friend (FOAF) specification [9] provides a machine-readable ontology for describing persons, activities and their interrelationships.² FOAF terms are grouped into five broad categories: FOAF basics, personal info, online accounts, projects and groups and documents and images. The FOAF project was started in 2000 by Libby Miller and Dan Brickley. It has been adopted by several services and tools and has been used in ROLE Ontology (see below).

IMS Enterprise. IMS Enterprise [17] specifies data about people and groups. Characteristics of persons that are maintained include name and address information and data on the role of a person within an organisation. The IMS Enterprise v1.1 final

¹<http://wiki.teria.no/display/inloc/Information+Model>

²<http://xmlns.com/foaf/spec/>

specification was released in July 2002 and XML bindings are available online.³ The specification is supported by various tools and services, including .LRN, Blackboard and Liferay.

IMS Learner Information Package. The IMS Learner Information Package (LIP) [18] specification provides a standard for information about learners. The specification is designed to enable the transfer of information about learners, including their progress to date and awards received, between different software applications. Version 1.0.1 was released in January 2005 and an XML binding is available online.⁴ The uptake of IMS LIP is limited, but parts of the specification have been reused by some applications.

The ROLE User Profile. The ROLE project has developed a user profile for learners using personal learning environments [14]. Since requirements in open personal learning environments are dynamic and new information can be identified at any time, the user profile model adopts a flat structure that allows storing unidentified future data. The main elements of this user model are competences and competence state, learning goals, learning history and progress, certificates, and pedagogical parameters. All of these elements refer to other, less complex elements. For example, the competence definition refers to domain concepts, learning tools, and learning activities. A REST service has been developed that enables storage and retrieval of user profiles and separate fields.

3.1.2 User profile proposal for Go-Lab

Go-lab will not rely on specific user profiles. Students will log in the system anonymously through a secret URL which will implicitly identify them as students. Teachers will have a regular username-password login. The implications of anonymous logins and secret URLs on learning analytics will be detailed in D4.2 and D5.2. The anonymous login is a necessity for enabling students to quickly log in and start their tasks as they will likely use the ILS only once. From the requirements to adapt the UI and content to the user's language. Language will be inferred from the context for teachers (e.g. their browser). Teachers will be able to set language, discipline, and educational level at the creation of ILS.

3.2 Metadata specifications

To enable the personalised delivery or discovery of learning resources (e.g. texts or video) labs, and ILS, metadata is often needed. This section discusses metadata specifications of learning resources, apps, ILS and labs. Metadata specifications affect numerous components of the Go-Lab architecture. Work on these specifications is ongoing and will be reported in upcoming deliverables such as D5.2 and D4.2. First, we will discuss existing specifications and then elaborate on the proposed specifications for Go-Lab.

³<http://www.imsglobal.org/enterprise/index.html>

⁴<http://www.imsglobal.org/profiles/index.html>

3.2.1 State of the art

Here we review the state of the art of metadata with respect to learning resources, apps, ILS, and labs.

Metadata for learning resources This section outlines three metadata standards that are frequently used in educational settings, namely IEEE LOM, the Dublin Core Metadata Element Set and MPEG-7.

IEEE LOM. IEEE Learning Object Metadata (IEEE LOM) [16] is a hierarchical metadata standard, published by the IEEE in 2002. Its purpose is to enable the description of learning content through attributes that include the type of object, author, owner, terms of distribution, and format, as well as pedagogical attributes, such as typical learning time or interaction style. IEEE LOM is widely adopted. XML (IEEE 1484.12.3) and RDF (IEEE 1484.12.4) bindings of the standard are available. The first version of the standard was published in 2002. In the meantime, a few corrigenda have been added to the standard.⁵

Dublin Core Metadata Element Set. The Dublin Core Metadata Element Set [30] is an ISO standard for metadata, intended for cross-domain resource description. The metadata standard includes two levels: Simple and Qualified. Simple Dublin Core comprises fifteen elements: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage and rights. Qualified Dublin Core includes three additional elements (audience, provenance and rights holder), and a group of element refinements that make the meaning of an element narrower or more specific. Dublin Core is a general purpose metadata schema that is widely adopted, but there is a working group active to develop an educational profile. Version 1.1 is the latest stable version.⁶ Dublin core is available in various bindings, such as XML, RDF, HTML and plain text bindings.⁷

MPEG-7. Where the previous MPEG standards mainly dealt with multimedia encoding, MPEG-7 [20] is an ISO/IEC XML-based metadata standard for the description of multimedia content. The standard structure is modular and consists of 12 parts, including specifications for the metadata. MPEG-7 is an XML-based specification and provides an XML Schema to enable the automatic validation. There also exist mappings from MPEG-7 to RDF/OWL in order to foster the utilisation of MPEG-7 multimedia semantics in the Semantic Web [15, 5]. MPEG-7 description schemes support the specification of high-level metadata as well as low-level feature descriptors on colours, textures, shapes, motion trajectories, audio signals, etc. In entirety, the MPEG-7 standard facilitates and specifies a wide set of multimedia indexing, filtering, access and search techniques, ranging from traditional text-based to content-based multimedia retrieval. The MPEG-7 standard offers an extensive set of multimedia metadata annotations, which is also an often-heard criticism – the standard is of an

⁵http://standards.ieee.org/findstds/standard/1484.12.1-2002-Cor_1-2011.html

⁶<http://dublincore.org/documents/dces/>

⁷<http://dublincore.org/specifications/>

impressive size, and while MPEG-7 is widely adopted most applications only implement parts. To address this problem, the MPEG-7 standard defines three profiles (cf. MPEG-7 Part 9: Profiles & Levels). Furthermore, the MPEG-7 standards also details specifications on usage and user interactions which can be of value for personalisation.

Metadata for apps. Here we review two metadata standards for apps, i.e., OpenSocial and the ROLE Ontology.

OpenSocial Metadata. The OpenSocial specification provides the 'ModulePrefs', which stores metadata about the apps. An example can be seen below:

```
<Module>
  <ModulePrefs title="Hypothesis Creator"
    title_url="http://groups.google.com/group/Google-Gadgets-API"
    height="400"
    author="Michael Jackson"
    author_email="michael@thejacksonfive.com"/>
  <Content>
    ... content ...
  </Content>
</Module>
```

The content of the 'ModulePrefs' node cannot be changed by users. The metadata is used for various purposes: descriptive metadata of the apps (e.g. title and author), software features (e.g. variably changeable height), authentication details, icons, internationalisation information, etc.⁸ The OpenSocial metadata can also be retrieved via the OpenSocial Javascript and REST API.⁹

ROLE Ontology. The ROLE Ontology consists of several concepts formalised into classes expressed in OWL (see Figure 2).¹⁰ The ontology allows the modelling of tools (which can be any computer program), apps (which is a specific tool), a space (which provides a setting where users can perform several activities using tools and resources), a bundle (which is a kind of space template that describes a set of resources and tools), a tool configuration (which is used to contextualise a tool and apply specific settings to it for a specific bundle or space) and a learning resource. Concepts of the FOAF ontology have been re-used in the ROLE Ontology.

Metadata for user activities. Another important type of data that can be used for personalisation is attention metadata [26]. Such 'attention metadata' describe not content as such, but what people do with it. We will not discuss attention metadata in detail in this deliverable although it is important. For an elaborate treatment on attention metadata we refer to D4.2, where attention metadata is discussed in relation with learning analytics and scaffolding applications.

⁸https://developers.google.com/gadgets/docs/xml_reference

⁹<https://cwiki.apache.org/SHINDIG/shindigs-metadata-call.html>

¹⁰<http://role-project.sourceforge.net/model/specs/terms.html>

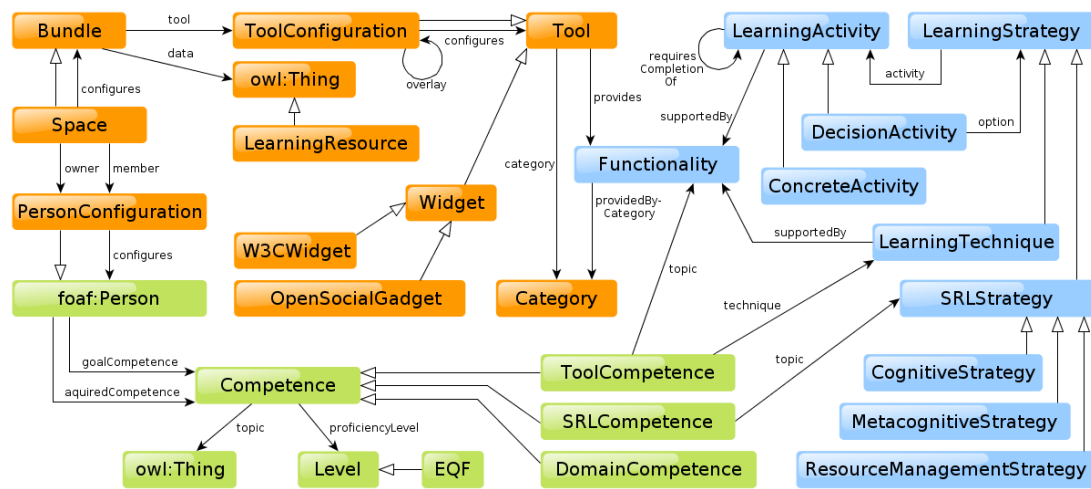


Figure 2: The ROLE Ontology overview.

Metadata for labs & ILS. In contrast to learning resources, where a number of accepted metadata schemes exist and are in use, as described above, a common metadata scheme to describe virtual and remote labs that would ease their annotation and discovery is currently not available [28]. Related efforts attempt to create indices or repositories of online labs, such as the Austrian Lab2Go project [8], or the LiLa project [28]. Of particular interest for the Go-Lab project are the efforts of the Global Online Lab Consortium (GOLC), which has been formed in 2010 with the aim of establishing an ontology and common metadata set for online labs. Recent results of the GOLC activities can be found in [28], and will be specifically considered in the Go-Lab project.

3.2.2 Metadata proposal for Go-Lab

Since Graasp, the Go-Lab ILS composer and exploitation platform, supports OpenSocial apps, Go-Lab obviously uses the OpenSocial metadata specification. Next to that, we are building a repository for labs, apps and ILS on top of ROLE Widget Store which makes extensive use of the ROLE Ontology to describe its apps (as Graasp does) and the OpenSocial metadata provided by developers to load apps into the store. Therefore, we plan to continue describing the Go-Lab apps with the ROLE Ontology. The specifications of the learning resources are still under discussion and will be detailed in Deliverable 5.2 on M12, where specific personalisation requirements will be taken into account. There is an on-going task within the technical cluster (WP4 and WP5) that aims to specify a metadata format to describe Go-Lab ILS. Metadata descriptions of ILS and labs combined with user activity metadata will enable various types of personalisation, such as recommendation. As mentioned in the previous section, existing metadata schemes, in particular the GOLC ontology and metadata will be considered in the design and application of lab metadata in Go-Lab. In Go-Lab, a number of additional aspects have to be considered in the design of lab metadata, e.g. information needed for the realisation of Smart Gateways (Task 4.2) or for the integration of scaffolds (Task 4.3) (see Deliverable 4.1). We aim to start development

and integrating of our architecture in parallel to defining the metadata specifications following an agile and flexible approach. The specific metadata particularities will be described in future deliverables.

4 Specifications for personalisation

In this section, we describe the specifications for personalisation techniques with respect to internationalisation and recommendation. These techniques can make use of the data models described in section 3.

4.1 Internationalisation

Internalisation refers to the adaptation of the Go-Lab portal and its entities (e.g. apps & resources) to different languages and regional differences. The following content could be adapted, related to:

- Language
 - Digital text (Short or long). The short text is the visible text in buttons, menus, tooltips, etc. The text is mostly displayed in the same way (the same font and style) Long text, with optional multimedia content. The long text is used for explanations, instructions, etc. It may use different fonts and styles and can contain multimedia content such as images, audio, video.
 - Graphical representation of text
 - Spoken text (audio)
 - Video and slides subtitles
- Culture
 - Images and color
 - Names and titles
 - Telephone numbers, addresses and international postal codes
 - Currency (symbols, positions of currency markers)
 - Units of value. The units of values can be different in different countries. For example most European countries use meter for size, while the United-Kingdom uses inch/foot/yard/mile.
- Writing conventions
 - Date/time format, including use of different calendars. The way date and time is shown is different in different countries. Currency: Currency is often country-dependent.
 - Time zones
 - Formatting of numbers (decimal separator, digit grouping). The way numbers are shown is language specific. For example, a number in English would be 12,345.67, while in Dutch it becomes 12.345,67.

- Differences in symbols (e.g. quoting text using double-quotes (“ ”), as in English, or guillemets (« »), as in French).
- The direction of the text, e.g. Hebrew is written from right to left.

4.1.1 The state of the art in internationalisation

This section discusses the research efforts and internationalisation techniques that are relevant for the Go-Lab project.

Internationalisation in OpenSocial. OpenSocial [27] provides the concept of ‘Message Bundles’, which are files that are separate from the apps definition file. Each file provides translation for one language of text visible to the user. Developers can provide a fallback option in case there is no appropriate translation present. An example for French is available below:

```
<messagebundle>
  <msg name="hello_world">
    Bonjour!
  </msg>
  <msg name="color">Couleur</msg>
  <msg name="red">Rouge</msg>
  <msg name="green">Vert</msg>
</messagebundle>
```

These message bundles are then referred to in the OpenSocial definition file. Based on the locale setting in the browser or apps container an appropriate message bundle will be selected. Furthermore, OpenSocial allows to create bi-directional user interfaces that dynamically change the direction of the content in a apps.¹ In conclusion, OpenSocial allows to internationalise the language and the direction of writing convention.

Internationalisation in Graasp. Graasp makes use of a Collaborative Translation Rails plugin.² This tool allows crowd-sourced translation of the Graasp user interface. Contributors can select a language, then they are presented with the list of strings used in Graasp’s UI, where they can add new translations or edit the current translation. Figure 3 illustrates the translation tool.³ In conclusion, Graasp currently only allows to internationalise the language.

Internationalisation in the SCY project. In the SCY project⁴ short text & numbers as well as static Web pages were internationalised.

Short text & numbers. The SCY project used the standard Java and JavaFX property files for the internationalisation of the text. The property files were organised at the

¹<https://developers.google.com/gadgets/docs/i18n>

²<https://github.com/vohtaski/crowd-translate>

³<http://graasp.epfl.ch/translation>

⁴<http://www.scy-net.eu/>



Figure 3: The Collaborative Translation Rails plugin in Graasp.

tool level. A special Java program was created to retrieve the text values from the property files. The same program was used for converting numbers to a display text. A tool was written to combine all language property files for all tools and exports them into an Excel file in order for users to edit the texts for all languages in a convenient way (see Figure 4). The tool could then convert the updated Excel file back to the separate language property files for each tool. This way language experts could edit the display values in one file, without the need to know technical details on how the keys and display values are stored and in which file.

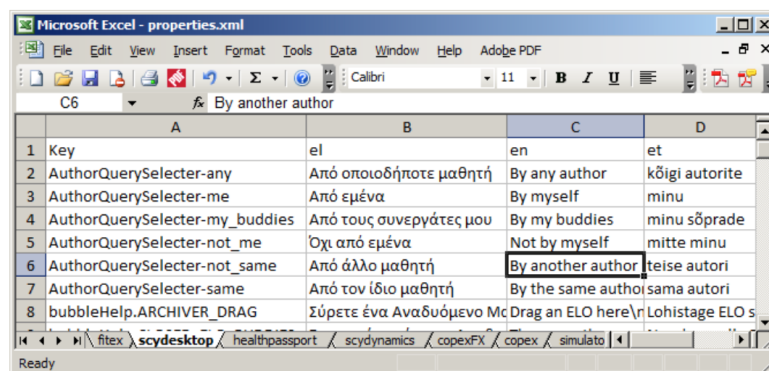


Figure 4: SCY language mapping.

Static Web pages. In SCY, static Web pages presented information and assignments related to the labs. For each language a separate Web page was created and stored in a language specific directory. For instance, English Web sites were stored in a directory named after the ISO 639-2 language code ('en' for English) on the server. When another language was needed, the ISO 639-2 language code 'en' was replaced in the url by the ISO 639-2 language code of the needed language. This is a similar system that is used in most CMS systems for websites (e.g. Drupal,⁵ but the content for a specific language is generated from the database of the CMS.

⁵<http://www.drupal.org>

4.1.2 Language detection mechanisms

To use the internationalisation techniques, the software needs to be aware of the user's preferred language. This can be done by asking the user for her language preference (e.g. when she first accesses the Go-Lab portal or when she sets up her user profile) or can be automatically detected. Typically in a browser, the user language can be automatically detected based on the HTTP request header. This header has a field 'Accept-Language' which contains a list of locales and a quality parameter that signifies the user's preference for this language. This list is determined by the user's browser and operating system. Based on this a language choice can be made and stored in the user profile or a cookie for future reference. Other possible options for detecting the user's locale is looking up the location, but this is obviously less accurate, as the user might be an expatriate or have different preferences. Preliminary participatory design activities have demonstrated that teachers would like to keep control of the language selection through preferences.

4.1.3 Internationalisation proposal for Go-Lab

In Go-Lab we want to personalise the language in the UI of the portal and the apps. In the above section 4.1.1, we have discussed various techniques. In this section, we discuss the choices made for Go-Lab. Graasp already has a mechanism for translating the UI. But, it might also be necessary to support internationalisation for learning resources. For instance, enabling that multiple language versions of a resource can be uploaded in Graasp. Note that Go-Lab does not provide support for right-to-left language direction. To internationalise apps, we can benefit from the built-in features of the OpenSocial apps. As these apps are often using numerical and measurement data, the internationalisation of numbers and units will be a necessity. For the units, we propose to use the International System of Units.⁶ This type of internationalisation will be handled by the apps and labs. We plan to have a module in the App Composer to enable teachers to translate the UI of apps (see D5.2).

The provided mechanisms together with the translation tools in the App Composer can also be used by teachers to adapt the UI and scaffolding apps to the specific needs of their students, e.g. providing the right language for a specific age group (for instance, English Biology terms for 8 to 10 year old). When translation the UI or providing content for scaffolding apps, teachers will be able to select the appropriate language and age group.

The Go-Lab portal (i.e. Graasp & the lab repository) might propose the user language based on the user preferences, settings or automatic language detection and will propagate this information to apps and display the UI and resources in the correct language. Teachers will still be able to change the language per space. The Go-Lab portal, where teachers can search for ILS, will be built on top of the ROLE widget store.⁷ This platform is build on Drupal, which contains extensive internationalisation features.⁸

⁶http://en.wikipedia.org/wiki/International_System_of_Units

⁷<http://www.role-widgetstore.eu/>

⁸<http://drupal.org/project/i18n>

4.2 Personalised recommendation

Recommender systems are important to provide various content to users and create responsiveness in the Go-Lab portal and in ILS. In this section we will first provide more detailed additional use cases than the ones described in section 2 and show the various types of content that can be recommended. Afterwards we will discuss the recommendation strategies and the current recommendation mechanism available in Graasp.

4.2.1 Use cases

As mentioned previously, recommendation will mainly assist teachers in their activities.

Recommendation for teachers. Recommendation for teachers covers simple entities such as resources and people and more complex ones such as apps, ILS and labs. This recommendation can be displayed for example during ILS discovery or ILS creation. Recommendation related to ILS discovery can be offered at the beginning of an ILS search on the Go-Lab portal. A specific query which might suit the needs of the teacher can be recommended. The query itself can also be personalised, to more easily find relevant result in very large search result sets. This is usually performed through query expansion [6]. Teachers can extend ILS with new *resources* (e.g. videos or slides). Relevant resources can be recommended based on the usage patterns of resources and ILS as well as similarity between teachers. When it comes to *people* recommendation, teachers can require technical assistance or can benefit from building a social network of experienced teachers. Recommending people can be performed naturally through user-based collaborative filtering as a ranked vector of user similarities. Useful *apps* can be recommended to teachers based on frequent adding of a specific app to an ILS. Similarly to apps, Spaces can be re-used by teachers and are therefore useful to recommend. Finally, an entire ILS can be recommended to teachers. As mentioned earlier, an ILS consists of the apps to control the online lab, other apps to support the learner, and a set of resources organized in IL phases. ILS can be recommended to teachers based on their profile (e.g. based on what they teach).

4.2.2 Recommendation state of the art

Hereafter we discuss several recommender systems that are relevant to Go-Lab, namely content-based recommenders, collaborative filtering, and hybrid recommenders.

Content-based recommender. In content-based recommenders, the similarity between two entities (e.g. people, resources & apps) can be computed by Latent Semantic Analysis (LSA) tools or other data mining tools.

Collaborative filtering. Collaborative Filtering (CF) adds relationships between entities. CF is a widely used approach for recommender systems [2]. Examples of mainstream platforms relying on such a mechanism include Amazon. One can distinct

user-based and item-based collaborative filtering. In user-based CF a user gets recommendation of items based on items that similar users use or like. In contrast, in item-based CF the similarity between items is calculated [25]. If a user uses an item, recommendation can be generated based on similar items that were used by users that also used the previous item. This method requires no user information.

Hybrid recommenders. To further improve recommendation quality hybrid approaches are used that combine different techniques. For instance, Graasp provides a personalized and contextualized recommender system combining content-based and collaborative filtering mechanisms. Graasp builds a weighted, directed and multi-relational graph (called a 3A graph in Graasp) to model its entity network [12]. In the 3A graph, the nodes represent the entities (i.e. Apps, Actors & Assets) and the edges are relations between the entities. The relations are built according to interactions between the entities. Content-based techniques are used as well. Namely, if two entities' similarity is beyond a certain threshold, they will be connected with a *similar* relation and appended to the 3A graph. After the graph is constructed, the 3A ranking algorithm is executed to rank the other entities according to their importance to the target. The algorithm is extended from personalized PageRank [22]. At last, through iterative computation, a ranking vector is extracted from the graph. The entities are ranked according to the possibilities that the target intends to visit the destinations. The top entities in the ranking vector are recommended to the target. Currently recommendation in Graasp are based on long term interests. However, user interest evolves over time and in some cases it might be interesting to increase the importance of recent activity over older ones [7].

4.2.3 Recommendation proposal for Go-Lab

In Go-Lab we aim for a hybrid system, using a powerful multi-relational graph model for fine-grained recommendation. Compared to a purely collaborative filtering approach, a hybrid approach using multiple relations solves the cold start problem, i.e. recommendations can only be calculated if there is sufficient user activity. We also consider to investigate the possibilities to integrate time-sensitive models to further fine-tune recommendation in the 3A graph. Finally, since Go-Lab entities will likely reside on different databases, we will investigate ways to provide a federated recommender system that can harvest user interaction data and content from different platforms and databases.

5 Inquiry learning apps

Inquiry learning apps are web applications, which assist learners in their work with online labs and support the inquiry-based learning pedagogical paradigm (see D1.1), e.g. students create a hypothesis, plan an experiment or draw conclusions and write a report. These tools will be integrated in the Go-Lab Portal and can be used as components in ILS. In this chapter we start by a description of these applications that will serve as a basis to derive specifications for the required functionalities within Go-Lab, such as Inter-widget communication, Space communication and Data storage-retrieval features. Finally, as these apps are expected to be not only used from desktops but also from mobile devices, we discuss ways to provide support for an adequate mobile user experience.

5.1 Examples of inquiry learning apps

Hereafter we present examples of inquiry learning apps, namely a concept mapping tool, a hypothesis creation tool, an experiment design tool, and a reporting tool.

5.1.1 Concept mapping tool

A concept mapping tool can help learners to establish a basic overview of a problem domain. At the heart, a concept map consists of nodes (concepts) and edges (relations between concepts). In Go-Lab, the choice of concepts and relations may be limited and tailored to the problem domain at hand and to the online lab being used. This feature can be realised by exchanging space metadata and lab metadata. In addition, concept maps (either predefined by a teacher or constructed by a learner) can serve as input to other inquiry learning apps, e.g. to provide variables and their relations to provide initial frames for hypotheses or to indicate a meaningful structure for an experiment. Figure 5 shows a screenshot of a prototype of the Go-Lab concept mapping tool.

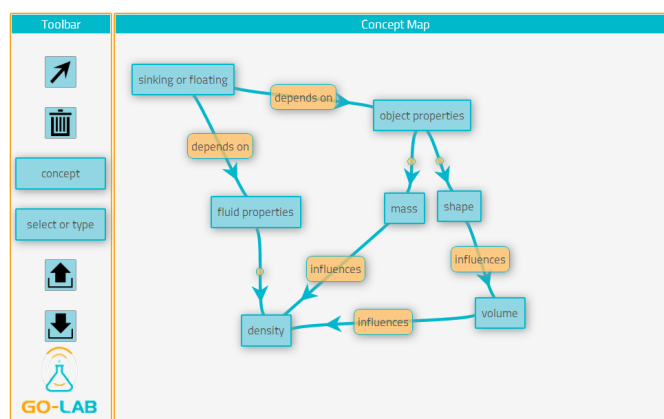


Figure 5: Go-Lab concept mapping tool propotype.

5.1.2 Hypothesis creation tool

This tool allows the learner to create hypotheses or research questions related to the current topic and to the used online labs. It provides sentence openers, dependent and independent variables and other building blocks to support the learner in the process. The hypothesis tool might make use of input from the concept mapping tool and provide output for the experiment design tool and for the Reporting tool.

5.1.3 Experiment design tool

The experiment design tool allows users to plan experiments in detail in Go-Lab. Using domain descriptions and lab metadata, this tool provides lists of dependent (manipulable) variables and independent (measurable) variables to construct a sequence of experience. Integrating data from the hypothesis tool, the experiment design tool guides the learner to create meaningful experiments to answer her questions. Figure 6 shows a screenshot of a prototype of the Go-Lab experiment design tool.

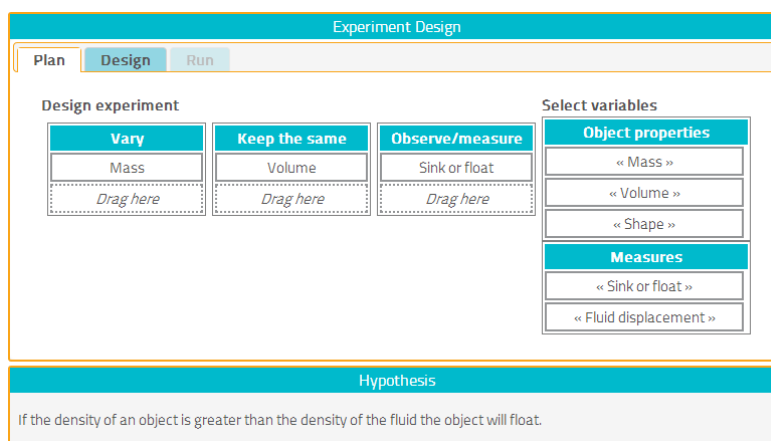


Figure 6: Go-Lab experiment design tool prototype.

5.1.4 Reporting tool

To support later inquiry phases, one or more tools are required to support summarising and presenting results. To this end, learners will be able to create a report by drawing conclusions from previous activities. A reporting tool can aid in this process by integrating the results of earlier phases, e.g. created hypotheses or data collected from online labs. Learner will be able to add their conclusions and critical discussions based on the collected evidence.

5.2 Specifications for inquiry learning apps

Inquiry learning apps are implemented as OpenSocial widgets. Next, we provide details about the OpenSocial specification and discuss how apps can communicate with their spaces, with other apps and also how they can save their specific data.

5.2.1 OpenSocial specification

The OpenSocial specification was launched by Google in 2007 and now represents an independent standardization organization - OpenSocial Foundation. The specification consists of three main parts. The first part describes the special widget standard (OpenSocial widget) that was first proposed and implemented by Google and later spread to other Web platforms. The second part of the specification standardizes the model for a social network elements (i.e. Person, App, Document) and connections between them. The third part standardizes a set of common REST and JavaScript APIs to retrieve elements of a social network from a Web platform. For example, a list of resources that a person uses or a list of her friends can be obtained through the APIs.

5.2.2 Space-to-app communication

Since the space concept did not exist in OpenSocial, we introduced it into the OpenSocial specification. The OpenSocial Space extension standardizes the space model (namely, a list of fields that a space can contain), the REST APIs and JavaScript APIs to work with spaces [3]. Graasp implements the OpenSocial Space extension APIs. Thus, inquiry learning apps can retrieve the information about the space in which they are executed via OpenSocial APIs. For example, the Person API can be used to retrieve all members of the space. The app can retrieve the list of the existing resources in the space and also a list of all space apps via the Document and the App APIs respectively.

5.2.3 App-to-app communication

To improve the user experience and to extend functionality, apps should be able to exchange data and events when they open side by side. In other words, the app-to-app communication has to be enabled. In this case, data does not leave the browser and the remote services are not involved. Data is transferred from the source app directly to the browser memory and then moved from there to the destination app. Zuzak et al. [31] provided a classification of the available techniques to achieve this type of communication. Our suggestion is to use OpenApp approach introduced by [19], developed in the ROLE project, for app-to-app communication due to its improved usability compared to other approaches and partial semantic interoperability features.

5.2.4 Data storage/retrieval

Inquiry learning apps can use the standard OpenSocial AppData APIs to store and retrieve their specific data. Each item of AppData represents a key-value pair that a widget can save on a per-space basis. These key-value pairs are persisted in the storage of the Web platform that hosts the apps. For example, a note taking client app can save the lab report of each student and a hypothesis creation app can store the hypotheses which can be used by an experiment design tool.

5.3 Mobile devices support

In order to assure a seamless desktop to mobile user experience transition, apps should be build on three fundamental pillars: responsive Web design, unified user interface frameworks and widespread system testing.

5.3.1 Responsive Web Design

Responsive Web Design (RWD) is a Web design and implementation approach for creating flexible Web sites that offer excellent viewing and navigating experience across a series of different device displays like mobile phones, tablets or PCs.¹ With this technique, while keeping the core HTML code intact and as minimal as possible, by utilizing CSS Media queries, flexible media (images) and fluid grids, it is possible to achieve a smooth cross-device user experience. CSS3 Media queries² are built upon the existing idea of media queries as found in CSS2 that only supported queries for media types, such as screen or print, but enhanced by the addition of several other parameters, such as browser type and device width and height, display orientation and resolution. With this, it is trivial to detect the viewport in which the Web page will be rendered and define custom rules for it. Moreover, the fluid grid based layouts offer content flexibility by reflowing and reformatting text or other content according to the detected screen size or browser window size. Finally, some best practices for the responsive design CSS recommend the use of screen percentages and font units (em) instead of pixels for formatting the content.³ UI Frameworks like JQuery mobile and Twitter Bootstrap offer the benefits of responsive design while also encompassing other utilities like form elements, theming functionality and widgets.

5.3.2 Unified user interface frameworks state of the art

The use of a unified user interface framework for all UI components such as form elements, commonly used widgets and navigation items, is considered to be mandatory for a coherent look and feel in Web sites. In most cases those frameworks offer a set of CSS templates and JavaScript libraries that should be utilized throughout the entire Web site in order to offer seamless user experience across various devices and browsers. Examples of such frameworks include JQuery Mobile and the Sencha touch. While both are very widely used and considered as the state of the art in mobile Web sites / Web apps development each one gets its focus on a slightly different UI segment.

JQuery Mobile. JQuery Mobile is based on the jQuery and jQuery UI libraries, offering great ease of use, good documentation and very large community support. The main idea behind it is to keep the html code as simple as possible and thereafter, with the use of jQuery to enhance the UI elements, by automatically wrapping them inside jQuery tags, and give them the desired *look and feel* and functionality. Aiming more in standard form and navigation elements and widgets, like buttons, input fields,

¹http://en.wikipedia.org/wiki/Responsive_web_design

²<http://www.w3.org/TR/css3-mediaqueries/>

³<http://view.jquerymobile.com/1.3.1/demos/intro/rwd.php>

collapsible grids, popup calendars, it demonstrates less out of the box capabilities when custom graphics are needed.

Sencha Touch. Sencha touch on the other hand, is based on the ext.js platform, which is somewhat less popular than jQuery and receives more criticism especially regarding the community support and the ease of use at the coding level. In general it requires less html code and more javascript/ext.js code but has very good documentation for that purpose. However, Sencha touch is more powerful in the graphics domain, as it offers greater flexibility for graphics driven applications that need features like charts or extended SVG support. It is best suited for demanding apps or Web sites that need that extra step in graphics or require a near-to-native app experience. Note that discussions are still going on to determine which libraries and design templates will be used in Go-Lab.

5.3.3 System Testing

The system-testing phase is extremely important for the overall quality of the final deliverable. We aim to organize the system testing as a two-step process.

In the first phase, we will take advantage of the official device simulators that are offered by the mobile platform companies like the iOS Simulator found in Xcode SDK by Apple or the Android OS simulator as in Android SDK (ADK). Those simulated environments offer operating system (OS) and hardware sets that in conjunction with the native Web browsers of each OS, Safari and Internet, are accurate enough in order to initialize the testing.

However, the main testing will be carried out in the second phase, where real devices will be taken into account; in most cases popular tablet devices with well-established OSs and Web browsers. This phase will also offer the opportunity to test in several other Web browsers as well, like Mozilla Firefox or Google Chrome, which will help to identify possible browser incompatibles.

A key point in device testing, beyond the apparent functionality testing, is the responsiveness of the Web interface in diverse viewports. With this series of tests we will also determine how efficiently our RWD template is able to handle different resolutions, orientation and aspect ratios. Finally, another important topic during the testing phase is the browser-to-browser incompatibilities regarding rendering of CSS3/HTML5 elements that are utilized as well as the behavior of the various touch events or gestures, like dragging, pinch-to-zoom, etc.

6 Conclusion

In this deliverable, we have presented our vision on personalisation in the Go-Lab front end solutions through use cases (see Section 2). Then we have investigated the various data sources that are required to achieve personalisation and how they can be represented by data standards and specifications. For apps, Go-Lab will follow the OpenSocial metadata specification and the ROLE Ontology. For other entities, such as ILS, resources and labs, specifications are still under discussion. Afterwards, we have discussed the internationalisation and recommendation techniques that we are using currently and are planning to use in the future. In terms of internationalisation Go-Lab will allow users to specify the language at ILS creation time. Go-Lab will also provide personalised recommendation via a hybrid recommender system using collaborative filtering and a multi-relational graph used in Graasp. Furthermore, we plan to investigate federated and time-sensitive recommendation systems. Finally, we have presented proposals for inquiry learning apps and have shown how they can be integrated in the Go-Lab portal. Inquiry learning apps will target both desktops and mobile devices and will be based on Open social standards for communication and data storage and retrieval. Discussions on library standards are still ongoing. An implementation of the proposals done in this deliverable will be available in D5.3 (M18) and the specifications will be further updated and finalised in D5.5 (M32).

References

- [1] CEN workshop. European e-Competence Framework 2.0 - executive overview. retrieved from http://www.ecompetences.eu/site/objects/download/5991_EUeCF2.0overview.pdf, 2010.
- [2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
- [3] Evgeny Bogdanov, Christophe Salzmänn, and Denis Gillet. Contextual Spaces with Functional Skins as OpenSocial Extension. In *4th International Conference on Advances in Computer-Human Interactions*, pages 158–163, 2011.
- [4] C2i. Retrieved from competency framework - computing and internet certificate (C2i): <http://www.c2i.education.fr/IMG/pdf/EN-D0C-Referentiel-C2i2e.pdf>, 2012.
- [5] Klamma R. Cao, Y. and M. Khodaei. A multimedia service with MPEG-7 metadata and context semantics. In *Proceedings of the 9th Workshop on Multimedia Metadata (WMM'09), Toulouse, France, March 19-20, 2009*.
- [6] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, January 2012.
- [7] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Communities of interest. In Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher, and Gabriela Guimaraes, editors, *Advances in Intelligent Data Analysis*, volume 2189 of *Lecture Notes in Computer Science*, pages 105–114. Springer Berlin Heidelberg, 2001.
- [8] M. Niederstätter D.G. Zutin, M.E. Auer. A repository to locate educational online laboratories, 2011.
- [9] Li Ding, Lina Zhou, Tim Finin, and Anupam Joshi. How the semantic web is being used: An analysis of FOAF documents. In *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04*, HICSS '05, pages 113.3–, Washington, DC, USA, 2005. IEEE Computer Society.
- [10] Mentzas G. Draganidis F. Competency based management: A review of systems and approaches. *Information Management and Computer Security*, pp. 51-64., 2006.
- [11] EIfEL. The elearning competency framework for teachers and trainers. retrieved from <http://www.eife-l.org/publications/competencies>, 2008.
- [12] Sandy El Helou. *The 3A Interaction Model and Relation-Based Recommender System*. PhD thesis, STI, Lausanne, 2010.

- [13] eTQF. ICT competency framework for teachers. retrieved from <http://etqfproject.ning.com/page/etqf-framework-1>, 2010.
- [14] Sten Govaerts, Katrien Verbert, Daniel Dahrendorf, Carsten Ullrich, Manuel Schmidt, Michael Werkle, Arunangsu Chatterjee, Alexander Nussbaumer, Dominik Renzel, Maren Scheffel, Martin Friedrich, Jose Luis Santos Odriozola, Erik Duval, and Effie L.-C. Law. Towards Responsive Open Learning Environments: the ROLE Interoperability framework. In Carlos Delgado Kloos, Denis Gillet, Raquel M. Crespo Garcia, Fridolin Wild, and Martin Wolpers, editors, *Towards Ubiquitous Learning - Proceedings of 6th European Conference of Technology Enhanced Learning, EC-TEL 2011*,, pages 125–138. Springer, September 2011.
- [15] J. Hunter. Adding multimedia to the semantic web - building an MPEG-7 ontology. In *International Semantic Web Working Symposium (SWWS), Stanford, July 30 - August 1, 2001*.
- [16] IEEE. *IEEE Standard for Learning Object Metadata*. 3 Park Avenue, New York, NY 10016-5997, USA., 2002.
- [17] IMS. IMS enterprise v1.1 final specification, 2002.
- [18] IMS. IMS learner information package version 1.0.1 - final specification. 2005.
- [19] E. Isaksson and M. Palmer. Usability and Inter-widget Communication in PLEs. In *Fifth European Conference on Technology Enhanced Learning (EC-TEL10), The 3rd Workshop on Mash-Up Personal Learning Environments (MUPPLE10)*, 2010.
- [20] Moving Picture Experts Group. *MPEG-7 Overview, 10th edition, October 2004*.
- [21] Department of Education and Training of West Australia. Competency framework for teachers. retrieved from <http://det.wa.edu.au/policies/detcms/policy-planning-and-accountability/policies-framework/guidelines/competency-framework-for-teachers.en?oid=com.arsdigita.cms.contenttypes.guideline-id-3738620>, 2004.
- [22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.
- [23] IEEE RCD. IEEE standard for learning technology-data model for reusable competency definitions, 2008.
- [24] Fytros D. Sampson D. Competence models in technology-enhanced competence-based learning. in Adelsberger H.H., Kinshuk, Pawlowski J.M., Sampson D. (Eds.), *International Handbook on Information Technologies for Education and Training* (pp. 157-176, Chapter 9), Springer, 2008.

- [25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web, WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.
- [26] Hans-Christian Schmitz, Uwe Kirschenmann, Katja Niemann, and Martin Wolpers. Contextualized attention metadata, in Claudia Roda (Hrsg.), *Human Attention in Digital Environments*, Cambridge University Press, S. 186-209, 2011.
- [27] OpenSocial Specification. Available on <http://opensocial.org/>, April 2013.
- [28] P. Grube T. Richter and D. Zutin. A standardized metadata set for annotation of virtual and remote laboratories, 2012.
- [29] UNESCO. ICT competency standards for teachers: Competency standards modules. retrieved from <http://cst.unesco-ci.org/sites/projects/cst>, 2008.
- [30] S. Weibel and T. Koch. The Dublin Core metadata initiative-mission, current activities, and future directions,. In *D-Lib Magazine*, vol. 6, no. 12, 2000.
- [31] Ivan Zuzak, Marko Ivankovic, and Ivan Budiselic. A Classification Framework for Web Browser Cross-Context Communication. In *CoRR*, 2011.