

Tools for Facilitating Idea Generation in Knowledge Practices Environment

Tania Vasileva¹, Ivan Furnadzhiev¹, Vassiliy Tchoumatchenko¹,
Minna Lakkala² and Merja Bauters³

Abstract – The paper outlines the design and implementation of end-user applications for supporting brainstorming and user interaction tightly integrated with shared spaces of Knowledge practices environment (KPE). It covers the features, graphic user interface and architecture of the Note editor and the Sketch pad. The Note editor supports quick brainstorming and production of ideas together and allows viewing, editing, versioning, comparing and printing of shared documents in flexible manner. The Sketch pad provides the functionality for drawing simple graphics for sketching synchronously or asynchronously in a collaborative manner. This tool enables easy in-context drawing to support brainstorming and externalizing ideas that are sometimes hard to explicate verbally. All implemented features are used for easy idea construction and elaboration, and hence support collaborative knowledge creation.

Keywords – Document centered collaboration; collaborative idea generation.

I. INTRODUCTION

The objective of the EU Project “Knowledge Practices Laboratory” (KP-Lab) [1] is to develop a novel theoretical approach, tools and services, and practical models to elicit deliberate creation of knowledge and advance corresponding knowledge practices. The project explores knowledge practices and innovative tool use in the creation and transformation/development of shared knowledge artifacts and practices. KP-Lab places reflective knowledge practices and deliberate advancement of shared knowledge artifacts to the forefront of technology-enhanced learning. KP-Lab partners carry out empirical studies in higher education and workplaces, focusing on interventions aimed to enhance and support knowledge creation activities mediated by tools developed in KP-Lab. Particularly solving of complex, authentic problems, calling for cross fertilization and boundary crossing, collaborative creation of epistemic artefacts as well as use of innovative technologies are addressed.

KP-Lab’s theoretical approach is based on assumptions of learning that involves collaborative efforts to advance ideas,

¹Tania Vasileva, Ivan Furnadzhiev and Vassiliy Tchoumatchenko are with the Department of Electronics at Technical University of Sofia, 8, Kliment Ohridski Blvd., Sofia 1000, Bulgaria, E-mail: tkv@tu-sofia.bg ; ihf @tu-sofia.bg; vpt@tu-sofia.bg

²Minna Lakkala is with Department of Psychology, University of Helsinki, FIN-00014, Finland, E-mail: minna.lakkala@helsinki.fi

³Merja Bauters is with the Metropolia University of Applied Sciences, Espoo, Finland, E-mail: merjab@metropolia.fi

create knowledge and of tool-use to transform existing practices as well as to clarify complex shared objects. This knowledge-creation perspective is called triological learning [2]. Triological learning incorporates use and development of innovative tools, and goes beyond approaches of learning as individual knowledge acquisition as well as social interaction without joint artifact development.

The following design principles (DP) were defined and used as guiding principles in the project to explicate central elements of the triological learning approach: DP1 – Organizing activities around shared “objects”; DP2 – Supporting interaction between personal and social levels; DP3 – Eliciting individual and collective agency; DP4 – Fostering long-term processes of knowledge advancement; DP5 – Emphasizing development through transformation and reflection between various forms of knowledge and practices; DP6 – Cross fertilization of various knowledge practices across communities and institutions; and DP7 – Providing flexible tool mediation. These design principles are rather abstract; therefore various other means have been used in the co-design process to mediate the design decisions between pedagogical researchers, technical developers and users, such as pedagogical or professional scenarios to describe the authentic practices, high-level requirements for tool design or mock-ups and prototypes. These different means have acted as boundary objects and aided in the formation of common ground between the partners [3].

The project explores tools in use, and develops a technological platform with tools and services relying on open standards for the semantic web, ontologies, web services, real-time multimedia, communication, ubiquitous access, and networked or wireless devices. The KP-Lab system is a modular, open, flexible and extensible system where users can customize Knowledge Practices environment (KPE) and its virtual spaces to support various kind of uses and contexts of use. Most development is open-source based.

The KPE is designed to provide specific affordances for joint development of concrete, epistemic objects as well as for planning, organizing and reflecting on related tasks and user networks [4]. The actual work occurs in virtual ‘shared spaces’ that have different views to visualise the shared artifacts, knowledge processes and social relations of the members. The shared knowledge objects can be visualized and searched in various ways to enable smooth collaboration around the created knowledge in shared spaces.

The paper describes two end-user applications as examples of KP-Lab tools that support easy access and intuitive ways for brainstorming and production of ideas together. These applications allow the viewing, editing, versioning, comparing, and printing of the shared documents as well as visual sketching of the ideas in a flexible manner. The Note

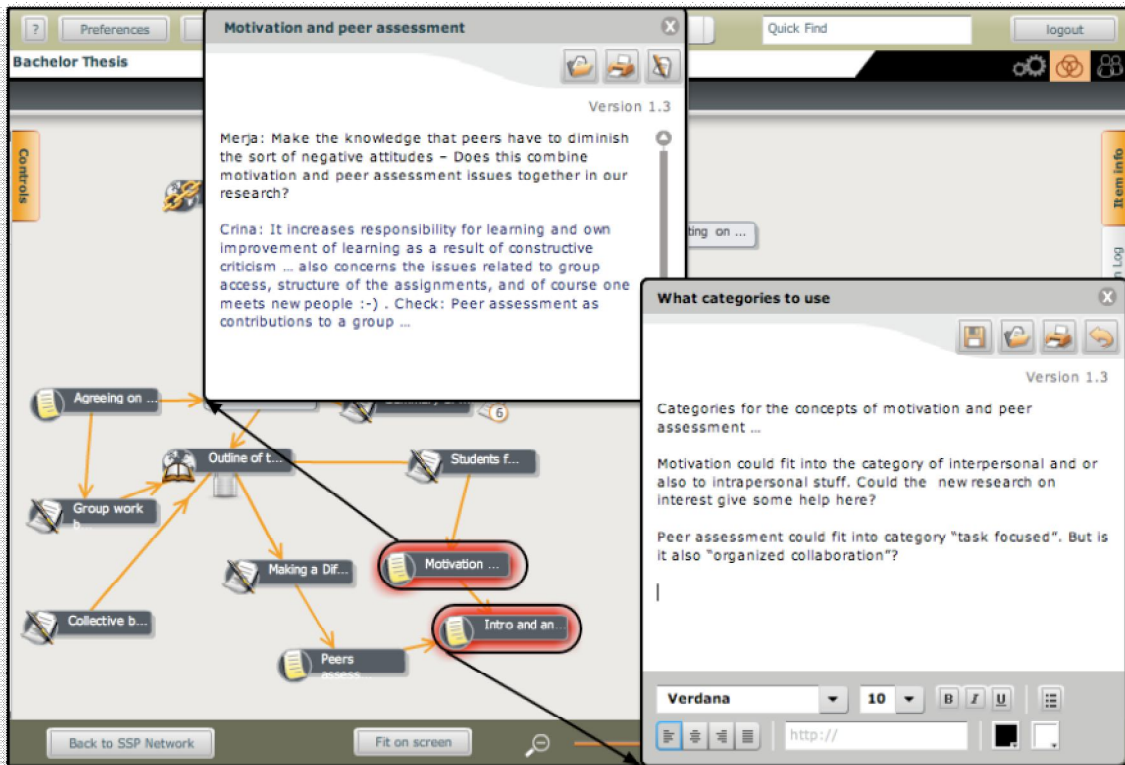


Fig. 1 An illustration of linked “note” objects in a closed, view and edit state in a shared space’s content view.

editor facilitates document centred collaboration (DCC) for writing of simple text objects, while the Sketch pad permits direct collaborative sketching of ideas through drawing. Both tools are tightly integrated in the Knowledge Practices Environment.

II. THE NOTE EDITOR

The “Note Editor” is an editor that enables writing and/or editing of a simple text object, referred to as a “note” object, without the need for an external application. The Note editor is not a replacement for applications such as MS Word, but rather a very simple editor that is tightly integrated into the already existing shared spaces of the KPE.. It is aimed for easy idea construction and elaboration, which is a central element in knowledge practices where a group of actors is aiming at innovatively and openly creating new solutions and viewpoints through writing. For such practice, word processing or Wiki tools are too clumsy, heavy, and inflexible tools which prevent simultaneous working on other artefacts central for idea development.

The “note” is seen as a new type of “Content Item” object in the Content View of a shared space. The other Content Item objects are classified as one of the following types: Wikipage, File, Weblink (URL), Vocabulary editor, Sketch pad, visual models. A note can be linked (associated) to an object in the Content View of the shared space or left unlinked. The notes can be moved around in the Content View by the users to be organised into meaningful visual arrangement. If the note is associated with a shared space object, a link is shown between

the note and the object when the note is selected with the mouse (Figure 1).

The Note editor is tightly integrated in to shared spaces of the KPE. The Note Editor Graphic User Interface (NE-GUI) is implemented as a shared space GUI component. It is activated by the SSGUI when the user chooses to create a new or open an existing note. A “note” can be created, edited or deleted by any user that is a member of a particular shared space. Creating a “note” equals the addition of an independent object within a Content View of the shared space. The individual “note” object has its own metadata specifications and can be used, linked, moved, commented, chatted about, etc. as any other object/content item in a shared space.

A. States of a Note Content Item

The notes can be in one of the following states: closed, opened and version state.

Opened State – When a note has been added to the Content View of a shared space, it can be “opened” from there. Multiple notes can be open on the screen at the same time (see Figure 1). The notes are scalable, namely the size of the note can be changed with the mouse. While the note is open, the other parts of the Content View can be worked on simultaneously. This supports brainstorming and sketching since the other shared objects can be viewed, acted on and organised to contribute to the idea generation. The Note can stay open as long as the user stays inside the shared space where the note is located. In the opened state, the user can choose from two modes: view mode and edit mode.

View Mode: Clicking on an already created note opens it. The content of the note appears in “view” (read-only) mode (see Figure 1. note titled “Motivation and peer assessment”). The user can have several notes in the view state simultaneously.

Edit mode: When a note is in “Open” state, it can be switched to the “Edit” mode from the “View” mode by means of an icon button. In this mode, the Note editor is available and the note object's content becomes editable. The user is able to control the following text attributes: font family, font size, style (bold, italic, underline), hyperlinks, horizontal alignment, and text color (see Figure 1). The background color of the “note” is also user controllable. Bulleted and numbered lists are supported. Each user can have only one “note” in “edit” state. Only one user can edit a particular “note” at a time; the other users of the shared space can see the content of the note but cannot edit it. An indication about who is editing the note is provided.

Closed state: When the “note” object is in the “Closed” state, the object exists in the Content View of the shared space in a similar fashion to other Content Items. The object is a node that can be moved around; it has coordinates that are stored in the same fashion as the node coordinates of other Content Items in the Content View. A note object will be automatically closed when a user leaves the confines of the shared space.

Version state – Versioning and keeping track of the modifications made to a note object by others is important. The user is able to see the latest version and the previous versions and compare them.. A “version state” icon button is used to show a list of versions of the note. The user can select a single version to be opened and viewed or he/she can select two versions and compare them by clicking on the “compare” button. The text which has been deleted from the later version is displayed in red, the text which has been added is displayed in green. Clicking on the “back” button takes the note back into the “versions state”.

The “note” is saved when its state changes from “edit” to something else or through an explicit user request. Saving a note creates a new version of it. All versions are available during the entire note lifecycle (i.e. until the note is deleted). The size of the note is stored as a view data. When a note is changed from a closed to open state, the stored size is used. A note can also be printed.

III. THE SKETCH PAD

The Sketch pad allows users to collaboratively view, create and edit sketches. It provides the functionality for drawing simple graphics (squares, circles, triangles, lines, arrows, free drawing with colors and with simple editing commands) with texts added in between for sketching in white-boarding manner. It is analogous to Note editor but for creating knowledge through drawing instead of writing. It provides another way of outlining tentative conceptions because it is often hard to verbalise emerging ideas. Users are able to produce and develop ideas, plans, sketches, etc. collaboratively and keep track of their history.

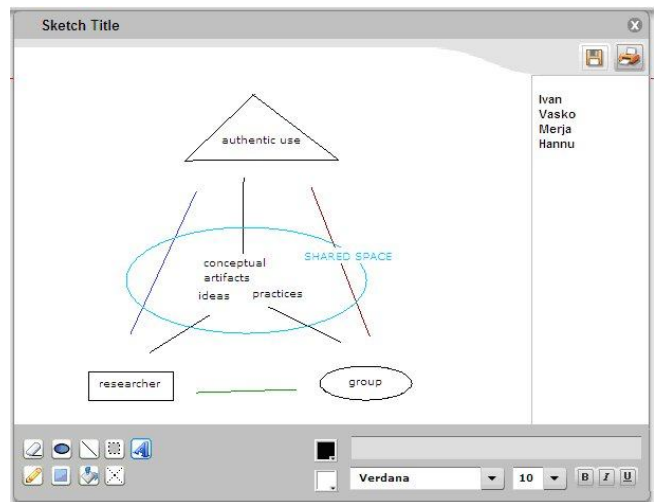


Fig. 2. Sketch pad GUI with opened existing sketch.

The functionalities and the graphic user interface have similarities to the Note editor, so that these two are easy to be used side by side. Figure 2 presents the screen design of the GUI of the Sketch pad. The title bar section displays the title of the drawing. The Menu bar contains buttons for common actions like save and print. The drawing pad is the place where the current sketch is visualized. Participants list visualizes the list of users that are currently working on the sketch. The Tool bar is the bar with tools for sketch creation.

The prototype implements the following drawing and collaboration functions: draw figure, free-hand drawing, add text, flood-fill with color, eraser, copy, move and erase the content of a rectangular region, selectable line thickness, and drawing synchronization between users. The user selects a drawing tool (line/arrow tool, polyline/polygon tool, rectangle/square tool, oval/circle tool, free hand tool, text tool) from the drawing tool GUI. The user creates a simple drawing in the drawing pad. The sketch is synchronized between the users at the end of every drawing operation. Users can undo the last drawing/editing operation and copies/moves or delete part of the sketch in the drawing tool GUI.

Users can open up several sketches at the same time. Sketches can also be easily used as background images in the content views of shared spaces, or the sketches can be downloaded as gif/jped/png files to other applications. The size of the drawing canvas can be chosen from certain fixed sizes. Drawn images can be saved as content items into shared spaces (with metadata) and can be used for semantic grouping/categorisation with shared objects that exist in the Content View of a shared space.

B. Architecture

The overall architecture of Sketch pad is presented on Fig.3. The Sketch pad is one of the shared space tools. It is invoked from the “Open” menu of the Content Item that represents the sketch.

The tool has its own synchronization service based on BlazeDS responsible for sketch synchronization between

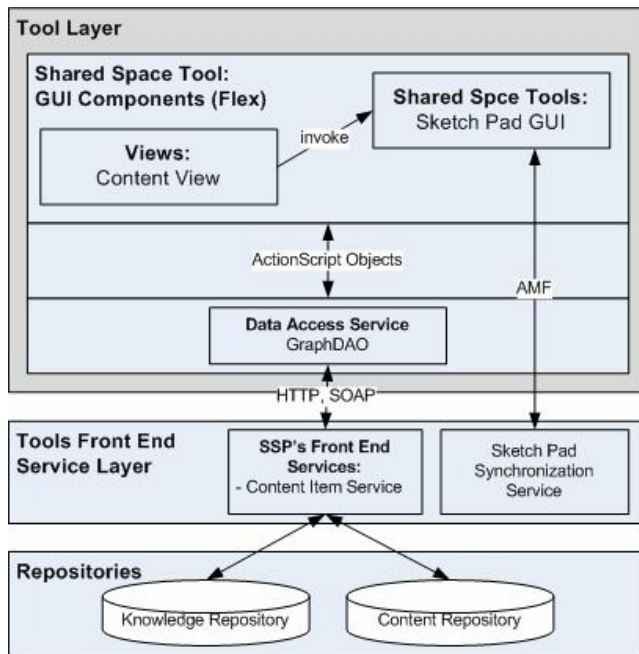


Fig. 3. Architecture of the Sketch pad

users working on it. The same service is used for keeping track on users currently working on the sketch.

The main Content Item operations are handled by the front-end services. The Content Item Service is called from GUI by the Data access services GraphDAO class, which is the Shared Space GUI's main data access class. The Content Item service is used to create/delete/modify the Content Item presenting the sketch in the Content View of the shared space and to retrieve/ update the content of sketch Content Item.

The Sketch pad is heavily dependent on the Trialogical Learning Ontology (TLO), because the sketch is presented as a Content Item in the Content View of a shared space. The drawn images can be used for semantic grouping/categorisation of the shared objects that exist in the Content View of a shared space.

IV. CONCLUSIONS

The paper describes two end-user applications as examples of KP-Lab tools facilitating document centred collaboration for writing of simple text objects (Note editor) as well as for direct collaborative sketching of ideas through simple drawing (Sketch pad). It covers the features, user interface and architecture of the tools, which are dedicated to support brainstorming and idea exchanging when working on shared knowledge objects.

These two tools are easy to be used side by side (with Note editor the user can externalize ideas as text notes and with the Sketch pad as graphics). Users can make simple sketches to support brainstorming and externalizing ideas with figures and short texts (while, for example, planning a structure for a text document, or planning upcoming tasks and work in a group). Both tools have their own front-end synchronization service based on BlazeDS, responsible for text or drawing

synchronization between the users. The same service is used for keeping track on users currently working on the note or sketch.

KPE has been piloted by second year Media Engineering students in a term project, through which they are expected to improve their practices in managing projects, while designing a product or service for a real client. In these courses the Note editor has been used by the students in various ways. Some students have used them as announcement notes to inform the whole group about important issues, or to divide tasks to each other, namely for organising the collaborative work. Others have used the notes to write, in place, the meeting minutes of important meetings and then arranged them visually in the Content view to represent the chronology of the meetings. The notes have also been linked to relevant other knowledge objects and tasks in order to explicate the relationships of the note contents to other materials and tasks that the group has in their shared spaces.

In the University of Helsinki, KPE was used in a bachelor year methodology course on semiotic studies. In the course, students wrote scientific reports in pairs. The Note editor was suggested to be used for co-constructing questions, ideas and versions of the report. According to the observations, the use of Note editor to formulate questions and to comment on others' questions did aid the students to grasp the relevance of generating research questions. In the course feedback, the students mentioned that they considered the functionalities to be useful for their inquiry task. We may conclude that the Note editor and linking functionalities both helped to organize the artifacts and promoted reflection. The Sketch pad has not been used yet in authentic contexts.

ACKNOWLEDGEMENT

The reported work is developed within the framework of the Knowledge Practices Laboratory (KP-Lab) project co-funded by the IST programme of the EU 6th R&D Framework programme.

REFERENCES

- [1] The KP-Lab website. [Online]. ICT-27490: Knowledge Practices Laboratory (KP-Lab), Available: www.kp-lab.org/
- [2] Paavola, S. & Hakkarainen, K. (2005). The Knowledge Creation Metaphor – An Emergent Epistemological Approach to Learning. *Science & Education* 14, pp. 535-557
- [3] Deliverable 2.2: Guidelines and models on implementing design principles of KP-Lab, application scenarios and best practices, v.2. Report for European Commission, KP-Lab project, IST-27490 [Online]. (2007) Available: http://www.kp-lab.org/publications/public-deliverables/documents-of-the-public-deliverables/KP-Lab_D2-2.pdf.
- [4] Markkanen, H., Holi, M., Benmergui, L., Bauters, M., & Richter, C. (2008). The Knowledge Practices Environment: a Virtual Environment for Collaborative Knowledge Creation and Work around Shared Artefacts. In J. Luca & E.R. Weippl (Eds.) *Proceedings of ED-Media 2008. World Conference on Educational Media, Hypermedia and Telecommunications*, Vienna 2008, pp. 5035-5040. Chesapeake, VA: AACE