Building the European Network
For Lifelong Competence Development

TENCompetence IST-2005-027087

Project Report

ID5.10: LearnWeb2.0 second cycle system evaluation results

Workpackage	WP5
Task	5.4
Date of delivery	**Contractual**: 31-07-2008 **Actual**: 30-07-2008
Code name	ID5.10 **Version**: 1.1 Draft ☐ Final ☒
Type of deliverable	Report
Security (distribution level)	Public
Contributors	
Authors (Partner)	Alessandro Mazzetti (Giunti Labs), Michele Dicerto (Giunti Labs), Mar Pérez Sanagustín (UPF), Patricia Santos (UPF), Alexander Grigorov (SU), Sergej Zerrr (UHANN), Rubén Sánchez Benítez (Altran), Atanas Georgiev (SU).
Contact Person	Alessandro Mazzetti (Giunti Labs).
WP/Task responsible	UPF
EC Project Officer	Martin Májek
Abstract (for dissemination)	
Keywords List	WP5, internal deliverable, Evaluation, LearnWeb2.0, code quality

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*	*12.06.08*	*Alessandro Mazzetti, Giunti Labs*
0.2	*Draft*	*15.07.08*	*Alessandro Mazzetti, Giunti Labs, Mar Perez, Patricia Santos*
0.3	*Draft*	*15.07.08*	*Alessandro Mazzetti, Giunti Labs, Rubén Sánchez Benítez, Michele Dicerto, Alexander Grigorov, Atanas Georgiev*
0.4	*Draft*	*24.07.08*	*Sergej Zerr, UHANN*
0.5	*Draft*	*25.07.08*	*Mar Pérez and Patricia Santos*
1.0	*Final*	*30.07.08*	*Alessandro Mazzetti, Giunti Labs*
1.1	*Revised*	*29.09.08*	*Alessandro Mazzetti, Giunti Labs*

Summary of Changes

Version	*Section(s)*	*Synopsis of Change*
0.1	*All*	*Draft: Table of Contents*
0.2	*-*	*Corrected Project Officer name*
	1,4,6	*Added*
	2	*Integrated contribution from UPF*
0.3	*3.5*	*Integrated contribution from Altran*
	3.6, 3.7	*Integrated contribution from Giunti*
	3.1, 3.2	*Integrated contribution from SU*
	3.3, 3.4	*Integrated contribution from UHANN*
	5	*Integrated contribution from SU*
	7	*Added annex*
0.4	*3.4*	*Fixed one errors with checkbox 3.4.1.3*
0.5	*2*	*Improvements from UPF*
1.0	*2.4, 2.5*	*Integrated the second cycle evaluation*
1.1	*all*	*Applied corrections suggested by reviewers*

Table of Contents

1 Introduction (Leader: GiuntiLabs)

The evaluation of LearnWeb2.0 is carried out following the methodology and the guidelines of the project. This deliverable refers to the following documents:

- TENCompetence Handbook page 66
- D 4.1 Pilot Evaluation Plan page 116, appendix 3
- ID5.9: KRSM first cycle evaluation outcomes (WP5 previous evaluation)
- DIP-3 version 1.1 page 41 (task description, deliverable definition)

The evaluation stage of LearnWeb2.0 is very important because it is a new product and it still suffers of immaturity.

The evaluation process has been organized in six phases:

1. Functionality proof carried out by real users
2. Code quality evaluation carried out by developers, exchanging their role
3. Improvements/enhancements identification
4. Implementation of main improvements and bug fixing
5. Second cycle proof carried out with developers
6. Identification of remaining improvements/enhancements

2 Validation of the LearnWeb2.0 (UPF)

In this section we present the main conclusions of the first evaluation of the LearnWeb2.0 tool. This evaluation has focused in two main aspects:

1. The evaluation of the system functionalities.
2. The evaluation of the utility of the LearnWeb2.0 tool in a real context.

In order to achieve these goals we develop the evaluation planning according to the scenarios developed in the internal deliverable ID5.12. We divided the evaluation in two main parts: a functional testing and an evaluation experience with real users.

The functional testing of the LearnWeb2.0 consists on a set of tests run by two software experts. The results of this testing show the main errors and faults of the system: LearnWeb2.0. They have been carefully collected in checklists including the observations and some comments to improve the system.

The evaluation experience was performed in La Verneda School for Adults (http://www.edaverneda.org/) with 14 real users. The results of this evaluation allow as to analyze those quality attributes such as reliability or usability.

The tests have been done with the application available at http://phpcake.it.fmi.uni-sofia.bg/.

2.1 Functional testing: Unit and Integration Checklist (UPF)

In order to provide a complete analysis of all the system functionalities we have developed a Unit and Integration tests. The Unit testing searches for defects in, and verifies the functioning of software that is separately stable. It has been run by a software expert different from the developer. The Integration testing consist in verifying if the units work evaluated in the Unit testing are correctly integrated and related among them: interfaces between units, complete processing chain and relation controls within the system including several modules and /or in combination with database. We also performed a test of the functionality requirements established in deliverable ID5.12. The results of these tests are indicated and explained in the following tables:

- **Table 1**: Unit and Integration Test: Calamities
- **Table 2:** Unit and Integration Test: Functionality
- **Table 3:** Unit and Integration Test: Integration
- **Table 4:** Unit and Integration Test: Storage
- **Table 5:** Functional Requirements Test

We haven't distinguish the aspects that correspond to the Unit test or the Integration test for providing an easy reading, however the distinctions and a more detailed description of each of them are explained in (D 4.1 Evaluation Plan, p. 118 and 119). In the observation column, we marked with Fail (in red) those aspects that doesn't work properly and have to be reviewed, with Ok (in green) those which work properly and with Ok (in orange) those which work fine but have to be reconsidered for different aspects.

Unit and Integration Test: *Calamities*	
Aspect: Monkey test	**Observations**
Main Menu	
Functionality *Home*	Fails. This link is not related to any (information) page.
Functionality *My HomePage*	Ok
Functionality *Search*	Ok
Menu Option *My HomePage*	
Functionality My Profile	Ok
Functionality Change Photo	Fails. It is not available
My Bookmarks	Ok
My Resources	Ok, it links correctly to the resource
Inside My Resources	- Resource button: Ok - Download: It opens a new web page but doesn't allow the user to save directly the resource. - Share: Fails - Upload a modified version: Fails

	- Rate: Ok - Insert your tag: Fails. It doesn't show to the user the list of tags associated to his account. - Insert your comment field: Ok
My Tags	Ok
Inside My Tags	- Tag button: Ok
My Comments	Ok
Inside My Comments	- Resource button: Ok
My Rates	Ok
Inside My Rates	- Resource button: Ok - Edit metadata: Ok
My Groups	Ok
Menu Option *Search*	
Selection menu	Ok, however, the categories don't vary. If the user has 200 categories, there will appear 200 categories in the selection menu?
Create/Upload resource Menu	
Create a new group	Ok
Upload a picture	It fails sometimes with the direct access to the application (it doesn't recognize the password added in the profile).
Upload and audio file	It fails sometimes with the direct access to the application (it doesn't recognize the password added in the profile).
Upload a video	It fails sometimes with the direct access to the application (it doesn't recognize the password added in the profile).
Upload a file	Ok
In Upload a file: File Browse button	Ok
In Upload a file: Date menu	Ok
Other buttons	
Logout	When clicking on this option, the system doesn't refresh the screen with the new information and leave the information from the previous functionality.

Unit and Integration Test: *Functionality*	
Aspect	**Observations**
Field type	Which values are accepted or refused and which length is supported
Login field	Ok. All values and any length except spaces.
Pass field	Ok. All values and any length.
URL field	Ok. It only support strings with an URL format.
Other metadata fields	Ok. All values and any length.
Validity test date fields	
Date in the Upload a file option	Ok. We have to clarify if it corresponds to the date in which the resources is uploaded or in which it is created. If we choose the last option it is ok, if we choose the other one, we have to delete the years before 2008.
Checking the overview screens	Checking if all the overview screens are present.
Home overview	Fails. It doesn't load anything.
My HomePage	Ok
Search	Ok
Resources overview	Ok
Paging up and paging down on overview screens	
Home overview	Ok
My HomePage	Ok
Search	Ok
Resources overview	Ok. Necessary to scroll down to see the space for comments when it is void of comments.
Completeness	Checking if all the buttons or fields are present
Change photo (in My HomePage)	Fails. It is not activated.
Share (in the Resources Overview)	Fails. It doesn't do anything.
Upload modified version (in the Resources Overview)	Fails. It doesn't do anything.
Delete buttons	Fails. No option for delete a resource is available.
Position on screen	Check if all the objects are on the right position on the screen
Menu My HomePage	Fails. It should be located in the centre of the screen.
Search Menu	Fails. It is not very visible, it should be reallocated.

Unit and Integration Test: *Integration*	
Aspect	**Observations**
Integration between different subsystems of the application	Ok.
Testing Broadcasts (internal messages) sent from one system to other systems	Ok. We have to review the access with the Fedora Data Base, is sometimes happens that, when uploading a resource and another user enters to the system, this new user doesn't find the resources uploaded by the other.

Unit and Integration Test: *Storage*	
Aspect	**Observations**
Deleting records	Fails. The system doesn't provide the option to delete a resource.
Deletion of multiple records at the same time	Fails. The system doesn't provide the option to delete a set of resources at the same time.
Deletion of all records in one action	Fails. The system doesn't provide the option to delete a set of resources at the same time with one action.(using only one button)
Check if the fields are stored correctly when storing a new record.	Ok. But we have to provide a shorter list of document possibilities and automatically detect the file type because if the user fails selecting the file type, he could not upload the resource.
Concurrent usage	Fails. It is not enough stable in this aspect. When two users have access to the same resource and one of them adds a comment to the document, the other user has problems with the visualization of the contribution to the resource of his colleague.
Time out	Fails. With good connections, it works slowly. With slow connections, the application doesn't run properly and the access to the resources becomes impossible because the session finishes before finishing the access.

Functional Requirements Test	
Aspect	**Observations**
Create a Knowledge Resource	Fails. The system doesn't provide the option to delete a resource.
Deletion of multiple records at the same time	Fails. The system doesn't provide the option to delete a set of resources at the same time.
Deletion of all records	Fails. The system doesn't provide the option to delete a set of

in one action	resources at the same time with one action.(only using one button)
Check if the fields are stored correctly when storing a new record.	Ok. But we have to provide a shorter list of document possibilities and automatically detect the file type because if the user fails selecting the file type, he could not upload the resource.
Concurrent usage	Fails. It is not enough stable in this aspect. When two users have access to the same resource and one of them adds a comment to the document, the other user has problems with the visualization of the contribution to the resource of his colleague.
Time out	Fails. With good connections, it works slowly. With slow connections, the application doesn't run properly and the access to the resources becomes impossible because the session finishes before finishing the access.

2.2 Quality Testing (UPF)

Quality test: Reliability ***Maintain a specified level of performance when use under specified conditions***	
Aspect	**Observations**
Maturity	- Ok. The tool doesn't fall down when an error is produced.
Fault tolerance	- Ok. The tool maintains a level of performance in cases of software faults.
Recoverability	- Fails. The tool doesn't maintain the information and the data introduced by the user when
Reliability	- Ok

Quality test: Usability ***Be understood, learner, used and attractive to the user, when used under specified conditions***	
Aspect	**Observations**
Understandability	- Fails. - The name of the titles and menus have to be reviewed in order to express better its contexts: "Upload a file" for "Store a resource". - The pages have to be refreshed according to user actions. - Main menu: Once the user clicks on a menu and, after this, he chooses another menu option the central page is not refreshed and the user sees the information listed with the previous menu. - The tree of the categories resource has to be clarified. - Add pop up windows in each of the functionalities.

	- Add an option in each page to come back to the previous page. - Reorganize the menus in the screen. - Increase the time of the session. - Maintain the fields configured in the Profile option from within sessions.
Learnability	- Fails. - Use more intuitive icons and a better organization of the functionalities (see comments in section 2.4). - Use constant structures and organizations of the menus
Operability	- Fails. - Add information messages. - Decrease the wainting times for loading pages. - Activate all the buttons and functionalities.
Attractiveness	- Ok - The design is coherent with the design of the TENCompetence tools. - Distribute correctly the menus on the screen to make it more attractive.

2.3 Validation proof with real users (UPF)

This section shows the results of a validation proof of the LearnWeb2.0 in La Verneda School for Adults in Barcelona (http://www.edaverneda.org/). The idea was to use the LearnWeb2.0 tool in a real context with non-expert users and with other technical characteristics.

For the evaluation we planned an activity of one hour in which 14 users. There were 14 students working in pairs and there was one computer per pair, this is, 8 computers. Each group has a different account to access the application. All the accounts' profiles were prepared before the experience by the evaluators and have access to the same accounts of each of the Web 2.0 services integrated in the tool. Thus, the accounts for the Web 2.0 were shared by all the users of the activity.

For the activity, the students received a printed document with the main instructions for performing the activity. Three evaluators wrote the main problems detected and their observations during the experience and a questionnaire for the users was designed (see annex Questionnaire Experience LaVerneda). The questionnaire include three questions related to the three main points to evaluate: 1) the tool as a support for stimulating the Knowledge resource sharing, 2) the tool as a support for educational contexts and 3) the too as a Knowledge Resources repository.

The activity was planned in different steps:

1. Login to the application LearnWeb2.0.
2. Access to *My HomePage* and navigate through the menu *Create/Upload a Resource.*

3. Go to the functionality *Upload a File* and store a picture from the desktop.
4. Go to the search functionality and look for the pictures of the rest of the group.
5. Comment some of the pictures using the functionality *Insert your comment.*

2.3.1 Observations from the experience

The experience with real users showed that there are some aspects of the tool that has to be improved. The first problem appear in the first step of the activity, when the users tried to login the application. Only three of the groups could access to the application. The other users have problems when loading the main page and screen and insert the user name and password. Then in the same page appears a message with the text “You are not logged”. They couldn’t continue the activity. One of the hypothesis is the low internet access quality level. Although in the previous hour they can use without any problem the following tools: Google, Flickr, Picasa and Slideshare.

From the three groups that could access to the application, two of them couldn’t see the icons in *My HomePage* menu and had to stop the activity at this point. Only one of the groups could access to the *Upload a file* functionality but, when clicking on it, a new window was open without any information.

The experience had to finish at this point after 30 minutes trying to access the application without success. We cannot do the questionnaires but we expect to do it in next validation proofs.

Here we transcribe the observations of one of the evaluators:

“The experience begins at 10:15 am. There are 14 students and some of they are in pairs to proof the tool. In summary, there are 8 groups per computer.
First, they have to put the address of the leanweb2.0 tool in the navigator, (one of the problems is the "-" between uni-sofia).
All the groups insert the link and they can see the home page of the tool.
Then they insert the user-name and password (eval 1.... eval8). The majority of the groups have problems with the login, because the tool after showing the window "login....", finally in the main window there are a message "you are not logged". They can't visualize the other functions.
Others groups that can login, have problems with the visualization of some of the icons of the tool.
Only one group can view all the icons, but when they press the button "Upload a File" a new window appears without any information.
The experience have to finalise because the students after try during 30 minutes to accessing to the tool, they can't.
In the previous hour the students had been using the following tools: flickr, picasa, slideshare and google

2.3.2 Conclusions and recommendations of the experience

The experience with real users showed some limitations of the tools that will be solved in next versions. In the following section we list some of them and recommendations for the improvement of the tool. The rest of the problems more related to usability or reliability aspects are included in section 2.2 of this document.

- Solve the concurrency problems when more than two users access to the application.
- Prepare a version of the tool for systems with low Internet connection.
- Add information messages for the user.

2.4 *Second Cycle Functional testing (UPF)*

The recommendations of the first evaluation cycle led to a quick bug fixing and the implementation of urgent enhancements.

A second cycle testing has been performed in collaboration with developers, for an immediate bug fixing and server tuning.

We have tried three different browsers:

- Mozilla 2.0.0.16
- Safari 3.1.2
- Internet Explorer 7.0.5730.11

The outcome is depicted in the following tables:

Unit and Integration Test: *Calamities*	
Aspect: Monkey test	**Observations**
Main Menu	
Functionality *Home*	Fails. This link is not related to any (information) page. Recommenadation: add a text explaining what is LearnWeb 2.0 and its uses. Add the last 5 comments, rates... that have been uploaded.
Functionality *My HomePage*	Ok
Functionality *Search*	Ok
Menu Option *My HomePage*	
Functionality My Profile	Fails with IE. Recommendation: show the credentials already stored by the user in previous sessions. **Java script error when filling the boxes in my profile and click on login button with IE:** Webpage Script Errors

	User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; InfoPath.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648) Timestamp: Wed, 23 Jul 2008 07:46:53 UTC 0. Message: 'length' is null or not an object Line: 301 Char: 21 Code: 0 URI: http://phpcake.it.fmi.uni-sofia.bg/users/login
Functionality Change Photo	Fails. It is not available
My Bookmarks	Ok The autentication functionalitiy doesn't work properly with Safari and Internet Explorer.
My Resources	Ok, it links correctly to the resource
Inside My Resources	- Resource button: Ok - Download: It opens a new web page but doesn't allow the user to save directly the resource. - Share: Fails. Not implemented yet. - Upload a modified version: Ok ○ **Recommendation:** all the metadata of a resource filled by the creator of the resource should be showed in the overview. Now, it is not possible to see the description of the resources written by its creator. This information can be included in an option for expanding the information about the resource. - Rate: Ok - Insert your tag: Ok - Insert your comment field: Ok
My Tags	Ok
Inside My Tags	- Tag button: Ok
My Comments	Ok
Inside My Comments	- Resource button: Ok
My Rates	Ok
Inside My Rates	- Resource button: Ok - Edit metadata: Ok
My Groups	Ok. **Recommendation:** the service should be open in another winddow.
Menu Option *Search*	
Selection menu	Ok, however, the categories don't vary.

	Recommendation: - When a resource is displayed, the category should be showed in the overview page. - When searching by categories, a link to a resource should be showed next to each category. - The user should select the where to search for resources: in fedora or in web 2.0 services. We should add this option.
Create/Upload resource Menu	
Create a new group	Ok
Upload a picture	Ok
Upload an audio file	Ok
Upload a video	Ok
Upload a file	Ok. **Recommendations:** The pop-up menu doesn't work properly for the option upload a file. It only works sometimes.
In Upload a file: File Browse button	Ok
In Upload a file: Date menu	Ok
Other buttons	
Logout	When clicking on this option, the system doesn't refresh the screen with the new information and leave the information from the previous functionality.

Unit and Integration Test: *Functionality*	
Aspect	**Observations**
Field type	Which values are accepted or refused and which length is supported
Login field	Ok. All values and any length except spaces.
Pass field	Ok. All values and any length.
URL field	Ok. It only support strings with an URL format.
Other metadata fields	Ok. All values and any length.
Validity test date fields	
Date in the Upload a file option	Ok. **Recommandation:** - The date corresponds to the date in which the resource is uploaded to Fedora and it should be filled automatically by the LearnWeb 2.0 application. The user should not be able to change it when uploading a resource, only when editing the metadata associated to it.
Checking the overview screens	Checking if all the overview screens are present.
Home overview	Ok
My HomePage	Ok
Search	Ok
Resources overview	Ok
Paging up and paging down on overview screens	
Home overview	Ok
My HomePage	Ok
Search	Ok
Resources overview	Ok. Necessary to scroll down to see the space for comments when it is void of comments. **Recommendation:** - Add an option for the user to select the number of resources to be shown per page and don't show all the resources withoun pagging them.
Completeness	Checking if all the buttons or fields are present
Change photo (in My HomePage)	Fails. It is not implemented yet.
Share (in the Resources Overview)	Fails. It doesn't do anything. **Recommendation:** - Since all the resources in the repository are public they are also shared. We suggest to delete this option from the resource overview.

Upload modified version (in the Resources Overview)	Ok .
Delete buttons	Fails. No option for delete a resource is available in this version.. It would be implemented in next versions.
Position on screen	Check if all the objects are on the right position on the screen
Menu My HomePage	Fails. **Suggestion:** It should be located in the centre of the screen.
Search Menu	Fails.
Unit and Integration Test: *Integration*	
Aspect	**Observations**
Integration between different subsystems of the application	Ok.
Testing Broadcasts (internal messages) sent from one system to other systems	Ok. We have to review the access with the Fedora Data Base, is sometimes happens that, when uploading a resource and another user enters to the system, this new user doesn't find the resources uploaded by the other.

Unit and Integration Test: *Storage*	
Aspect	**Observations**
Deleting records	Fails. The system doesn't provide the option to delete a resource. It would be implemented in next verisons.
Deletion of multiple records at the same time	Fails. The system doesn't provide the option to delete a set of resources at the same time.
Deletion of all records in one action	Fails. The system doesn't provide the option to delete a set of resources at the same time with one action.(using only one button)
Check if the fields are stored correctly when storing a new record.	Ok. But we have to provide a shorter list of document possibilities and automatically detect the file type because if the user fails selecting the file type, he could not upload the resource.
Concurrent usage	Ok. The system supports now 10 concurrent users but it will be extended in the next versions.
Time out	Ok

Functional Requirements Test	
Aspect	**Observations**
Create a Knowledge Resource	Ok. The system allows the user to upload to the repository any type of file and share it with other users.
Deletion of multiple records at the same time	Fails. The system doesn't provide the option to delete a set of resources at the same time. I will be implemented in next versions.
Deletion of all records in one action	Fails. The system doesn't provide the option to delete a set of resources at the same time with one action (only using one button). I will be implemented in next versions.
Check if the fields are stored correctly when storing a new record.	Ok. **Recommendation:** - The next version will detect authomatically the type of file that the user is uploading.
Concurrent usage	Ok. The system supports 10 users concurrently. It will be extended in next versions.It is not enough stable in this aspect.
Time out	Ok

2.5 Second cycle quality testing (UPF)

Quality test: Reliability ***Maintain a specified level of performance when use under specified conditions***	
Aspect	**Observations**
Maturity	- Ok. The tool doesn't fall down when an error is produced.
Fault tolerance	- Ok. The tool mainatin a level of performance in cases of software faults.
Recoverability	- Ok
Reliability	- Ok

Quality test: Usability ***Be understood, learner, used and attractive to the user, when used under specified conditions***	
Aspect	**Observations**
Understandability	- **Recommendations:** o The name of the titles and menus have to be reviewed in order to express better its contexts: "Upload a file" for "Store a resource". o The pages have to be refreshed according to

	user actions. ○ Main menu: Once the user clicks on a menu and, after this, he chooses another menu option the central page is not refreshed and the user sees the information listed with the previous menu. ○ The tree of the categories resource has to be clarified. ○ Add pop up windows in each of the functionalities. ○ Add an option in each page to come back to the previous page. ○ Reorganize the menus in the screen. ○ Increase the time of the session. ○ Maintain the fields configured in the Profile option from within sessions.
Learnability	- **Recommendations:** ○ Use more intuitive icons and a better organizations of the functionalities (see comments in section 2.4). ○ Use constant structures and organizations of the menus
Operability	- **Recommendations:** ○ Add information messages. ○ Decrease the wainting times for loading pages. ○ Activate all the buttons and functionalities.
Attractiveness	- **Recommendations:** ○ The design is coherent with the design of the TENCompetence tools. ○ Distribute correctly the menus on the screen to make it more attractive.

2.6 Summary of the evaluation results and recommendations (UPF)

This section presents a summary of the results extracted from the analysis of the functional evaluation and the validation proof with real users a list of recommendations. We have organized them according with the basic requirements extracted from Scenarios of the document ID5.12.

Search

The application allows two types of search:

1. Simple search

   It is performed by the functionality “Search”. The user can introduce any type of word or sentence and a list of resources associated to it are showed. The evaluators have found that when introducing more than one word, the functionality search only look for those resources associated to the first one and doesn't consider the second one.

2. Advanced Search

It is performed by using the menu located on the right of the search field in which the user can select the category he/she wants to look for. This functionality doesn't work properly and the evaluators found that it would be necessary to allow the search by the fields included in the metadata, such as author or date in this same overview page. Another aspect that fails is, once the user adds a new category, this is not added to this search menu. Moreover, the evaluators considered that this functionality has to be reviewed in next versions. It has to be analized if the application should list all the categories that the user has created or only 5 categories, at least (maybe the most used).

The user has also the possibility to search in his/her resources using the functionalities *My Tags, My resources....* These options show the list of resources organized by tags, rates .... according to the functionality chosen.

Recommendations:

- Solve the search option when the user uses more than one word for the search.
- Review the advanced search considering only the most used categories. (or establishing a generic categories)
- Implement a search by the metadata information of the resources.

Browse

It is performed using the functionality "Search". The user can search new resources from other users in the repository. When clicking on it, a tree of categories appears. The user can search by the different metadata fields of the resource by clickin on it. The evaluators noticed that, when browsing for category, any resource was listed and it was not possible to look for the different fields in the metadata. They also found that the tree representation was not enough clear.

Recommendations

- Add the possibility of searching by the fields in the metadata directly in the search overview page.
- Review the tree representation of categories providing a more informational and intuitive representation.
- Add information messages for the users when it is not possible to look by author or rating if there are no resources related to this category.
- Avoid showing those categories that doesn't have any resource associated.

Discovery of related resources

According to the document ID5.12, when looking for a resource the user should visualize a resource with a cloud of tags and a list of communities associated to it. This is not yet implemented in this version of the application.

Social Bookmarking

The LearnWeb 2.0 tool provides a means for interoperability with the existing Web 2.0 tools Del.icio.us for managing bookmarks and GroupMe for configuring groups. The first one is accecible by the option *My Bookmarks* and the second one by *My Groups*. When using this last one, the application redirects the user to the GroupMe page without opening a new page and the main LearnWeb 2.0 overview page is lost. The tests also show that there are problems with the direct acces of both applications. It seems that there are problems with the login and password storage.

Recommendations

- The link *My Bookmarks* should be be opened in a new window.

Aggregation of resources

There are two different type of agregation of resources: to aggregate a resource to the repository and to aggregate a resource to a Web 2.0 service.

- ***Aggregate a resource to the repository***

The current version of the LearnWeb 2.0 supports the aggregation of the Knowledge Resource using the option in My HomePage overview called "Upload a file". This brings the user to a metadata editor in which she/he has to fill all the information about the resource, which can be any type of file (a .pdf, an audio file, a video...). The tests show that uploading a resource is successful when the user fills correctly all the fields. However, if the user makes an error when filling the information message explaining which have been his/her error (e.g. when filling the URL if this is not correct, there is no message explaining the reason). The evaluators also found confusing the name of the option "Upload a file". Since the other options are also "Upload a Video" or a picture, it is confusing for the user because, using this functionality the user can upload any type of file. They also found the system for selecting the type of file is too difficult and he/she doesn't know sometimes which type of file should choose.

- ***Aggregate a resource to a Web 2.0 service***

The tests showed that the option of creating a group always work but the options of uploading a video, a picture or an audio fail in some cases. It seems that the login and passwords related to these services are not well stored. The evaluators also observed that it was confusing to group all this functionalities under a menu called "Create/Upload a Resource" in which it is also included the option to store a resource into the repository.

Recommendations:

- Separate the functionalities of create a group and upload a video, a picture and an audio from the option upload a file and allocate them into a menu called "Upload your resource to a Web 2.0 service" (delete the word "create" because it can be confusing. The user cannot create, can upload an already existing resource). Add

the *Upload a file* option separately from the options in which the user upload a file to a web 2.0 services.
- Maintain the titles "Upload an audio", "Upload a video" and "Upload a picture" but include some icon of the web 2.0 service in which they are going to be uploaded.
- Add information messages when a field in the metadata is not correctly added.
- Change the name "Upload a file" by "Store a Resource"
- Simplify the file type selection. Recognize automatically the type of file that the user is uploading.

Delete a Knowledge Resource
The current version of the LearnWeb 2.0 tool doesn't support the delete functionality. But is planned to add it in the next version when the authorization mechanisms provided by WP3 would be included.

Recommendations:
- Create a new title menu "Store a Resource" (see the above section "Store a Knowledge Resource").

Sharing
The sharing concept is treated in the tool from different perspective. A user can share his/here resource with the others, can recommend a resource, edit the resource of other user... We list here the different forms of sharing that the LearnWeb 2.0 tool offers.

- ***Share***

The share functionality is accessed directly from the overview page of the resource by clicking on *Share*.. It allows the user to share the resource with the other users of the system. This functionality doesn't work.

- ***Collaborative creation/modification of Resources: Edit a knowledge resource***

The user edit a knowledge resource by using the option "Upload modified version" in the overview page of a resource. However the evaluators found it confusing because this option don't allow the user to upload a new resource but change its metadata associated. Another aspect to review is the integration of tools for collaborative creation, such as "Google Docs".

- ***Discussions around resources***

The tool supports a mechanism to set up discussions around resources through the functionality *Insert your comment* associated to each of the resources in the repository. Some problems were detected when two users accessed to the same resource because the comments were not refreshed correctly.

Recommendations:

- Change the name "Upload modified version" by "Edit Resource" or "Change description of the Resource".
- Consider the integration of collaborative edition tools such as Google Docs.

Recommendation of resources

- ***Rating***

The application includes the option of rating a resource. When the user access to the overview page of a resource he/she has the possibility of rating the resource by moving the mouse over a set of stars indicating the quality level. No problems found with this option. The user can also to look up the resources that he/she has rate using the functionality "My Rates".

- ***Tagging***

The user can tag a resource using the option *Tag*. This functionality works correctly.

- ***Comment***

The users can comment any resource and see other comments. The evaluators detected some problems when two users accessed to the same resource and edit a comment.

Organize my Knowledge Data
The LearnWeb 2.0 tool allows the user to organize its Knowledge data by using the functionalities: "My Bookmarks", "My Resources", "My Tags", "My comments", "My Rates" and "My Groups".

All this options work properly and allow the user to organize his/her resources in relation to his/her tags, however, the evaluators have seen that the functionalities My Bookmarks and My Groups sometimes fails. Again, it the system seems to have problems remembering the login and the passwords to the systems.

Download (retrieve) a resource
Resources found in the system can be downloaded by the user to use, modified or change it. This can be done using the functionality Download in the resource overview page. When clicking on it, a new page is open with the resource and the user has to click over it using the right button of the mouse to save it. The evaluators found that it was necessary to consider the possibility of downloading directly the file when clicking on the option.

Recommendations
- Review the download functionality for those resources that have been uploaded as a file resource type. The system should allow the user to save this resources directly.

Identity Management
One of the main characteristics of this version of the tool is the integration of Web 2.0 services. In order to make this integration transparet for the user, the tool includes a

profile functionality in which the user write down the login and the passwords to the different web 2.0 services integrated in order to avoid the indentifications for each functionality. This can be done through *My profile*, in *My HomePage* overview page. The evaluators notice that the button Decrypt doesn't work and that, once the user introduce his/her data, the information is not showed for other sessions. The user cannot see which are the fields that he/she has already filed in past sessions.

Recommendations:

- Review the *Decrypt* button. When you fill in password and press decrypt usernames' fields are initialised with usernames but not password fields.
- Consider to maintain the information added in the profile from one session to another.

General look and feel of WebLearn 2.0

The comments regarding these aspects are explained and developed in section 2.2 of this document.

3 Coding quality

3.1 MyHome page (SU)

3.1.1 Correct use of Object Oriented programming

3.1.1.1 Efficiency - Are the constructs efficiently designed?

[✓ Yes]

☐ No (add your remarks)

Remarks:

3.1.1.2 Complexity – Do the constructs increase the architectural complexity?

☐ Yes (add your remarks)

[✓ No]

Remarks:

3.1.1.3 Understandability - Does the design increase the psychological complexity?

☐ Yes (add your remarks)

[✓ No]

Remarks:

3.1.1.4 Reusability - Does the design quality support possible reuse?

☒ Yes

☐ No (add your remarks)

Remarks:

3.1.1.5 Testability/Maintainability - Does the structure support ease of testing and changes?

☒ Yes

☐ No (add your remarks)

Remarks:

3.1.1.6 Proper use of try/catch and managing of Exceptions

☐ Yes

☒ No (add your remarks)

Remarks: There is no error trapping

3.1.1.7 What is the quality of the javadoc (or javadoc like) documentation of the code?

☒ Not required/possible (explain why)

☐ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but does not exists (add your remarks)

3.1.1.8 Inline comments

☒ Not required/possible (explain why)

☐ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but do not exists (add your remarks)

3.1.2 Code conventions

3.1.2.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

☐ Yes

☐ Could be improved (add your remarks)

3.1.2.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.1.2.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.1.2.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.1.2.5 Are variable names human readable?

☒ Yes

☐ Not in all necessary cases (add your remarks)

Remarks:

3.1.2.6 Over aspects of coding quality

- [x] Perfect
- [ ] Problematic (add your remarks)

Remarks:

3.1.3 Overall code quality

3.1.3.1 Does the dead code exist?

- [ ] No
- [x] Yes (add your remarks)

Remarks: edit_metadata **method in myhome_controller**

3.1.3.2 Is the code efficient?

- [x] To my best knowledge - yes
- [ ] Not always (add your remarks)

Remarks:

3.1.3.3 Are some items, that might be changed in the future hardcoded?

- [x] No
- [ ] Some of them (add your remarks)

Remarks:

3.1.3.4 How is the English quality within code and comments?

- [x] Very good
- [ ] Good
- [ ] Acceptable
- [ ] Poor

Remarks:

3.1.4 Summary

The code is very well written and conforms to the standards used in the CakePHP framework. Proper naming conventions are used and the names of variables, member functions and classes are self explanatory. Classes are JavaDoc style commented and also proper inline comments are used. Some commented lines can be removed.

Some dead code (edit_metadata method in myhome_controller) should be removed.

There are no hard-coded items and the configuration data is separated in the configLWComponent class. However, we suggest that the configurations constants about the URLs of the severs (Web services and PCM sever) to be moved to a configuration file (for example bootstrap.php) in app/config in order to make the installation of LearnWeb easier.

Another recommendation is to use try-catch constructs when calling the Web services and when processing the XML. In certain cases, the Web services may not return a proper XML and this can cause the DOM XML parser to raise an exception which should be caught in try-catch statements.

Our conclusion is that the quality and efficiency of the code are very good.

3.2 *User profile page (SU)*

3.2.1 Correct use of Object Oriented programming

3.2.1.1 Efficiency - Are the constructs efficiently designed?

[✓ Yes]

☐ No (add your remarks)

Remarks:

3.2.1.2 Complexity – Do the constructs increase the architectural complexity?

☐ Yes (add your remarks)

[✓ No]

Remarks:

3.2.1.3 Understandability - Does the design increase the psychological complexity?

☐ Yes (add your remarks)

☒ No

Remarks:

3.2.1.4 Reusability - Does the design quality support possible reuse?

☒ Yes

☐ No (add your remarks)

Remarks:

3.2.1.5 Testability/Maintainability - Does the structure support ease of testing and changes?

☒ Yes

☐ No (add your remarks)

Remarks:

3.2.1.6 Proper use of try/catch and managing of Exceptions

☐ Yes

☒ No (add your remarks)

Remarks: There is no error trapping

3.2.1.7 What is the quality of the javadoc (or javadoc like) documentation of the code?

☒ Not required/possible (explain why)

☐ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but does not exists (add your remarks)

3.2.1.8 Inline comments

☒ Not required/possible (explain why)

☐ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but do not exists (add your remarks)

3.2.2 Code conventions

3.2.2.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

[✓ Yes]

☐ Could be improved (add your remarks)

3.2.2.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

[✓ Perfect]

☐ Could be improved (add your remarks)

Remarks:

3.2.2.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

[✓ Perfect]

☐ Could be improved (add your remarks)

Remarks:

3.2.2.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

[✓ Perfect]

☐ Could be improved (add your remarks)

Remarks:

3.2.2.5 Are variable names human readable?

[✓ Yes]

☐ Not in all necessary cases (add your remarks)

Remarks:

3.2.2.6 Over aspects of coding quality

[✓ Perfect]

☐ Problematic (add your remarks)

Remarks:

3.2.3 Overall code quality

3.2.3.1 Does the dead code exist?

[✓ No]

☐ Yes (add your remarks)

Remarks:

3.2.3.2 Is the code efficient?

[✓ To my best knowledge - yes]

☐ Not always (add your remarks)

Remarks:

3.2.3.3 Are some items, that might be changed in the future hardcoded?

[✓ No]

☐ Some of them (add your remarks)

Remarks:

3.2.3.4 How is the English quality within code and comments?

☐ Very good

[✓ Good]

☐ Acceptable

☐ Poor

Remarks:

3.2.4 Summary

The code is very well written and conforms to the standards used in the CakePHP framework. Proper naming conventions are used and the names of variables, member functions and classes are self explanatory. Classes are JavaDoc style commented and also proper inline comments are used. Some commented lines can be removed.

There are no hard-coded items.

We recommend to use try-catch constructs when calling the Web services and when processing the XML.

Our conclusion is that the quality and efficiency of the code are very good.

3.3 Viewer + Metadata + Upload page (Uhann)

3.3.1 Detailed report (table form)

3.3.2 Correct use of Object Oriented programming

3.3.2.1 Efficiency - Are the constructs efficiently designed?

☒ Yes

☐ No (add your remarks)

Remarks:

3.3.2.2 Complexity – Do the constructs increase the architectural complexity?

☐ Yes (add your remarks)

☒ No

Remarks:

3.3.2.3 Understandability - Does the design increase the psychological complexity?

☐ Yes (add your remarks)

☒ No

Remarks:

3.3.2.4 Reusability - Does the design quality support possible reuse?

☒ Yes

☐ No (add your remarks)

Remarks:

3.3.2.5 Testability/Maintainability - Does the structure support ease of testing and changes?

☒ Yes

☐ No (add your remarks)

Remarks:

3.3.2.6 Proper use of try/catch and managing of Exceptions

☒ Yes

☐ No (add your remarks)

Remarks:

3.3.2.7 What is the quality of the javadoc (or javadoc like) documentation of the code?

☐ Not required/possible (explain why)

☐ Exists in good quality

☐ Could be improved (add your remarks)

☒ Required but does not exists (add your remarks)

3.3.2.8 Inline comments

☐ Not required/possible (explain why)

☐ Exists in good quality

☒ Could be improved (add your remarks)

☐ Required but do not exists (add your remarks)

3.3.3 Code conventions

3.3.3.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

☒ Yes

☐ Could be improved (add your remarks)

3.3.3.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.3.3.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.3.3.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

☐ Perfect

☒ Could be improved (add your remarks)

Remarks:

See in summary

3.3.3.5 Are variable names human readable?

☒ Yes

☐ Not in all necessary cases (add your remarks)

Remarks:

3.3.3.6 Other aspects of coding quality

☒ Perfect

☐ Problematic (add your remarks)

Remarks:

3.3.4 Overall code quality

3.3.4.1 Does the dead code exist?

☐ No

☐ Yes (add your remarks)

Remarks:

3.3.4.2 Is the code efficient?

☒ To my best knowledge - yes

☐ Not always (add your remarks)

Remarks:

3.3.4.3 Are some items, that might be changed in the future hardcoded?

☐ No

☒ Some of them (add your remarks)

Remarks: redirections

3.3.4.4 How is the English quality within code and comments?

☒ Very good

☐ Good

☐ Acceptable

☐ Poor

Remarks:

3.3.5 Summary

The code is based on the cakePHP framework. The framework gives the needed architecture and the keeps code design clean and good understandable. Thanks to cakePHP division into controller and view models, components can (and are actually) be reused by other developers. Class declarations follow the cakePHP standard and are good presented. The code itself is well written and unnecessary complexity is avoided. Typical OOP features like exception handling are used properly to my best knowledge. Variable names are self explaining. I could not find unused parts in the code, however there are some, I guess, temporarily commented lines, which could be removed.

Unfortunately we did not agreed on name conventions in advance. Authors use underscore-divided words as variable names, which is quite unusual. Another weak point is the little number of comments, and the absence of javadoc-like comments, although existing comments are well formatted and have a good English quality in my eyes. Some redirections are hardcoded, however we should first decide how to avoid it in the future at one of our meetings.

My conclusion is that the quality and efficiency of the code are very good. However the comments have to be improved.

3.4 KRService (Uhann)

3.4.1 Correct use of Object Oriented programming

3.4.1.1 Efficiency - Are the constructs efficiently designed?

[✓ Yes]

☐ No (add your remarks)

Remarks:

3.4.1.2 Complexity – Do the constructs increase the architectural complexity?

☐ Yes (add your remarks)

[✓ No]

Remarks:

3.4.1.3 Understandability - Does the design increase the psychological complexity?

☐ Yes (add your remarks)

[✓ No]

Remarks:

3.4.1.4 Reusability - Does the design quality support possible reuse?

[✓ Yes]

☐ No (add your remarks)

Remarks:

3.4.1.5 Testability/Maintainability - Does the structure support ease of testing and changes?

[✓ Yes]

☐ No (add your remarks)

Remarks:

3.4.1.6 Proper use of try/catch and managing of Exceptions

☒ Yes

☐ No (add your remarks)

Remarks:

3.4.1.7 What is the quality of the javadoc (or javadoc like) documentation of the code?

☐ Not required/possible (explain why)

☒ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but does not exists (add your remarks)

3.4.1.8 Inline comments

☐ Not required/possible (explain why)

☒ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but do not exists (add your remarks)

3.4.2 Code conventions

3.4.2.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

☒ Yes

☐ Could be improved (add your remarks)

3.4.2.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.4.2.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.4.2.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

☒ Perfect

☐ Could be improved (add your remarks)

Remarks:

3.4.2.5 Are variable names human readable?

☒ Yes

☐ Not in all necessary cases (add your remarks)

Remarks:

3.4.2.6 Other aspects of coding quality

☒ Perfect

☐ Problematic (add your remarks)

Remarks:

3.4.3 Overall code quality

3.4.3.1 Does the dead code exist?

☒ No

☐ Yes (add your remarks)

Remarks:

3.4.3.2 Is the code efficient?

☒ To my best knowledge - yes

☐ Not always (add your remarks)

Remarks:

3.4.3.3 Are some items, that might be changed in the future hardcoded?

☒ No

☐ Some of them (add your remarks)

Remarks:

3.4.3.4 How is the English quality within code and comments?

☒ Very good

☐ Good

☐ Acceptable

☐ Poor

Remarks:

3.4.4 Summary

The software is well written and to my best knowledge efficient. The whole code is understandable and split into short functions that can be easily maintained and reused. OOP paradigms like exceptions handling are properly used. Variable names are understandable and follow the java name conventions. I could not find dead code, however some out-commented parts that could be deleted. Nevertheless comment quality is very good from a formatting point of view as well as from the English quality.

3.5 Identity management + Grouping + Upload (Altran)

3.5.1 Correct use of Object Oriented programming

3.5.1.1 Efficiency - Are the constructs efficiently designed?

Yes

3.5.1.2 Complexity – Do the constructs increase the architectural complexity?

No

Remarks:

3.5.1.3 Understandability - Does the design increase the psychological complexity?

No

Remarks:

3.5.1.4 Reusability - Does the design quality support possible reuse?

Yes

Remarks: Most items are standard names

3.5.1.5 Testability/Maintainability - Does the structure support ease of testing and changes?

Yes

Remarks: Flow of code its clear

3.5.1.6 Proper use of try/catch and managing of Exceptions

Yes

Remarks:

3.5.1.7 What is the quality of the javadoc (or javadoc like) documentation of the code?

Exists in good quality

3.5.1.8 Inline comments

Could be improved (add your remarks)

Remarks: Maybe if there is a person who doesn’t know the application, it becomes a little bite short of comments

3.5.2 Code conventions

3.5.2.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

Yes

3.5.2.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

Perfect

3.5.2.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

Perfect

3.5.2.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

Perfect

3.5.2.5 Are variable names human readable?

Yes

3.5.2.6 Over aspects of coding quality

Perfect

3.5.3 Overall code quality

3.5.3.1 Does the dead code exist?

Yes (add your remarks)

Remarks: There are comments with alerts in the code, but in my opinion they are useful for programming

3.5.3.2 Is the code efficient?

yes

3.5.3.3 Are some items, that might be changed in the future hardcoded?

No

3.5.3.4 How is the English quality within code and comments?

Good

3.6 Drivers (image, video, audio, generic) (Giunti)

3.6.1 Correct use of Object Oriented programming

3.6.1.1 Efficiency - Are the constructs efficiently designed?

Yes

Remarks:

3.6.1.2 Complexity – Do the constructs increase the architectural complexity?

No

Remarks:

3.6.1.3 Understandability - Does the design increase the psychological complexity?

No

Remarks:

3.6.1.4 Reusability - Does the design quality support possible reuse?

Yes

Remarks:

3.6.1.5 Testability/Maintainability - Does the structure support ease of testing and changes?

Yes

Remarks:

3.6.1.6 Proper use of try/catch and managing of Exceptions

Yes

Remarks:

3.6.1.7 What is the quality of the javadoc (or javadoc like) documentation of the code?

☐ Not required/possible (explain why)

☐ Exists in good quality

☐ Could be improved (add your remarks)

☐ Required but does not exists (add your remarks)

Remarks: Needs to be added

3.6.1.8 Inline comments

Not required/possible (explain why)

Remarks: The code is quite simple.

3.6.2 Code conventions

3.6.2.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

Yes

3.6.2.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

Perfect

Remarks:

3.6.2.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

Perfect

Remarks:

3.6.2.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

Perfect

Remarks:

3.6.2.5 Are variable names human readable?

Yes

Remarks:

3.6.2.6 Over aspects of coding quality

Perfect

Remarks:

3.6.3 Overall code quality

3.6.3.1 Does the dead code exist?

No

Remarks:

3.6.3.2 Is the code efficient?

yes

Remarks:

3.6.3.3 Are some items, that might be changed in the future hardcoded?

No

Remarks:

3.6.3.4 How is the English quality within code and comments?

- ☐ Very good
- ☐ Good
- ☐ Acceptable
- ☐ Poor

Remarks: Not present

3.7 *Search+found+order page (Giunti)*

3.7.1 Correct use of MVC programming

3.7.1.1 Efficiency - Are the constructs efficiently designed?

Yes

Remarks:

Complexity – Do the constructs increase the architectural complexity?

No

Remarks:

3.7.1.2 Understandability - Does the design increase the psychological complexity?

No

Remarks:

3.7.1.3 Reusability - Does the design quality support possible reuse?

No (add your remarks)

Remarks: Some parts of the code could be included in API or control.

3.7.1.4 Testability/Maintainability - Does the structure support ease of testing and changes?

Yes

Remarks:

3.7.1.5 Proper use of try/catch and managing of Exceptions

No (add your remarks)

Remarks: Partially yes. In some place others try catch should be added

3.7.1.6 What is the quality of the javadoc (or javadoc like) documentation of the code?

Required but does not exists (add your remarks)

3.7.1.7 Inline comments

Required but do not exists (add your remarks)

3.7.2 Code conventions

3.7.2.1 Comments formatted correctly?

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc4.html#385

Yes

Remarks:

3.7.2.2 Class/Interface declarations

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc5.html#2991

Perfect

Remarks:

3.7.2.3 Statements quality

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc6.html#430

Perfect

Remarks:

3.7.2.4 Naming Conventions

Hints: http://java.sun.com/docs/codeconv/html/CodeConventions.doc8.html#367

Could be improved (add your remarks)

Remarks: more java style then php stype, some name in spanish

3.7.2.5 Are variable names human readable?

Yes

Remarks:

3.7.2.6 Over aspects of coding quality

Perfect

Remarks:

3.7.3 Overall code quality

3.7.3.1 Does the dead code exist?

No

Remarks:

3.7.3.2 Is the code efficient?

Not always (add your remarks)

Remarks:

3.7.3.3 Are some items, that might be changed in the future hardcoded?

Some of them (add your remarks)

Remarks: some xml namespace and url

3.7.3.4 How is the English quality within code and comments?

Good

Remarks:

4 Impact analysis (GiuntiLabs)

The impact of LearnWeb2.0 in the panorama of life-long learning is articulated: first of all LearnWeb2.0 may conflict with "smart" engines, like Google and Wikipedia, that today are widely used for learning purposes as well.

One of the base requirements of LearnWeb2.0 is the crowding of tags/comments/rates, that constitute the real added value with respect to search engines.

A possible side effect of using LearnWeb2.0 is the possibility to reach the learning objectives without using any LMS, but simply browsing resources and navigating among tags/comments.

Another important impact is the possibility to automatize the knowledge management, by means of the Web Services exposed by KRService. This allows the development of “bright” tools benefitting of LearnWeb2.0 social resources.

From a technical point of view, a particular impact is due to the web architecture: no more installations needed at client side will enable a wide range of users. This does not prevent the possibility to install different servers, for obtaining separate resources networks.

A particular impact (being addressed in the future) is the confidentiality issue. The most important aspect is the fact that the resources content is stored on the web (YouTube, Flickr,...) and this may conflict with privacy and/or IPR (Intellectual Properties Rights). A powerful authorization mechanism should be set up to face this problem.

Another impact is the modificability of resources. The modification of a resource should not be allowed, because it can be referred inside Learning Objects and related to other resources to form a lesson. This may have an impact over the intuitive belief that the owner can modify his stuff.

5 Improvements/enhancements for next release (SU)

The evaluation results of LearnWeb2.0 tool and the recommendations from the validation and from the code reviews will be carefully analysed:

- small bugs and errors have been already corrected;
- recommendations that require more coding, but are critical for the pilots, should be implemented and tested before the start of the pilots;
- others recommendations, that require significant change in models, Web services and/or LearnWeb2.0 Web tool will be considered as improvements/enhancement for the next release.

Here is a list of suggestions for improvement/enhancements for the next release besides the recommendations from the evaluation and code reviews.

Modifications of existing and development of new web services

The existing web services will be analysed and some of them modified in order:

- to implement some required functionality;
- to improve the efficiency of the web services and/or LearnWeb2.0 web tool.

Also some new web services will be developed based on recommendations during the evaluation. Such services are for example for advanced search, deletion of resources, search for user, etc.

Authentication and authorization

The implemented authentication will be reviewed whether it complies with the accepted authentication approach in TENCompetence. User authorization will be implemented together with WP3. This can solve many problems with the user access to resources and privacy issues.

Deletion and modification of resources

Deleting and modifying resources, comments, categories, etc., is a complex problem and that's why this is left for the next release. Here are some examples.

If an author is allowed to delete a resource, this will lead also to the deletion of the associated comments, ratings and tags made by other people. And in certain cases the comments can be more valuable for the community than the resource itself. If the resource has no comments, rating and tags the delete operation is OK.

Modifying the content of a resource also can cause problems - the existing ratings, comments and tags are for the old content, not for the new one. We have implemented and tested Upload Modified Version of a resource, but we must be aware of this problem.

If the user is allowed to delete or modify his own comment this can cause that the next comments to make no sense if they are comments or answers to the deleted/modified comment.

The same is for categories. Categories should be created by administrators only, but the current version still has no authorization. Also the deletion and renaming of a category can not be allowed if there are resources within this category.

Different scenarios will be developed for the next release in order to decide what and when the user is allowed to delete or modify.

Multi language support

The LearnWeb2.0 Web tool will be redesigned and translated to support multiple languages (English, Italian, Spanish, Bulgarian, etc.).

Load tests and improving the performance

Load tests will be designed and performed in order to test the performance of the LearnWeb2.0 server in a concurrent situation. The analysis of the results will help the improvement of the efficiency of the web services and the web tool. Also a caching mechanism will be implemented to improve the overall performance of the server.

6 Conclusion (GiuntiLabs)

The evaluation of LearnWeb2.0 has been a good opportunity to discover several imperfection of the system, both from the designing point of view and from the implementation one.

The system is now in a youth stage: several bugs have been already fixed and the list of improvements/enhancements is the starting point for future release, to be designed in autumn 2008 and developed in winter 2009.

7 Appendix – Real users' evaluation tests (UPF)

(1)	<u>**Evaluation as a tool for stimulating knowledge sharing**</u> **¿Le ha parecido una herramienta útil para aprender de los archivos y los comentarios que han añadido sus compañeros?** Muy útil [ ] Útil [ ] Normal [ ] Poco útil [ ] Muy poco útil [ ] ***Do you think that LearnWeb 2.0 is a useful tool for learning from the comments of your partners?*** ***Very useful [ ]*** ***Useful [ ]*** ***Normal [ ]*** ***Not very useful [ ]*** ***Not useful [ ]***
(2)	<u>**Evaluation the tool as a support for educative contexts**</u> **¿Cree que las herramientas escogidas para compartir fotografías, videos, documentos, etc… con sus compañeros, son los más adecuadas?** **Si [ ]** **No[ ]** *Escriba aquí los programas que le gustaría usar para la gestión de los archivos (ej. Blog, Wikipedia ...)* **………**

	Do you think that the web 2.0 services that the tool integrates are the well-selected? ***Yes [ ]*** ***No [ ]*** ***Write down the services that you would like to include in the tool*** **..**
(3)	<u>**Evaluating the tool as a repository**</u> **¿Cómo valora esta herramienta como espacio para almacenar recursos que se hayan utilizado durante el curso?** Muy útil [ ] Útil [ ] Normal [ ] Poco útil [ ] Muy poco útil [ ] ***How do you evaluate the tool as an space for store the resources related to the course?*** ***Very useful [ ]*** ***Useful [ ]*** ***Normal [ ]*** ***Not very useful [ ]*** ***Not useful [ ]***

Si quiere añadir alguna sugerencia y/o tiene algún comentario, por favor indíquelo a continuación:

Muchas gracias por su colaboración.