

iCamp

innovative, inclusive, interactive & intercultural
learning campus

Final portfolio of iCamp Building Blocks

Deliverable Number:	D2.5
Contractual Date of Delivery:	November 2008
Actual Date of Delivery:	January 2009
Work Package(s) Contributing:	WP2
Nature of the Deliverable:	Report / Prototype
Status:	Final
Security (Distribution Level):	Public
Editor(s):	Terje Våljataga (TLU)
Project start: 1. Oct. 2005	Duration: 39 months STREP / IST

The iCamp Consortium consists of:

Centre for Social Innovation (CSI)	Coordinator	Austria
Jožef Stefan Institute (JSI)	Contractor	Slovenia
University of Leicester (ULE)	Contractor	United Kingdom
Universidad Politécnica de Madrid (UPM)	Contractor	Spain
Vienna University of Economics and Business Administration (VUE)	Contractor	Austria
University of Science and Technology (AGH)	Contractor	Poland
Kaunas University of Technology (KTU)	Contractor	Lithuania
Işik University (ISIK)	Contractor	Turkey
Tallinn University (TLU)	Contractor	Estonia
Tomas Bata University in Zlín (TBU)	Contractor	Czech Republic
Siemens AG (SIE)	Contractor	Germany

Amendment History

Version	Date	Editor(s)	Modification
V0.1	13.11.2008	Terje Väljataga	First version of the deliverable
V0.2	04.12.2008	Terje Väljataga	Inclusion of the comments from peer reviewers
V1.0	17.12.2008	Terje Väljataga	Final version

Contributors

Name	Institution
Terje Väljataga	Tallinn University (TLU)
Evaldas Karazinas	Kaunas University of Technology (KTU)
Fridolin Wild	Vienna University of Economics (VUE)
Stefan Sobernig	Vienna University of Economics (VUE)
Felix Mödritscher	Vienna University of Economics (VUE)
Robert Koblischke	Vienna University of Economics (VUE)
Pjotr Savitski	Tallinn University (TLU)
Vahur Rebas	Tallinn University (TLU)
Karsten Ehms	Siemens AG
Mart Laanpere	Tallinn University (TLU)

List of Acronyms

Acronym	Description
TEL	Technology Enhanced Learning
VoIP	Voice over Internet Protocol
SIP	Session Initiation Protocol
LGPL	Lesser General Public License
API	Application Programming Interface
SQL	Simple Query Interface
HTML	Hypertext Markup Language
RSS	Really Simple Syndication

Table of Contents

Executive summary	6
1. iCamp Space	7
2. iFolio as a container of iCamp building blocks	7
2.1. Description of iFolio	8
2.2. Using iFolio	10
2.3. Technical specifications of iFolio	12
2.3.1. General Specification.....	12
2.3.2. Technical Specification	14
2.3.3. iFolio Content-type.....	14
2.3.4. Tool Content-type	15
2.3.5. User Content-type.....	16
2.3.6. Affordance Content-type	16
2.3.7. AJAX application side	16
3. iCamp Building Blocks	18
3.1. Tools and services chosen for the third iCamp field trial	19
3.1.1. iLogue, a web tool to manage personal learning contracts	19
3.1.2. ObjectSpot, the academic federated search engine	21
3.1.3. Videowiki.....	22
3.2. Additional tools and services	24
3.2.1. XoWiki, a multi-purpose Wiki system	24
3.2.2. Mash-UP Personal Learning Environments (MUPPLE)	26
3.2.3. myDENTITY, an email-based social network analysis tool.....	29
4. Conclusions	31
A. Appendix 1. Complete list of iCamp building blocks	33
B. Appendix 2. Handbook	34
C. Appendix 3. Installation instructions of iFolio	35

Executive summary

The main goal of the Workpackage 2 was to develop a portfolio of iCamp building blocks and integrate them into the iCamp Space – an augmented virtual environment, which brings together people with their resources, activities and heterogeneous tools and services. iCamp building blocks embrace a set of TEL services and tools. Some of the iCamp building blocks are the existing social software applications with some modifications and adaptations done for iCamp purposes while other building blocks are developed from scratch for supporting iCamp pedagogical challenges: self-directing learning and change projects, social networking and cross-cultural collaboration.

This deliverable focuses on the last version of the iCamp building blocks that were implemented and validated by the students and facilitators in the third iCamp field trial as a reference set of building blocks. In addition, Deliverable 2.5 takes a closer look at the experimental development of tools and services.

Deliverable 2.5 focuses on the following iCamp building blocks that were developed further and used in the third iCamp field trial: iLogue as a personal learning logbook, Objectspot as an academic federated search engine and Videowiki as a videoblog for recording videoclips. xoWiki as a multi purpose Wiki environment; Mupple for empowering students to design their personal learning environment; myDENTITY for providing students an overview of their communication partners were the experimental and work in progress developments that are described in this deliverable.

All the iCamp building blocks are gathered into the web-based system called iFolio. iFolio is a help system with the purpose to support the selection of iCamp building blocks as well as other open source tools and services according to the user's needs.

This deliverable starts with revisiting the iCamp Space and iFolio system by giving an updated description of the iFolio system and its technical specification. The second part of this deliverable provides descriptions of the last set of iCamp building blocks. Pedagogical scenarios of the final set of iCamp building blocks as well as the landscape of these building blocks formed in three iCamp trials are presented in the separate document, the iCamp Handbook. The third section gives an overview of the experimental, work in progress developments of the tools and services. Fourth section concludes the work done in Workpackage 2.

1. iCamp Space

iCamp follows the vision of bringing geographically and temporarily „distributed“ people and resources together in an augmented virtual environment - the iCamp Space. The “Space” metaphor refers to the heterogeneous and distributed environment, where people and resources are connected through technology. It incorporates multiple interoperable systems and tools, various resources and human capital combined with a set of activities.

Specific educational goals, such as the advancement of self-directed learning competences, determine these activities, which can be applied for mediated collaborative cross-national teaching and learning initiatives in combination with the provided tools portfolio. The iCamp Space provides a network of web-based activities related to technologically enhanced learning experiences. An activity is understood as a pattern of: persons with objectives, rules and regulations, tools and artefacts, and events they are involved in.

The iCamp Space does not consist of a single software system, but will be rather a web-based activity space for technology-enhanced teaching and learning projects composing of various interoperable tools and platforms. Users have seamless access through their own learning platforms or tools to services and artefacts offered by remote systems enabling collaboration on a group or individual level. iCamp Space is not static in terms of structure, architecture or centralised platform, rather it is in constant change according to the activities of people who act in the Space and their need to extend or replace it with new tools and services, resources and artefacts.

A selection of tools and services in the iCamp Space provided by iCamp project as well as many other open source tools and services are collected to the system called iFolio - a system for supporting individual or group selection of heterogeneous tools and services for learning as well as teaching purposes.

2. iFolio as a container of iCamp building blocks

In this deliverable a second prototype of iFolio as a container of iCamp building blocks is described. Theoretical framework of the iFolio based on the advanced Activity Theory and the concept of affordances as well as the main principles of using it can be found in Deliverable 2.2.

This chapter provides the reader with the short updated description of the iFolio system, user manual and technical specification of the iFolio second prototype.

2.1. Description of iFolio

iFolio (URL: <http://www.htk.tlu.ee/iFolio>) is developed for serving users who are challenged to find and choose the right tools and systems for their specific learning purposes. iFolio is a support system for students and facilitators enabling to handle the growing variety of TEL systems and tools with the purposeful and effective ways.

iFolio provides to an user two different views:

- tools' section with the detailed description of every tool from pedagogical and technical point of view (see Figure 1);

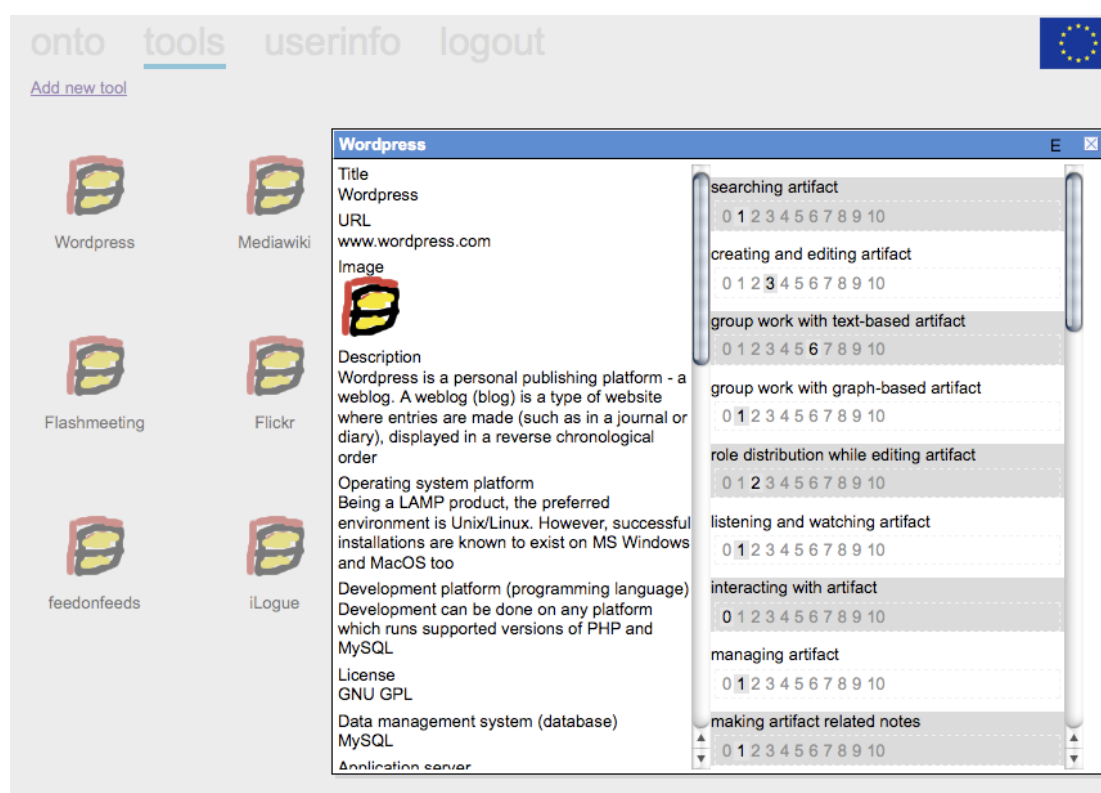


Figure 1: iFolio tools' section

- ontology section, where a user can select and save his/her perspectives to tools and systems in iFolio system; and look at either the tools view or users view.

The ontology section can be used as an explorative decision support tool. The selector tool thus enables a user to investigate different TEL systems and tools, which have been added to the portfolio beforehand according to different pre-selected perspectives. Perspectives in iFolio are presented as a list of possible activities with pre-determined numerical values. The values reflect the TEL systems and tools' capabilities to support these activities.

The ontology section comprises two views: *tools view* (see Figure 2), where a user can see how the tools and systems are presented concerning their values of activities; *users view* gives an overview of other users' perspectives (see Figure 3).

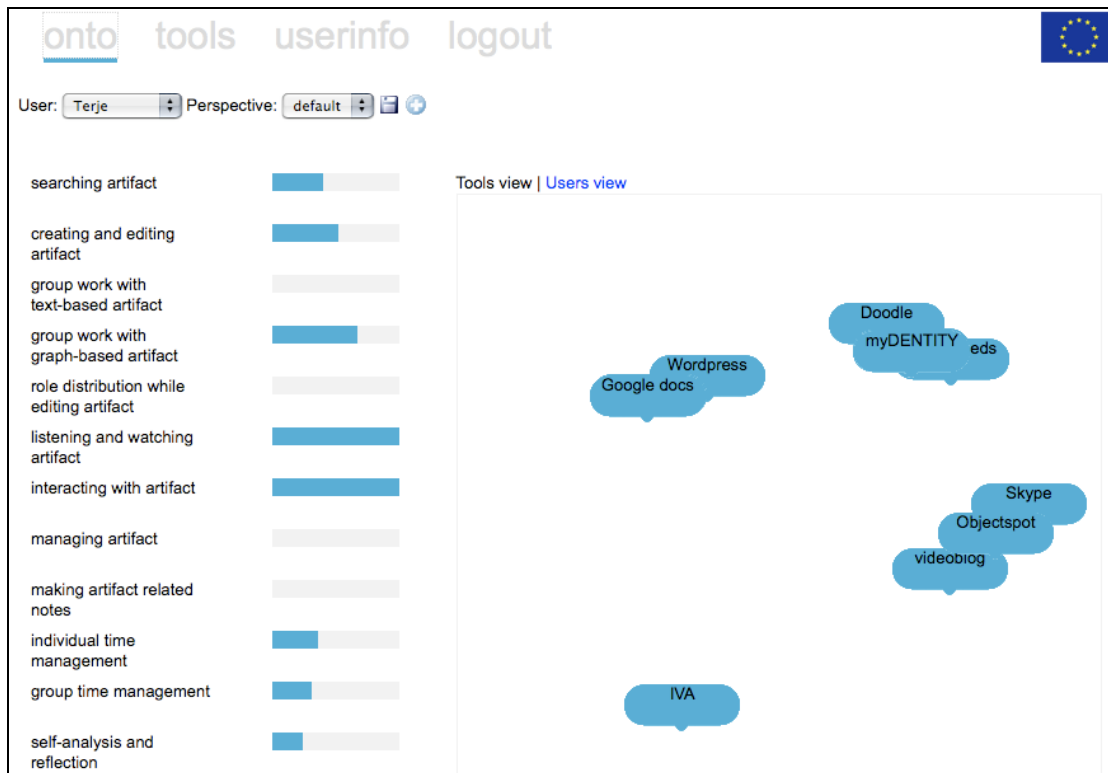


Figure 2: iFolio ontological section, tools view

The tools view comprises a list of names of TEL systems and tools covered by the iFolio. For each of them, activities are stored with the respective numerical value, which represents the average value given by users.



Figure 3: iFolio ontological section, users view

The applications of iFolio are twofold: it can be used as the basis for instructional design by facilitators and students in different e-learning settings and it could support the software designers with the knowledge for developing the learning software towards the formation of the flexible selectable interoperable sets of tools and systems for different learning challenges.

2.2. Using iFolio

The following section gives a short overview of how to use iFolio system for comparison and evaluation of TEL systems and tools. As an authenticated user one has the option to edit and add TEL systems and tools as well as activities to the iFolio.

On entering iFolio one can choose between the Ontology section or the Tools section.

The Ontology section (see Figure 2, 3) presents the learning systems and tools in the iFolio with their names and the list of activities. From the list of users, a user can choose his/her name and start playing with the iFolio system. Ontology section consists of two views: tools and users view.

On *tools view*, left-side provides an initial list of activities with sliders. Names of TEL systems and tools are arranged randomly on the right part of the screen. Sliders enable the user to manually change the weight of each activity by moving the slider from left (minimum) to the right (maximum), resulting in changed positions of the learning systems and tools on the right part of the screen (see Figure 2 as an example).



Figure 4: iFolio ontological section, average values of the activities with respect to Wordpress

Moving the cursor over one tool on the right part of the screen results in a change of colour of the activities displayed on the left part of the screen.

Red colour on the sliders shows the actual average value of the particular activity according to the chosen tool (see Figure 4). By clicking on the one of the names of the tools on the right part of the screen, a box with the technical and pedagogical description appears.

Users view gives an overview of the registered users and their saved perspectives. Moving the cursor over one user on the right part of the screen results in a change of colour of the activities displayed on the left part of the screen. Red colour on the sliders shows the perspectives of the particular user and the blue colour shows the perspectives of the current user (see Figure 5).

In tools' section users can look at the technical description and values of activities of every TEL tools and systems added to this iFolio by clicking on the tool's name.

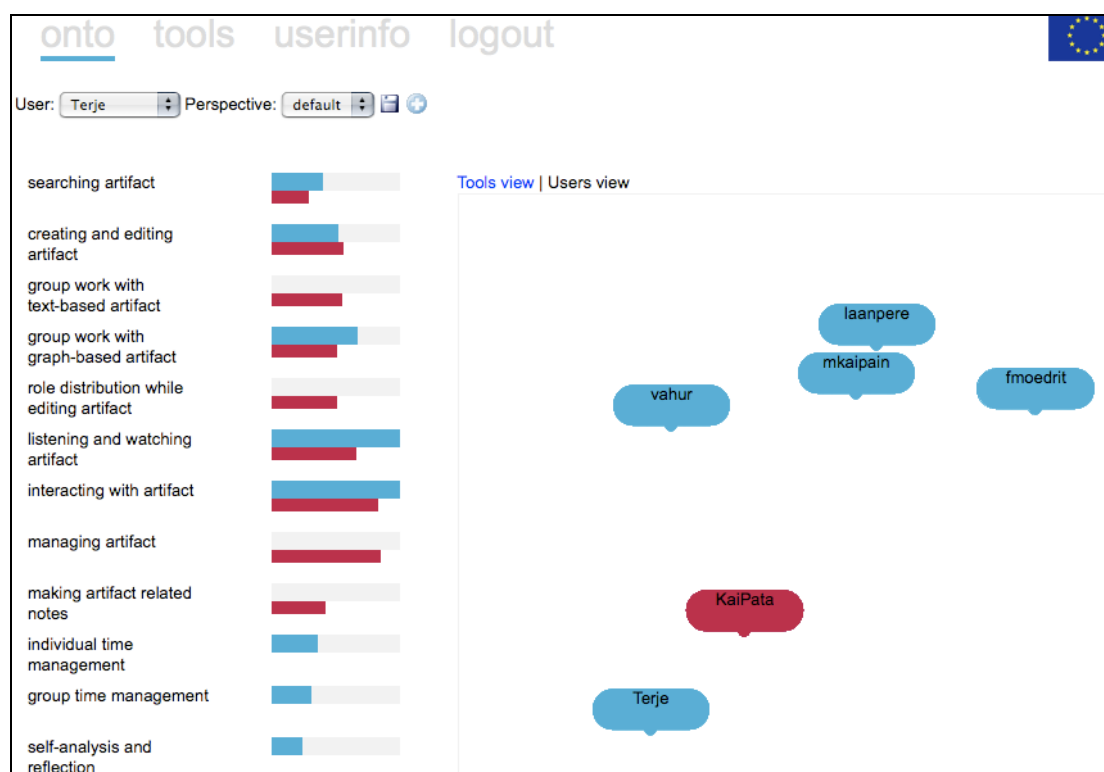


Figure 5: iFolio ontological section, perspectives of the user KaiPata

iFolio has been tested in the context of post-graduate course with a group of master students in Tallinn University. The purpose was to get some data and feedback from the students in order to analyze the applicability and usability of the iFolio. The main idea was to find out if the product meets users' needs e.g. if it facilitates the user to find study partners based on the similar selection of activities as well as if it facilitates the comparison and evaluation of various TEL services and tools for selecting the most appropriate to support one's activities.

The students carried out the following tasks with the iFolio: First the students were asked to think of their first homework, choose activities in iFolio and evaluate to what extent they would need these activities to carry out their first homework.

Secondly, the students were asked to save their perspectives and choose some of the tools and services that most support the chosen activities and that they want to work with. Thirdly, the students were asked to find other students who have similar preferences in terms of chosen activities and their values.

Users' general feedback and experiences with the iFolio system was positive. The user interface of the iFolio system was rather intuitive for the students and they understood the structure of the system. However some drawbacks occurred. First, in onto section the location of the tools and services on the right part of the screen confused the students as it was every time different when they logged in. Secondly, in order to give values to the activities it was not that clear for the students how to move sliders. Thirdly, it was not very clear how to interpret the location of the tools and services on the right part of the screen as soon as the activities were chosen and values to them were given. The location of the tools and services on the right part of the screen was confusing as it did not indicate directly which tools and services actually support the chosen activities and which not. But as soon as they mastered the techniques the use of the system went smoothly and the students enjoyed it a lot. With respect to find study partners for certain task the iFolio system seems to have a great potential. The students claimed to find study partners with similar learning habits. However as iFolio system is a work in progress, further development and user testing of the system is needed.

2.3. Technical specifications of iFolio

iFolio is an open-source standalone Zope product. iFolio is a standard Zope product that can successfully co-exist with some other products inside same Zope instance. iFolio is using several standard Zope packages for user authentication and rights control purposes. iFolio consists of two main components: Zope product written in Python and AJAX application side, integrated into Zope page-template mechanisms. Templates and AJAX application side together form a graphical User Interface of iFolio. The AJAX applications side is built using The Yahoo! User Interface Library (YUI). For more information see <http://developer.yahoo.com/yui/>.

2.3.1. General Specification

iFolio object is a folderish content-type that is able to store several additional types of objects (shipped with iFolio product; in Figure 6 these are called content-type classes). One of the main object types is called Tool (e-learning systems and tools that are used as building blocks for constructing various distributed e-learning landscapes). New Tools can be added into the iFolio by any user of iFolio instance. Additional object type is called Affordance. Affordances are used for evaluating the Tools by registered users. Affordances can only be added by members with manager or administrator privileges, managers will add affordances from Zope Management Interface (not from iFolio graphical interface). One more content-type is called User, this one stores information about user perspectives and tools evaluations (values a user can save for evaluating the importance of affordances or evaluating tools). In addition to these several folderish objects are defined: AffordanceManager, ToolsManager and UsersManager. These are used as storages for main content-types and have several helper methods that provide needed data for other components.

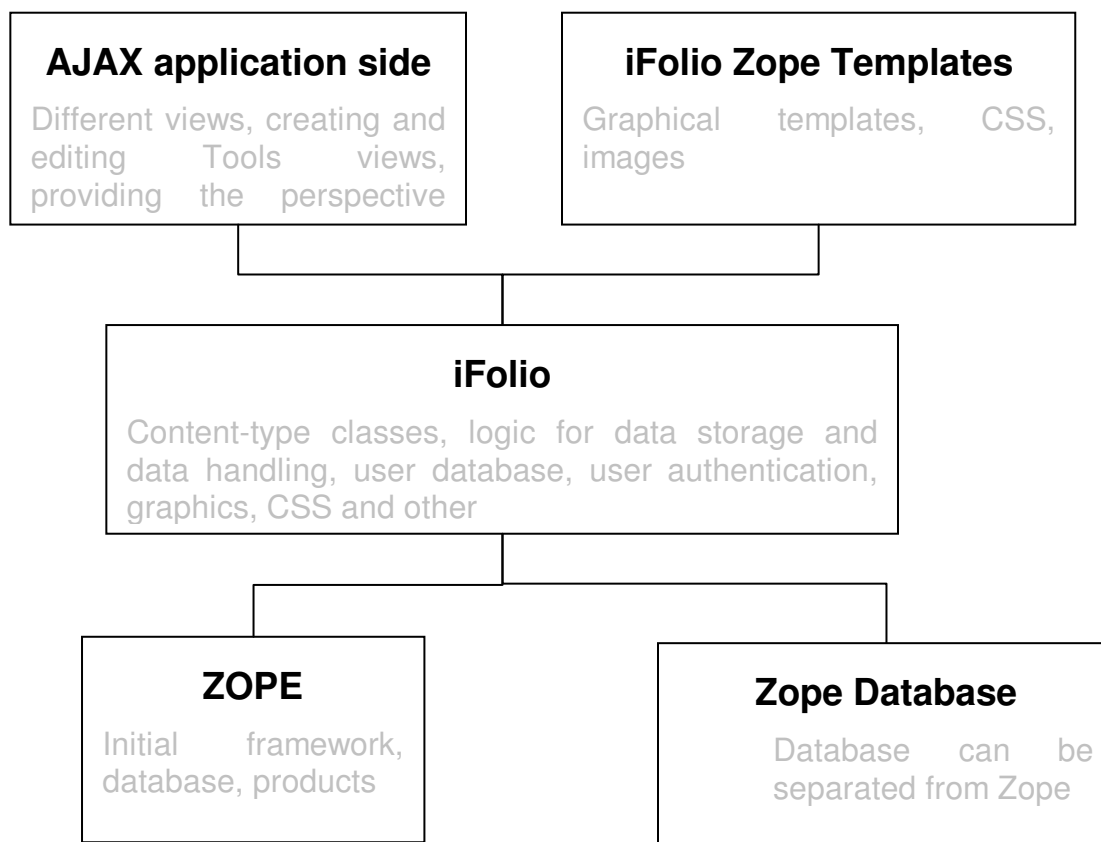


Figure 6: Schematic presentation of iFolio product integration

Every Tool object has two types of parameters:

- static parameters (technical characteristics) that are used to store valuable information about different aspects of the Tool in a text form (these can be specified upon creation of the Tool, later on Tools can be edited and these parameters changed);
- dynamic parameters (pedagogical affordances) that can be given a numerical values between 0 and 10 (provided by registered users for every standalone Tool).

Every User object has two additional types of storages in addition to user data:

- personal data stores numeric values for user perspective;
- tool data parameters that user specifies when evaluating tools.

User Interface is divided onto several sections:

- onto section is the place a user can see other users perspectives (by selecting their usernames from the drop-down menu), evaluate importance of affordances and save own perspective, user can switch between two views Tools view and Users view;
- tools section is the place users can view all existing tools and edit them, they can also create some new tools here;

- register, login and logout sections can be used for registering a user account in iFolio, logging in with username and password and logging out of the system (closing the browser window should also be enough to logout).

The system is using a numerical values of the dynamic parameters in order to build an Ontological views (as many users can specify their values for the parameters – the system is using an average values calculated from these for every parameter), where each Tool is represented as a dot on the square-shape Screen. The location of each Tool is calculated with Multi-Dimensional Scaling (MDS) algorithm, so that two Tools with similar sets of affordance values are displayed next to each other while the most different pair of tools will be moved as far from each other as possible. In addition to Tools, Users will also be shown in the Users view. In this case valued from their perspectives will be used.

2.3.2. Technical Specification

All the data is situated in Zope database (as shown in Figure 6) that is usually integrated into a Zope instance, but it can be separated from it and can run in a separate mode.

Four main content-types or object types are:

- iFolio
- Tool
- User
- Affordance

Each of these content-types is a class based upon some super class provided by the system. Super classes have a lot of helper methods and parameters defined; these are common for the whole system. In addition several specific methods are defined for every content-type. Some of them are used by AJAX application side to make communication with the storages.

iFolio product is using Zope user permission framework and PluggableAuthService for user authentication purposes.

2.3.3. iFolio Content-type

As mentioned earlier iFolio is a folderish type of object inside which all additional types shipped with the product can be added. It has three subfolders: affordances, tools and users. All the other objects are added into these according to their type.

In addition, it also stores a PluggableAuthService instance. This handles user registration and authentication. All new registered users will be sent an e-mail with the password generated by the system.

All the functions that get and set parameters for the user are specified on the Zope (Python programming language) side and are just called out by the AJAX application side or Zope objects themselves. All calculations that determine position of components on the screens are done on the Zope/Python side of the iFolio product. The results of these calculations are provided for the AJAX application side mechanisms, which deal with User Interface side.

All methods used by the views are also specified here. These methods are combining together all the values of Tools and Users that are needed by the AJAX application side to work and making a dictionary with predefined structure out of them (Tools view is only using Tools as the accepted types; Users view is only using perspective data that has been specified or saved by users). The values for User perspectives and Tools are stored based upon the system from 0 to 10.

`getUserData` method returns a string that has a form of dictionary, which is used by AJAX application side for building Users view. There is one predefined key in this dictionary: `userid`s. Its value is an array of IDs for all registered users. All the other keys are User IDs, a value for each of them is a dictionary. This dictionary has two predefined keys: `x_title` and `x_url`. These will provide a name of the user and possible homepage. All the other keys are Affordance IDs, which have values from the saved perspective.

Example: `{"userid": ["pjotr", "mari"], "pjotr": {"x_title": "pjotr", "x_url": ""}, "affordanceID": value, "affordanceID": value,...},...`

`getToolsData` method returns a string that has a form of a dictionary, which is used by AJAX application side for building Tools view. There is one predefined key in this dictionary: `toolid`s. Its value is an array of IDs for all created tools. All the other keys are Tool IDs, a value for each one of them is a dictionary. This dictionary has two predefined keys: `x_title` and `x_url`. These will provide the name of the tool and possible homepage. All the other keys are Affordance IDs, which value is an average from the values of all the Users that have evaluated the Tool.

Example: `{"toolid": ["plone", "zope"], "plone": {"x_title": "Plone", "x_url": "http://plone.org"}, "affordanceID": value, "affordanceID": value,...},...`

2.3.4. Tool Content-type

Tool is a content-type with predefined static parameters containing information about technical characteristics and some image file for the tool. In addition to that it also has sub-storage for evaluation of the Tool by the usage of Affordances values that are that values provided upon creation.

This storage is a dictionary where every key is Affordance ID, the value for the element is the initial value specified upon creation of the tool.

Example: `{'affordanceID': 'value', 'affordanceID': 'value', ...}`

Every user can only specify one set of value for the Affordances of the Tool. This can be done by editing the Tool and selecting values from drop-down lists. If the user has already saved some values, then he will be given these. This way user can only specify one set of values and later on these values can be changed by just selecting some new values and saving the Tool. Evaluation data for all the tools is stored within the User object in iFolio (the structure of it will be covered in the next chapter).

If some new Affordances have been added that have not existed during the initial evaluation, then they will just be given a value of 0 by the getter-method. All the affordances deleted from the systems will just be omitted by the get methods.

In addition there are several methods that will construct HTML, which is used by the AJAX side of the application for constructing view and edit forms for tools.

2.3.5. User Content-type

User is a content-type which holds most of the user-specific information. In addition to that it also has sub-storages for User Perspective and Tools evaluations data.

User Perspective storage is a dictionary where every key is some Perspective ID, its value is another dictionary that has Affordance ID as a key and a value, which user saved for that Affordance. At the moment the system will only allow user to save the default Perspective, but the dictionary itself is ready to handle cases of multiple Perspectives.

Example: {'default':{'affordanceID':'value','affordanceID':'value',...},...}

Tool evaluation storage is a dictionary where every key is Tool ID, its value is another dictionary that has Affordance ID as a key and a value, which user has saved when evaluating the Tool.

Example: {'toolID':{'affordanceID':'value','affordanceID':'value',...},...}

All methods for setting and setting Perspective and Tool data are also defined here.

2.3.6. Affordance Content-type

Affordance is a simple content-type that only has a description and title fields. Affordance is used by the system to make evaluation of the Tools available to users.

2.3.7. AJAX application side

AJAX application side together with Zope page-templates forms a graphical User Interface of iFolio. The AJAX applications side is built using The Yahoo! User Interface Library (YUI). For more information see <http://developer.yahoo.com/yui/>.

Several types of events are used for handling data, performing actions and forming application UI:

loadPerspectiveData – requests perspective data from the Zope side of iFolio application provides username to get the required data for.

loadToolData – requests data for all the existing tools from the Zope side of iFolio application.

showToolDataEvent – when user enters any tool on ontology view with mouse, this event will show the values for affordances situated under user perspective data on the left of the screen.

hideToolDataEvent – when mouse leaves any selected tool on ontology view, this event will hide values for affordances situated under user perspective data on the left of the screen.

calculateEvent – upon changes in perspective data, viewing perspective of other user or entering the ontology view, this event will initialize calculator functionality situated on Zope application side (making calculations for positioning of the objects within ontology view area).

savePerspectiveEvent – sends values of affordances evaluation of the user to the Zope application side and initializes saving of that data into corresponding user object in the database.

initFinished – when user or tool data is loaded this event gets initialize, this event is called to initialize the calculateEvent if the is needed. This will also set a delay of

one second, so that calculations can be finished before applications will start positioning.

workingEvent – will show user a notification in case application is processing any requests of saving or retrieving the data (also calculating positioning of the object for ontology view).

workFinishedEvent – will hide notification shown by the workingEvent as processing of some request has been finished.

loadUserData – requests data for all the users from Zope side of iFolio application.

clearOntoData – removes all the elements situated inside ontology view positioning area (initialized when users wants to switch between viewing users or tools).

flashEvent – shows user notification if some functionality is not available yet, creating flashing animation.

All requests from iFolio User Interface are triggered and processed by the AJAX side, only then information is sent to the Zope side of the application. It is also handling any notifications shown to the user.

Tabbed browsing is also created by AJAX, long with determining which tabs should be shown to what type of user (for example: register and login tabs will be shown to Anonymous user, but will be hidden once login operation is performed).

Calculation of the element positioning or the ontology view has been moved to the Zope side and written with Python, because it turned out to be working quite slow in if done within the AJAX side itself. The calculation on the Zope side is working several times faster.

3. iCamp Building Blocks

One of the main tasks for iCamp WP2 was to select and test the most appropriate open-source services and tools, but also to adapt and customize some of these tools and services according to the iCamp pedagogical challenges and the nature of three iCamp field trials. The final list of the iCamp building blocks that were used in three iCamp field trials was the following:

Tools and services chosen to support the iCamp pedagogical challenges in the first iCamp field trial:

Flickr for photo sharing
Delicious for social bookmarking
Skype for (a)synchronous communication
Wordpress for personal logbook
Flashmeeting for synchronous communication

Tools and services chosen and developed to support the iCamp pedagogical challenges in the second iCamp field trial:

iLogue for personal learning logbook
xoWiki for personal learning contract
ObjectSpot for searching learning materials
Feedonfeeds for merging information from different sources to one place
Videowiki for recording videoclips
Scuttle for social bookmarking
myDENTITY for providing students an overview of their communication partners
Doodle for scheduling meetings and carrying out votings
iCamp SIP server for communication
Wordpress for personal logbook
iCamp help center, tutorials of the tools' use

Tools and services chosen and developed to support the iCamp pedagogical challenges in the third iCamp field trial:

iLogue for personal learning contract
Objectspot for searching learning materials
FeedonFeeds for merging information from different sources to one place
Videowiki for recording videoclips
Scuttle for social bookmarking
Doodle for scheduling meetings and carrying out voting
Wordpress for personal or group logbook
iCamp help center, tutorials of the tools' use

Overview of the particular tool (non)use by the students and facilitators in three iCamp field trials can be found in Deliverable 4.2 (first trial), Deliverable 4.3 (second trial), Deliverable 4.4 (third trial).

This deliverable presents the last customized set of the iCamp building blocks (iLogue, ObjectSpot, Videowiki) as well as experimental tools' developments that are currently work in progress (xoWiki, Mupple, myDENTITY). The tools and services that are not presented in this Deliverable, but were used in the third iCamp field trial are described in Deliverable 2.3, as no further development or customisation has been done since publishing Deliverable 2.3.

While Deliverable 2.5 concentrates more on the general description of the iCamp building blocks, iCamp Handbook (available on the iCamp project website: <http://icamp.eu>) as a part of this deliverable gives an overview of the tools' landscape in the context of the iCamp three field trials and possible pedagogical scenarios of the most useful building blocks.

3.1. Tools and services chosen for the third iCamp field trial

Following sections present the iCamp building blocks that were chosen and developed further for the third iCamp field trial and are therefore validated by the students in an authentic learning context.

3.1.1. iLogue, a web tool to manage personal learning contracts

Learning contracts allow a learning scenario, in which learners can assemble, compile and aggregate learning actions, resources, and thoughts on a more operational level and specify what they want and how to learn and how to evaluate learning. Personal learning contracts (PLCs), as introduced and discussed by Harri-Augstein and Webb (1995), are instruments to allow for somehow structured conversations or rather (inner) dialogues of people actively participating in formal learning processes about their activities and artifacts as result of these formal learning processes. Normally, a weblog- and wiki-style authoring environment is utilized for PLC purposes.

General description and educational purpose

iLogue, a prototypical implementation of a PLC, aims at supporting personalized learning experiences for self-directed and self-organized learners. Thus, it fosters an active engagement in the learning process from start to finish, enabling shared responsibility of the planning and learning experiences. It can help learners to build on past experience and determine needs as they carry out learning activities. Learning contracts can be also used to provide a means for negotiating reconciliation between the need and expectations of organizations or professions and the learner's internal need and interests. On the other hand learning contracts allow making the learning objectives clear and explicit for both learners and facilitators. By participating in the process of analyzing personal needs, setting up objectives, identifying resources, choosing strategies, and evaluating accomplishments the learner develops a sense of ownership of the plan.

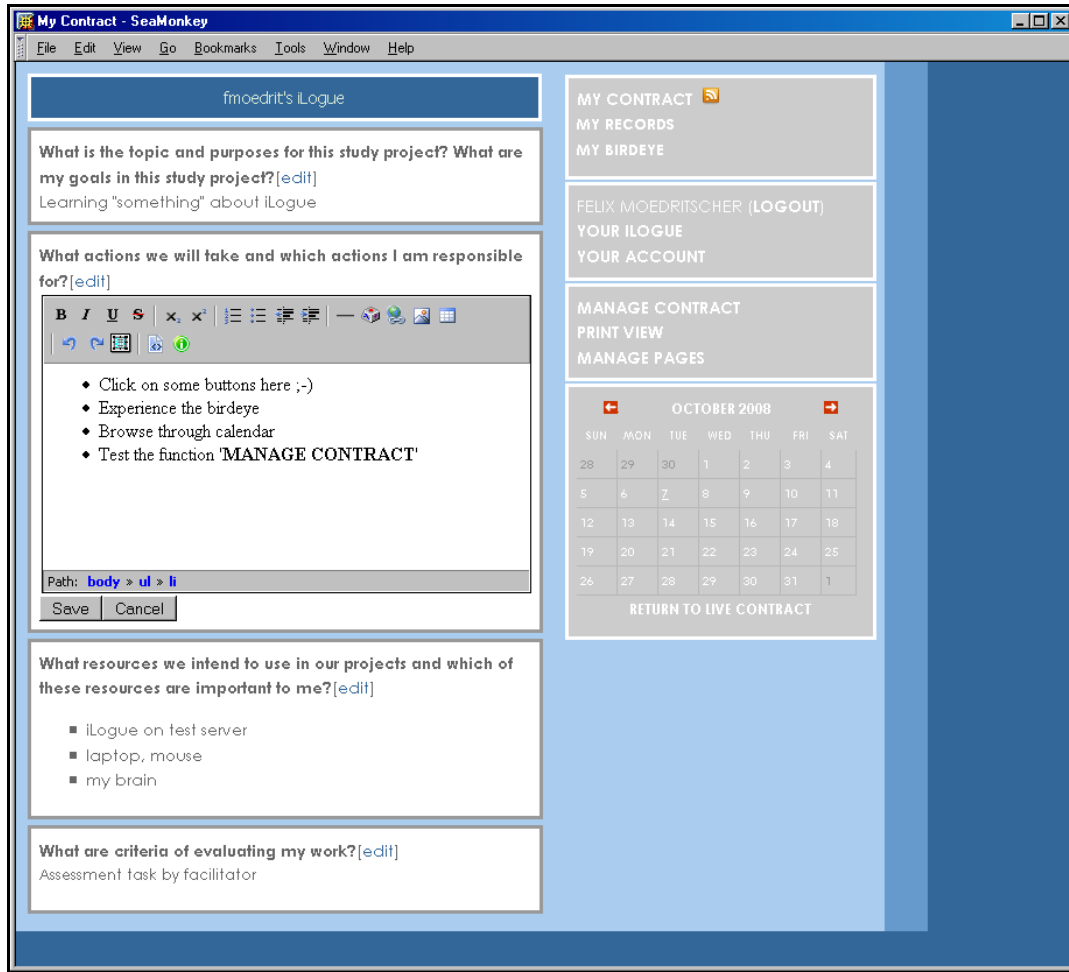


Figure 7: Example of a personal learning contract within iLogue

In figure 7 an example for managing a personal learning contract with iLogue is presented. The user interface distinguishes between three feature sets: First, features for negotiating and developing learning contracts (section 'MY CONTRACT') aim at establishing a contract in a textual form, i.e. by offering basic categories of contract items (purpose/goal, actions, resources, evaluation criteria) that can be used as guideline. Editing these items is supported with an embedded WYSIWYG editor, as indicated with the item 'What actions we will take and what actions I am responsible for?' Second, features for creating records of actions ('MY RECORDS') comprise functions to track user actions which are produced as a side-effect when going through a formal learning process, but can be used for a more profound self-reflection later on.

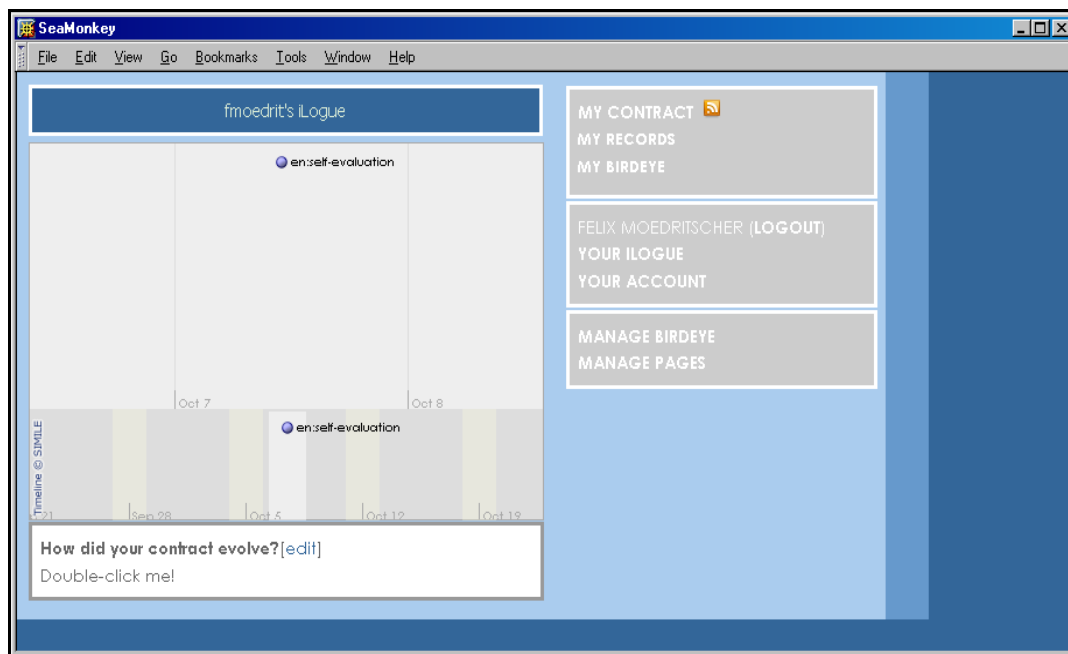


Figure 8: Viewing mode 'bird-eye' of an iLogue personal learning contract

Third, features for allowing and support for a review phase ('MY BIRDEYE') focus on visualizing changes to the personal learning contract entered by learners. Using a timeline as shown in figure 8, enables learners to articulate self-reflective steps throughout learning. iLogue is not meant to be a stand-alone and, therefore, support application. It should interface with the personal tool environment of individuals and groups of individuals and provide an enriched view on artifacts produced in these environments.

Prerequisites and technical details

iLogue is realized as an OpenACS component. Thus, technological prerequisites and the installation steps are equal to the ones of XoWiki. iLogue can be deployed on top of an OpenACS installation which includes the package XoTcl. iLogue is supported by modern browsers.

License and availability

iLogue is under the BSD license and available at the iCamp code base (<http://sourceforge.org/projects/icamp>). A demonstrator can be found under the URL <http://teldev.wu-wien.ac.at/ilogue>.

3.1.2. ObjectSpot, the academic federated search engine

ObjectSpot facilitates academic and scientific research by providing access to multiple professional research databases such as ACM, Citebase, Citeseer, EBSCO, IEEE and many others, via one search interface. Additionally these target repositories are complimented by several learning object repositories of iCamp partner universities. As the federated search engine is able to access any SQL-compliant repository, ObjectSpots repository pool can easily be extended. One may access the search engine by following the link www.objectspot.org or by configuring your own embeddable ObjectSpot portlet version at www.objectspot.org/configure.

Prerequisites and technical details

To access ObjectSpot all you need is an up-to-date version of your favorite browser. To embed your own configured version of ObjectSpot, simply add the generated html code from the configurator to your website. Should you want to run ObjectSpot entirely on your own server, you need to have an OpenACS-ready system, download the code from Sourceforge and follow the installation instructions.

As for the technical details, ObjectSpot is a federated search engine, meaning it queries multiple search engines and merges the results into a single list. ObjectSpot can communicate with repositories providing the Simple Query Interface (SQI, see Simon et al. 2005) using SOAP messages as a means of transportation. Should you need to run ObjectSpot on your own server you will need a running OpenACS installation as ObjectSpot is implemented as an OpenACS package. It needs several packages and Tcl libraries installed as explained in more detail in the installation instructions. The portlet itself is running mostly on javascript and iframes while using AJAX calls and the Prototype framework.

iCamp improvements/adaptations

For iCamp ObjectSpot has greatly improved in performance and monitoring possibilities to ensure a more stable and reliable operation. Several new features have also been added to the portlet interface such as more flexible ranking and filtering options as well as improved cross-browser compatibility. The number of available repositories has grown and an easy to use Moodle interface was implemented to help facilitate the creation of new repositories (i.e. from iCamp partner universities) containing learning objects provided by the Moodle system in only a few simple steps.

License and availability

ObjectSpot is distributed under the BSD license available for download at Sourceforge http://sourceforge.net/project/showfiles.php?group_id=191261&package_id=255776 under

The SQI targets themselves are implemented separately and the content they provide often requires valid accounts with the data repositories. Nevertheless ObjectSpot provides you with pointers to the information you seek in either case.

3.1.3. Videowiki

Videowiki is a video repository - wiki, where users' video is recorded from webcam and put in wiki page. All videos in one page, which is described by namespace, constitute information about specific topic. Page and its resources can be accessed directly by URL, and if it doesn't exist, it is created and new video is recorded. This is a repository of videos, which can be accessed through other sites, blogs, and wikis by embedding special HTML code. There is a possibility to subscribe RSS to all wideoviki repository, or only some specific pages (see Figure 9).

Prerequisites and technical details

To access Videowiki all that is needed is an up-to-date version of a favorite browser. and a flash plug-in. If you'd like to put there your video, you also need a webcam. If

you'd like to have your own videowiki repository, download the code from Sourceforge and follow the installation instructions.

As for the technical details, there's also a need to install Red5 flash server on your machine. You also need FFmpeg - a complete solution to record, convert and stream audio and video - to get video image out of recorded video. Videowiki is a PHP project done on top of Zend framework.

License and availability

Videowiki is distributed under the BSD license available for download at a svn repository in <http://icamp.svn.sourceforge.net/viewvc/icamp/videowiki/>

Sourceforge under Videowiki can be accessed and successfully used by following the link <http://distance.ktu.lt/videowiki>

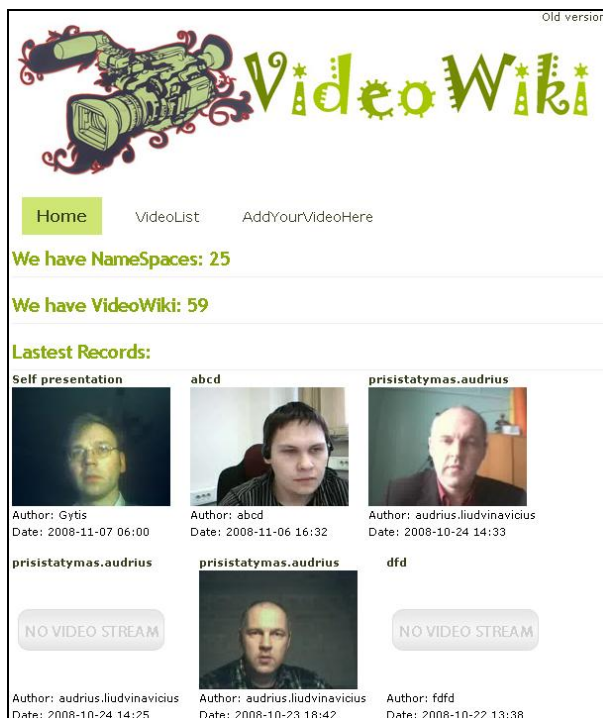


Figure 9: Example of a videowiki interface

3.2. Additional tools and services

Following sections present the iCamp building blocks that were developed as experimental applications during the iCamp project, but were not validated in iCamp field trials due to the time issue, tight schedule and the specific settings of the iCamp field trials.

3.2.1. XoWiki, a multi-purpose Wiki system

Collaborative creation and maintenance of knowledge artifacts is one of the emerging phenomena of online communities on the Internet, as evidenced with prominent examples like Wikipedia.org, MediaWiki.org, LyricWiki.org, Microformats.org, Wikitravel.org, and the forth. Additionally, a collection of web pages (a so-called Wiki) can be useful for teaching and learning, for instance if learners should collaborate on certain topics or facilitators want to develop and share their learning content with others. Consequently, a contemporary approach to technology-enhanced learning requires tools, which enable learners to work on artifacts collaboratively, either by allowing them to publish smaller messages for all or by providing real Wiki functionality.

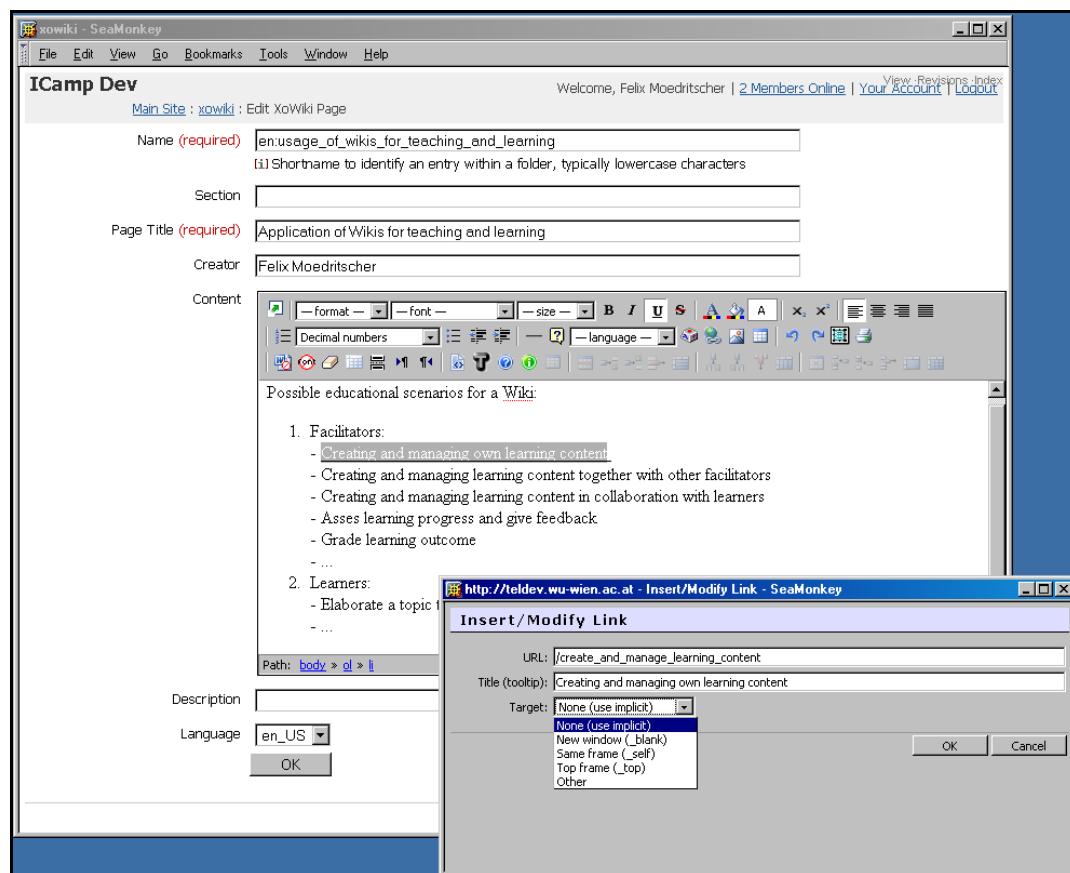


Figure 10: Creating and editing a Wiki page using a WYSIWYG editor

General description and educational purpose

XoWiki is such a Wiki implementation, which is realized as a component of OpenACS (Open Architecture Community System), a framework for building scalable, community-oriented web applications. XoWiki includes a rich text editor for creating and editing Wiki pages in a comfortable way and provides facilities for structuring, commenting, tagging and visualizing Wiki-based content. Figure 10 shows the form for creating and editing Wiki pages. Here, facilitators and learners can use the WYSIWYG ('What You See Is What You Get') editor for entering web-based content. Furthermore, users can create hypertext structures by creating links to other Wiki pages, as indicated in Figure 10.

In figure 11 an example page is displayed. Beside the common Wiki functionality (e.g. specifying the font, the text style, the alignment, numbered and item lists, and so forth), XoWiki provides community-enabling features, like the functions for tagging and commenting a page at the bottom. Thus, learners can add value to the learning content by discussion Wiki pages and structuring it according to their own vocabulary. Using the underlying access control mechanism, XoWiki also supports the definition of different roles in collaborative learning activities.

The screenshot shows a web browser window titled 'xowiki - SeaMonkey'. The page content includes:

- Header: 'ICamp Dev' with navigation links: 'Welcome, Felix Moedritscher | 1 Member Online | Your Account | Logout' and 'Main Site : xowiki : Scenarios for Creating and Managing own Learning Content; Edit · Revisions · New Page · Delete · Admin · Notifications · Search · Index'.
- Notification box: '1 registered user in community xowiki in last 10 minutes: Felix Moedritscher 1 minute ago'.
- Page title: 'Scenarios for Creating and Managing own Learning Content'.
- Purpose: 'Showing how facilitators can be supported in creating and managing own learning content by using XoWiki'.
- Scenario A: 'Copying existing content into a Wiki and formatting it'.
- Diagram: A diagram showing two equivalent circuit representations. The left representation shows a trapezoidal block with two input ports labeled I_0 and I_1 , and one output port labeled 'out'. A control signal 'sel' is shown entering from the bottom. The right representation shows a similar block with a dashed line indicating a different internal configuration or state, also with inputs I_0 , I_1 and output 'out', and control signal 'sel'.
- Tags: 'Your Tags: content-authoring, facilitator (edit tags, popular tags)'. The current tags are 'content-authoring' and 'facilitator'.
- Comments: A comment by Felix Moedritscher on 09/22/08: 'Concerning scenario A by Felix Moedritscher on 09/22/08'. There is an 'Add comment' link.
- Footer: 'Install locales' link.

Figure 11: Example Wiki page with embedded image, on the bottom typical community features like tagging and commenting

Finally, XoWiki exceeds other Wiki systems by dynamic, includable elements for the content, so-called 'includelets', to be utilized by users. These includelets can be inserted into the content in a simple way by using a special syntax. An overview of the includelets available is given by adding '{{available-includelets}}' into a Wiki

page. Typical examples for includelets are the inclusion of an RSS feed, a tag cloud, meta-information of a page, a timeline, a graph, or a portlet implementation of a different system. Additionally, this includelet mechanism also allows to divide a Wiki page into different sections, e.g. for arranging the portlets.

From a pedagogical point of view, XoWiki can be used in many ways for educational purposes. Some examples are listed below according to Duffy and Bruns (2006), Bergin (2002) and Lamb (2004):

- Collaboration (writing research proposal and projects, joint articles, group works, brainstorming, presentations, etc.) as it enables dialog and share information among learners as well as facilitators;
- Knowledge construction as it documents work and keeps the history of the document making pages created earlier accessible;
- Monitor the work of learners and analyze the progress of their work;
- Create a list of bibliography, reference lists or outlines;
- Make summaries of learners thoughts and suggestions as well as facilitators contributions;
- Course handout and resources with the opportunity to give and get feedback, comment and edit it by learners;
- Notebook that peer-learners can comment, edit or add their notes;
- Learners homepages.

Prerequisites and technical details

XoWiki (current stable version 0.6) has being developed within the OpenACS community (OpenACS, 2007). Technologically, XoWiki is implemented as an OpenACS package and following an object-oriented extension of the scripting language TCL, namely XoTcl. To deploy an own XoWiki instance, an AOLServer plus an OpenACS installation (including the XoTcl package) must be available. Subsequently, the OpenACS administrator can install and deploy the XoWiki instance. Concerning the end user, XoWiki can be accessed with the most common modern browsers.

License and availability

The XoWiki software is under the BSD license and available at 'cvs.openacs.org' (to be checked out manually). A demonstrator is available under the URL <http://teldev.wu-wien.ac.at/xowiki>.

3.2.2. Mash-UP Personal Learning Environments (MUPPLE)

Aiming at lifelong learning and second road education, personal learning environments (PLEs) comprise the concept of learners designing their learning environment (Van Harmelen, 2006) in order to connect to a network of actors, artifacts, and activities (Koper et al., 2005). In contrast to didactic-awareness and learning management systems (LMSs), PLE-driven approaches focuses on the perspective of learners, i.e. the end-user development of a learning environment, interoperability of learning tools and services, as well as the support of learners in terms of social competencies and PLE design capabilities. Consequently, personalized learning is not achievable through instructional design on the basis of

top-down, ex-ante models, but through adaptation effects and recommendations in networked learning ecologies.

General description and educational purpose

The idea of Mash-UP Personal Learning Environments (MUPPLE) represents PLE-based, activity-driven collaboration in learning networks. A first MUPPLE prototype aims at the following three aspects of learning environment design (Wild et al., 2008): First, ‘learning to learn’ within the context of a subject is preferable to ‘transferring’ domain-specific knowledge. Additionally, it is important to emphasize the acquisition of transcompetences, i.e. social, self, and methodological competences. Second, the learning environment is considered as an important part of the learning outcome instead of being an instructional condition. Third, MUPPLE bases on end-user development and the design for emergence instead of rule-based personalization, as the models involved are simpler while achieving the same effect.

Extending the ideas of educational tool mashups, figure 13 shows what a mash-up personal learning environment look like. After authenticating to the system, users have access to their ‘activity space’ (list of activities at the top of the navigation area

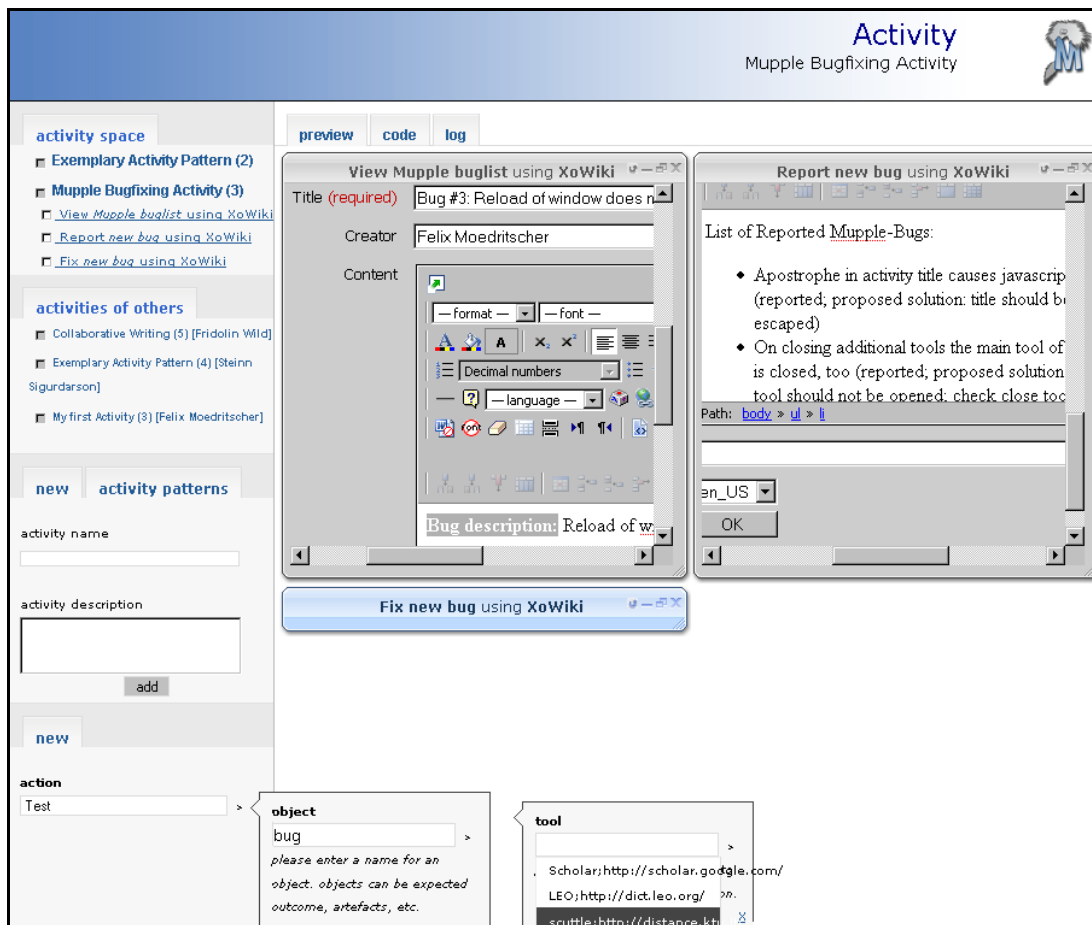


Figure 12: MUPPLE page for an example activity (bug-fixing), which consists of three (learner) actions, each one bound to one outcome and one tool

on the left side); furthermore, administrators receive an overview of all existing activities (‘activities of others’). Right below the activity space, a user has facilities

for creating new activities, either from the scratch or from a pre-defined activity pattern which formerly was derived from a finished activity by a peer or the facilitator. If a MUPPLE page is currently accessed a list of all learner-defined actions are listed within the activity space, and the content of this page is loaded into the main area (below the header and next to the navigational elements).

At the bottom of the navigation area users can insert new actions into the MUPPLE page opened. In doing so, they are supported through recommendations of action-outcome-tool triplets which are derived from other activities.

For a MUPPLE page three different viewing modes are possible: The preview mode (cf. Figure 13) presents the mashup of the learning tools binding each tool to a learner-defined action and an outcome. Users can rearrange, minimize, maximize, and close the windows bordering these tools. Closing a tool is equivalent to stopping the corresponding action. The learner interactions are materialized within the MUPPLE page in the form of script, the so-called 'Learner Interaction Scripting Language' (LISL). All kind of user interactions with a MUPPLE page are stored as one specific line of code into the script. The second viewing mode ('code') provides an in-line code editor for such a LISL script, while the third viewing mode ('log') displays the web-based interpreter for it (cf. Figure 14).

The screenshot shows the MUPPLE web interface in 'log' mode. The page title is 'Activity Mupple Bugfixing Activity'. The left sidebar shows 'activity space' with 'Mupple Bugfixing Activity (3)' selected, and 'activities of others' with 'My first Activity (3) [Felix Moedritscher]' selected. The main area has tabs for 'preview', 'code', and 'log', with 'log' active. The log shows LISL code and error messages: 'Error in line 5: Action name 'Fix' is already reserved!', 'Warning in line 10: No result by interpreter given!', and 'Error in line 11: Tool name 'Google' is already reserved!'. The code includes definitions for objects, actions, and tools, and drag actions for tool placement.

Figure 13: MUPPLE page displayed in the viewing mode 'log', showing the web-based interpreter for the 'Learner Interaction Scripting Language' (LISL)

This interpreter executes the LISL code, creates the underlying learning activity models as well as the tool mashup visualization, gives feedback on errors (red lines plus adequate messages), and allows entering new scripting code via web interface. On revisiting a MUPPLE page, the LISL script is interpreted again, and the latest state of the tool mashup is restored.

Prerequisites and technical details

The MUPPLE component is realized as an OpenACS component and dependent on the XoWiki package. Thus, technological prerequisites and the installation steps are equal to the ones of XoWiki. MUPPLE can be deployed on top of an OpenACS installation which includes the packages XoTcl and XoWiki. For end-users MUPPLE is supported by modern browsers supported by the Ajax library 'Prototype Windows'.

License and availability

MUPPLE is, like other OpenACS component, under the BSD license and available at the iCamp code base (<http://sourceforge.org/projects/icamp>). A demonstrator can be found under the URL <http://mupple.org>.

3.2.3. myDENTITY, an email-based social network analysis tool

Due to the trend to 2.0-driven ecologies and online communities on the Web, social network analysis (SNA) is one of the emerging areas in the few years. Beside the big players in the Internet, like Google, Facebook, Xing, etc., industry and research have started to investigate in this issue, for instance to create expert networks and support knowledge workers. In context of learning, SNA can be used for reflecting the learning process and empower learners to find adequate support by peers. Thereby, a tool which visualizes social relationships can be valuable.

General description and educational purpose

myDENTITY helps individuals in managing their own professional social network. For example, it bears the potential to help to discover hidden relationships, connect with the in-crowd or assist loners. Using myDENTITY requires simply a valid email address for registering. Subsequently all contacts should be asked to write mails only to the new myDENTITY address (which can be eased through the use of reply-to headers). Whenever someone writes to the user of myDENTITY address, myDENTITY keeps track. Whenever someone writes back, myDENTITY keeps track as well. Using myDENTITY does not require any software installation as it bases on the existing mail account and mail programme.

myDENTITY is an intelligent mail forwarding system to the user's 'old' address. By doing so, the individuals' own mail interaction networks can be visualized, as shown with an example in figure 15. The individuals in this social network are either anonym, e.g. if not within one of the trusted organization or a private contact of a colleague, or labeled with their names. Clicking on a name directly opens the mail-client in order to write an email to this user.

Prerequisites and technical details

In terms of hardware there are no specific requirements. On the server side, myDENTITY is based on the LAMP technologies (Linux, Apache, MySQL, and PHP). Besides, the open source 'Postfix' mail-server needs to be installed and configured. For end-users myDENTITY is independent of the operating system, works with any browser and has a web-based user interface.

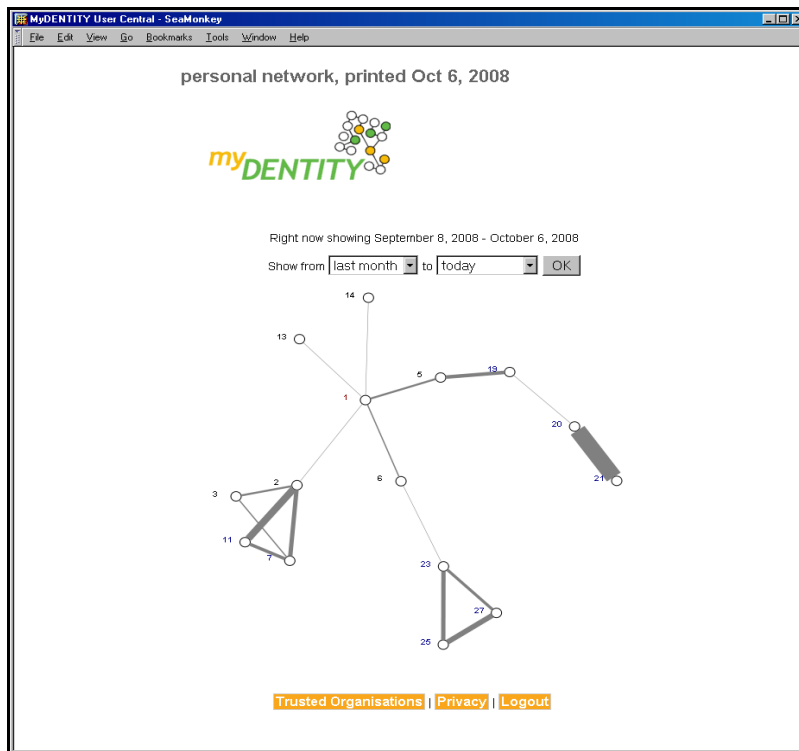


Figure 14: Example of a personal network visualized with myDENTITY

License and availability

myDENTITY is under the GPL license and can be accessed at the iCamp code base (<http://sourceforge.org/projects/icamp>). A demonstrator is available under <http://mydentity.eu>.

4. Conclusions

The main objective of the iCamp WP2 was to design an iCamp Space - an open virtual learning environment that integrates people, resources and activities through different TEL tools and services. Existing open source tools and services were investigated and selected or developed from scratch to support the iCamp pedagogical challenges and adapted a set of them accordingly for integration into the iCamp Space.

This deliverable starts with revisiting the concept of iCamp Space with its final portfolio of updated iCamp building blocks. The reader is provided with more detailed description of the iCamp building blocks that were developed or customised and implemented for the third iCamp field trial (iLogue as a personal learning logbook, Objectspot as an academic federated search engine and Videowiki as a videoblog for recording videoclips). In addition, experimental developments of the tools and systems (xoWiki as a multi purpose Wiki environment; Mupple for empowering students to design their personal learning environment; myDENTITY for providing students an overview of their communication partners) were carried out during the iCamp project and are presented here.

All the selected iCamp building blocks are collected to iFolio – a support system for students and facilitators for finding the most appropriate tools and services that most contribute to their learning needs and specific educational purposes. The second prototype of the iFolio, its user manual, technical specification and installation instructions are presented in Deliverable 2.5.

References

- Bergin, J. (2002). Teaching on the wiki web. Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, ITICSE'02, Aarhus, Denmark: June 24-26. Available at: <http://portal.acm.org/citation.cfm?id=637610.544473> (2008-10-03).
- Duffy, P., Bruns, A. (2006). The use of blogs, wikis and RSS in education: A conversation of possibilities. Proceedings of the Online Learning and Teaching Conference 2006, Brisbane: September 26. Available at: https://olt.qut.edu.au/udf/OLT2006/gen/static/papers/Duffy_OLT2006_paper.pdf (2008-10-03).
- Harri-Augstein, S., Webb, I.M. (1995). Learning To Change. A Resource for Trainers, Managers, and Learners based on self-organised learning. McGraw-Hill Book Company Europe, United Kingdom.
- Koper, R., Rusman, E., Sloep, P. (2005). Effective Learning Networks. Lifelong Learning in Europe, 1, pp. 18-27.
- Kulathuramaiyer, N. (2007). Mashups: Emerging Application Development Paradigm for a Digital Journal. Journal of Universal Computer Science, 13(4), 531-542.
- Lamb, B. (2004). Wide open spaces: Wikis, ready or not. EDUCAUSE Review, 39(5) (September/October), 36-48. Available at: <http://www.educause.edu/pub/er/erm04/erm0452.asp?bhcp=1> (2008-10-03).
- Mödritscher, F., Neumann, G., García-Barrios, V.M., Wild, F. (2008). A Web Application Mashup Approach for eLearning. Proceedings of the OpenACS and .LRN Conference 2008, Guatemala City: Februar 12-16.
- OpenACS (2007). XoWiki. Available at: <http://openacs.org/xowiki/xowiki> (2008-10-03).
- Scharl, A., Tochtermann, K. (2007). The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society. Springer: London.
- Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., Miklós, Z. (2005): A Simple Query Interface for Interoperable Learning Repositories. Proceedings of the 1st Workshop on Interoperability of Web-based Educational Systems, Chiba, Japan.
- Väljataga, T., Pata, K., Laanpere, M., Kaipainen, M. (2007). Theoretical Framework of the iCampFolio: New Approach to Comparison and Selection of Systems and Tools for Learning Purposes. Proceedings of 2nd European Conference on Technology Enhanced Learning.
- Van Harmelen, M. (2006). Personal Learning Environments. Proceedings of the International Conference on Advanced Learning Technologies (ICALT).
- Wild, F., Mödritscher, F., Sigurdarson, S.E. (2008). Designing for Change: Mash-Up Personal Learning Environments. eLearning Papers, 9.

A. Appendix 1. Complete list of iCamp building blocks

xoWiki: <http://teldev.wu-wien.ac.at/xowiki>
xoMashup: <http://teldev.wu-wien.ac.at/xomashup>
Mupple: <http://mupple.org>
iLogue: <http://teldev.wu-wien.ac.at/ilogue>
myDENTITY: <http://myidentity.eu>
Videowiki: <http://distance.ktu.lt/videowiki/>
FeedonFeeds: <http://distance.ktu.lt/feed-on-feeds/>
ObjectSpot, the iCamp Federated Search: www.objectspot.org
Scuttle: <http://distance.ktu.lt/scuttle>
Doodle: <http://www.doodle.ch/main.html>
Wordpress: <http://blog.course.isikun.edu.tr/>
iCamp SIP server: <http://sip.icamp.eu/>

B. Appendix 2. Handbook

iCamp handbook (How to Use Social Software in Higher Education) gives a pedagogical overview of the iCamp building blocks and possible scenarios of every building block. The handbook is accessible from the iCamp website: <http://icamp.eu>.

FOREWORD

This handbook is a result of the iCamp project, a three-year EC-funded research project that set out to encourage innovative educational practices within European higher education.

iCamp's vision is to support competence development in self-organizing intentional learning projects, in collaborating and in social networking by making systematic use of interoperable, networked tools and services.

When we started the project about three years ago, the attitude towards Social Software was rather different from today, both at European Higher Education Institutions as well as amongst the community of TEL (Technology Enhanced Learning) researchers. People were suspicious of this technological shift and many regarded it as just another gimmick, which would fade away within a few years. Now it seems that these critical voices have been proven wrong. Social Software technologies are here to stay and have already started to penetrate into educational landscapes. iCamp has been one of the pioneers in promoting the use of Social Software within higher education.

Our project has been strongly motivated by the three trials that were performed in the "wild" world, where we were facing many challenges that had nothing to do with our core research questions, being largely concerned with organizational aspects. We experienced some highly restrictive institutional policies with regards the use of technologies, which in our opinion hinder any attempt to educate autonomous and self-organized citizens. The democratization of institutions remains an important move that we need to take in Europe and we believe that the iCamp approach may be one small step in this process.

iCamp promotes the effective integration of emerging Social Software technologies within the changing landscapes of European higher education. This handbook illustrates possible strategies for making good use of these technologies, guided by a constructivist pedagogical vision.

I would like to take this opportunity to thank everyone who has contributed to the success of iCamp: our team members in the consortium; the European Commission and, in particular, our project officer, who always believed in our project; our cooperation partners in the trials; our external reviewers and anyone else who has worked with us on this project.

We hope that you find this handbook useful for your practices and that it can encourage educationalists to experiment, engage and become excited about Social Software!

Barbara Kieslinger

C. Appendix 3. Installation instructions of iFolio

iFolio is an open source product (released under LGPL license) that can be downloaded from <http://sourceforge.net/projects/icamp>. Currently iFolio is installed on the server at the Centre for Educational Technology in Tallinn University, but in the future it is possible and recommended to have multiple instances of iFolio in different locations (although these instances will not be able to exchange data between them automatically).

Prerequisites for installing iFolio on a server (to be installed in the same order as listed below) are as follows:

1. Python 2.4.x depending on the requirements of Zope version (see installation guide at http://www.diveintopython.org/installing_python/index.html),
2. Zope 2.10.X family (see <http://www.zope.org/Products/Zope/2.10.6>),
3. GenericSetup (see <http://www.zope.org/Products/GenericSetup>)
4. PluginRegistry (see <http://www.zope.org/Products/PluginRegistry>)
5. PluggableAuthService (see <http://www.zope.org/Products/PluggableAuthService>)

Installation of iFolio (user should be logged in to Zope as Manager):

1) Make sure that iFolio product is situated in your Zope /Products directory and all the other required components are also present. Follow the instructions you can find in INSTALL file situated in /iFolio product folder.

2) After you have made all the steps required by installation instructions, start your Zope instance and log-in as Manager.

Create new instance of iFolio product. This can be done in the following way: log in to Zope management Interface, select "iFolio" from pull-down menu and click on the button "Add", then inset a title for your new iFolio instance (you can also specify the Object Name, which would be the unique identifier of your iFolio instance).

3) After you have successfully created new iFolio instance you should go inside that instance by clicking on it. You should see several folders present there now. You should add some Affordances to the /affordances folder.

4) Now you should go to iFolio graphical User Interface and add at least two new tools there, for the positioning system to get to work. You could use the current account with Manager Permission, but it is better to register a normal iFolio user for that purpose.

5) iFolio will initially send to login section. Click on the register link and fill the form, this will create a user for you and send your password to the e-mail address you have provided.

6) Check your mail, message with password should arrive quite soon. Go back to iFolio and perform the login procedure.

7) Now you should be able to add new Tools and save your perspective information. Go to tools section and create a couple of new tools by clicking on „Add new tool“ link. You need to fill-in the Title field and as many other fields as you need. Do not forget to provide initial values for the Affordances (you can use values from 0 to 10 for that). If you want to change these later, then you can do that by editing tool from Zope Management Interface. Click the save button as soon as you are ready (it is situated in the top right hand corner just in front of the close button).

8) Now there should be some tools shown in the tools section. If you want to see information about the tool – double-click on its title. If you want to make any changes to tools, you can do that by pressing the edit button (it is situated in the top right hand corner just in front of the close button). It is probably a good idea to set some of the affordances for this Tool. Values can be from 0 to 10.

9) iFolio has now got all the needed components. Other people can now register and add new tools, modify them, save their perspectives and look at perspectives from others.