

Project Report

ID5.14: Core services requirements

Workpackage	WP5
Task	5.2
Date of delivery	Contractual: 31-03-2008 Actual: 24-04-2008
Code name	ID5.14 Version: 1.0 Draft <input type="checkbox"/> Final <input checked="" type="checkbox"/>
Type of deliverable	Report
Security (distribution level)	Public
Contributors	
Authors (Partner)	Alessandro Mazzetti (Giunti Labs), Michele Dicerto (Giunti Labs), Linda Napoletano (Giunti Labs), Elena Demidova (UHANN), Sergej Zerrr (UHANN), Krassen Stefanov (SU), Alexander Grigorov (SU), Carlos Mendez (Altran), Ayman Moghnieh (UPF), Mar Perez (UPF)
Contact Person	Alessandro Mazzetti (Giunti Labs), Michele Dicerto (Giunti Labs).
WP/Task responsible	Giunti Labs
EC Project Officer	Martin Májek
Abstract (for dissemination)	
Keywords List	WP5, internal deliverable, core services, specification

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

Version	Status	Date	Author(s)
0.1	Draft	11.03.08	Linda Napoletano, Giunti Labs
0.2	Draft	03.04.08	Alessandro Mazzetti Giunti Labs
0.3	Draft	11.04.08	Alessandro Mazzetti Giunti Labs
0.4	Draft	13.04.08	Sergej Zerr, UHANN
0.5	Draft	14.04.08	Alexander Grigorov, SU
0.6	Draft	14.04.08	Sergej Zerr, UHANN
0.7	Draft	15.04.08	Mar perez, UPF
0.8	Draft	21.04.08	Alessandro Mazzetti, GIUNTI
0.9	Final	24.04.08	Alessandro Mazzetti, GIUNTI
1.0	Revised	29.09.08	Alessandro Mazzetti, GIUNTI

Summary of Changes

Version	Section(s)	Synopsis of Change
0.1	All	Draft: Table of Contents
0.2	All Section 4 Section 2	"KRSM" substituted with "LearnWeb2.0" Improved but not yet finished (Mazzetti) Added (Moghnieh)
0.3	Section 4.x	Moved matrix Resources Mngmt from 4.3.4 to 4.1.2 Added section 4.2 (Architecture) Made several improvements all over section 4
0.4	Section 4.1.6 Section 4.3.3.5	Added definition of a group Described search within groups
0.5	Section 4.1.3	Made improvements over section 4.1.3
0.6	Section 4.1.2	Extended the WebTool table.
0.7	Section 2.X	Reformulation of the text in the section 2.X Extended WebTool table with Uols, Competences... Extended of the appendix with a table relating the tools with the pedagogical contexts Final table of the web 2.0 tools selection to be review for the developers.
0.8	All	Integration of all contributions
0.9	Chapter 5	added
1.0	all	Applied corrections suggested by reviewers

Table of Contents

1	INTRODUCTION (LEADER: GIUNTILABS)	5
2	EXISTING TOOLS (LEADER: UPF)	5
2.1	TOOLS MAP	5
2.1.1	<i>Mapping pedagogical needs to primitive activities</i>	6
2.1.2	<i>Defining activity sets for K.M</i>	10
2.1.3	<i>Compatibility of existing web 2.0 services with the interaction design schema</i>	14
2.2	CLASSIFICATION OF THE TOOLS	16
2.2.1	<i>Description of the tools benchmarked and selected</i>	18
2.2.2	<i>Application of the selection criterion in a generic list of tools</i>	18
2.2.3	<i>Final list of web 2.0 tools selected for their integration in the KMS</i>	21
3	INTEGRATION OF THE TOOLS	23
4	CORE SERVICES: DESIGN SPECIFICATIONS (LEADER: GIUNTI)	23
4.1	ENTITIES	23
4.1.1	<i>Users</i>	23
4.1.2	<i>Resources</i>	24
4.1.3	<i>Metadata</i>	25
4.1.4	<i>Categories</i>	31
4.1.5	<i>Groups</i>	31
4.1.6	<i>Tags</i>	31
4.1.7	<i>Comments</i>	31
4.1.8	<i>Rates</i>	31
4.1.9	<i>Popularity</i>	31
4.2	SOFTWARE ARCHITECTURE	32
4.3	FUNCTIONALITIES	33
4.3.1	<i>User Login</i>	33
4.3.2	<i>My Home Page</i>	33
4.3.3	<i>Search</i>	34
4.3.4	<i>View and manage</i>	37
4.3.5	<i>Bookmark</i>	38
4.3.6	<i>Download</i>	39
4.3.7	<i>Upload</i>	39
4.3.8	<i>Share</i>	39
5	CONCLUSIONS	40
6	REFERENCES	40
7	APPENDIX A - LIST OF KNOWLEDGE OBJECTS	40
8	APPENDIX B - WEB-SERVICES EXPOSED BY KRSERVER	46
9	APPENDIX C - KRSM ACCESS-API-LITE	47
9.1	USER	47
9.1.1	<i>Services</i>	47
9.1.2	<i>Retrieve user metadata</i>	47
9.1.3	<i>List User configurations</i>	47
9.1.4	<i>List user tags</i>	48
9.1.5	<i>List user comments</i>	48
9.1.6	<i>List user ratings</i>	48
9.1.7	<i>Retrieve users list</i>	49
9.1.8	<i>List user's resources</i>	49

9.2	CATEGORY	49
9.2.2	<i>List subcategories</i>	50
9.2.3	<i>List category resources</i>	50
9.3	RESOURCE	50
9.3.2	<i>Retrieve resource file</i>	51
9.3.3	<i>List resource tags</i>	51
9.3.4	<i>List resource comments</i>	52
9.4	TAG	52
9.4.1	<i>List resources by tag</i>	52
9.5	SEARCH RELATED SERVICES	53
9.5.1	<i>Search by keyword based query</i>	53
9.5.2	<i>Search by category</i>	54
9.5.3	<i>Search by tag</i>	54
10	APPENDIX D - KRSM MANAGEMENT-API-LITE	56
10.1	USER	56
10.1.2	<i>Add User</i>	56
10.1.3	<i>Add User configuration</i>	56
10.1.4	<i>Add User Tag</i>	56
10.2	CATEGORY	57
10.3	RESOURCE	57
10.3.2	<i>Upload resource file</i>	58
10.4	RATING	58
10.5	TAGGING	58
10.6	COMMENTING	59

1 Introduction (Leader: **GiuntiLabs**)

This document describes in detail the core services of Knowledge Resources Sharing and Management, focusing on the main features of the system:

- Exploit Web2.0 existing tools
- Satisfying pedagogical needs
- Offering an interactive user interface
- Offering base services for other tools (from other WPs)

After a deep analysis of existing tools, a selection of the most suitable tools is performed, keeping in mind the need of future modifications.

The software architecture is a major issue of this system, in order to allow a flexible interoperability and a scalable re-use of modules.

The basic point of the system is the resource repository, playing a central role in supporting social functionalities.

2 Existing tools (Leader: **UPF**)

2.1 *Tools Map*

The integration of the web 2.0 services in learning environments should not only support the technical demands of the system in terms of functionalities, but also address the pedagogical necessities of the learners in that context. Based on this, selecting web 2.0 services for integration in learning environments should be based on functional necessities, pedagogical needs, and the focus on providing a collection of compatible services that offers the desired user experience.

We propose hereafter a framework that relates web 2.0 services for integration in learning environments with the different pedagogical needs that the learners express in different interaction contexts. A selection criterion is drawn from the framework and applied on a collection of interesting Web 2.0 services to choose the most adequate bundle offering the required functionalities for a knowledge management environment (KME). In the following, we explain step by step the construction of this framework that we have called Tools Map. It is important to remark that this is an initial approach that should be refined in a cyclic process in the near future, taking into account the conclusions from the evaluations of an integrated KME with users in future prototypes.

2.1.1 Mapping pedagogical needs to primitive activities

The first step to organize and categorizes web 2.0 tools in a learning environment is understanding and analyzing the main pedagogical needs of the learners in such as context, and identifying the related activities required to address these needs.

With the purpose of understanding the pedagogical needs and the activities related to them we analyse and merge documents that provide information about the pedagogical model definitions in TENCompetence (D2.2 [1], the Roadmap [2] and the recommendations of pilot evaluations [3]) with the scenarios and usage profiles for WP5 and the activities that conform them (D2.2 [1] and the scenarios c). This undertaking clarifies how the generic needs of system users map to sets of primitive activities supported by web 2.0 services. This also provides a clear understanding of the objectives behind using the KME in an eLearning context, and the functional role(s) that this environment is ought to play in assisting learners while they address their pedagogical needs. For example, The personalization of knowledge elements can be related to a limited set of predefined activities involving bookmarking and tagging.

By atomizing the scenarios and use cases' activities into what we consider as scenario activities, we derive a global set of activities related to the KME's general context. These activities are listed in the following table (table 1) and contrasted with other primitive activities whose definitions are generic and simple, and can be incorporated in a more abstract set of scenarios. There functional definitions have been implicitly standardized as they became are supported by a wide range of different web 2.0 services. Relating the scenarios' activities to primitive activities helps in assessing the compatibility of a given collection of web 2.0 services with the functional needs inherent in pedagogical needs reflected by the scenarios' definitions. For example, when the user Explores categories s/he is implicitly filtering and sorting the information that will be relevant for his exploration, and hence requires filtering and sorting services to accomplish the scenario's tasks.

Activities	Primitive Activities	Description of a situation
Search resources	Search/Find/Explore	Search for a concrete resource that the user already knows. (Search and find). Search for something that you don't know.
Explore categories	Filter/Sort	Click on categories or tags depending on the search in order to accurate the search or exploration.
Publish	Publish/Upload/Share	The user can upload his resources but not necessary share it with the others. He can also decide to public them and allow the others to contribute.

Bookmark	Bookmark	The user would need to save the resources he found interesting in a given moment but not classify them neither tag them, just save them.
Edit resource	Edit/Write/Create	The user can write or create a new resource. The user can contribute on adding information into an existing resource by editing or writing on it.
Rate resource	Rate	The user could rate the resources in order to share his opinion about them or to have his own ranking list.
Add tag	Tag	The user add a label to a resource with a representative concept understandable for anybody. The user add a label to a resource with a representative concept for his own classification.
Comment resource	Comment	The user could add comments in relation to a resource. He could share or not share them.
Download	Download	The user could download those resources that he find interesting.
Search per tags	Filter per tags	The user could quickly search for tags when he were searching for a particular resource or just exploring.

Table 1 Relating the scenarios' activities with the primitive activities inferred.

After analysing the main activities related to the descriptive scenarios and use cases, we study their compatibility with the set of functionalities required to fulfil the pedagogical needs that the KME is meant to answer. The deliverable D2.2 [1] details on the functional requirements of these pedagogical needs in a descriptive manner. These requirements are identified and grouped according to the pedagogical need(s) that they cover (Fig. 1). In the context of knowledge management, these pedagogical needs are generally clustered into three general domains being:

- **Knowledge mining:** is the domain where learners search, filter, and sort the resources inherent in the system or provided by it. These resources are generally treated as knowledge objects with pedagogical values that the system should be able to communicate clearly to its users.
- **Knowledge transfer:** is related with activities that contribute to the expansion of the collections of resources published in the system, and to the development of their ontological values. Activities involve adding information to an existing resource or creating a new one for sharing or publishing.
- **Knowledge personalisation:** encompasses the activities that the user performs in order to organize and sort collected resources to reach his/her objectives.

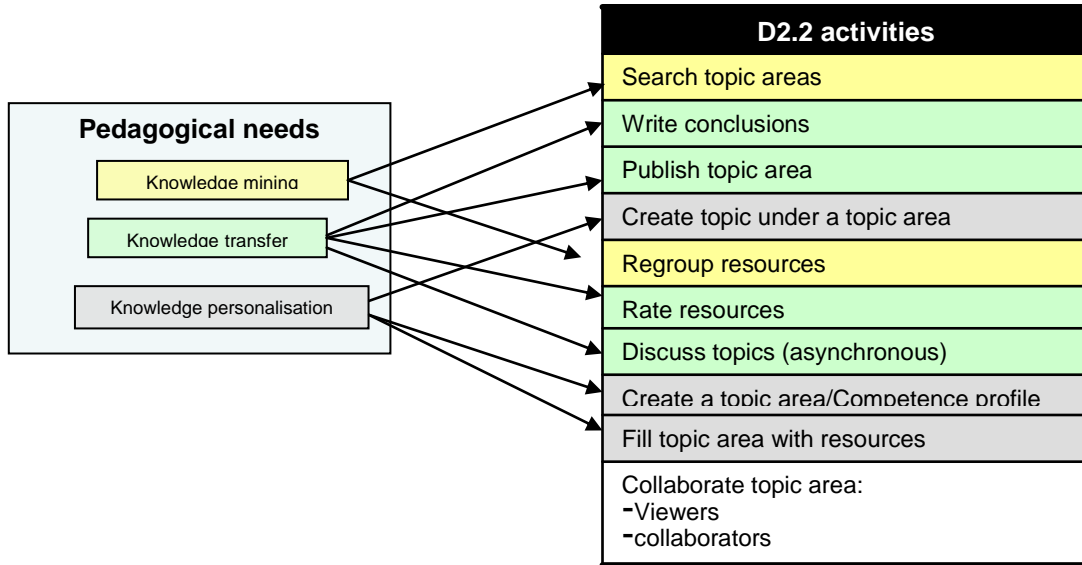


Figure 1 Identifying pedagogical activities from the pedagogical needs definitions

By decomposing each pedagogical activity in D2.2 into its primitive constituents from the list of primitive activities identified in the scenarios and use cases, the prime focus of the functional requirements with pedagogical motives becomes clear. For each domain, an exclusive set of activities can be drafted for its definition as shown in figure 1. The pedagogical needs related to the KME context do not explicitly require social collaboration and peer interaction among users. This activity hence does not belong to the list of required functionalities for supporting the KME in addressing the pedagogical needs.

We eliminate the scenario activities (in this case, collaborate) that do not link to the pedagogical needs related to the KME, and contrast the remaining activities with the primitive activities that are generically supported by web 2.0 services. This is shown in the following figure (figure 2). First, the only two primitive activities that are not included among scenario activities address the retention of knowledge units (bookmark and download). Two primitive activities, being publish/upload/share and edit/write/create, are referenced by more than one scenario activity. These activities deal with the creation and transfer of knowledge among KME users. The remaining primitive activities can be referenced as knowledge manipulation activities and their usage is usually specific to the context of use addressed in their definitions.

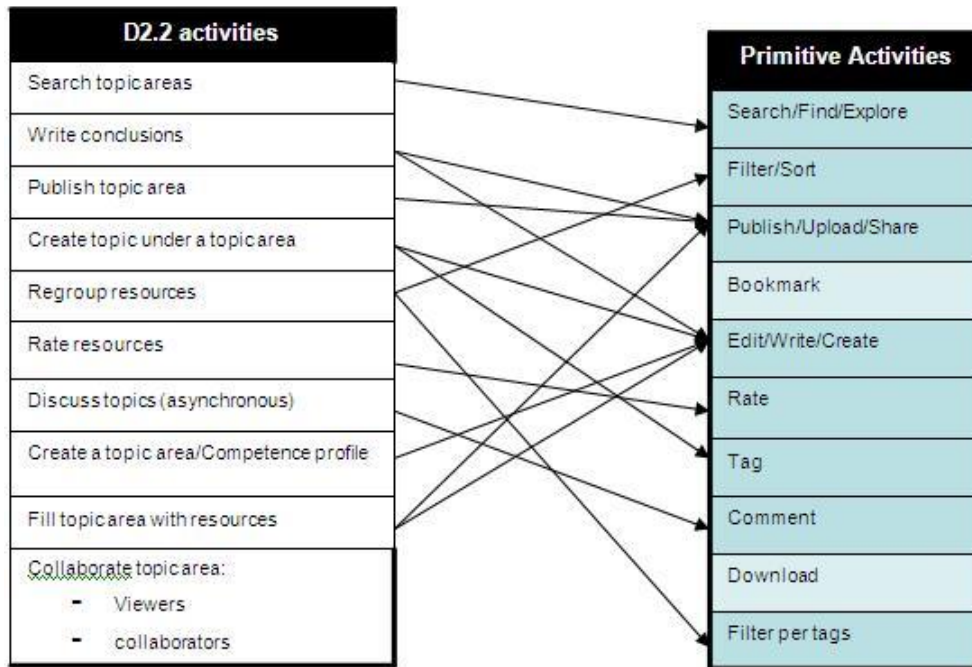


Figure 2 Relation of the activities defined in D2.2 and the implicit primitive activities

By breaking down the pedagogical needs behind the development of the KME to their functional constituents (figure 1) with the help of the drafted scenarios and use cases, it became possible to link these needs to primitive activities supported by web 2.0 services, by linking these scenario activities to primitive activities (figure 2). This mapping is shown in the following figure (figure 3) where the primitive activities have been coloured according to their mappings.

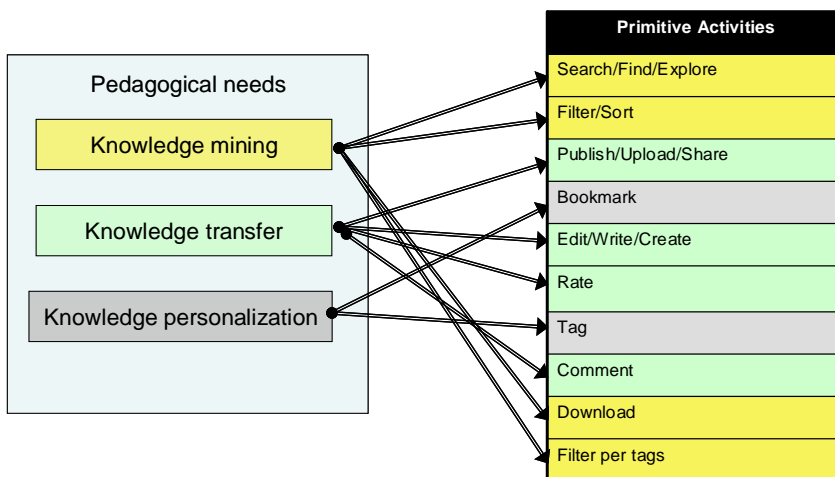


Figure 3 Mapping the pedagogical needs behind the KME to the primitive activities in web 2.0 services

Each of the three domains characterizing the pedagogical needs at hand represents a potential interaction context where the users aim to fulfil a predefined set of goals related to the domain’s definition. By default, the activities that the users performs in each of these three interaction contexts are those from the set of primitive activities that map to the corresponding domain. Hence, with each of the three interaction contexts drafted from the pedagogical needs, we associate an activity set that encompasses the primitive activities that support the users in achieving their goals in this context. This is shown in the following figure (Figure 4).

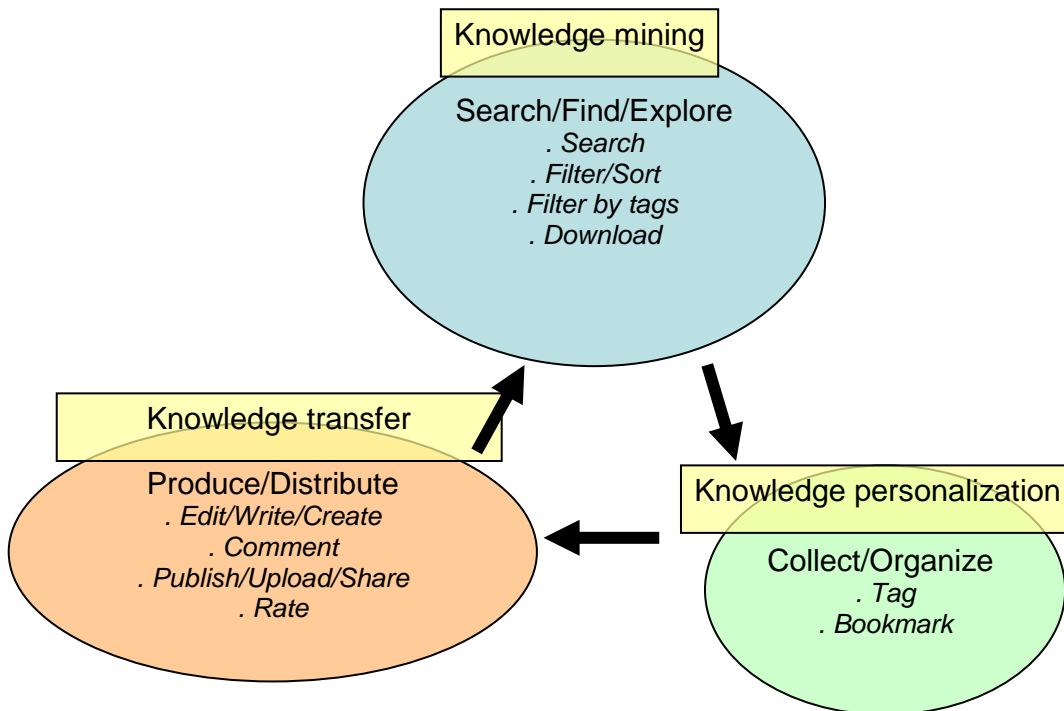


Figure 4 First definition of the interaction contexts and their related activity sets.

This first definition of the main interaction contexts is the first step for defining the interaction design part of the framework that allows the selection and classification of web 2.0 services for integration in the KME. The scenarios and use cases provided for the KME describe the steps that users take to accomplish the specific goals defined in each scenario or use case. They constitute a trial battery for refining the interaction design schema by insuring mutual independence between the interaction contexts and better compositions of the related activity sets. This process is presented in the following section.

2.1.2 Defining activity sets for K.M

An interaction context is a cluster of functionalities presented in a contextual interface container to serve a specific user objective with which the context is associated.

Examples of interaction contexts include among others, wizards, forms, and graphic canvases. As previously argued, each pedagogical need represents a context of interaction where actions and activities concentrate on answering the inherent need and its constituents. Hence, we start by defining three activity sets in correlation with the three previously discussed pedagogical needs. Next, primitive activities are clustered according to their relation with three activity sets. Such clustering maps each primitive activity to one activity set and represents a first draft of an interaction design centered on pedagogical needs (Figure 4).

Figure 5 shows the draft interaction design for the KME as deduced from the definition of the pedagogical needs and their functional requirements. This draft contains the following three interaction contexts with their activities sets that have been identified:

- Knowledge mining: An interaction context where users search for knowledge units in the system and on the web, and refines the results obtained from their formulated queries.
- Knowledge personalization: A context where users personalize the knowledge units they mined or created in the system. Such personalization consists of bookmarking the unit for future reference by system users, and tagging it for ontological classification.
- Knowledge transfer: Users work in this context to produce knowledge units and or add knowledge to existing ones. This can be accomplished through several actions being the adding of comments and/or ratings for existing units and editing them, and the production of new units (write and/or create). The units can later be made public by publishing or sharing with the rest of the user community.

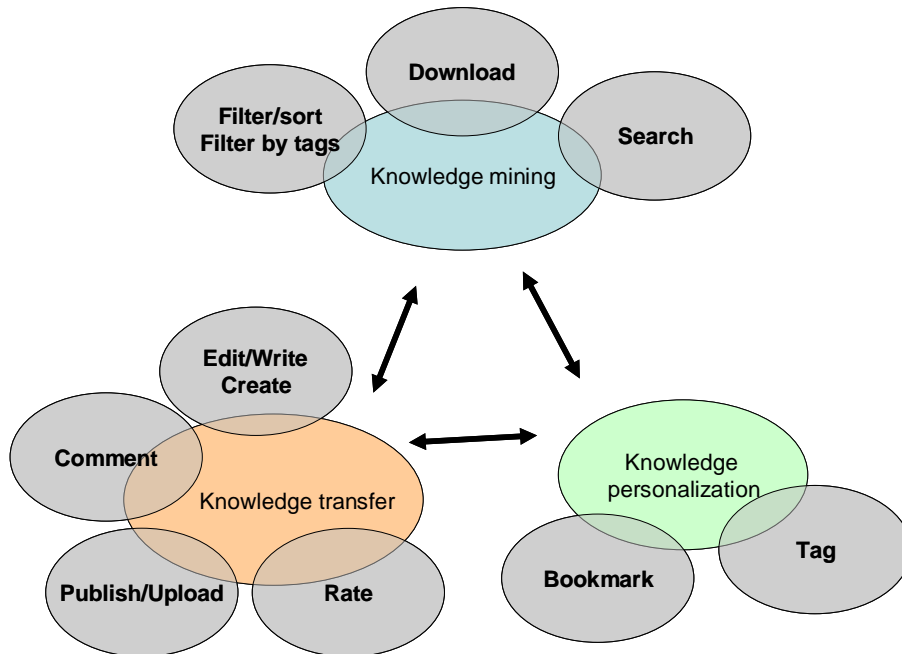


Figure 5 Draft interaction design with defined activity sets.

A successful interaction design should abide by the following rules:

- 1 Interaction contexts should be of comparable weights in term of functionalities
- 2 Each interaction context should have a unique set of functionalities
- 3 Interdependencies among interaction contexts should be minimized
- 4 The maximum number of interaction contexts should not exceed 6
- 5 The optimal number of contexts is between 3 and 5

Rule number 1 and 2 specify the relations among the interaction contexts in terms of functionalities. Each context has its own unique set of functionalities whose role is to provide users with the kind of support they expect while working on achieving goals related to this context. In order to accomplish the goals related to the KME, users may have to switch between several interaction contexts where they actuate steps along the way. Hence, the number of times that a user might change from context to another while pursuing the same goal should be minimized as stated by the third rule. The last two rules deal with the size of the conceptual model associated with the KME toolset. Users may optimally manage a system with 3 to 5 proportionally comparable interaction contexts as stated. The battery of scenarios and use cases provided for the KME is used in analyzing the effectiveness of the interaction design schema and refining it further.

Each scenario consists of a series of activities that we refer to as the scenario’s activity path. When an activity path is traced onto the interaction design schema, it reveals possible design errors such as strong dependencies along the interaction contexts and identifies activities misplaced in the wrong interaction context. When that happens, the designer has the choice of moving an activity from one context to another, creating a new context, or merging two contexts. After changing the interaction design accordingly, the new design is re-evaluated with the same battery of scenarios and progress is measured.

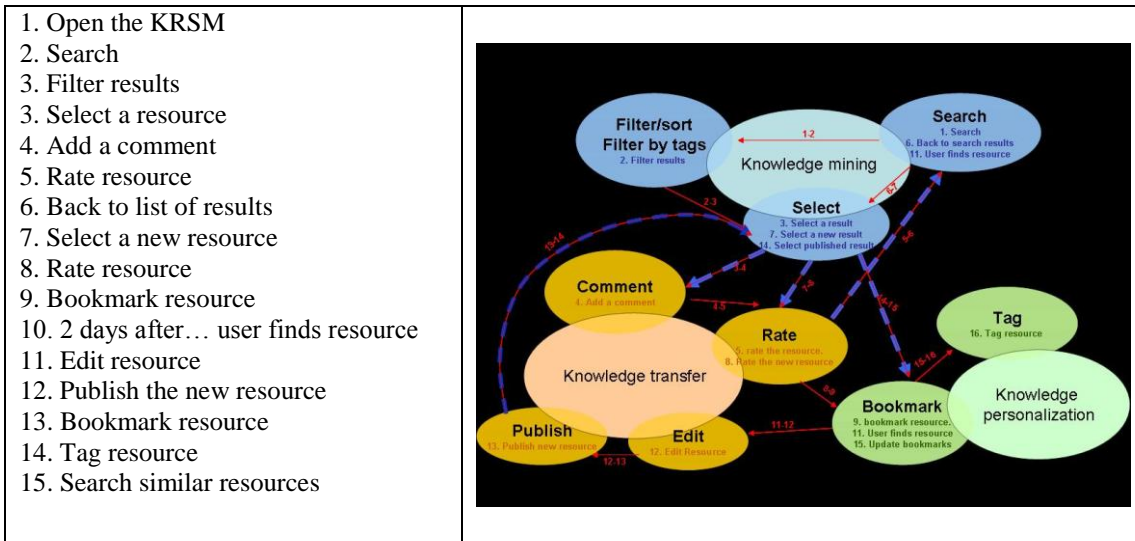


Table 1 Mapping a scenario to the activity set definition. There are many dependencies within scenarios.

In the table above, a comprehensive scenario is drawn and traced upon the draft interaction design (Table 1). This reveals strong dependencies among the contexts as the user changes contexts eight times for fourteen activities. It is also noticeable that the activity Select is the most related to changes of contexts, and that the activities Rate and Comment are more used in the context of Knowledge Personalization rather than Knowledge Transfer. These inconsistencies have also been discovered and confirmed in other scenarios.

A new interaction context is hence created to alleviate the dependencies among contexts and incorporate functionalities to visualize results and select those of interests among them. The selection of results is later transferred to one of the three other interaction contexts for processing. This change has placed the set of knowledge units that are related to the user’s overall objectives in the centre of the interaction design. The activities Comment and Rate were also moved to Knowledge Personalization where they seem to cluster better with the rest of the activities. The design was tested again with the same scenarios and finalized (Fig. 6).

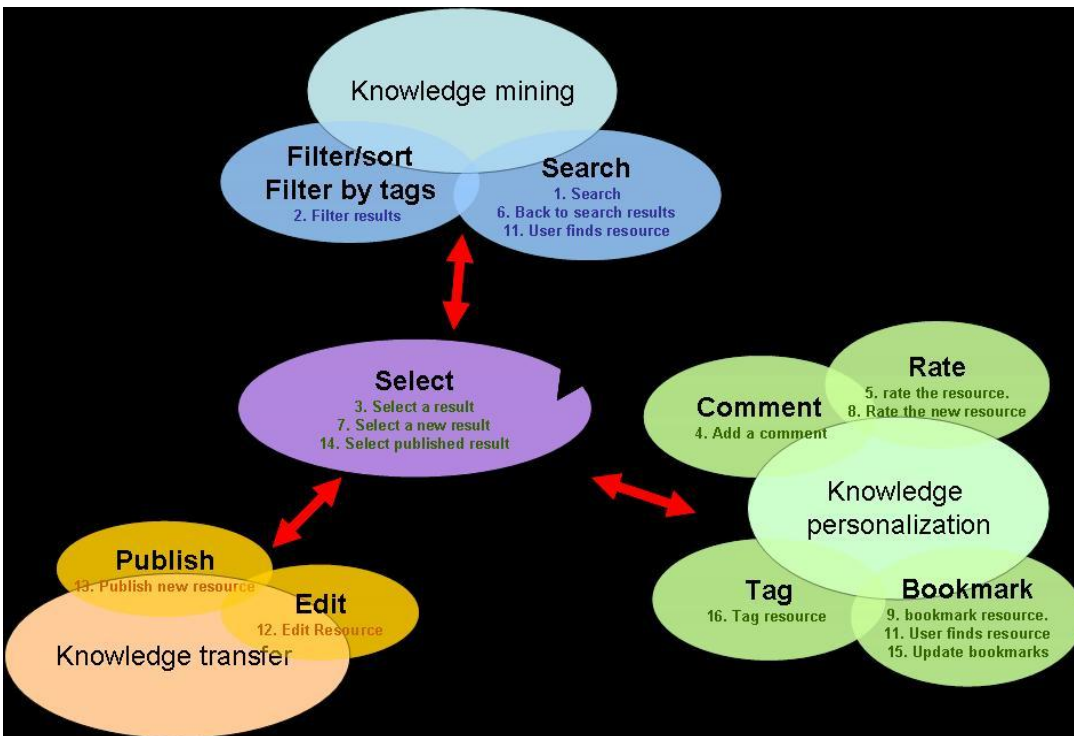


Figure 6 Final interaction design schema for KRSM

2.1.3 Compatibility of existing web 2.0 services with the interaction design schema

After that the pedagogical need of knowledge management have been translated into a theoretical interaction design schema, the compatibility of this schema with existing web 2.0 tools is studied and conclusions on its potential use are drawn. Hence, in order to proceed, it's important to define the components of compatibility being treated in the KME context. The level of compatibility of a given web 2.0 services or tool reflects its ability to cover specific functions within a unique interaction context and its aptitude to manipulate knowledge units treated in this same context. Compatibility with the interaction design schema also includes consideration about inter-compatibility levels among the web 2.0 services selected for integration.

Once we have defined the different contexts of interaction and the activities related to them, we also take into consideration the different stages of lifelong competence development described in section 3.4 in the D2.2. After an analysis of the elements that should be manipulated in each of these stages, we propose a set of Knowledge Objects (KO) that would be treated in the different interaction contexts defined above. We will assume for the sake of simplicity that the set of knowledge objects treated is the same for all of the interaction contexts.

The definition of these Knowledge Objects is an important asset for the classification and selection criterion upon which web 2.0 services considered for integration in the KMS will be evaluated. These objects are the core of the user interaction and are manipulated in different contexts and with different objectives. For example, a simple document can be treated in many different ways. When the user wants to share it with others it will be treated as an object from the Knowledge Transfer context. However, when the user wishes personalize it by adding a comment, it will become an object of the Knowledge Personalization context.

KO Stages	<i>O</i> rientatio <i>n</i>	<i>P</i> lanning	<i>R</i> eflection (self- ..	<i>K</i> nowledge <i>t</i> ransfer	<i>L</i> earning <i>a</i> ctivities	<i>P</i> eer <i>A</i> ssesse <i>m</i> e
UOLs	--	XX	XX	XX	XX	--
Activities	--	XX	XX	XX	XX	--
LPD	XX	XX	XX	XX	--	--
Questions	--	--	XX	XX	--	XX
Questionnair	--	--	XX	XX	--	XX
Videos	--	--	--	XX	XX	--
Audios	--	--	--	XX	XX	--
Images	--	--	--	XX	XX	--
Files	--	--	--	XX	XX	--
Discussions	XX	XX	XX	--	XX	--
Competence	XX	XX	XX	XX	--	--

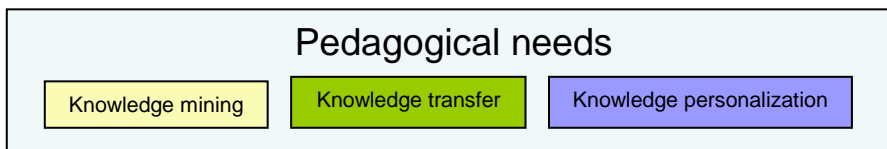


Table 2 Table of Knowledge Objects treated in each interaction context.

the Knowledge Objects related to the stages of lifelong competence development are listed in Table 2 above (more information about the definition of the KOs in the appendix). We mark with a cross those KO that we consider that have direct relation with each of the different stages. To maintain the coherence, we also classify each of the stages according to the different interaction contexts previously defined (colours in table 2).

Once the KO have been determined and associated with each of the interaction contexts, the technical characteristics of each are analyzed in order to derive a list of technical requirements for the web 2.0 services (Table 3). This list is another asset for the classification and selection criterion.

	<i>Formats</i>	<i>Tools required</i>	<i>Examples</i>
UOLs	Zip package, XML, IMS-LD	authorized engine, player XML parser	Coppercore, APIS and Sled player.
Questions	XML, IMS QTI specification	Authorized engine, player XML parser	APIS Sled
Questionnaires	XML, IMS QTI specification	Authorized engine, player XML parser	APIS Sled
Videos	FLV (AVI, DivX, MPEG, WMV, ASF, Mov) ...	Video player (with codecs) FLV encoder	VideoLan Player
Images and graphic representations	JPEG, GIF, PNG, Pixar, TIFF, PCX, SCT, BMP, RAW.	Image browser	Gimp ImageMagick
Audio	WAV,OGG,MPC,FLAC,AIFF,R AW,AU,GSM,DCT,VOX,ACC, MP4.MP3.WMA.WAV.RA.RA	Audio player	VLC player
Files	Zip, txt, pdf, doc, ppt, rtf, html,RSS ...	Editors	Adobe pdf OpenOffice 7-zip
Discussions	Txt, xml	DocReader XML parser	OpenOffice
Competences	XML Competence specification	Authorized engine, player XML parser	Digital cinema UOLs
LPD	XML, LPD specifications	authorized engine, player XML parser	TCLPE

Table 3 Technical characteristics of the knowledge
objects

Besides the definition of the interaction schema and its relevant contexts and activity sets, the definition of knowledge objects related to the KME completes the information necessary for the classification and selection of web 2.0 services. The following section details on how such a step is performed.

2.2 Classification of the tools

Now that the interaction schema is defined, we study the mapping of the web 2.0 services benchmarked in [4] to the interaction contexts defined in this schema. This document contains a descriptive compilation of popular web 2.0 tools and services along with their usage profiles and technical characteristics. Our aim is to classify all these tools in relation to the classification and selection criterion and then select those that fit best the requirements inherent in this criterion. The following table (Table 4) maps the list of tools to the activities of each interaction context. The Select context has been left out since its functionalities mainly rely on services provided by the KME core functionalities and developed by the KME development team.

Tool	Knowledge Mining		Knowledge Transfer		Knowledge Personalization				Score
	Filter/Sort/Filter by tags	Search	Publish	Edit	Rate	Comment	Tag	Bookmark	
GroupMe		X					X	X	1/0/2
Stud.IP		X		X					1/1/0
Toodledo	X	X		X			X		2/1/1
LinkedIn	X	X				X			2/0/1
SMF	X	X	X	X		X	X		2/2/2
Leo	X	X				X			2/0/1
Wordreference	X	X				X			2/0/1
FeedRaider	X								1/0/0
Drupal	X	X	X	X	X	X	X		2/2/3
Juice	X	X	X						2/1/0
MediaWiki			X	X	X	X			0/2/2
YaBB		X	X			X			1/1/1
Elgg	X	X	X	X		X	X		2/2/2
ILIAS	X	X	X	X		X	X		2/2/2
Moodle	X	X	X	X		X			2/2/1
CMap			X	X					0/2/0
LiveJournal		X	X	X		X			1/2/1
Blogger	X	X	X	X		X	X		2/2/2
Skype		X	X			X			1/1/1
Zoho	X	X	X	X		X	X		2/2/2
Delicious	X	X			X	X	X	X	2/0/4
Facebook		X	X			X	X		1/1/2
Flickr	X	X	X			X	X		2/1/2
Youtube	X	X	X		X	X	X		2/1/3
Slideshare			X	X					0/2/0
Diigo						X	X	X	0/0/3

Table 4 Mapping Web 2.0 tools to the functional requirements of the KMe interaction design schema

The table shows that nearly all tools do not map clearly to the contexts of the interaction schema as defined. Most tools map partially to several contexts with few exceptions. For example, the functionalities offered by *Slideshare* maps perfectly to the activities performed in the Knowledge Management context. *FreeRaider* maps partially to the Knowledge Mining context, and so does *Diigo* in respects to the Knowledge Personalization context.

This situation is expected because, to the extent of our knowledge, there is no of-the-shelf set of web 2.0 services that befit the functional requirements of knowledge management in learning environment. Hence, the focus of this work is to find the “best fit” solution for the KME requirements rather than identifying the correct one from a set of candidates. Therefore, a list of selection and classification principles has been drafted to aid in mining the most proper set of web 2.0 tools that befit the interaction schema. For each tool selected for integration, the following list of conditions should apply:

1. The tool is selected in correspondence with the activities defined in one interaction context.
2. The functionalities provided by the tool should only cover the activities of the context where it is integrated, in order to minimize the availability of functionalities not required by the KME.
3. The tool should be able to handle the defined knowledge objects treated in the relevant interaction context
4. Each group of web 2.0 tools selected to cover an activity set should address all the functional requirements of the set as exclusively as possible
5. Groups of tools considered for a certain activity set need to cover the maximum knowledge objects treated.
6. Number of tools selected for a any activity set has to be minimized.

2.2.1 Description of the tools benchmarked and selected

In this section we propose a set of selection principles to organize and group the selected tools by score and inter-compatibility within the requirements of each of the knowledge objects that they can manage and the technical characteristics of the own tool.

2.2.2 Application of the selection criterion in a generic list of tools

We here take the list of web 2.0 tools and relate them with each of the activity sets. For each activity set we select the minimal group of tools that cover all knowledge objects while limiting their functional scope to the activities involved in this set as much as possible. Notice, that this is the strongest constraint that has to be accomplished always when selecting a group of tools.

When applying these principles for choosing a set of activities from a list, the user can either be interested in looking for a tool for managing a concrete type of Knowledge Object or searching a tool that covers the functionalities for an specific activity. For this reason, we distinguish within two different ways of applying the criteria: activity-centered or ko-centered.

- Activity-centered selection:
 1. The tool has to cover the required functionalities in the selected activity.
 2. The best tool would be that one that covers the maximum number of activities within the activity set selected.
 3. The best tool would be that one that covers the maximum number of knowledge objects within the context.
 4. Follow steps 1, 2, and 3, till you reach the constraint.
- KO-centered selection:

1. The tool has to cover the required functionalities in the selected KO.
2. The best tool would be that one that covers the maximum number of KOs within the activity set selected.
3. The best tool would be that one that covers the maximum number of activities within the context.
4. Follow steps 1, 2, and 3, till you reach the constraint.

Let's explain this through a simple example:

- **Activity-centered selection**

- We want to select tools for the activity set in the Knowledge Personalization context.
- This context has associated a set of activities and a set of knowledge objects (figure 5 and table 2, respectively) that the tools have to completely cover.

Tool	Knowledge Personalization				Score
	Rate	Comment	Tag	Bookmark	
GroupMe			X	X	2
Stud.IP					0
Toodledo			X		1
LinkedIn		X			1
SMF		X	X		2
Leo		X			1
Wordreference		X			1
FeedRaider					0
Drupal	X	X	X		3
Juice					0
MediaWiki	X	X			2
YaBB		X			1
Elgg		X	X		2
ILIAS		X	X		2
Moodle		X			1
CMap					0
LiveJournal		X			1
Blogger		X	X		2
Skype		X			1
Zoho		X	X		2
Delicious	X	X	X	X	4
Facebook		X	X		2
Flickr		X	X		2
Youtube	X	X	X		3
Slideshare					0
Diigo		X	X	X	3

- Using table 8 in this case, we select those tools that cover the maximum number of activities required in this context. We should

order the previous list by score, however, in this example; we will just use the first four activities.

Tool	Knowledge Personalization				Score
	Rate	Comment	Tag	Bookmark	
Drupal	X	X	X		3
Delicious	X	X	X	X	4
Youtube	X	X	X		3
Diigo		X	X	X	3

- In order to reduce the list, we then see which are the tools that cover the maximum knowledge objects related to this context. In this case, there are no tools that can cover all the KO, but we choose 1 that covers the knowledge. But there are still some objects that need to be managed. Then, we will also choose a tool that just use these missing objects.

Tool	Rate	Comment	Tag	Bookmark	Score	KO
Delicious	X	X	X	X	4	Files,discussions
Drupal	X	X	X		3	UOL,Questions, Questionnaires, videos,files,images,audio,discussions,competences, LPD
Youtube	X	X	X		3	Videos
Diigo		X	X	X	3	Files,discussions

- Finally, when applying the selection criteria “activity-centered” to this example, we’ll be recommended to use the two first tools in the list above: Delicious+Drupal.

When applying the KO-selection over the same list of tools, the result will differ in the order and the recommended final list will be different, but still covering all the activity set in the context and all the KOs.

In this document, we have proposed a general framework for selection web 2.0 tools based on three different interactions contexts. Each of these contexts contain a set of basic activities that have been defined in order to cover the pedagogical needs of the learner when using the systems.

In this framework we have defined a selection criterion that could be used in the future as an automatic way for a recommendatory.

2.2.3 Final list of web 2.0 tools selected for their integration in the KMS

Once defined the framework above, we here present a list of tools that have been finally selected considering the selection criterion exposed and the technical constraints provided from the developers of this WP.

The main interest of the technicians' partners is to have tools able to manage the different file formats that the KRMS has to consider. They have provided us with a list of tools for these formats as a suggestion considering also the technical characteristics for facilitating the integration with the system (Open APIs...).

In order to relate this list of services and formats with the different scenarios' activities defined in the framework above, we analyze which are the activities that each of the proposed tool covers and add to the list, if necessary, new tools for considering those activities that belong to each context and cannot be solved with the proposed list. We have performed here a *KO-centered selection*, in which the knowledge objects are defined by the format file that the KRMS should manage.

In the table of the appendix "*Relation within tools and the pedagogical scenarios*" we show the set of tools selected for covering all the necessary file types and all the activities in each of the pedagogical context determined. Notice that there are file types that can be manage with two different tools (the images can be manage with *Flickr* and *Ipernity*), however we select for each format only one tool (green shadow in the table).

Notice that in the table we have also marked the service *Del.ici.ous* because it is the service that is common to all the format types and cover a set of activities that other tools don't cover.

After this first approach we can get finally the list of tools that should be integrated (table 9). For each tool we have showed the type of files that they treat the technical requirements for managing these resources, the place where these would be located and, regarding the pedagogical scenarios, the contexts where they are mapped and the functionalities related with this context.

When listing the resources and selecting a concrete one, the functionalities listed in table 10 should be showed next to the Thumbnail related to this resources. When selecting a resource, next to it there should be a set of functionalities that link to the tool that manages these functionalities. For example, when an image is selected, next to it there must: *Edit*, *Share* and *Search*. When the user selects *Edit*, he will be redirected to the tool *Picture2Life* and to *Flickr* when his selection is *Share* or *Search*.

Tool	Format	Tec. Req.	Storage	Contexts	Functionalities
Scribd	Doc Ppt	API to search,	Scribd	Knowledge Transfer	Edit Publish/Share/Edit

	Xls Pdf Plain text	get, reference		Knowledge Mining	Collaboratively Search
Flickr	Image	API to search, get, reference	Flickr	Knowledge Transfer Knowledge Mining	Publish/Share Search
Picture2Life	Image	API to search, get, reference	Flickr	Knowledge Transfer	Edit (Image)
Youtube	Video	API to search, get, reference	YouTube	Knowledge Transfer Knowledge Mining	Publish/Share Search
Ipernity	Audio	API to search, get, reference	Ipernity	Knowledge Transfer Knowledge Mining	Publish/Share Search
	Recognized binary File And Unrecognized binary file				
VirtualDub Video	Video editing	API to search, get, reference	YouTube	Knowledge Transfer	Edit (Video)
Donwloader	Zip and binary files	API to search, get, reference	Fedora	Knowledge Mining	Download
Del.ici.ous	Bookmarking for all type of docs	API to search, get, reference	Del.ici.ous	Knowledge Personalization	Bookmark
GroupMe!	All resources	API to search, get, reference	GroupMe!	Knowledge Mining	Sort (Organize your resources)
Wiki	.doc	API to search, get, reference	Wiki	Knowledge Transfer	Publish/Share/ Edit Collaboratively

Table 9 Final selected web 2.0 tools.

3 Integration of the tools

With respect to the activities supported and the requirements elicited in the ID5.12, the tools and services benchmarked have been selected in order to choose those suitable for the **LearnWeb2.0**.

Activity	Tools	Remarks
1. Simple search	- SQI - RMI (repository selection) - COI (download) - Google API	
2. Sharing	- SPI	
3. Edit (documents/images)	- google shared docs (.doc, .ppt, .exl) - snipshot (images) - custom protocols - (.pdf hammer)	
4. Metadata Tagging	- fedora	The metadata tagging includes resources tagging and categorization
5. Rating/recommendation	- fedora - taste	
6. Aggregation, Grouping	- Group Me	
7. Identity Management	- UMI - Single Sign On	
8. Metadata extraction and Indexing	- fedora? - Aperture & Lucene based component	

4 Core services: Design specifications (Leader: **Giunti**)

4.1 ENTITIES

4.1.1 Users

Each user of LearnWeb2.0 Web Tool is identified by a nickname, and has access credentials.

The user is univocally recognized by his/her id. The email address is also a unique key in user's database.

4.1.2 Resources

A knowledge resource can be:

- a text (html, txt,...)
- a document (doc, xls, csv, pdf,...)
- an image (jpg, bmp, png,...)
- a video (mpg, avi, mp4, 3gp,...)
- an audio (mp3, wav,...)
- Unit of Learning, Competence, profile,...
- whichever file (even binary)

Each resource is composed by two elements:

- the content (the file)
- the metadata (attributes describing the resource and pointing to the content)

The above elements can be stored in different places (e.g. an image may be stored in Flickr and its title may be stored in Fedora)

Each resource has the following attributes depending on the format:

- thumbnail: a little icon for an easy visual identification of the resource
- player: the application for using the resource
- upload storage: the space that physically stores the resource
- access: the modality for manipulating the resource

The following matrix defines the attributes, and will be a parametric table used by the software (“property” file).

Format	Thumbnail	Player	Upload to (storage)	Access
Plain text	First N characters	Browser on popup window	Fedora	API to, search, get reference,...
Html text	Associated Icon	Browser on popup window	Fedora	API to, search, get reference,...
Document “.doc”	Associated Icon	Associated application	Fedora	API to, search, get reference,...
Spreadsheet “.xls”	Associated Icon	Associated application	Fedora	API to, search, get reference,...
Presentation “.ppt”	Associated Icon	Associated application	Fedora	API to, search, get reference,...
Document “.pdf”	Associated Icon	Associated application	Fedora	API to, search, get reference,...
Image	Miniature	Browser on popup window	Flickr	API to search, get reference,...

Video	Shockwave-Thumbnail	got from youTube	YouTube	API to, search, get reference,...
Audio	Shockwave-Thumbnail	got from Iperinty	Iperinty	API to search, get reference,...
Grouped resources	Group miniature if available, icon otherwise	Browser GroupMe! on popup window	GroupMe!	API for search, get reference.
Recognized binary file (zip, exe...)	Associated Icon	Downloader	Fedora	API to, search, get reference,...
Unrecognized binary file	Generic Icon	Downloader	Fedora	API to get reference,...
Collaborative document	Icon showing collaboration (e.g. handshake)	Browser	Wiki (?)	Api to get reference
UoLs	Icon showing a zip file with a explicative text about the contents of the UoL	From OpenDock	OpenDock	API to get the reference
Competence	Icon of the competence according to the PCM icons	PCM	Fedora	API to get the reference
Communities	Icon of the competence according to the PCM icons	PCM	Fedora	API to get the reference
Competence Profile	Icon of the competence according to the PCM icons	PCM	Fedora	API to get the reference

4.1.3 Metadata

Metadata are used to facilitate the understanding, characteristics, and management usage of resources. The metadata required for effective resource management varies with the type of data and context of use.

LearnWeb2.0 uses the metadata standards developed by the Dublin Core Metadata Initiative (DCMI). The **Dublin Core** metadata element set [5] is a standard for cross-domain information resource description. It provides a simple and standardised set of conventions for describing resources online in ways that make them easier to find. Dublin Core is widely used to describe digital materials such as video, sound, image, text, and composite media like web pages. Implementations of Dublin Core typically make use of XML and are Resource Description Framework based. Dublin Core is defined by ISO in 2003 ISO Standard 15836, and NISO Standard Z39.85-2007

The Dublin Core Metadata element set [5] is described in Table 11.

Term Name: contributor	
Label:	Contributor
Definition:	An entity responsible for making contributions to the resource.
Comment:	Examples of a Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.
Term Name: coverage	
Label:	Coverage
Definition:	The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.
Comment:	Spatial topic and spatial applicability may be a named place or a location specified by its geographic coordinates. Temporal topic may be a named period, date, or date range. A jurisdiction may be a named administrative entity or a geographic place to which the resource applies. Recommended best practice is to use a controlled vocabulary such as the Thesaurus of Geographic Names [TGN]. Where appropriate, named places or time periods can be used in preference to numeric identifiers such as sets of coordinates or date ranges.
References:	[TGN] http://www.getty.edu/research/tools/vocabulary/tgn/index.html
Term Name: creator	
Label:	Creator
Definition:	An entity primarily responsible for making the resource.
Comment:	Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
Term Name: date	
Label:	Date
Definition:	A point or period of time associated with an event in the lifecycle of the resource.
Comment:	Date may be used to express temporal information at any level of granularity. Recommended best practice is to use an encoding scheme, such as the W3CDTF profile of ISO 8601 [W3CDTF].

References:	[W3CDTF] http://www.w3.org/TR/NOTE-datetime
Term Name: description	
Label:	Description
Definition:	An account of the resource.
Comment:	Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource.
Term Name: format	
Label:	Format
Definition:	The file format, physical medium, or dimensions of the resource.
Comment:	Examples of dimensions include size and duration. Recommended best practice is to use a controlled vocabulary such as the list of Internet Media Types [MIME].
References:	[MIME] http://www.iana.org/assignments/media-types/
Term Name: identifier	
Label:	Identifier
Definition:	An unambiguous reference to the resource within a given context.
Comment:	Recommended best practice is to identify the resource by means of a string conforming to a formal identification system.
Term Name: language	
Label:	Language
Definition:	A language of the resource.
Comment:	Recommended best practice is to use a controlled vocabulary such as RFC 4646 [RFC4646].
References:	[RFC4646] http://www.ietf.org/rfc/rfc4646.txt
Term Name: publisher	
Label:	Publisher

Definition:	An entity responsible for making the resource available.
Comment:	Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
Term Name: relation	
Label:	Relation
Definition:	A related resource.
Comment:	Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.
Term Name: rights	
Label:	Rights
Definition:	Information about rights held in and over the resource.
Comment:	Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.
Term Name: source	
Label:	Source
Definition:	A related resource from which the described resource is derived.
Comment:	The described resource may be derived from the related resource in whole or in part. Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.
Term Name: subject	
Label:	Subject
Definition:	The topic of the resource.
Comment:	Typically, the subject will be represented using keywords, key phrases, or classification codes. Recommended best practice is to use a controlled vocabulary. To describe the spatial or temporal topic of the resource, use the Coverage element.
Term Name: title	
Label:	Title

Definition:	A name given to the resource.
Comment:	Typically, a Title will be a name by which the resource is formally known.
Term Name: type	
Label:	Type
Definition:	The nature or genre of the resource.
Comment:	Recommended best practice is to use a controlled vocabulary such as the DCMi Type Vocabulary [DCMITYPE]. To describe the file format, physical medium, or dimensions of the resource, use the Format element.
References:	[DCMITYPE] http://dublincore.org/documents/dcmi-type-vocabulary/

Table 11. The Elements of the Dublin Core Metadata

An example of using RDF/XML for describing a resource with Dublin Core is the following:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description rdf:about="http://media.example.com/audio/guide.ra">

    <dc:creator>Rose Bush</dc:creator>
    <dc:title>A Guide to Growing Roses</dc:title>
    <dc:description>Describes process for planting and nurturing
different kinds of rose bushes.</dc:description>
    <dc:date>2001-01-20</dc:date>

  </rdf:Description>
</rdf:RDF>
```

LearnWeb2.0 uses a Fedora digital repository [6] to store knowledge resources (i.e. metadata in Dublin Core, the URL of the resource or the content of the resource). LearnWeb2.0 extends the standard Dublin Core metadata by providing additional author-oriented and user-oriented information:

- owner/publisher
- list of categories
- list of tags

- list of comments
- ratings of resources
- ratings of comments
- popularity of resources
- etc.

This is achieved by designing and implementing an Object Model in Fedora as shown on Figure 6.

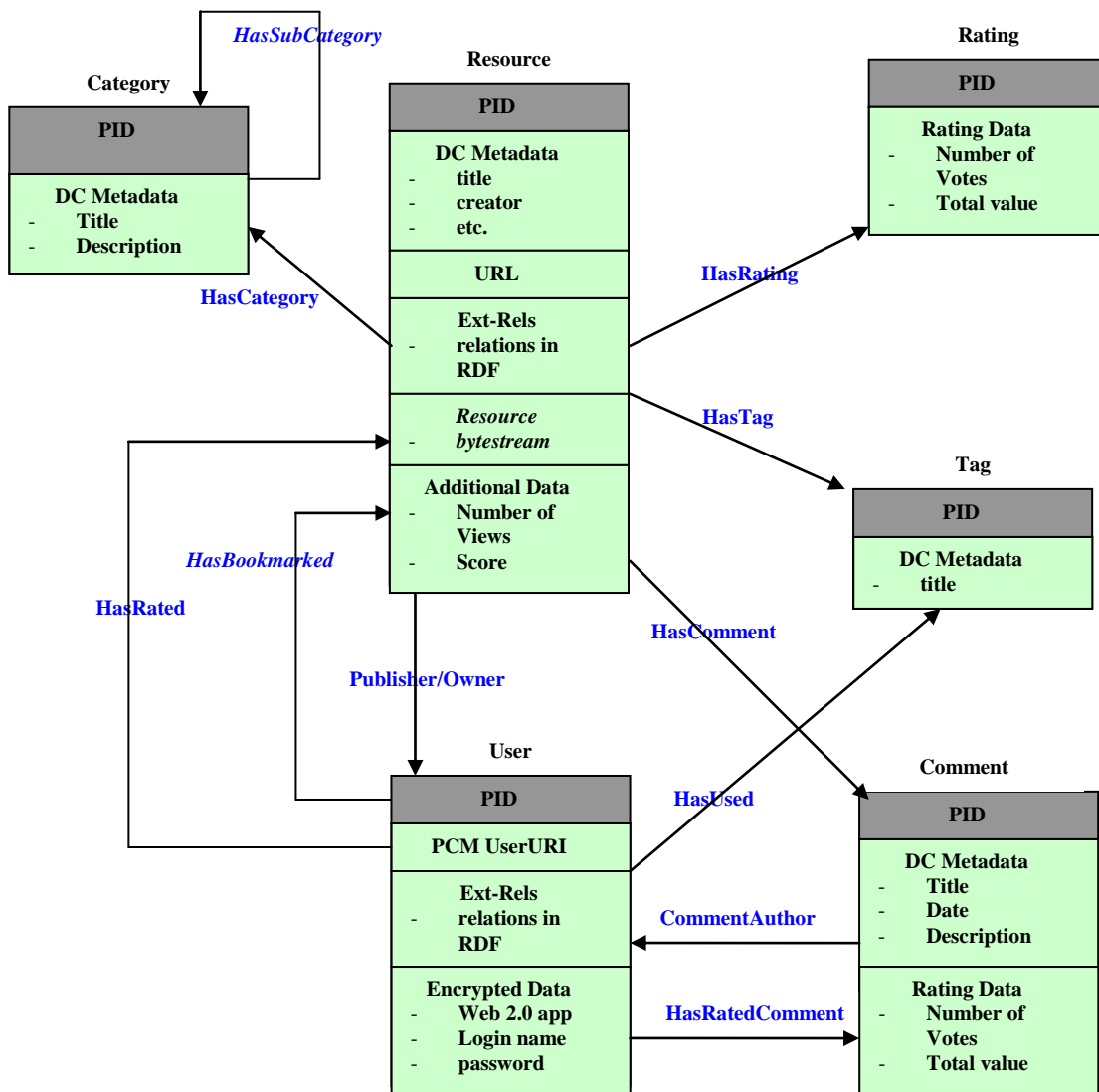


Figure 6. LearnWeb2.0 Fedora Object Model

4.1.4 Categories

Categories are fixed and a-priori defined.

Categories can be modified by administrators only.

Categories may include sub-categories, e.g. Sports->Nautical->Sailing

The belonging of a resource to a category is defined by the resource owner.

4.1.5 Groups

A group aggregates a number of resources. The belonging of a resource to a group is defined by the user. Groups are defined dynamically by the user. A group can contain a number of nested groups. A user can assign a number of keywords (tags) to a group.

4.1.6 Tags

Tags are user defined.

A tag is a word or a brief word sequence, e.g. "Sailing precedences"

Tags are only syntactically handled, without semantic ("Sailing" is different from "Sail" or "Voile")

4.1.7 Comments

Comments are user defined.

A comment is a sentence describing the opinion of a user, e.g. "This video is a good explanation about how to enter a harbour, but doesn't consider nocturnal sailing"

A comment belongs to a single user.

Each comment may be rated by other users.

4.1.8 Rates

Rates are user defined.

A rate is a number in a range (e.g. 1-5) expressing the appreciation of a resource from a user.

The global rate of a resource is the arithmetic mean of the users rates about the resource.

4.1.9 Popularity

The popularity of a resource is the number of time the resource has been used.

Popularity is computed using the following rule:

- visiting a resource yields 1
- bookmarking a resource yields 2
- downloading a resource yields 2
- modifying a resource yields 3
- commenting a resource yields 4

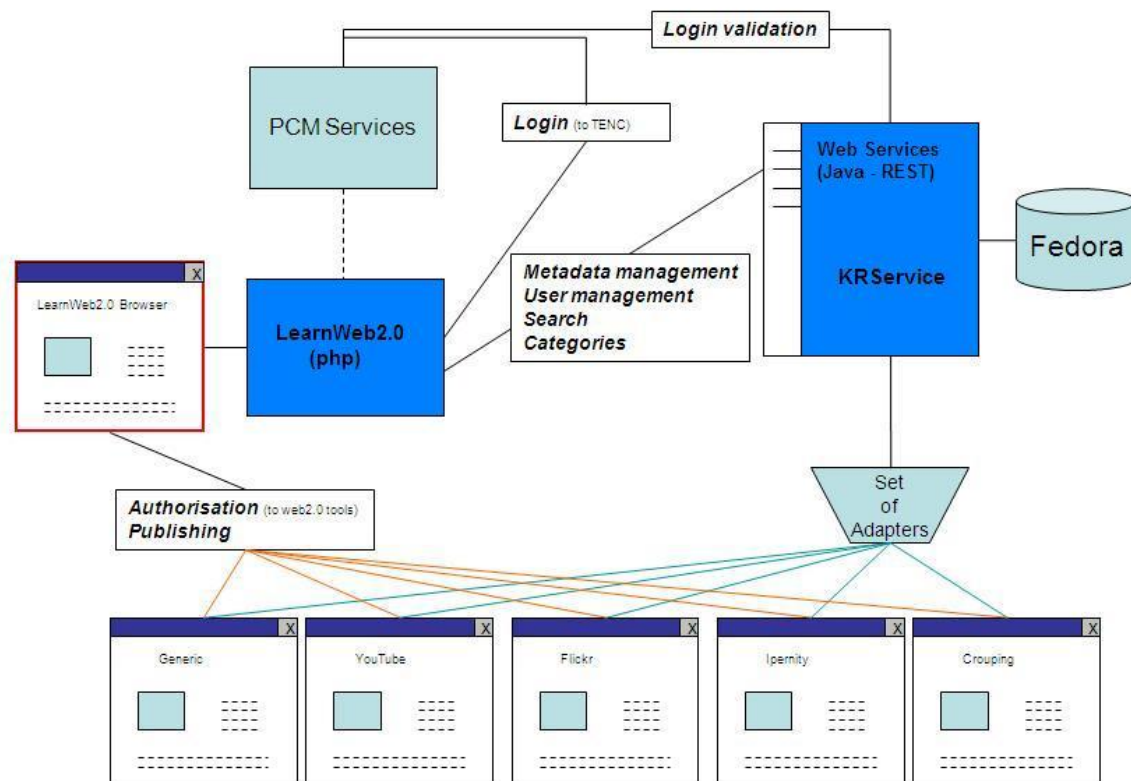
4.2 SOFTWARE ARCHITECTURE

The system is composed by the following elements:

- LearnWeb2.0: a web application (Php) for interactively manage knowledge resources
- KRService: a Web Service (Java, REST) for automatically manage knowledge resources
- Adapters: a set of drivers for managing web2.0 tools through APIs (Java or other)
- Fedora: the metadata repository (based on Java, Tomcat, Rest-WS)
- Metadata manager: a tool for metadata editing and management (Java)
- Identity managers: a set of tools for login/validation/authorization handling (Javascript, Rest-WS,...)
- Publishing managers: a set of tools for uploading management (Javascript)
- Toolbars: for open source browser (Firefox, Javascript)

The main issue of this architecture is re usability/re-usability by other TenCompetence tools (PCM,...)

The below figure shows the software architecture:



The complete list of Web Services exposed by KRService is shown in Appendix 7.2
 The complete list of Driver (Adapters) functions is shown in Appendix 7.3

The communication protocol between web application and services functions is shown in Appendix 7.4

4.3 FUNCTIONALITIES

4.3.1 User Login

The system may be used without requiring user login.

The functionalities available prior to login are:

- searching (whichever mode)
- downloading

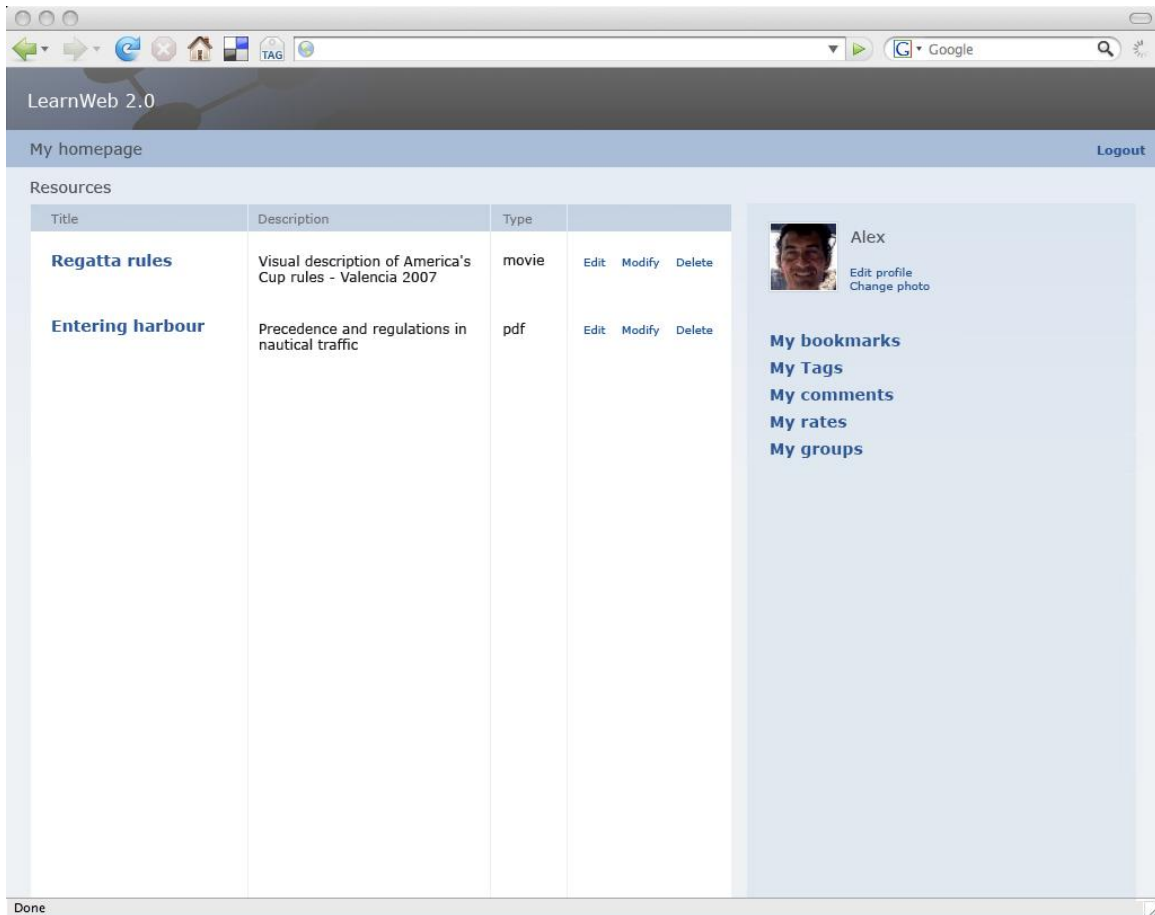
Functionalities enabled by logging-in

- bookmarking (i.e. tagging)
- sharing (i.e. sending an email)
- rating
- commenting
- publishing (uploading)

If the user is not yet logged, the disabled functionalities are visible but unclickable. On mouse moving, a balloon will invite to log-in.

4.3.2 My Home Page

LearnWeb2.0 homepage allows the management of “my resources”, as shown in below figure.



The homepage allows the following functions:

- Managing my profile (nick, password, photo, email,...)
- Managing my resources (search/view/edit/modify/delete)
- Managing my attributes (bookmark, tags, comments, rates, groups,...)
- Access to my bookmarks

The esthetics will be aligned to the one of other TenCompetence tools, sharing the same Stylesheet.

4.3.3 Search

4.3.3.1 Simple Search

Simple search is done by keys, with a Google-Style (ANDing words, quoting phrases,...)

The search is performed into various targets:

- the repository (Fedora)
- images (e.g. Flickr)
- videos (e.g. YouTube)
- other (to be defined)

Results coming from the different targets are mixed.

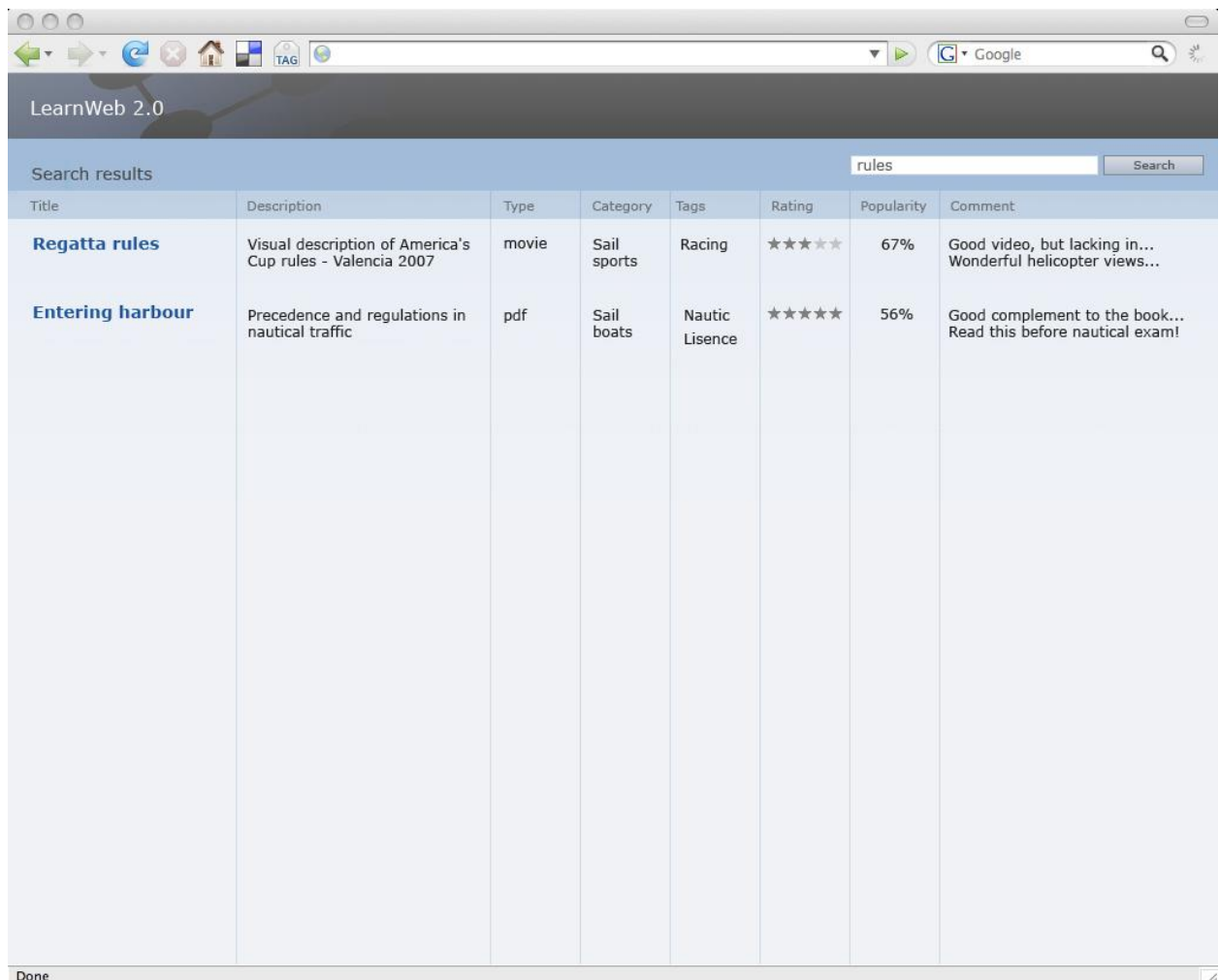
Each result item may include:

- title
- description
- format
- date creation/modif
- popularity (if available in Fedora)
- author (if available in Fedora)
- rate (if available in Fedora)
- tags (if available in Fedora)
- category (if available in Fedora)
- comments (if available in Fedora)

Results are presented in a tabular way having default order by rate (see fig. 1). Results can be reordered by whichever of the above items.

There are two ways to access simple search:

- from homepage
- quick-search from toolbar (see figure below)



Title	Description	Type	Category	Tags	Rating	Popularity	Comment
Regatta rules	Visual description of America's Cup rules - Valencia 2007	movie	Sail sports	Racing	★★★★★	67%	Good video, but lacking in... Wonderful helicopter views...
Entering harbour	Precedence and regulations in nautical traffic	pdf	Sail boats	Nautic Lisence	★★★★★	56%	Good complement to the book... Read this before nautical exam!

4.3.3.2 Search by Categories

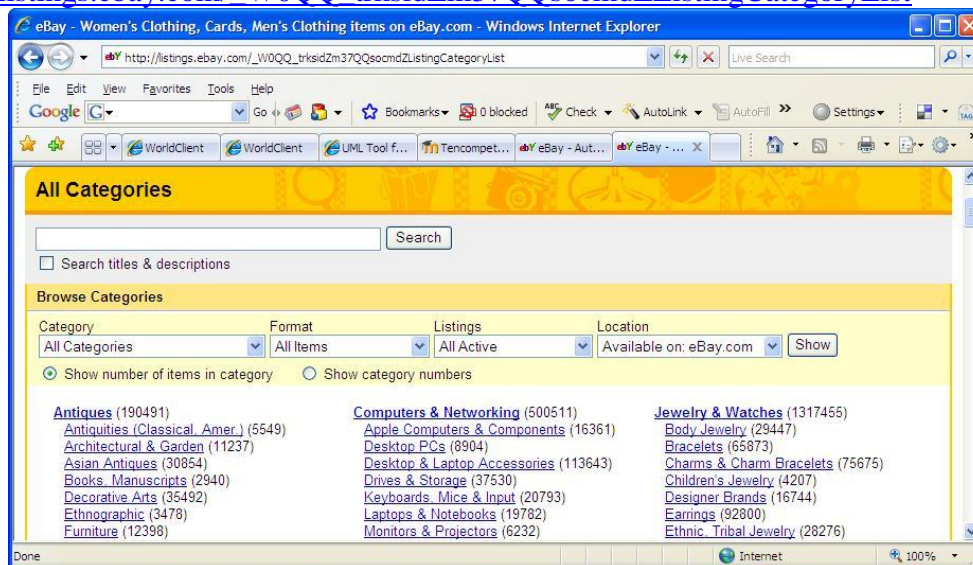
Search by categories may be performed in two ways:

- just from category
- refinement of a simple search

The refinement of a simple search is activated clicking on the category item of a previously retrieved resource. In this case the user will obtain the resources matching both the keys and the category. Example: key “tactic” in category “sports”.

The just-from-category search is activated through a specific button, that shows a category-tree, with an eBay-Style, example:

http://listings.ebay.com/_W00Q trksidZm370QsocmdZListingCategoryList



Category search is obviously performed only in LearnWeb2.0's repository (Fedora).

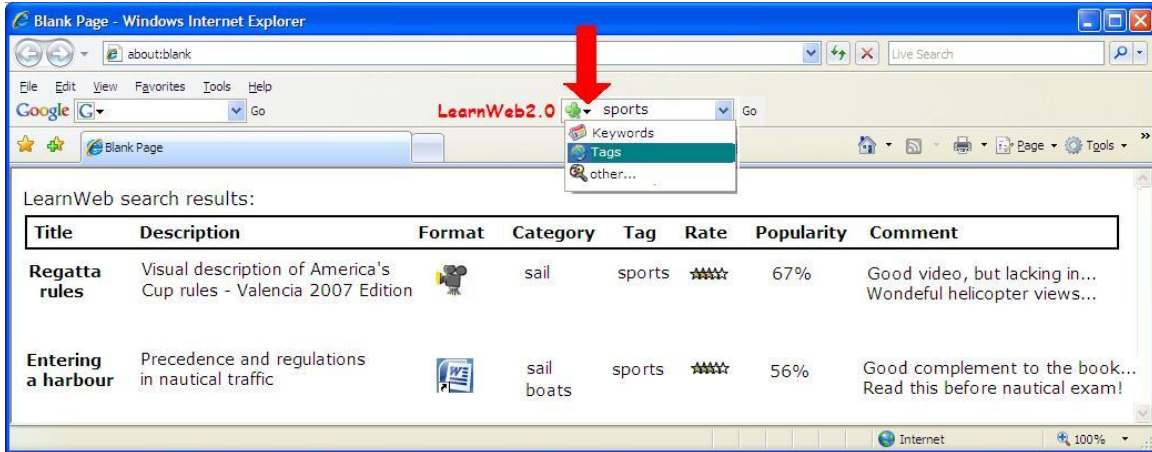
4.3.3.3 Search by Tags

Search by tags may be performed in two ways:

- just from tag
- refinement of a simple search

The refinement of a simple search is activated clicking on the tag item of a previously retrieved resource. In this case the user will obtain the resources matching both the keys and the tag. Example: key “tactic” with tag “racing”.

The just-from-tag search is similar to simple search from the user point of view, as depicted in the above figure.



The main difference is the fact that the search is performed only in LearnWeb2.0's repository (Fedora), where tags are.

4.3.3.4 Search by Bookmark

This is not a real search.

This is done by simply entering MyHomePage and selecting one of my bookmarks.

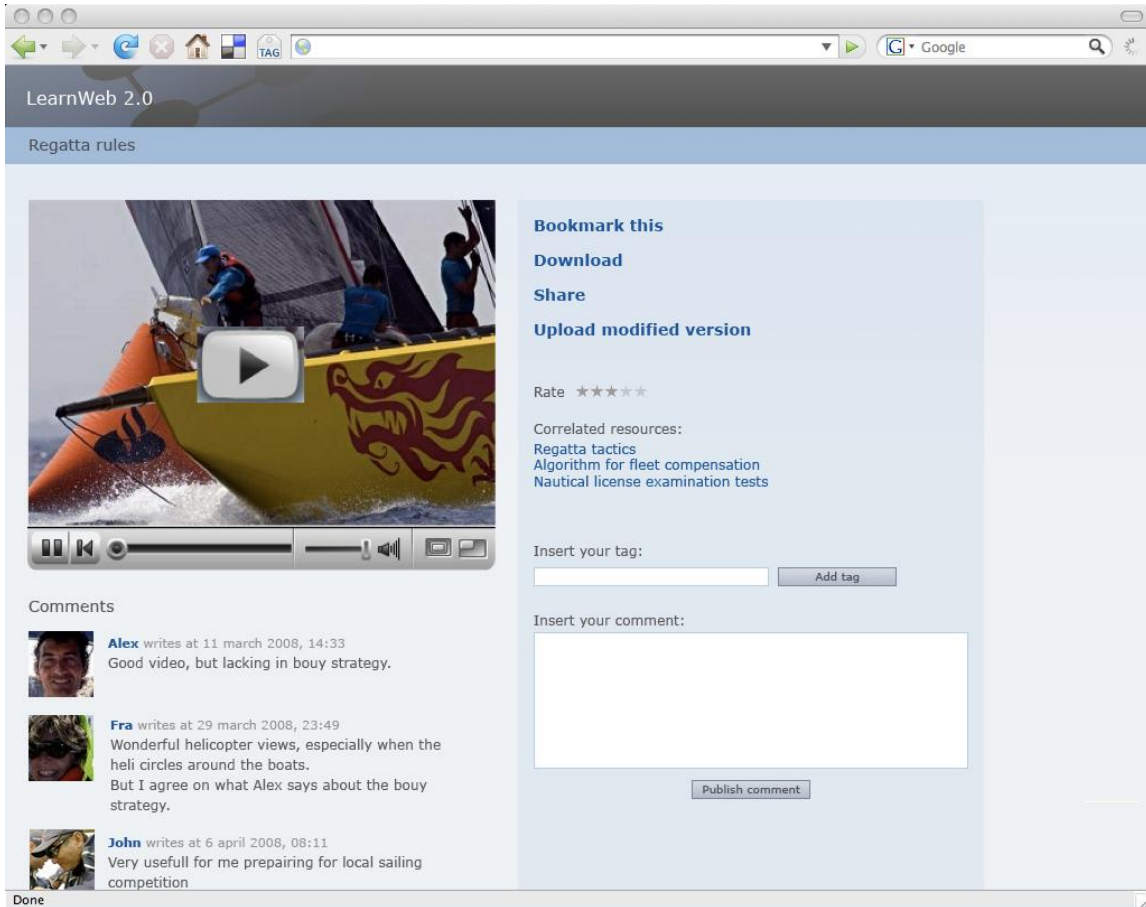
The bookmarks are handled by Del.icio.us.

4.3.3.5 Search by Group

A group can have a list of associated keywords (tags). These keywords can be used by search. Thereby the user query is matched against the tags attached to the group. Query result is a list of groups.

4.3.4 View and manage

Selecting a specific resource (by clicking its title), you get a resource thumbnail with its social functionalities: comments, share, rate, bookmark,...



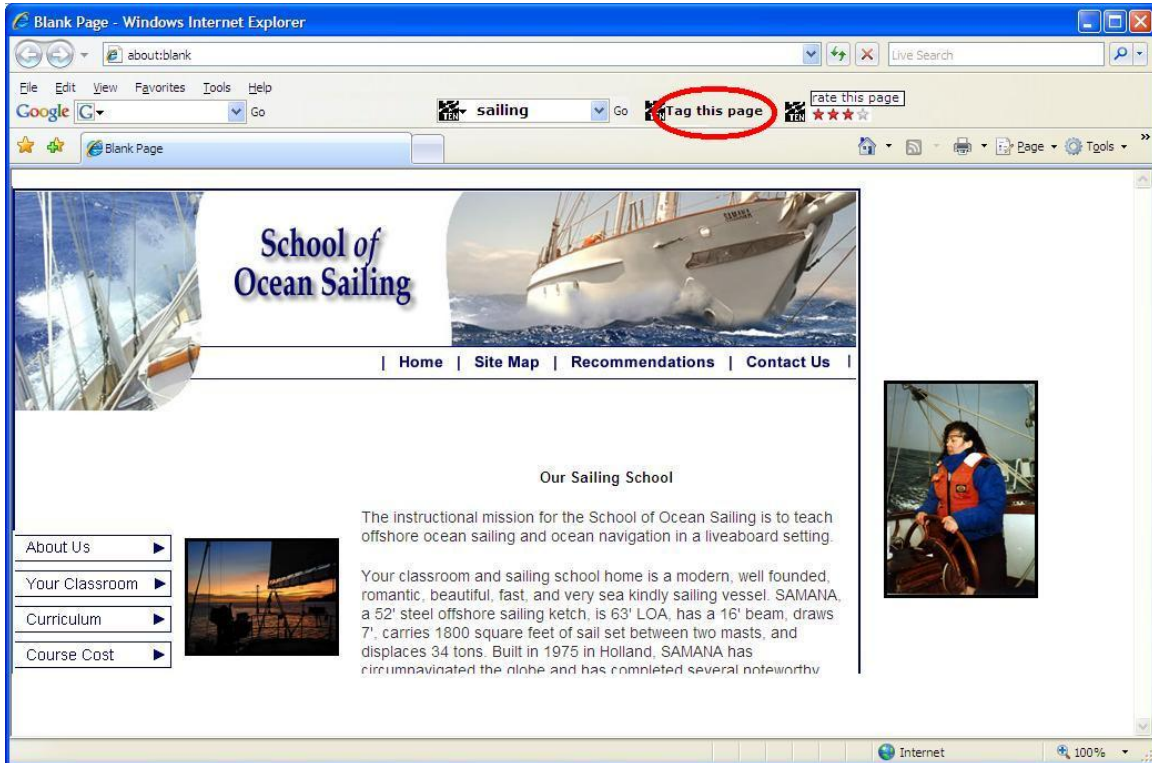
The screenshot shows a web browser window with the address bar set to 'Google'. The page title is 'LearnWeb 2.0' and the content is 'Regatta rules'. A video player is embedded, showing a yellow and orange sailboat. To the right of the video is a sidebar with the following options: 'Bookmark this', 'Download', 'Share', and 'Upload modified version'. Below these is a rating section with five stars and a 'Rate' label. Underneath are 'Correlated resources' listed as 'Regatta tactics', 'Algorithm for fleet compensation', and 'Nautical license examination tests'. There is also a 'Insert your tag:' field with an 'Add tag' button and a 'Publish comment' button at the bottom of the sidebar. The 'Comments' section on the left shows three entries: Alex (11 march 2008, 14:33) saying 'Good video, but lacking in bouy strategy.', Fra (29 march 2008, 23:49) saying 'Wonderful helicopter views, especially when the heli circles around the boats. But I agree on what Alex says about the bouy strategy.', and John (6 april 2008, 08:11) saying 'Very usefull for me preparing for local sailing competition'. The browser status bar at the bottom says 'Done'.

4.3.5 Bookmark

There are two ways to bookmark a resource:

- From the LearnWeb2.0 preview page (see above)
- From outside LearnWeb2.0 through the toolbar (quick-bookmarking)

The Quick-Bookmarking through toolbar is depicted in the below figure:



Quick-Bookmarking implies the automatic insertion of associated tags into repository

4.3.6 Download

Resources can be downloaded and stored on the local file-system.
Some resources cannot be downloaded, e.g. YouTube videos.

4.3.7 Upload

Resources uploading depends on the format, as shown in the matrix of section 4.1.2.

Uploading is performed in two ways:

- Uploading into web2.0 tools (Flickr,...): a popup window will be opened pointing to the specific web2.0 tool uploading page
- Uploading into generic web space: a LearnWeb2.0 page will be opened.

Uploading a resource includes metadata loading into Fedora.

4.3.8 Share

Sharing a resources is intended by simply sending an email containing the URL of the resource.

5 Conclusions

LearnWeb2.0 will be a useful tool both for an interactive social use, and for providing services for other eLearning tools.

The first part of the task (till may 2008) will address all main functionalities.

The second part of the task (from may 2008 on) will address secondary topics:

- Collaborative resources: handled by GoogleDocs or Wiki (the latter easier to integrate). Probably it will be sufficient to add a link into MyHomepage
- Sticking with Diigo (see ID5.12-sect 3.3.1.6)
- Likert (see ID5.12-3.3.1.7)

6 References

- [1] D2.2- Updated high level Use Cases and pedagogical model definitions.
- [2] RoadMap (Linda)
- [3] Recommendations of pilot evaluations (D4.2)
- [4] Helena's deliverable
- [5] Dublin Core Metadata Element Set, Version 1.1
<http://dublincore.org/documents/dces/>
- [6] Fedora Open Source Repository Software
<http://www.fedora.info/>
- [7] Web-services exposed by KRServer

7 Appendix A - List of Knowledge Objects

UoL

What is it?	<p>An UoL is a zip package that contains a file called imsmanifest, and some different resources.</p> <p>With this object we can structure a course, a lesson, a group of activities, a competence... with their associated resources.</p> <p>Imsmanifest: is an XML file written following the IMS LD specification, this file contains all the information associated with the organization and structure of the Unit of Learning.</p> <p>Resources: a different formats file.</p>
Format	<p>Zip package</p> <p>XML</p> <p>IMS Learning design specification</p> <p>IMS QTI specification</p>

Tools required	<p>An XML parser.</p> <p>A zip software.</p> <p>To interpret the imsmanifest file, we need:</p> <p>An authorized engine. (Example : Coppercore , APIS)</p> <p>To visualize the contents of the UoL, we need:</p> <p>An authorized player. (Example : Sled)</p>
----------------	---

QTI Questions/ items

What is it?	<p>An individual item contains the information associated with the question, the response process to be applied to the candidates (only one possible interaction) and the feedback to score the candidate. It is the smallest exchangeable object within QTI. The content of an individual assessmentItem , can be: text, images, graphics... and since the 2.x version other XHTML or MathML content, for example we can use maps with the integration of the web 2.0 service of GoogleMaps.</p>
Format	<p>We need a tool that interprets:</p> <p>XML</p> <p>IMS QTI specification</p>
Tools required	<p>An XML parser.</p> <p>To interpret the imsmanifest file, we need:</p> <p>An authorized engine. (Example : APIS)</p> <p>To visualize the contents of the UoL, we need:</p> <p>An authorized player. (Example : Sled)</p>

QTI Questionnaires

What is it?	<p>A QTI test is an organized set of QTI items, some XML files, organized in some sections (one section as a minimum) or sub-sections. We can only have one assessmentTest for each instance. We can use the results of this sequencing of items and calculate the outcome values.</p>
Format	<p>We need a tool that interprets:</p> <p>XML</p> <p>IMS QTI specification</p>
Tools required	<p>A XML parser.</p> <p>To interpret the imsmanifest file, we need:</p> <p>An authorized engine. (Example : APIS)</p> <p>To visualize the contents of the UoL, we need:</p> <p>An authorized player. (Example : Sled)</p>

Video

What is it?	<p>A video is a several storage formats for moving eye pictures.</p>
Format	<p>- FLV (AVI, DivX, MPEG, WMV, ASF, Mov)</p>
Tools required	<p>Video player (with codecs).</p> <p>FLV encoder</p>

Images and graphic representations

What is it?	A file – from a digital camera or scanner or computer – which contains sets of numbers that specify the colors of a rectangular grid of pixels.
Format	Jpeg Gif Png Pixar TIFF PCX SCT BMP RAW
Tools required	An image browser.

Audio

What is it?	A Digital file with sounds, music or voice.
Format	- WAV OGG MPC FLAC AIFF RAW AU GSM DCT VOX ACC MP4 MP3 WMA WAV RA RAM DSS MSV DVF
Tools required	Audio player

Files

What is it?	A set of textual information and images.
Format	Txt Pdf .doc ppt rtf html zip ...
Tools required	Editors. A zip software.

Discussions

What is it?	Logs that capture the evolution of a discussion in a blog or a forum
Format	Txt XML
Tools required	DocsReader XML parser

Competences

What is it?	Is like a Learning Unit or a UoL, but the difference is the organization of the learning resources oriented to competences. Normally is a zip package that contains some different resources. With this object we can structure their associated learning resources. Resources: different formats of a file and imsmanifest file.
Format	XML Competence specification
Tools required	- An XML parser. To interpret the specification, we need: An authorized engine. To visualize the contents of the competence, we need: An authorized player.

Relation within tools and the pedagogical scenarios												
File Type	Tool/Service	Player	Storage	Knowledge Mining		Knowledge transfer		Knowledge Personalization				Technical requirements
				Filter/Sort/ Filter by tags	Search	Publish/ Share	Edit	Rate	Comment	Tag	Bookmark	
.doc .ppt .xls .pdf	Associated app	X										
	Free web space	.	X			X						http://www.freewebspace.net/php/ List personal webspace providers.
	Del.ici.ous			X	X			X	X	X	X	
.doc	Wiki					X						Document sharing
.doc .ppt .xls .pdf Image Plain text	Scribd	X	X	X	X	X	X	X	X	X		http://www.scribd.com/platform/documentation Embedded for any websiter owner
Image		Browser on popup window										
	Flickr		X	X	X	X		?	X	X		
	Picture2Life						X					http://www.picture2life.com/ Compatible with Flickr
	Scribd	X	X	X	X	X	X	X	X	X		http://www.scribd.com/platform/documentation Embedded for any websiter owner
	Del.ici.ous			X	X			X	X	X	X	
Video	Youtube	From YouTube	X	X	X	X		X	X	X		
	Del.ici.ous			X	X			X	X	X	X	
	Ipernity	From Ipernity	X	X	X	X			X	X		
	VirtualDub Video						X					http://virtualdubbook.packtpub.com/ Open source tool
Audio	Ipernity	From Ipernity	X	X	X	X			X	X		http://www.ipernity.com/
	Del.ici.ous			X	X			X	X	X	X	

Recognized binary File And Unrecognized binary file	Donwloader	X										
	Del.ici.ous			X	X		-	X	X	X	X	
All type of resources	GroupMe!	Browser GroupMe!	X	X	X				X	X		API for search get, reference
UoL	OpenDoc	Browser on popup window	X									<u>We have to manage the rest of functionalities in the application</u>

8 Appendix B - Web-services exposed by KRServer

The API for KRSM services are modelled using the REpresentational State Transfer (REST) approach. See http://en.wikipedia.org/wiki/Representational_State_Transfer for a detailed explanation of REST.

The following summarises salient features of REST based services on KRSM:

1. Each “CRUD” operation (Create / Read / Update / Delete of an object) is matched to an HTTP method, according to the table below.

CRUD operation	REST (HTTP method name)
Create	POST
Read	GET
Update	PUT
Delete	DELETE

2. Every object is regarded as a resource and has its own unique URI. The HTTP method determines what action is performed on the resource (create it, read it, ...). The type of each entity in URIs can be recognized by the format of the corresponding guid (resource:nnn, user:nnn, category:nnn, etc.).
3. The clients have the responsibility to request the Fedora server to generate unique identifier before creating URI.

The following types of objects are stored on the server:

- **User** – a person who uses the system.
Guid format: user:nnn;
- **Category** – contains other categories and/or resource.
Guid format: category:nnn;
- **Resource** – a resource stored on the server. Each resource has metadata in Dublin Core format (<http://dublincore.org/documents/usageguide/elements.shtml>) that describes the resource. The content of the resource can be stored on the server or anywhere on the Web (in this case the URL of the resource is stored on the server). Resources have tags, comments, popularity and rating.
Guid format: resource:nnn;
- **Tag** – used for tagging resource.
Guid forma: tag:nnn;
- **Comment** – for commenting resources. The comments can be rated by users.
Guid format: comment:nnn.

The API for KRSM web services are divided into two groups: Access-API-Lite and Management-API-Lite, and are described in the next sections.

9 Appendix C - KRSM Access-API-Lite

9.1 User

9.1.1 Services

- Retrieve user metadata
- Retrieve user configurations list
- Retrieve user tags list
- Retrieve user comments list
- Retrieve user ratings list
- Retrieve users list
- Retrieve user resources

9.1.2 Retrieve user metadata

9.1.2.1 URL syntax

GET: `http://{krsm-server}/{krsm-user-guid}`
retrieve user metadata

9.1.2.2 XML structure

```
<user>
  <canonical>
    <link type="user" href="KRSM User URI">
      {krsm-user-guid}
    </link>
    <link type="web-resource" href="PCM User URI">
      {pcm-user-guid}
    </link>
  </canonical>
  <config id="{krsm-config-guid}">
    <title>Display title</title>
  </config>
  ...
</user>
```

9.1.3 List User configurations

9.1.3.1 URL syntax

GET: `http://{krsm-server}/{krsm-user-guid}/configurations`
list all user's configurations

GET: `http://{krsm-server}/{krsm-user-guid}/configurations/{krsm-config-guid}`
retrieve user's configuration metadata

GET: `http://{krsm-server}/{krsm-user-guid}/configurations{krsm-config-guid}/content`
retrieve user's configuration encrypted data and return the datastream as a mime-typed application/octet-stream stream.

9.1.3.2 XML structure

```
<configurations>
  <config>
    <link type="configuration" href="{krsm-config-guid}">
      Display title
    </link>
  </config>
  ...
</configurations>
```

```
</configurations>
```

9.1.4 List user tags

9.1.4.1 URL syntax

```
GET: http://{krsm-server}/{krsm-user-guid}/tags
      retrieve all user's tags metadata
```

9.1.4.2 XML tructure

```
<tags>
  <tag>
    <link type="tag" href="{krsm-tag-guid}">
      Display title
    </link>
  </tag>
  ...
</tags>
```

9.1.5 List user comments

9.1.5.1 URL syntax

```
GET: http://{krsm-server}/{krsm-user-guid}/comments
      retrieve all user's comments
GET: http://{krsm-server}/{krsm-user-guid}/comments/{krsm-comment-guid}
      retrieve user's comment metadata
```

9.1.5.2 XML structure

```
<comments>
  <comment>
    <link type="comment" href="{krsm-comment-guid}">
      Comment display title
    </link>
    <link type="resource" href="{krsm-resource-guid}">
      Display title
    </link>
    <text>comment text</text>
    <published>date</published>
    <updated>date</updated>
  </comment>
  ...
</comments>
```

9.1.6 List user ratings

9.1.6.1 URL syntax

```
GET: http://{krsm-server}/{krsm-user-guid}/ratings
      list all user's rated resources
```

9.1.6.2 XML structure

```
<ratings>
  <rating>
    <value>xxx</value>
    <link type="resource" href="{krsm-resource-guid}">
      Display title
    </link>
    <published>date</published>
```



```

        <updated>date</updated>
    </rating>
    ...
</comments>

```

9.1.7 Retrieve users list

9.1.7.1 URL syntax

GET: `http://{krsm-server}/users`
retrieve all krsm users

9.1.7.2 XML result structure

```

<users>
  <user>
    <link type="user" href="xxxx">
      Display title
    </link>
  </user>
  ...
</comments>

```

9.1.8 List user's resources

9.1.8.1 URL syntax

GET: `http://{krsm-server}/{krsm-user-guid}/resources`
retrieve all published from user resources

9.1.8.2 XML structure

```

<resource-list>
  <resource>
    <link type="resource" href="{krsm-resource-guid}">
      Display title
    </link>
  </resource>
  ...
</resource-list>

```

9.2 Category

9.2.1.1 Services

- Retrieve
- Retrieve subcategory list
- Retrieve resource list

9.2.1.2 URL syntax

GET: `http://{krserver}/categories`
retrieve root category metadata
 GET: `http://{krserver}/{krsm-category-guid}`
retrieve category metadata

9.2.1.3 XML structure

```

<category xmlns="http://www.tencompetence.org/krsm/api/v1.0">
  <canonical>
    <link type="category" href="KRSM Category URI">
      {krsm-category-guid}
    </link>
    <title>Category title</title>
  </canonical>

```

```

<description>Category description</description>
<parent>
  <link type="category" href="KRSM Category URI">
    {krsm-category-guid}
  </link>
  <title>Category title</title>
</parent>
<children>
  <child>
    <link type="category" href="{krsm-category-guid}">
      Display title
    </link>
    <title>Category title</title>
  </child>
  ...
</children>
</canonical>
</category>

```

9.2.2 List subcategories

9.2.2.1 URL syntax

GET: `http://{krserver}/{krsm-category-guid}/subcategories`
retrieve list of subcategories

9.2.2.2 XML Result

```

<subcategories-list>
  <category>
    <link type="category" href="{krms-category-guid}">
      Display title
    </link>
  </category>
  ...
</subcategories-list>

```

9.2.3 List category resources

9.2.3.1 URL syntax

GET: `http://{krserver}/{krsm-category-guid}/resources`
retrieve all resources from a category

9.2.3.2 XML Result

```

<resource-list>
  <resource>
    <link type="resource" href="{krms-resource-guid}">
      Display title
    </link>
  </resource>
  ...
</resource-list>

```

9.3 Resource

9.3.1.1 Services

- Retrieve metadata
- Retrieve file
- Retrieve ratings
- Retrieve tags

- Retrieve comments

9.3.1.2 URL syntax

```
GET: http://{krsm-server}/{krsm-resource-guid}
      retrieve resource metadata
GET: http://{krsm-server}/resources
      retrieve a list with all resources
```

9.3.1.3 XML structure

```
<resource>
  <canonical>
    <link type="resource" href="{krsm-resource-guid}">
      {krsm-resource-guid}
    </link>
    <link type="web-resource" href="URL">
      Source Title (YouTube, Flickr)
    </link>
    <dc>
      <title>Resource title</title>
      <description> Resource description</description>
      ...
    </dc>
    <content>
      <type>File resource/Web resource</type>
      <link type="resource/web-resource" href="web-resource-
url/{krsm-resource-guid}">
        Display title
      </link>
    </content>
    <category>
      <link type="category" href="{krsm-category-guid}">
        Display title
      </link>
      <title>Category title</title>
    </category>
    <owner>
      <link type="user" href="{krsm-user-guid}">
        {krsm-user-guid}
      </owner>
    <tag-list>
      <tag>
        <link type="tag" href="{krsm-tag-guid}">
          Display label
        </link>
      </tag>
      ...
    </tag-list>
    <rating>0.nnn</rating>
  </canonical>
</resource>
```

9.3.2 Retrieve resource file

9.3.2.1 URL syntax

```
GET: http://{krsm-server}/{krsm-resource-guid}/content
      retrieve a resource file
```

9.3.3 List resource tags

9.3.3.1 URL syntax

GET: `http://{krsm-server}/{krsm-resource-guid}/tags`
retrieve all resource's tags

```
<tag-list>
  <tag>
    <link type="user" href="{krsm-user-guid}">
      Display title
    </link>
    <link type="tag" href="{krsm-tag-guid}">
      Display label
    </link>
  </tag>
  ...
</tag-list>
```

9.3.4 List resource comments

9.3.4.1 URL syntax

GET: `http://{krsm-server}/{krsm-resource-guid}/comments`
retrieve all resource comments

9.3.4.2 XML result

```
<comment-list>
  <comment>
    <link type="user" href="{krsm-user-guid}">
      Display title
    </link>
    <link type="comment" href="{krsm-comment-guid}">
      Display title
    </link>
    <text>comment text</text>
  </comment>
  ...
</comment-list>
```

9.4 Tag

9.4.1 List resources by tag

9.4.1.1 URL syntax

GET: `http://{krsm-server}/tags/{krsm-tag-guid}`
retrieve all resources tagged by tag

9.4.1.2 XML result

```
<resource-list>
  <resource>
    <link type="resource" href="{krsm-resource-guid}">
      Display title
    </link>
    <link type="user" href="{krsm-user-guid}">
      Display label
    </link>
  </resource>
  ...
</resource-list>
```

9.5 Search related services

9.5.1 Search by keyword based query

9.5.1.1 URL syntax

GET: `http://{krsm-server}/search/{query-string}`
 Search by keyword based query

9.5.1.2 XML result

```
<results>
  <resource>
    <canonical>
      <link type="resource" href="{krsm-resource-guid}">
        {krsm-resource-guid}
      </link>
      <link type="web-resource" href="URL">
        Source Title (YouTube, Flickr)
      </link>
      <dc>
        <title>Resource title</title>
        <description> Resource description</description>
        ...
      </dc>
      <content>
        <type>File resource/Web resource</type>
        <link type="resource/web-resource" href="web-resource-
url/{krsm-resource-guid}">
          Display title
        </link>
      </content>
      <category>
        <link type="category" href="{krsm-category-guid}">
          Display title
        </link>
        <title>Category title</title>
      </category>
      <owner>
        <link type="user" href="{krsm-user-guid}">
          {krsm-user-guid}
        </owner>
      <tag-list>
        <tag>
          <link type="tag" href="{krsm-tag-guid}">
            Display label
          </link>
        </tag>
        ...
      </tag-list>
      <rating>0.nnn</rating>
    </canonical>
  </resource>
  ...
</results>
```

Comments:

The query string is the same as in Fedora search:

<http://www.fedora.info/download/2.2.1/userdocs/server/webservices/apialite/index.html#d0e600>

Restriction – the query should use **terms=TERMS**.

9.5.2 Search by category

9.5.2.1 URL syntax

GET: `http://{krsm-server}/{krsm-category-guid}/search/{query-string}`
 Search by category

9.5.2.2 XML result

```
<results>
  <resource>
    <canonical>
      <link type="resource" href="{krsm-resource-guid}">
        {krsm-resource-guid}
      </link>
      <link type="web-resource" href="URL">
        Source Title (YouTube, Flickr)
      </link>
      <dc>
        <title>Resource title</title>
        <description> Resource description</description>
        ...
      </dc>
      <content>
        <type>File resource/Web resource</type>
        <link type="resource/web-resource" href="web-resource-
url/{krsm-resource-guid}">
          Display title
        </link>
      </content>
      <category>
        <link type="category" href="{krsm-category-guid}">
          Display title
        </link>
        <title>Category title</title>
      </category>
      <owner>
        <link type="user" href="{krsm-user-guid}">
          {krsm-user-guid}
        </owner>
      <tag-list>
        <tag>
          <link type="tag" href="{krsm-tag-guid}">
            Display label
          </link>
        </tag>
        ...
      </tag-list>
      <rating>0.nnn</rating>
    </canonical>
  </resource>
  ...
</results>
```

Comments:

The query string is the same as in Fedora search:

<http://www.fedora.info/download/2.2.1/userdocs/server/webservices/apialite/index.html#d0e600>

Restriction – the query should use **terms=TERMS**.

9.5.3 Search by tag

9.5.3.1 URL syntax

GET: `http://{krsm-server}/{krsm-tag-guid}/search/{query-string}`
 Search by tag

9.5.3.2 XML result

```

<results>
  <resource>
    <canonical>
      <link type="resource" href="{krsm-resource-guid}">
        {krsm-resource-guid}
      </link>
      <link type="web-resource" href="URL">
        Source Title (YouTube, Flickr)
      </link>
      <dc>
        <title>Resource title</title>
        <description> Resource description</description>
        ...
      </dc>
      <content>
        <type>File resource/Web resource</type>
        <link type="resource/web-resource" href="web-resource-
url/{krsm-resource-guid}">
          Display title
        </link>
      </content>
      <category>
        <link type="category" href="{krsm-category-guid}">
          Display title
        </link>
        <title>Category title</title>
      </category>
      <owner>
        <link type="user" href="{krsm-user-guid}">
          {krsm-user-guid}
        </owner>
      <tag-list>
        <tag>
          <link type="tag" href="{krsm-tag-guid}">
            Display label
          </link>
        </tag>
        ...
      </tag-list>
      <rating>0.nnn</rating>
    </canonical>
  </resource>
  ...
</results>

```

Comments:

The query string is the same as in Fedora search:

<http://www.fedora.info/download/2.2.1/userdocs/server/webservices/apialite/index.html#d0e600>

Restriction – the query should use **terms=TERMS**.

10 Appendix D - KRSM Management-API-Lite

10.1 User

10.1.1.1 Services

- Create
- Update
- Upload configuration encrypted data

10.1.2 Add User

10.1.2.1 URL syntax

```
http://{krsm-server}/{krsm-user-guid}
    create user with posted metadata with identifier {krsm-user-guid}
```

10.1.2.2 XML structure

```
<user>
  <canonical>
    <link type="web-resource" href="PCM User URI">
      {pcm-user-guid}
    </link>
  </canonical>
</user>
```

10.1.3 Add User configuration

10.1.3.1 URL syntax

```
http://{krsm-server}/{krsm-user-guid}/{krsm-config-guid}
    add new user configuration and store metadata
```

10.1.3.2 XML structure

```
<config id="{krsm-config-guid}">
  <title>Display title</title>
</config>
```

```
PUT, POST: http://{krsm-server}/{krsm-user-guid}/{krsm-config-guid}/content
    store user's configuration encrypted data
```

10.1.4 Add User Tag

10.1.4.1 URL syntax

```
http://{krsm-server}/{krsm-user-guid}/{krsm-tag-guid}
    add new tag
```

10.1.4.2 XML structure

```
<tag>
  <label>Display label</label>
```


</tag>

10.2 Category

10.2.1.1 Services

- Create
- Update

10.2.1.2 URL syntax

`http://{krserver}/{krsm-category-guid}`
create category with posted metadata

10.2.1.3 XML structure

```
<category xmlns="http://www.tencompetence.org/krsm/api/v1.0">
  <canonical>
    <title>Category title</title>
    <description>Category description</description>
    { <parent>
      <link type="category" href="{krsm-category-guid}">
        {krsm-category-guid}
      </link>
    </parent> } *
  </canonical>
</category>
```

**Note: If element "parent" is omitted, category will be added at root level*

10.3 Resource

10.3.1.1 Services

- Create
- Update

10.3.1.2 URL syntax

`http://{krsm-server}/{krsm-resource-guid}`
create resource with posted metadata

10.3.1.3 XML structure

```
<resource>
  <canonical>
    <dc>
      <title>Resource title</title>
      <description> Resource description</description>
      <subject>DC property</subject>
      <type>DC property</type>
      <source>DC property</source>
      <coverage>DC property</coverage>
      <creator>DC property</creator>
      <publisher>DC property</publisher>
      <contributor>DC property</contributor>
      <rights>DC property</rights>
      <date>DC property</date>
      <format>DC property</format>
```

```
        <identifier>DC property</identifier>
        <language>DC property</language>
    </dc>
    <link>
        <title></title>
        <url>web-resource url</url>
    </link>
    <category>
        {krsm-category-guid}
    </category>
    <owner>
        <uri>{krsm-user-guid}</uri>
    </owner>
</canonical>
</resource>
```

DC elements description:

<http://dublincore.org/documents/usageguide/elements.shtml>

10.3.2 Upload resource file

10.3.2.1 URL syntax

`http://{krsm-server}/{krsm-resource-guid}/content`
upload a file for a resource

10.4 Rating

10.4.1.1 Services

- Create
- Update

10.4.1.2 URL syntax

POST: `http://{krserver}/rating`
add comment to a krsm resource

10.4.1.3 XML structure

```
<rating>
    <value>xxx</value>
    <user>
        <uri>{krsm-user-guid}</uri>
    </user>
    <resource>
        <uri>{krsm-resource-guid}</uri>
    </resource>
</rating>
```

10.5 Tagging

10.5.1.1 Services

- Create
- Update

10.5.1.2 URL syntax

POST: `http://{krserver}/tagging`

tag a resource

10.5.1.3 XML structure

```
<tagging>
  <tag>
    <uri>{krsm-tag-guid}</uri>
  </tag>
  <user>
    <uri>{krsm-user-guid}"></uri>
  </user>
  <resource>
    <uri>{krsm-object-guid}</uri>
  </resource>
</tagging>
```

10.6 Commenting

10.6.1.1 Services

- Create
- Update

10.6.1.2 URL syntax

POST: `http://{krserver}/commenting`
comment a resource

10.6.1.3 XML structure

```
<comment>
  <title>comment title</comment>
  <text>comment text</text>
  <user>
    <uri>{krsm-user-guid}"></uri>
  </user>
  <resource>
    <uri>{krsm-object-guid}</uri>
  </resource>
</comment>
```