## Project Report

### ID5.12: Interaction models and requirements for a knowledge sharing scenario

<table>
<thead>
<tr>
<th>Workpackage</th>
<th>WP5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>5.1</td>
</tr>
<tr>
<td>Date of delivery</td>
<td><strong>Contractual:</strong> 31-01-2008</td>
</tr>
<tr>
<td>Code name</td>
<td>ID5.12</td>
</tr>
<tr>
<td>Type of deliverable</td>
<td>Report</td>
</tr>
<tr>
<td>Security (distribution level)</td>
<td>Public</td>
</tr>
</tbody>
</table>

### Contributors

**Authors (Partner)**
Linda Napoletano (Giunti Labs), Michele Dicerto (Giunti Labs), Elena Demidova (UHANN), Krassen Stefanov (SU), Carlos Mendez (Altran), Alessandro Mazzetti (Giunti Labs), Alexander Grigorov (SU)

**Contact Person**
Alessandro Mazzetti (Giunti Labs), Michele Dicerto (Giunti Labs).

**WP/Task responsible**
Giunti Labs

**EC Project Officer**
Hans-Jürgen Westhof

### Abstract (for dissemination)

**Keywords List**
WP5, internal deliverable, scenarios, requirements

---

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 – Fax: +31 - Email: ten.competence@ou.nl
Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Date</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Draft</td>
<td>26.02.08</td>
<td>Linda Napoletano, Giunti Labs</td>
</tr>
<tr>
<td>0.2</td>
<td>Draft</td>
<td>03.03.08</td>
<td>Linda Napoletano, Giunti Labs</td>
</tr>
<tr>
<td>0.3</td>
<td>Draft</td>
<td>11.03.08</td>
<td>Linda Napoletano, Giunti Labs: Elena Demidova, UHANN</td>
</tr>
<tr>
<td>0.4</td>
<td>Draft</td>
<td>26.03.08</td>
<td>Alessandro Mazzetti Giunti Labs</td>
</tr>
<tr>
<td>0.5</td>
<td>Final</td>
<td>31.03.08</td>
<td>Alessandro Mazzetti Giunti Labs</td>
</tr>
<tr>
<td>1.0</td>
<td>Revised</td>
<td>29.09.08</td>
<td>Alessandro Mazzetti Giunti Labs</td>
</tr>
</tbody>
</table>

Summary of Changes

<table>
<thead>
<tr>
<th>Version</th>
<th>Section(s)</th>
<th>Synopsis of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>All</td>
<td>Draft: Table of Contents</td>
</tr>
<tr>
<td>0.2</td>
<td>All</td>
<td>Improvement of the Table of Contents</td>
</tr>
<tr>
<td>0.3</td>
<td>Sections 2-3-4</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>Section 1.2.3</td>
<td>Added</td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>Deleted un-useful yellow lines</td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>Where applicable, KRSM has been substituted with LearnWeb2.0</td>
</tr>
<tr>
<td></td>
<td>Section 4.1</td>
<td>Added</td>
</tr>
<tr>
<td></td>
<td>Section 5</td>
<td>Added</td>
</tr>
<tr>
<td>0.5</td>
<td>Section 3.3</td>
<td>Added from WP5_ID5_12_V0.2-SU2.doc (Grigorov)</td>
</tr>
<tr>
<td></td>
<td>Section 3.1.1</td>
<td>Updated (Demidova)</td>
</tr>
<tr>
<td></td>
<td>Section 2.2</td>
<td>Merges previous sections 2.2, 2.3, 2.4 (Mazzetti)</td>
</tr>
<tr>
<td>1.0</td>
<td>all</td>
<td>Applied correction suggested by reviewers</td>
</tr>
</tbody>
</table>
# Table of Contents

1 **INTRODUCTION (LEADER: GIUNITILABS)** ................................................. 4
  1.1 WP5 Scope in DIP3 ................................................................................. 4
  1.2 DESCRIPTION OF THE DESIGN PROCESS.................................................. 4
    1.2.1 Strand one: Benchmarking .................................................................. 5
    1.2.2 Strand two: Scenarios ....................................................................... 5
    1.2.3 Strand three: Aggregation ................................................................. 5

2 **KNOWLEDGE SHARING (LEADER: UHANN)** ....................................... 5
  2.1 WRAPPING OF THE WORK DONE SO FAR IN THIS DIRECTION................. 6
  2.2 USER AND APPLICATION CONTEXTS ....................................................... 7

3 **SCENARIOS AND REQUIREMENTS (LEADER: US+ UHANN)** ................... 7
  3.1 THE SCENARIOS PRODUCED ................................................................... 7
    3.1.1 Scenario Higher Education / Need for Interoperability ....................... 7
    3.1.2 Scenario “Collect Information with Peers” ...................................... 9
    3.1.3 Scenario “Display a Simple search and Information Associated” ........ 10
    3.1.4 Scenario “Add a Resource to My Favourites (LearnWeb2.0 toolbar)” .... 11
    3.1.5 Scenario “Discovery Resources” ..................................................... 12
    3.1.6 Scenario “Grouping Web Resources” .............................................. 12
    3.1.7 Scenario “Choosing a Good Java Textbook Using Recommendations” .. 14
    3.1.8 Scenario “Sharing resources for collaborative authoring using Google Docs” .. 15
    3.1.9 Scenario “Download a Resource” ................................................... 16
    3.1.10 Scenario “Upload/Publish/Share a Resource” ................................ 17
    3.1.11 Scenario “Identity Management” .................................................. 17
    3.1.12 Scenario “Improving Knowledge in Statistics” ................................ 19
  3.2 ELICITATION OF REQUIREMENTS ....................................................... 20
    3.2.1 Challenge 1: Integration and Interoperability .................................... 20
    3.2.2 Challenge 2: Identity Management................................................. 21
  3.3 BASIC REQUIREMENTS EXTRACTED FROM SCENARIOS ....................... 22
    3.3.1 Functional Requirements ............................................................... 22
    3.3.2 Non-Functional Requirements ....................................................... 26

4 **INTERACTION MODELS (LEADER: GIUNTI + ALTRAN)** ....................... 28
  4.1 THE ISSUES ........................................................................................... 28
    4.1.1 Usefulness ...................................................................................... 28
    4.1.2 Reliability ....................................................................................... 28
    4.1.3 Stimulation ..................................................................................... 28
  4.2 THE CONCEPT ..................................................................................... 28

5 **CONCLUSIONS** ......................................................................................... 29

6 **REFERENCES** .............................................................................................. 29
1 Introduction (Leader: GiuntiLabs)

1.1 WP5 Scope in DIP3
WP5 aims at stimulating knowledge sharing, knowledge management and the conversion of information into knowledge in communities of practices. Building on the work carried out over the first two years of the project, with a stronger orientation on repurposing available web2.0 tools (see the year-2 Activity Report), the following core services will be integrated and where necessary developed: discovery, aggregation of resources, recommendation, categorising and tagging, setting up discussions, download and upload resources. Towards the implementation of these services we will focus on two main issues: managing the identity of users and conceiving grouping strategies.

1.2 Description of the design process
This task's aim is twofold: devising scenarios for supporting knowledge sharing in communities of practices and modelling their interactions (people-resources-people). Envisioning scenarios to narratively describe the features of the services giving the idea of how they could be implemented to support people's activities. Models of interactions will disclose and represent the possible relations among users and resources aiming at supporting sharing activities. The final aim will be to trigger and support user participation in knowledge sharing, such as rating and commenting quality, usefulness and relation to competence level, allowing users to group resources, topics and topic areas as well as setting up discussions around resources/topics.
1.2.1 Strand one: Benchmarking
The first strand of this process is devoted to selecting available tools and services for knowledge sharing, focussing on already existing suitable services (e.g. Google Doc, Flickr,...)

1.2.2 Strand two: Scenarios
Devise envisioning scenarios to trigger and support user participation in knowledge sharing.

1.2.3 Strand three: Aggregation
Converge towards the aggregation of the services into a compelling web client supporting knowledge sharing scenarios.

2 Knowledge sharing (Leader: UHANN)
During the last three years, the Web has entered into a second phase, known as Web 2.0. New services and software have transformed the Web from being a predominantly read-only medium
to one where anyone can publish and share web contents. Web 2.0 tools promote different types of communication: one-to-one, one-to-many, or many-to-many, synchronous and asynchronous, and can be used to search, share and create different media: from text (Blogs and Wikis) to images in Flickr, audio, podcasting and video in YouTube. Given the information overload that is created by the exponential growth of content on the Web, other tools help learners filter and manage information (social bookmarking and RSS feeds). The use of these services has provided new means to share knowledge, exchange ideas and easily publish work.

Social software at its core is based on supporting individuals to interact socially and to achieve their personal goals, together with people who have similar interests. It works bottom-up: people sign up to a system and form communities through personal choice and actions. Their desire to organize themselves into groups and to collaborate by advancing personal interests contrasts with more traditional approaches where people are placed into organizationally or functionally-defined groups.

In contrast, traditional LMS still approach group membership in a top-down fashion. In current learning environments and in corporate settings, it is hard to imagine a single person acting without some specifically assigned membership (in a class, a working group, a team or a division). Social software will change the traditional way in which learning systems, groupware and other project-oriented collaboration tools work. People start using social software individually; they advance their own biases and connections, and reflect them in social relationships in everyday life. This process is not organized in terms of a single, clearly defined project; rather, it is a people-driven world, in which social interactions are inductive, passing from individual to a group, to other people and other groups. This approach may appear untidy and approximate, but often is a better method towards forming strongly motivated groups and working teams.

In our project context we want to address the following questions:

- What happens, if social software is used in formal learning or work environments, and how can it extend the functionalities of traditional learning or work environments?
- How can the essential elements of social software be incorporated into more conventional software solutions, ultimately transforming learning communication and working collaboration, and which challenges do we have to address to achieve this integration?

The use of Blogs, Wikis, media-sharing services, and other social software, has been shown to create exciting new learning opportunities for people, and to support creation of social networks and communities of practice among company employees [1, 5]. The learner is seen as a participant who is actively engaged through a rich set of interactions within these communities. At the same time, the worker must fulfill three workplace roles: working, learning and collaborating with other colleagues. In this paper, we sketch, in a scenario-oriented way, how people can interact in their working environment to create, search and share knowledge resources [3, 4, 5].

### 2.1 Wrapping of the work done so far in this direction

To integrate models and tools for creation, storage and exchange of knowledge resources, in the first project stage we implemented the KRSM infrastructure [2] making information accessible to better support lifelong learning and enhance learning experience. This infrastructure brings together information stored on institutional servers, centralized repositories, locally on learner desktops (by means of P2P technology) and online community-sharing systems like Flickr and YouTube. The KRSM architecture is depicted in Figure 1.
In the next project phase we plan to extend the integration of Web 2.0 applications to support a variety of scenarios, one of which is described in the next section. Whereas in the current KRSM architecture Web 2.0 applications like Flickr and YouTube are only considered to be information sources for existing services, in the next project stage we aim to combine functionalities of existing Web 2.0 applications (tagging, bookmarking and commenting) into an integrated LearnWeb 2.0 platform for sharing, discussing as well as for (possibly collaborative) creation of knowledge resources.

2.2 User and application contexts

User participation in knowledge sharing is mainly individual. Each user has an account and a profile. Each user can belong to a group (e.g. a school class), as well as each resource can belong to one or more groups (categories).

The application context is informal and friendly. Users should be stimulated to participate as much as possible in order to improve the effectiveness of sharing.

Supported activities include: discovery and recommendation of resources (based on ratings and comments of quality, usefulness and/or difficulty), aggregation (based on groups of resources, topics or topic areas), collaborative creation/modification of resources (e.g., via wikis or google docs) set of discussions around resources or topics as well as download and upload resources from/to Internet.

3 Scenarios and requirements (Leader: US+ UHANN)

3.1 The scenarios produced

3.1.1 Scenario Higher Education / Need for Interoperability

Our main actors in this scenario are ICT technicians who support different projects and people in a university environment. ICT is used at different levels of the organization: work (store and share knowledge resources), teaching (present and provide learning materials), learning (for
workers and students). Dynamic changes in the technical infrastructure, like e.g. hard- and software development require technicians to develop new competences to cope with continuously evolving environment. Thereby information sharing plays a central role to allow for speed up of the required competence development.

Each technician works on several tasks, but communication among colleagues is only performed in person or via e-mail, without any synchronization or support. Too often, a technician does not know what the others are doing, even if their work is relevant for her tasks. Resources are stored in different databases, which are neither linked nor interoperable; discussions take place ad hoc and are not stored; best-practice transfer is manual and ad hoc, and is not supported by any system.

Figure 2 depicts some of the technicians’ tasks and contextualizes some possible applications of Web 2.0 tools to foster information exchange and thus speed up competence development. For instance, Blogs and Wikis can provide cross-project communication, creating a useful knowledge repository and allowing easier review, reporting and sharing of activities. A technician could then browse and subscribe to them through RSS to keep updated on relevant news. Web 2.0 tools can be used to foster interactivity, communication and collaboration.

Other examples include the adoption of Instant Messaging to communicate and share information within the technicians’ group at lower costs (compared with telephone calls) and the advantage of saving data in a chat. RSS feeds and social bookmarking can help to track changes and news, replacing the staff newsletter. Blogs and social annotation tools support project discussions and development of technical plans. The instant, secure and constant accessibility of data in searchable format, which Blogs provide, can be a huge productivity improvement in sharing information. Wikis and mind mapping can help in creating a knowledge base of good practices and preparing the agenda for delivering the minutes. Forums can be used to keep track of courses and exams procedures as well as Podcasts as an alternative or supplement to traditional face-to-face training activities, to facilitate IT support for university employees and part-time students.

Also using new means of sharing between different institutions can provide useful insides by adapting new technologies:

- social bookmaking in existing online library services (including online catalogues and online information resources such as e-journals)
- video-conferencing and content sharing to customize university courses
- reuse and sharing knowledge resources and tools between different universities to improve the quality of learning
- forums to allow students to share their experiences with evolving learning environment.

Traditional information systems like LMS should blend with Web 2.0 applications in order to create new environments that reshape information processes and flows and connect competences. The objective is to allow users to invest as much of the available effort as possible in the production of rich interaction, resulting in an optimal collaborative load. The use of social
Software applications fosters the sense of community and group motivation, supporting lifelong competence development.

Providing integration and sharing among these different kinds of tools is crucial, though. Already with current ERP solutions, integration of diverse systems turned out to be a challenge. Applications "do not converse"; they do not share data and do not concur to re-use services or applications in a uniform / interoperable way.

With Web 2.0, we have to integrate new application types into this already complicated environment, to provide functionalities for knowledge resource sharing and exchange. Retrieval of heterogeneous knowledge resources among different tools and social network services is still too difficult. We would like to collect relevant information about different knowledge resources, gathering them in an integrated environment from where they can easily be accessed. This should be provided via a distributed and modularized infrastructure, but allow some means of centralized user authentication or Single Sign On (SSO) functionalities, to avoid authenticating several times for each tool integrated in our environment.

3.1.2 Scenario “Collect Information with Peers”

Description:

John wants to write a report about the effect of Web 2.0 on the infrastructure of their university. He creates a topic area “Impact of Web 2.0” in LearnWeb2.0 and asks his colleagues to share and discuss information with him.

Initially, John finds some interesting resources on the Web. While viewing the resources in the browser he bookmarks them (adds the resources to the LearnWeb2.0 repository) with a simple click on a button on the LearnWeb2.0 toolbar. Also he highlights some interesting paragraphs and makes some comments using sticky notes. All this is done in the browser using the LearnWeb2.0 toolbar while viewing the resources.

Peter, a friend of John, has many bookmarks about Web 2.0 tools on his local browser. Using the LearnWeb2.0 toolbar he imports these bookmarks in LearnWeb2.0.

Sue, interested in social networks, adds some new resources and makes useful comments on John’s ideas using sticky notes. She rates and tags resources with ease using the toolbar. Other colleagues also collaborated by adding more resources, commenting, rating and tagging resources.

All friends continuously discuss the topic in the forum.

John writes a draft version of the report and shares it in LearnWeb2.0 or in GoogleDocs. His friends make some comments and suggestions for improvements. Finally, the report is written and John thanks his colleagues for the collaboration.

Activity model:

1. Create a topic area
2. Bookmark (add) resources
3. Share resources
4. Highlight parts of resources
5. Comment (annotate) resources
6. Tag resources
7. Rate resources
8. Discuss in forum

9. Collaborative authoring

Actors and Roles:
- John – a PCM user
- Colleagues – PCM users who collaborate in collecting information and writing the report

Keywords:
- Social Annotation - highlight, annotate, share and interact on any webpage
- Tagging and rating
- Collaborative authoring

Enabling technologies:
- Browser toolbar or bookmarklet, which allows bookmarking (adding resources), highlighting, commenting, tagging and rating resources while viewing resources in browser (see Diigo for examples).
- Forum
- Collaborative authoring tool

3.1.3 Scenario “Display a Simple search and Information Associated”

Description:
John is an undergraduate student at the University. For his thesis in “Media and Society” he decides to gather some literature about the “web2.0” and its most important features. He tries the LearnWeb2.0 web tool. He downloads the application and install the LearnWeb2.0 toolbar on his browser.

He types in the word “web 2.0” in the space for search of the homepage. The list of results appears. The default order of presentation is per rating, however per each resource founded are also displayed the following information: the title, a short description (origin), the format (jpg, .pdf, …), the date of creation/modification, the popularity, the author, the tags associated, the TENC communities which share the resource, the people who saved the resource.

When clicking the information associated to the resource (the title, a short description, the format -jpg, .pdf, …-, the popularity, the author, the tags associated, the TENC communities which share the resource, the people who saved the resource), the results are re-ordered according to the information:

<table>
<thead>
<tr>
<th>Title</th>
<th>Format</th>
<th>Author</th>
<th>Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>alphabet order</td>
<td>blocks of resources of the same format</td>
<td>alphabet order</td>
<td>alphabet order</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>People</th>
<th>Popularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources sharing the same tag (explore the possible visualisation in the “look and feel” scenarios)</td>
<td>by date (displaying comments)</td>
<td>from the most clicked to the lesser</td>
</tr>
</tbody>
</table>

Activity model:
- Install the LearnWeb2.0
- Set preferences
- Search
• Browse the information found

**Actors and Roles:**
• User (perform a search)

**Keywords:**
• keyword search
• information associated

**Enabling technologies:**

### 3.1.4 Scenario “Add a Resource to My Favourites (LearnWeb2.0 toolbar)”

**Description:**
John is an undergraduate student at the University. For his thesis in “Media and Society” he decides to gather some literature about the “web2.0” and its most important features. He tries the LearnWeb2.0 web tool. He downloads the application and install the LearnWeb2.0 toolbar on his browser.

He digits the words “web 2.0” in the space for search. The list of results appears. John clicks on a .pdf paper and the document opens. In the LearnWeb2.0 toolbar John sees an active “working area”, a toolbox displaying the editing possibilities he has: bookmark, download, comment.

He decides to bookmark the article. A pop up appears showing different fields to fill in:

<table>
<thead>
<tr>
<th>Name of the resource (automatically filled, but editable)</th>
<th>Tags associated (editable: add/delete tags)</th>
<th>Share resource (by default) can be deselected to maintain the resource private</th>
</tr>
</thead>
</table>

John set his LearnWeb2.0 preferences to add his bookmark to delicious, so the resource is automatically added to delicious.

**Issue: how can John bookmark if he is not a delicious user?**

**Activity model:**
• Install the LearnWeb2.0
• Set preferences (add bookmarked to delicious)
• Search
• Fill in the resource description
• Add to delicious

**Actors and Roles:**
• User

**Keywords:**
• bookmark
• edit information

**Enabling technologies:**
• How to connect to delicious?
3.1.5 Scenario “Discovery Resources”

Description:

John is an undergraduate student at the University. For his thesis in “Media and Society” he decides to gather some literature about the “web2.0” and its most important features. He tries the LearnWeb2.0 web tool. He points to the LearnWeb2.0 web site, downloads the application and install the LearnWeb2.0 toolbar on his browser.

He searches for web 2.0 and a list of results appears. John notices that beside the list, there are two boxes: the first contains a list of communities and a cloud of tags related to the search he did. He quickly skims the results, but then he decides to explore the resources in a more casual way, and he starts following the tags. Once he clicks on one of the tag cloud words, the list of results is updated. The new list of results determines also the update of the cloud and the communities.

In this way John follows the other people comments, tags and gathers different resources useful for his work, in a more explorative way.

Activity model:

- Install the LearnWeb2.0
- Set preferences (add bookmarked to delicious)
- Search
- Visualise the resource
- Click a tag associated
- Visualise the new results
- Click a community
- Visualise the resources associated to the community

Actors and Roles:

- User

Keywords:

- explore
- discovery approach

Enabling technologies:

3.1.6 Scenario “Grouping Web Resources”

Description:

Tom teaches a course about "Semantic Web" at the university and is using LearnWeb2.0 to provide the students with useful learning material. For each lecture he creates at least one group using LearnWeb2.0 GroupMe! component for resource aggregation, in which he adds the lecture slides, exercises, and other relevant (multimedia) resources, e.g. Tom's “Semantic Web introduction” group contains a YouTube video, in which Tim Berners-Lee, the inventor of the Semantic Web, gives an introductive interview. The group also contains an image of the Semantic Web architecture, an essential article, important web sites, etc.

Simplicity of the LearnWeb2.0 system and specifically LearnWeb2.0 GroupMe! component is a main reason why Tom appreciates the system. Groups are created via drag & drop operations: Resources of interest can be easily dragged into a group and then moved to a specific location. Furthermore, Tom can add Web resources via a browser button while navigating the internet.
Maria is a student and participates in Tom's course. She enjoys the visual presentation of GroupMe! groups. Resources are visualized according to their media type. Hence, Maria can immediately watch videos, view images, check the latest headlines of news feeds, etc. For short, she is able to grasp the group's content at a glance.

In context of the Semantic Web course, Maria has built her own groups that consist of resources (and groups) Tom has published and moreover of new resources she has discovered. Maria's groups reflect her personal view on a particular topic.

LearnWeb2.0 GroupMe! component also enables Maria to subscribe to groups she is interested in, so that she will be notified at their personal GroupMe! page or within her news reader application whenever the group is modified.

GroupMe!'s tagging functionality allows Maria and Tom to organize their own content or content they are interested in. Based on the tags Maria and Tom are using, they retrieve recommendations for new groups and resources that have been added to the system. Furthermore, the novel group structure allows advanced strategies to explore and search for content.

**Activity model:**

1. create a group about a topic
2. add arbitrary multimedia Web resources to the group
3. arrange Web resources with a panel (move and resize them)
4. assign tags to the group and its resources
5. subscribe to groups of other users
6. retrieve notifications whenever something happens
7. retrieve recommendations for interesting new resources/groups/users
8. modify and improve own groups

**Actors and Roles:**

**Tom:**
- expert on a certain topic
- creates high-quality groups about this topic

**Maria:**
- student who wants to learn about a topic
- consumes content and subscribes to groups
- creates her own groups (= personal view) about a topic

**Keywords:**
- social resource sharing
- social bookmarking
- social annotations
- social recommendations
- improved folksonomy-based search

**Enabling technologies:**

- Browser button / Bookmarklet, which enables to add resources to groups and tag them while browsing the internet
- Advanced (possibly personalized) folksonomy-based search and ranking algorithms
- novel UI paradigms
- GroupMe! system:
Interaction models and requirements for a knowledge sharing scenario

- Ajax-based user interface
- J2EE application
- Database backend

3.1.7 Scenario “Choosing a Good Java Textbook Using Recommendations”

Description:
Robert is a second year student in IT and a very good programmer in C++. As he is interested in Web programming he wants also to learn Java. Robert is quite familiar with programming techniques and object-oriented concepts so he is seeking for a good textbook for Java that briefly introduces Java so he can see the differences with C++ and then discusses more advanced topics.

The search in Amazon.com returns a huge list of books for Java. Knowing that the Tencompetence site has an advanced LearnWeb2.0 web tool with recommendations and social networks (he used it when he studied C++) Robert logs in the TenCompetence site where he finds a topic area “Java programming”. Then he browses the resources in LearnWeb2.0 associated with the topic area. First he filters the resources by type (book) but still the list of resources is quite large.

Then Robert sorts the list of resources by their user rating (he is interested in good books). But the resources with high ratings are still too many. Robert notices that besides the general user rating resources have additional Likert scale ratings such as: easy/difficult, beginner/advanced, useful/useless, popular/scientific, etc. These ratings and the user tags associated with the resources help Robert to skip out introductory Java books. Finally, with a simple click Robert views the user comments as he is interested in what other users say about the selected resources. This helps Robert to find the desired Java textbook.

After reading the book Robert also rates and tags the resource in LearnWeb2.0 and gives useful comments using browser toolbar buttons/bookmarklets.

Activity model:
1. Create a topic area
2. Upload document to Google Docs
3. Share document in Google Docs and LearnWeb2.0
4. Comment (annotate) resources
5. Tag resources
6. Rate resources
7. Discuss in forum
8. Collaborative authoring

Actors and Roles:
- Robert - a second year student in IT who wants to find a good Java textbook

Keywords:
- Ratings
- Likert scale ratings
- User comments

Enabling technologies:
- Advanced Ajax-based Web interface
3.1.8 Scenario “Sharing resources for collaborative authoring using Google Docs”

Description:

The teacher Richard wants his students to write collaboratively a report on the social effect of Web 2.0. He creates a topic area “The effect of Web 2.0” in the TenCompetence site. Then he asks his students to join in, search the Web for relevant resources and bookmark (add) them to LearnWeb2.0 (this is done easily through the LearnWeb2.0 toolbar/bookmarklet when viewing the resource in a browser).

Richard creates a MS Word draft document containing the structure of the report. Then he uploads it to Google Docs through LearnWeb2.0 (implemented using Google Documents List Data API). Alternatively, if Richard has installed the LearnWeb2.0 MS Office plug-in, he can upload the document from within MS Word by clicking on button on the toolbar. The URL of the uploaded document is added automatically to the resources in LearnWeb2.0 and so the students can open the document through a single click.

Richards shares the report in Google Docs by inviting his students as collaborators. He also opens a topic in the TenCompetence forum to discuss the report. To do this Richards simply clicks a link (button) and LearnWeb2.0 automatically creates a topic in the forum with title - the name of the resource.

The students find and add relevant resources to LearnWeb2.0, rate and tag them, add comments (sticky notes) to them, discuss the topic in the forum and collaboratively write/modify the shared report in Google Docs.

Activity model:

1. Create a topic area
2. Upload document to Google Docs
3. Share document in Google Docs and LearnWeb2.0
4. Comment (annotate) resources
5. Tag resources
6. Rate resources
7. Discuss in forum
8. Collaborative authoring

Actors and Roles:

- Richard – a teacher who initiates sharing and collaborative authoring
- Students – collaborators in writing the report

Keywords:

- Sharing
- Collaborative authoring
- Discussion

Enabling technologies:

- Google Docs
- Google Documents List Data API – allows: authentication, retrieving a list of documents, uploading documents, trashing a document, searching documents
- Google Data APIs
• MS Office plug-in

### 3.1.9 Scenario “Download a Resource”

**Description:**

John is an undergraduate student at the University. For his thesis in “Media and Society” he decides to gather some literature about the “web2.0” and its most important features. He tries the LearnWeb2.0 web tool. He downloads the application and install the LearnWeb2.0 toolbar on his browser.

He digits the words “web 2.0” in the space for search. The list of results appears showing different fields.

<table>
<thead>
<tr>
<th>Name of the resource</th>
<th>Tags associated</th>
<th>Categories associated</th>
</tr>
</thead>
</table>

John clicks on a category. A new list or results appears. John doesn’t find on the results anything related to his interests and goes back to the initial list. He clicks on a tag. A new list of results associated with this tag appears. John click on a .pdf paper and the document opens and it is related to his interests.

He decides to download the article.

**Issue:** John can look for a .pdf resource but he can also search for a web 2.0 tool that allows him to bookmark his literature (ex. delicious).

**Activity model:**

![Activity model diagram]

**Actors and Roles:**

- User

**Keywords:**

- search/find/explore
- filter/sort
- download

**Enabling technologies:**
3.1.10 Scenario “Upload/Publish/Share a Resource”

Description:
John is an undergraduate student at the University. For his thesis in “Media and Society” he writes a summary about the Web 2.0 tools in e-learning. He wants to upload the file in order to access it from anywhere.

He decides to add the tags e-learning and web 2.0 and categorize this file in his own category called “Thesis-Docs”.

He uploads the article.

Activity model:

Actors and Roles:
- User

Keywords:
- tag
- filter/sort
- upload/publish/share

Enabling technologies:

3.1.11 Scenario “Identity Management”

Description:
Marco has just finished his computer science study at the university and he is looking for a job as a software engineer. As a job seeker, he must arm himself with as much information as possible about potential jobs, including salary comparison data and company research along with tips on everything from writing his cover letter to nailing the first interview. He uses the Web to look for job advertisements and information about companies’ requirements (knowledge and skills) and LearnWeb2.0 system to find out how much he has to learn to be fit for the role. In general it is not difficult to find job offers in computer science industry, but usually specific skills are required:
Marco was a good student and had extensive Java courses, learned GUI and software design, however he is not fully versed as regards acronyms or the required software skills and technologies (e.g. RDF, JESS, Cocoon, Apache Turbine, Enhydra). Now he would like to deepen his knowledge about associated matters. So he uses different components of the LearnWeb2.0 system to find the answers he needs. Marco is a scrupulous youngster and wants to verify every potential employer’s legitimacy whether at an individual or corporate level before submitting personal information via interviews or job applications. He uses LearnWeb2.0 system to find information about specialization/postgraduate courses, software applications or Web agencies that offer advices and tips about career and salary.

Marco realizes that social software is a good mean for sharing informal knowledge resources and collecting information from other people. The reason for Marco to use LearnWeb2.0 system is its simplicity and seamless integration of different Web 2.0 components and services. This way Marco is able to use many different Web 2.0 applications like e.g. Facebook or LinkedIn without logging in separately to each of them. All integrated applications are available through the LearnWeb2.0 system, such that Marco is only required to log in once and the system automatically authenticates him by the required application using his personal login information.

Using LearnWeb2.0 system Marco is on the lookout for new avenues and new information and joins some developer communities to improve his knowledge on relevant technologies and associated matters (Financial Services, Travel Services, Judicial, and Government (state / local / federal) http://nytimes.monster.com).

Like many other university graduates Marco lacks the practical work experience and thus knowledge of what exactly an employee is supposed to do in a particular job, as such knowledge is usually not provided to students by the university courses. Marco realizes that traditional Internet searches give mostly formal data and a specific description of what a company needs (e.g. advertisement), but it is not easy to find detailed information about the job itself.

Thanks to social network applications such as Facebook or LinkedIn integrated into the LearnWeb2.0 system, he keeps in touch with many old and new friends and begins to learn about other people’s experiences. In this way he learns about interesting resources he failed to discover by himself (e.g. "A day in the life": an online collection of a great series of what various professionals do after they wake up, fall out of bed, and drag a comb across their heads).

Marco knows that social software is a powerful information and communication resource, but he finds it difficult to find relevant information distributed across different applications. There is no common rule for collecting and defining data, information is not well-indexed, so it is not possible to rank pages and make structured searches. Ranking rules used for traditional web pages do not really work in a Blog and are not applicable to social software applications. Therefore he uses LearnWeb2.0 system which fills this gap providing access to the relevant resources through the central place, and supports indexing of distributed resources. Finally, Marco shares his own experiences in his Blog through the LearnWeb2.0 system. The system extracts metadata from resources provided by Marco and publishes them in his favourite application.

**Activity model:**
1. authenticating with different applications  
2. searching for information  
3. communicating with people  
4. publishing information  

**Actors and Roles:**  
Marco:  
- university graduate searching for a position  
- sharing own experiences  

**Keywords:**  
- identity management  
- metadata extraction  
- indexing  

**Enabling technologies:**  
- Single Sign On  
- Facebook, LinkedIn applications  
- Aperture metadata extractor  
- Lucene indexing library  

### 3.1.12 Scenario “Improving Knowledge in Statistics”

**Description:**  
Susan is a researcher student in Information Science. In her department a prototype of new library tool was developed. She receives an assignment to design and conduct a user study to explore whether the new tool can help users to perform their tasks more efficiently. Susan is modestly familiar with some basic concept in statistics, but she doesn’t feel confident how to design experiment and questionnaires on her own. She wants to learn more how raise research hypothesis, design the related experiments and questionnaires.  

Susan logged in the TEN-competence web site to search for related resources. A general search in all categories returns a long list of related recourse. After viewing a few of them she decided that probably she could not be able to learn it on her own and decided to attend a course. Then she selected the category “courses” and searches again. The search system suggested her three forthcoming courses related to “experiment design” entitled as follows: 1. “Introduction to Statistics”; 2. “Experiment design and statistical data analysis”; 3. “Practical data analysis”.  

She studied the syllables of the above course. All of them were teaching to some extent experiment design but she notices that the target students of the courses are as follows: 1. - Applied math students; 2 - Natural science student; 3. - Social science student. The first one seems to be very theoretical and she decided that it is not appropriated to her. However she couldn’t decide which one of the rest two courses is more appropriated for her and send a message to data analysis discussion group. She receives a few recommendations from other student, which were rather contradictory recommendations. Then she contacted one of the lecturers in these courses for advice which one to select. She arranged an online meeting with him.  

**Activity model:**  

1. Search for appropriate recourse (course, tutorial, text books and etc.)
2. select/decide
3. study
4. exercises
5. pass the test
6. if satisfactory then finish
   else go to 1
   - collaborate with colleagues in 1-4.

Actors and Roles:
- The R. Student
- Colleagues (TENC users)
- Instructor/lecturer

Keywords:
- semantic search
- social recommendations
- ambient intelligence
- personalization

Enabling technologies:
- Semantic web
- Data & text mining

### 3.2 Elicitation of requirements

Web 2.0 is a challenging environment, in which knowledge resources are distributed among a set of heterogeneous online storage tools, each providing specific functionalities. Whereas each online application supports a limited set of pre-defined tasks (like storage, editing or discussing of resources), our LearnWeb 2.0 integrated environment aims at offering a rich set of functionalities over the whole virtual working space containing the entire set of distributed resources, without unnecessary boundaries.

Existing Web 2.0 tools differ in programming languages, granularity degree of their APIs, and licensing system. Among the great number of available tools, only a few are delivered with an open source license, which allows them to be customized and seamlessly integrated in a centralized environment. Whereas many tools are copyrighted, some of them deliver their API, which allows for integration of their services in LearnWeb 2.0. When neither the source code nor an API is available, they have to be linked as external tools.

Our work on LearnWeb 2.0 will address two main challenges in the coming months: a) Integration and Interoperability and b) Identity Management. We discuss these two challenges and possible solutions in the following sections.

#### 3.2.1 Challenge 1: Integration and Interoperability

Our LearnWeb 2.0 infrastructure aims to provide rich functionalities within a single environment as a combination of services provided by Web 2.0 systems.

Content provided on Wikis, Blogs, Forums, Podcasts and other tools need to be integrated in a way that makes access to these distributed resources as easy as to learning materials in a conventional LMS, and also provide the entire set of required functionalities. Technical integration of different Web 2.0 applications can be performed at different levels. We consider three possible integration degrees: basic, partial and complete. An example of basic integration is linkage of resources provided by one application (for example, a photo in Flickr) from another
application (such as a document in Google docs). This basic integration level does not require lots of implementation effort (in fact some existing Web 2.0 applications provide such basic integration by means of links to external resources), but does not really help the user to reduce manual efforts, as all references need to be created manually. A more tight (partial) integration can be achieved by putting one common application on top of the APIs provided by the different Web 2.0 tools. Unfortunately, most of the available APIs are application specific and functionally limited, making even partial integration difficult.

Full integration (the most difficult to achieve) would result in a common system that provides the entire set of functionalities of all applications in an integrated manner. For example, in such a fully integrated system we could drag and drop a Flickr picture to a document written in Google docs. Although at the first glance full integration seems to be the most preferable choice, it needs to be performed in a modular way, preserving the ability of future updates of integrated tools. One possible approach to achieve such technical integration is to define a set of common interfaces for the core services, such as SQI [6] for search. However, heterogeneous APIs require creating specific wrappers, like SQI wrappers implemented in the LearnWeb2.0 system for YouTube and Flickr integration. One disadvantage of this approach is that such wrappers need to be created for every application to be included. Another problem is that not all required functionality of the Web 2.0 applications can be accessed through their APIs. For instance, the YouTube API does not allow for video upload.

To address this problem in LearnWeb 2.0, we currently investigate a set of core services to be fully integrated and look for suitable tools providing them. In the next phase we will extend the list of supported functionalities by adding new tools using tool- and service dependent degrees of integration. In order to support the core discussion functionality described in our scenario, we will first install freely available Wiki and Forum software on a LearnWeb 2.0 server and then connect services of the other tools like Flickr using their APIs.

Apart from integrating components on an operational level, semantic interoperability has to be provided. Currently, most of the search facilities of the available Web 2.0 applications rely on keyword search using tags. We expect more semantic search features to be added in the future, raising the question of semantic interoperability. Also, some of the tools provide more expressive query languages than the others. Some allow only retrieving single resources, whereas others like Flickr or GroupMe² support resource aggregation.

3.2.2 Challenge 2: Identity Management

LearnWeb 2.0 will need to provide means of seamless user authentication for every application it integrates. Having to log into a multitude of separate applications would, besides the generated nuisance, slow down search and learning processes significantly.

One of the popular approaches for Single Sign On in a university environment is Shibboleth³, which supports cross-institutional sharing of access controlled web resources. Unfortunately, targeted Web 2.0 tools as well as the users of the LearnWeb 2.0 are typically not a part of any specific organization, reducing the applicability of Shibboleth as a solution candidate.

Another interesting approach for SSO in a Web environment is provided by OpenID⁴, an open, decentralized, free SSO system for user-centric digital identity. Using OpenID-enabled sites, Web users do not need to remember traditional authentication tokens (username, password) for every site they want to visit. Instead, they only need to be previously registered on a Website with an OpenID identity provider. As OpenID is decentralized, any Website – regardless its institutional affiliation – can employ OpenID for users to sign in. OpenID takes advantage of already existing Internet technology (URI, HTTP, SSL, etc.) and employs identities that people have already

² GroupMe: http://groupme.org/GroupMe/
³ Shibboleth: http://shibboleth.internet2.edu/
⁴ OpenID: http://openid.net/
created for themselves in their blog, photostream, profile page, etc. OpenID does not solve all problems, though. Although many sites already support it, in order to be useful for a fully integrated environment, OpenID needs to be integrated in every included site. Thus selecting OpenID still restricts possible choices for the tools to be integrated.

An important consideration for designing LearnWeb 2.0 is that most users already have their personal accounts with many applications to be integrated. These user accounts can be accessed through the Web interface of the specific tool. Users need to access their own resources, contacts, and bookmarks already available in the Web 2.0 applications through the new integrated environment. One possibility to provide SSO for LearnWeb 2.0 is to keep authentication data required by each application encrypted in a single place (locally by the user or on a trusted server). In this way, the data for a specific application can be decrypted and used to authenticate the user, only requiring the user to provide one password for decryption. The advantages of this approach are its simplicity and independence of the target application.

3.3 Basic Requirements Extracted from Scenarios

3.3.1 Functional Requirements

The functional requirements extracted from the scenarios are based on the following activities of the user with the LearnWeb2.0:

3.3.1.1 Search

Simple Search

The user can perform a Simple Search from the home page (or any other web page) of the LearnWeb2.0. She/he enters a search phrase into the search field on the web page and clicks the search button. A list of results is displayed. Per each resource the following information can be displayed: title, author, format, date of creation/modification, popularity, rating, tags, etc. The default order of the found resources is by rating. The user can reorder the result list of knowledge resources by a simple click.

- Title → alphabetic order
- Format → blocks of resources of the same format
- Author → alphabetic order
- Communities → alphabetic order
- Tag → resources sharing the same tag (explore the possible visualisation in the “look and feel” scenarios)
- People → by date (displaying comments)
- Popularity → from the most clicked to the lesser
- etc.

If the user has installed the LearnWeb2.0 browser toolbar he can use it to perform Simple Search. This allows the user to search for resources from any web page he is browsing without first visiting the LearnWeb2.0 web site.

Advanced Search

The LearnWeb2.0 also provides Advanced Search capabilities based on the metadata (title, author, subject, format, publisher, etc.) associated with the resources. From any web page the user can select Advanced Search to adjust the search.

3.3.1.2 Browse

The user can browse the resources stored in the LearnWeb2.0 repository (or other repository). When browsing she/he must select the community, topic area (competence profile) and topic
(competence). A list of results is displayed. Per each resource the following information can be displayed: title, author, format, date of creation/modification, popularity, rating, tags, etc. The user can reorder the result list by title, author, format, date, etc. in a similar way as in Search.

3.3.1.3 Discovery of Related Resources

The LearnWeb2.0 also supports the user in the discovery of resources. Beside the list of search/browse results the LearnWeb2.0 displays two boxes: the first one contains a list of communities and the second a cloud of tags related to the search. The user can explore the resources in a more casual way by following the tags. Once he clicks on one of the tag cloud words, the list of results is updated. The new list of results determines also the update of the cloud and the communities.

3.3.1.4 Social Bookmarking

The LearnWeb2.0 should provide means for interoperability with existing Web 2.0 social bookmarking tools such as del.icio.us, Digg, reddit, etc. When performing browse or search of resources the system displays a list of resources. For each resource, besides the basic information (title, author, format, etc.), the system displays “Bookmark with” links and with a simple click the user can add the resource to the preferred bookmarking Web 2.0 tool.

An alternative way for bookmarking a resource is by using the LearnWeb2.0 browser toolbar. The user opens the resource in the browser and clicks the “Bookmark with” button. A popup appears showing different fields to fill in:

- checkboxes for selection of preferred bookmarking tools;
- name and URL of the resource (automatically filled in);
- associated tags;
- Share resource (by default) can be deselected to maintain the resource private;
- other additional information such as comments, annotations, etc.

After filling the needed information the resource is automatically added to the selected bookmarking tools using the corresponding tool APIs.

Where possible, the LearnWeb2.0 system should provide means for importing resources that are already bookmarked by the user in del.icio.us, Digg, reddit or other Web 2.0 tool.

3.3.1.5 Aggregation of Resources

The aggregation of the resources in LearnWeb2.0 is based on groups of resources, topics or topic areas.

The standard grouping of resources in the LearnWeb2.0 repository uses the TENC topic areas (competences profiles) and topics (competences). When the user adds a resource to LearnWeb2.0 she/he must specify a corresponding topic area and topic. The LearnWeb2.0 should provide means for easy reorganization of the resource – creation of a new topic area or topic, moving selected resources into another topic area or topic, etc.

In addition, the LearnWeb2.0 system can include a GroupMe! Component to organize the resources into groups. Groups can be created via drag & drop operations: Resources of interest can be easily dropped into a group and then moved to a specific location. Furthermore, the user can add Web resources via a browser button while navigating the internet. The resources in GroupMe! are visualized according to their media type. Hence, the user can immediately watch videos, view images, check the latest headlines of news feeds, etc. and she/he is able to grasp the group's content at a glance.
3.3.1.6 Sharing

The aim of the LearnWeb2.0 system is to trigger and support user participation in knowledge sharing. The user can share a resource by simply adding the resource to the LearnWeb2.0 repository. By default, the resource is shared between the members of TENC community – all of them can view/download the resource. The owner of the resource has the possibility to set rights to specific persons (or all the community) to change information or to only read information.

The LearnWeb2.0 browser toolbar can be used to simplify the sharing of the resource. While viewing a resource in the browser the user bookmarks it (adds the resource to the LearnWeb2.0 repository) with a simple click on a button on the LearnWeb2.0 toolbar.

The knowledge sharing also includes activities such as collaborative creation and modification of resources (using Web 2.0 tools like GoogleDocs, Wikis, etc.) and setting up discussions around resources/topics (using forums, blogs, social annotations, etc.).

Collaborative Creation/Modification of Resources

The LearnWeb2.0 system should include some basic support for collaborative authoring of documents using GoogleDocs and Wikis. This can be achieved in the following way. First, the user creates a draft document in GoogleDocs or a Wiki system and defines who from colleagues/friends has the rights to edit/view the document. Then she/he adds the document (in fact the URL of the document) to the LearnWeb2.0 repository. In this way the document is shared between the members of the TENC community. Whenever a user clicks the resource (the document) in the LearnWeb2.0, the document is opened in the browser in GoogleDocs or in the corresponding Wiki system. Then the user can view or edit the document if she/he has enough rights.

The process of uploading a document to GoogleDocs can be automated to some extent by using Google Document Lists Data API.

Discussions around Resources

The LearnWeb2.0 system should support an easy way to set up discussions around topics and resources. This can be achieved for example by adding a “Discuss It” link (button) next to each resource in the list of results when browsing or searching. When the user clicks the link the LearnWeb2.0 system automatically creates a topic in the corresponding TENC forum and the forum is opened in the browser. The name of the topic is automatically filled in with the title of the resource. After the topic is created other users can discuss the resource in the forum by simply clicking the “Discuss It” link (button). The users can use also use the LearnWeb2.0 toolbar to set up discussions in a more convenient way when viewing the resource in the browser.

Topics and resources can be discussed also in blogs. A “Blog It” link (button) can be added to the LearnWeb2.0 and toolbar. When the user clicks it a corresponding Blog system will automatically be loaded in the browser.

3.3.1.7 Recommendation of Resources

Recommendations of resources in LearnWeb2.0 is achieved by user ratings, tags and comments (annotations).

Ratings

Each topic area, topic and resource has a rating. Each user can easily rate the quality of a resource by clicking on the “Rate It” link (button) in the LearnWeb2.0 or toolbar. When browsing/searching resources the can easily sort the resources by their rating.
Besides the general rating, the LearnWeb2.0 system can use also additional Likert scale ratings for resources such as: easy/difficult, beginner/advanced, useful/useless, popular/scientific, etc.

To implement recommendations using ratings LearnWeb2.0 can use recommending systems for users’ preferences via collaborative filtering like ‘Taste’.

**Tagging**

Collaborative tagging (also known as folksonomy, social classification, social indexing, and social tagging) is the practice and method of collaboratively creating and managing tags to annotate and categorize content. Many Web 2.0 systems (Flickr, YouTube, del.icio.us, etc.) make use of this method.

The LearnWeb2.0 system also support user tagging. Each user can add tags to a resource or group of resources by clicking on a “Tag It” link (button) in the LearnWeb2.0 or toolbar.

Tagging can be used for recommendations of similar resources. If the user selects a resources or group of resources LearnWeb2.0 can suggest similar resources based on their tags (see also Discovery of Related Resources).

When importing a resource from a Web 2.0 tools that uses tagging LearnWeb2.0 should also import the associated tags using tool API if available.

**Comments (Social Annotations)**

The user can add comments (annotations) to each resource by clicking on the “Add Comment” link (button) in the LearnWeb2.0 or toolbar. When browsing/searching resources the user can view the comments associated with a resource by clicking on “Add Comment” link (button).

The LearnWeb2.0 system can implement comments and social annotations in a similar way as Diigo. The social annotation service introduced by Diigo allows users to add highlights and sticky notes, in situ, on any web page they read. Imagine a giant transparency overlaying on top of all the web pages. Users can write on the transparency as they wish, as private notes or public comments. And they can read public comments on the transparency left by other readers of the same page.

**3.3.1.8 Download and Upload**

When browsing/searching resources the user can download/open a resource by simply clicking on the title. Depending on the format of the resource, it can be opened in the browser, downloaded to the file system of the user and/or opened with the associated program (MS Word, MS Excel, Acrobat Reader, etc.).

When uploading a resource to LearnWeb2.0 (adding the resource to the LearnWeb2.0 repository) the user also is prompted to select topic area and topic, fill in / modify metadata information (some of this information should be automatically fill in) and add user tags.

**3.3.1.9 Personalization of the Results**

The LearnWeb2.0 uses the user preferences such as number of resources displayed, ordering, which repositories (LearnWeb2.0 repository, other repositories) and Web 2.0 tools and applications (Flickr, YouTube, etc.) to search in, etc. With simple clicks the user can change his preferences.

**3.3.1.10 Identity Management**

The identity management of the users is extremely important for the LearnWeb2.0 system and it is discussed in details in section 3.2.2 Challenge 2: Identity Management.
3.3.1.11 General Look and Feel of LearnWeb2.0

The LearnWeb2.0 should have a good, attractive graphical design, interaction design and ease of use. It should be usable for new, not so well informed users.

The web tool should use AJAX for the user interaction and will mimic the functionality that is usually found in Web 2.0 applications.

Additional requirements for the user interface are given in the non-functional requirements of LearnWeb2.0.

A LearnWeb2.0 browser toolbar and a bookmarklet should be implemented to simplify some of the interactions with LearnWeb2.0 system.

3.3.2 Non-Functional Requirements

3.3.2.1 Usability & Accessibility

- LearnWeb2.0 toolbar should allow novice users to install and operate it with little or no training.
- The end user shall be able to access any page within maximum five seconds.
- Implementations to the user interface should utilize AJAX
- Common CSS styles should be used in conjunction with XHTML
- The Accessibility issue is a must. Therefore, W3C WAI specifications should be met.
- Interface has to support different groups - The UI has to support different groups (learner / teachers / authors) and therefore should provide different views on the content (+ intuitive overviews). In other words, according to the different kind of users some activities are allowed or not depending on the rights of the user in accessing the content itself and / or the repositories where the content (i.e. the resources) is stored.
- The User Interface (UI) has to be flexible - UI should be flexible enough to allow and / or foresee future use in different contexts and delivery modes as, e.g., not only web GUIs can be devised but also Tangible User Interfaces (TUI).
- System must keep balance between individualized views and a common view - The UI should be adapted to user’s needs provided that common and standard basic features should be always available and reachable.
- The System should provide a reasonable amount of personalization from the beginning - To avoid the cold start problem it may provide soliciting data or no personalization at all.
- Cross-browser compatibility - The website should display correctly in all the major website browsers in addition to Internet Explorer e.g. Mozilla Firefox, Opera, Konquerer.
- Web 2.0 compliance architecture –
  - application interfaces, should include pre-built components and widgets that can operate independently from one another. Any event associated with these components can trigger a request for new data.
  - Events that occur within these component parts can trigger the loading of new data without a full page refresh. Only the part of the interface displaying the changed data is reloaded. The fixed portions of the interface stay put.
  - The interface is independent of data. Changes to data don’t require that the entire interface is refreshed.
Data can be served from a different domain to the interface. The server delivers the interface once and has an ongoing role in providing data.

**Issue:** In addition, since pages are updated incrementally, AJAX applications may not have distinct uniform resource identifiers (URIs) that can be bookmarked. This can also lead to issues with the Back button not functioning as expected.

### 3.3.2.2 Reliability

- fail-fast when error or mis-configured
- helpful error reporting
- continuous operation (no chronic memory leaks)
- Consistency in the view on the system - Changes in the system must not change the view of the users on the system completely in order to avoid confusion. The user must be keeping able to work on his planned activities and the system should keep track of the changes in the system to give the user a better overview
- Securing of data - Data, e.g. the domain model must be secured in such a way that no user (e.g. a teacher) can destroy the domain model by accident. This refers also to preventing from making resources repositories not available by accident with insecure access rights handling.
- Synchronization of data - In order to preserve actual data sets for the proper execution of algorithms (regarding navigation, positioning etc). This is particularly relevant for the LearnWeb2.0 system since the information regarding the resources available and the repositories reachable should be always up-to-date.
- Rollback Mechanisms - As in conventional information systems, some operations should be considered as atomic and therefore a rollback mechanism should be available.

### 3.3.2.3 Performance

- All Web pages must download within three seconds during an average load, and five seconds during a peak load.
- While executing a search, the system must be able to display 500 search results per page.
- At least the most recently used resource should be cached in order to make it readily available for further processing. Depending on resources sizes and on performance parameters that can affect LearnWeb2.0 system fruition on the local user system, it could be set a limited number of resources to be cached.
- The systems must be able to handle one hundred concurrent users with no detrimental effects on performance.
- Preview disabled by default - Previewing is normally disabled in order to speed up LearnWeb2.0 system response time (e.g. loading a resource player can be heavy in terms of system workload in a mobile scenario). But it can be set on at User will.

### 3.3.2.4 Security

The LearnWeb2.0 system for User authorisation and authentication mechanisms (e.g. in order to perform Administration activities) should be provided by the overall TENC system (i.e. WP3).

- User authentication shall be via the Single Signon system.
- Only authorized access to administrative screens. User must be authenticated through username and password control.
3.3.2.5 Internationalization /Localizability

- LearnWeb2.0 toolbar should have the ability to make adaptations due to regional differences e.g. language, date/time etc.

4 Interaction Models (Leader: Giunti + Altran)

4.1 The issues

4.1.1 Usefulness

In order to be a useful tool, LearnWeb2.0 should:

- Find resources with the same effectiveness of Google (remember: Google toolbar is few millimetres far)
- Crowded of comments, tags, rates (otherwise Google would be preferable)
- Allow a rapid resource access (few clicks from search to display)
- Show the resource content (image, text, video,...) in the same page of its attributes (tags, rates, comments,...). This strongly depends from the format: for images and video is not difficult, for Html pages or binary file it would be hard.
- Allow quick access to frequent functions (toolbar)
- Allow complete access to complex functions (MyHomePage)

4.1.2 Reliability

In order to be a reliable tool, LearnWeb2.0 should:

- Support the qualification of metadata:
  - A user can assert: “this tag is misleading”
  - A user can assert: “this comment is wrong”
- Defend from abuse:
  - “this resource should be censured”
  - “this user is unfair”

4.1.3 Stimulation

In order to have plenty of metadata the LearnWeb2.0 should:

- Reward raters/taggers/commenters
- Reward resource uploaders

4.2 The concept

Interacting with resources:
- The user accesses a personal space
- The user can access a community space
- The system provides a space to the user to browse and import resources
- The user activity onto the system impacts two levels:
  1. user space
  2. resources space
The first level is the personal space of the user. This space is characterised as a set of activities supported by the LearnWeb2.0.
The second level is that of the resources:
The resources can be stored:
   1. in the FEDORA database
   2. externally (other repositories)

The user acts at the first level. All his activities are carried on through the user interface. The system is enriched by the user's activities since each time he adds a resource to his preferences.

5 Conclusions
The web 2.0 technology is already mature for supporting the “happy-end” scenarios depicted in the above sections.

Knowledge sharing should be supported by a tool having the following features:
- Usefulness
- Effectiveness
- Pleasantness

The LearnWeb2.0 tool should also have a flexible architecture, in order to be used both as a stand-alone tool and as a service for the eLearning main tool.

6 References