

TOOL FOR SYNCHRONOUS COMMUNICATIONS IN COLLABORATIVE ENVIRONMENT

**Ivan Haralampiev Furnadziev, Vassiliy Platonovich Tchoumatchenko,
Tania Krumova Vasileva**

Department of Electronics, Technical University of Sofia,
8, Kliment Ohridski St., 1756 Sofia, Bulgaria, phone: +35929653731, ihf@tu-sofia.bg

The paper outlines the design and implementation of end-user applications for networking and community building in shared collaborative environment. It covers features, architecture and user interface of the Chat tool, dedicated for synchronous communications in Shared Space. Chat tools aim to facilitate contextualized user communications and ideas exchanging when working on shared objects. All implemented features help in collaborative knowledge creation and facilitate reflection of shared knowledge and practices.

Keywords: Collaborative learning, synchronous text communication

1. INTRODUCTION

The EU Project “Knowledge Practices Laboratory” (KP-Lab) [1] is an integrated project aiming to facilitate innovative practices of sharing, creating and working with knowledge in education and workplaces based on triological approach to learning [2]. This approach builds on the assumption that learning is not just individual knowledge acquisition (monological) or social interaction (dialogical), but activity is organized around transforming, or creating shared knowledge objects (Fig.1)

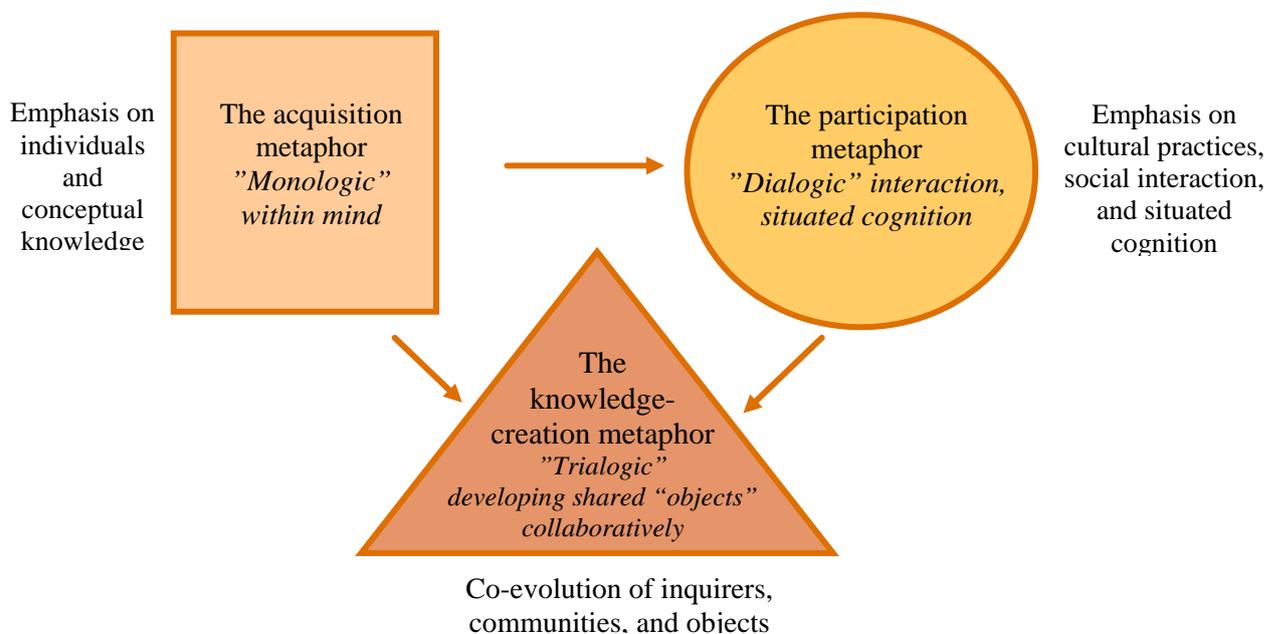


Figure 1. Three metaphors of learning.

The KP-Lab project focuses on developing comprehensive set of tools to manage knowledge processes, to work with knowledge artifacts, and to collaborate with peers. It utilizes emerging Semantic Web technologies and refers to applications that help groups to create, modify, discuss about, comment on, build on and rise above shared objects of activity (Fig.2).

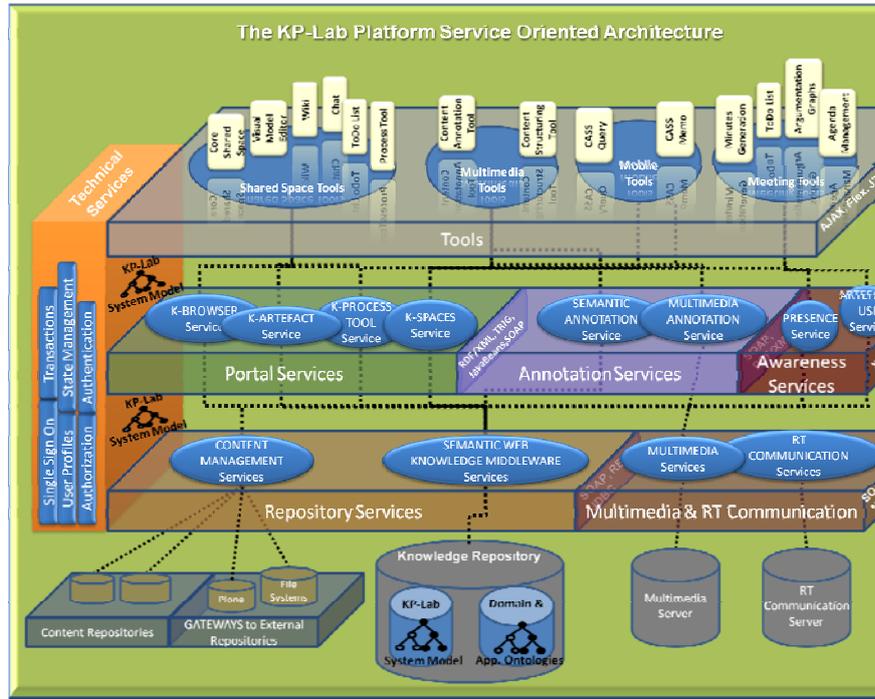


Figure 2. KP-Lab platform architecture.

The Shared Space Application (fig.3) is a central application in KP-Lab and most of the other end-user applications are integrated into it. The shared knowledge objects can be visualized and searched in various ways to enable smooth collaboration around the created knowledge.

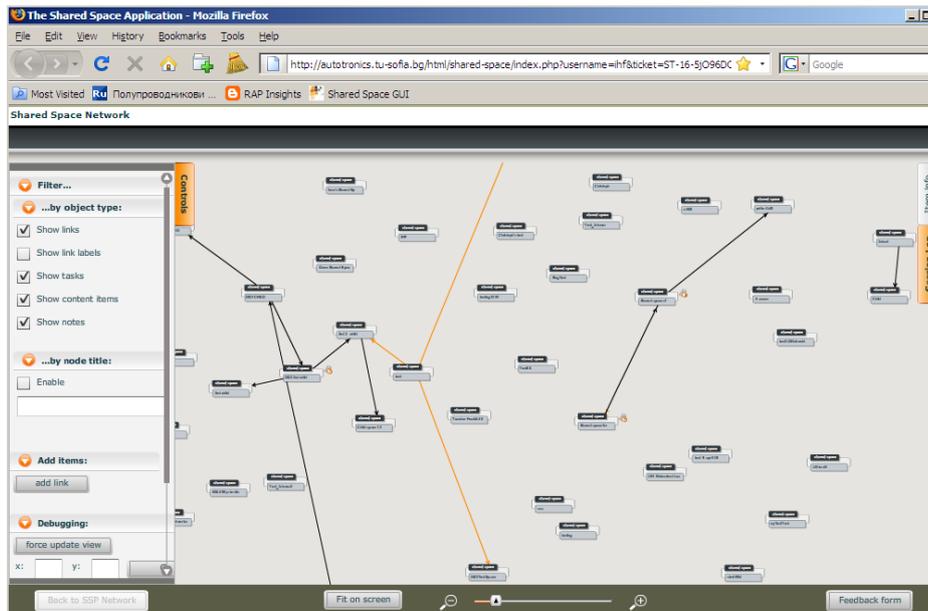


Figure 3. The Shared Space Application

The paper describes end-user application for synchronous communication within a Shared Space to facilitate user interaction in a contextualized manner (*Chat tool*). The process of developing the functional specifications of SS end-user applications has followed the co-design process based upon both the pedagogical and professional scenarios [3].

2. CHAT TOOL

The Chat Tool is tightly integrated in the Shared Space Graphic User Interface (GUI). There are three types of chat “channels”:

- General chat is part of the Shared Space. It is saved as a Content Item in SS. There is only one “General chat” per SS.
- One-to-One chat (“private”) is part of the Shared Space. It is saved as a Content Item in SS and is visible for chat’ participants ONLY.
- Context-based chat is part of the Shared Space. It is as a Content Item in SS. Only one context-based chat associated with one SS object is allowed.

All type of chats can send message to existing chat session, receive message from current chat session; indicate author, date and time of the message; permit images in the chat message; show all online users; create contact list from all existing users. The chat window can be resized. The chat session can be in one of the following states: closed and open, not available, do not disturb. Log user status is provided in a message area to indicate who joins or leaves the chat session.

In *General chat* the chat session object is automatically created when first message is post to SS General Chat. The chat session object can be moved around the SS by the users. The coordinates of the chat session are stored by the SS. The chat session is saved when someone post new message to it. Chat history is kept and can show only messages from the past 1 day, 1 week Log user status in message area – join or leave chat.

One-to-one chat is available inside or outside the Shared Space. It opens new “private” chat session and invites chat participants. It saves the chat session inside SS as a “private” content item (only the chat participants should be possible to view it) or outside SS in custom content repository. This chat keeps chat history and shows “private” chats history in read only mode.

Context-based chat is associated with an object in the personal shared space created by user in Shared Space Application. The link to the discussed object is automatically created. The context based chat title is defined. This chat also keeps the chat history and shows log user status.

2.1 Data Model and Architecture

The chat tool data model is shown in figure 4. The Chat Session is a subclass of the ContentItem. The Chat Session Object (General or Context-based) text (content) is stored in the same Content Repository (CR) like the other Content Items.

The chat tool architecture is given in figure 5. The Chat Tool Service (CTS) is responsible for sending and receiving of chat messages, retrieval of chat’s history and visualization of user’s online status.

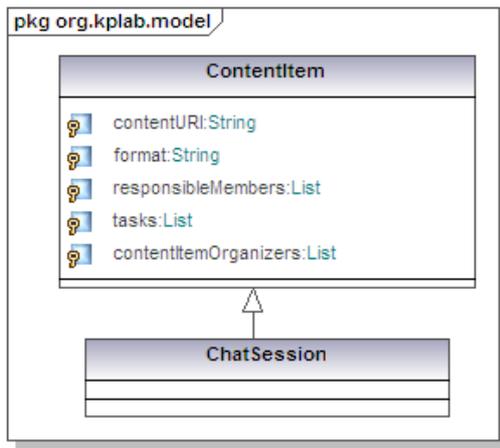


Figure 4. The Chat Tool data model

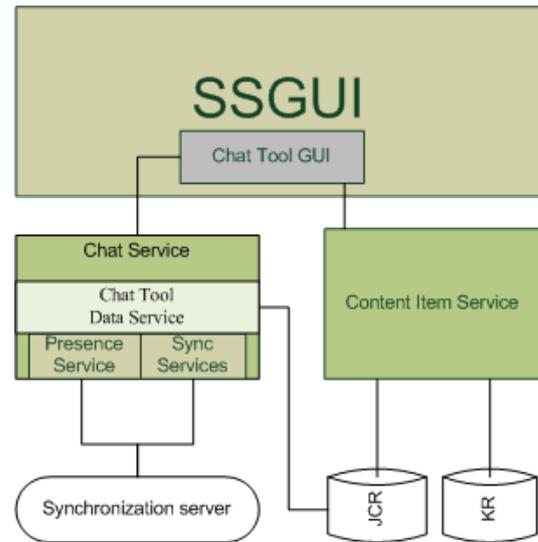


Figure 5. The Chat tool architecture

It uses three addition services for data retrieval:

SSGUI PresenceService is responsible for the SS users' status;

SSGUI SyncServices is responsible for broadcast of the messages;

Chat Tool Data Service is responsible for additional helping activities as: getting server synch time, getting user' chats history, append message to already existing chat, getting all messages for the defined period of time. Histories form all context and general chats are stored as ContentItems via the Contentitem service.

The Chat Tool Service public methods are shown in figure 6. Some of them are exposed as web services as well.

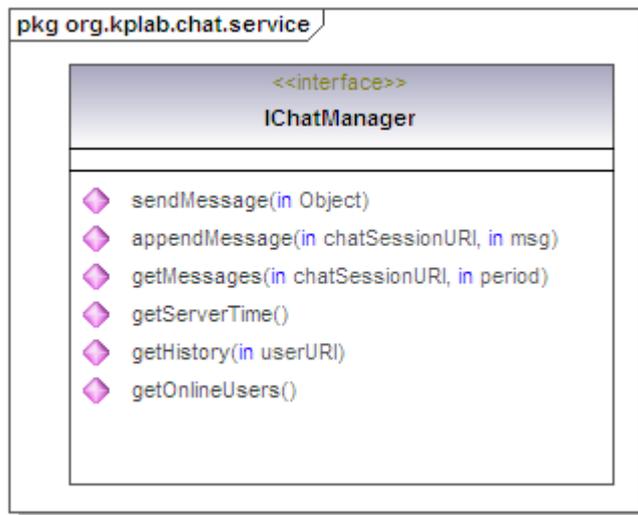


Figure 6. The Chat tool public methods.

2.2 User Interface and Interactions

The Chat Tool GUI is part of the Shares Sspace GUI. It is organized as tabs – one contacts tab, one history tab and chat sessions tabs (0 or more).

Using tabs help to separate the various chat windows, list of contacts and chat' session history, so it is easy way to minimize the number of chat windows open.

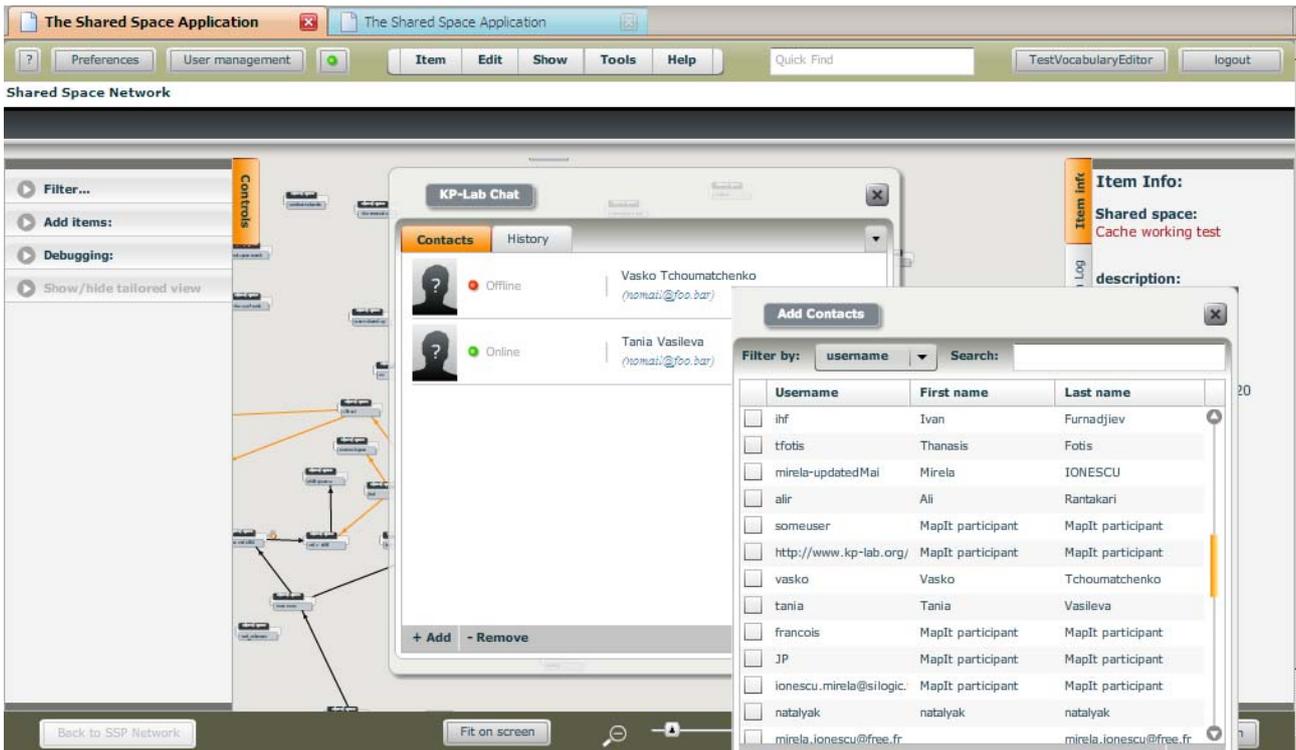


Figure 7. The general chat windows integrated in Shared Space Graphic User Interface.

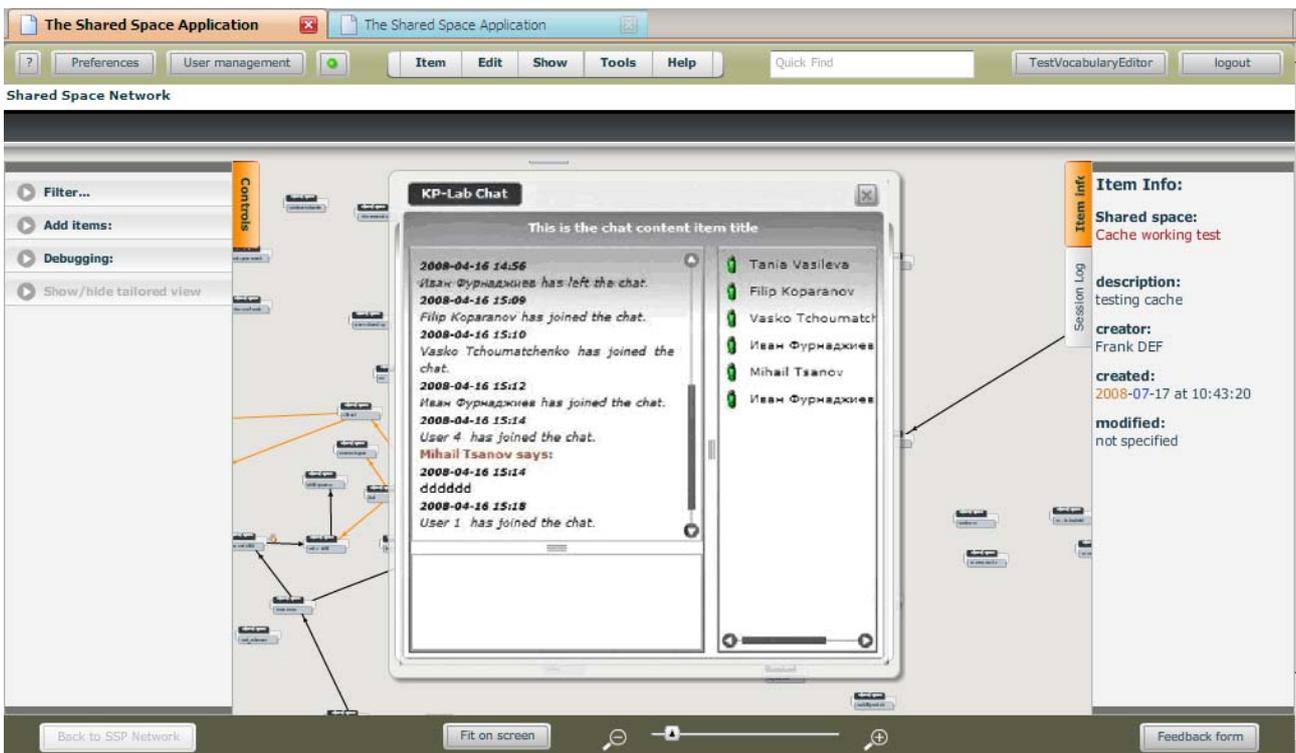


Figure 8. Context based chat window

Figure 7 shows the general chat window with the contacts tab as the default tab. User can select person to chat with, see the chat history, or add or remove chat contacts. Log status of chat participants is also visible. When “add/remove contact”

option is chosen users can be filtered by first or last names or by e-mail. Search option is also available.

A Context based chat GUI with the log status of the participants is illustrated in figure 8. All windows, message and status areas are user-resizable.

3. CONCLUSIONS

The paper discusses a Chat tool as examples of KP-Lab end-user application that is specifically designed for facilitating idea-generation, collaboration and contextual communication between collaborating partners.

Three types of chat tools (general, one-to-one and context based) support communications between shared space users. All types of chats can send message to existing chat session, receive message from current chat session; indicate author, date and time of the message; permit images in the chat message; show all online users and their log status; and create contact list from all existing users. They are tightly integrated in Shared space and aim to support working with the common objects across contexts, objectives and organizational boundaries.

All implemented features help in organizing learning activities around shared objects and in providing means for interaction between personal and community levels of activity.

4. ACKNOWLEDGEMENTS

The reported work is developed within the framework of KP-Lab project. The KP-Lab Integrated Project is sponsored under the 6th EU Framework Programme for Research and Development. The authors are solely responsible for the content of this article. It does not represent the opinion of the KP-Lab consortium or the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.”

5. REFERENCES

- [1] The KP-Lab website. [Online]. ICT-27490 (IP): Knowledge Practices Laboratory (KP-Lab), Available: www.kp-lab.org/
- [2] (2005) Paabola, S. & Hakkarainen, K. The Knowledge Creation Metaphor – An Emergent Epistemological Approach to Learning. *Science & Education* 14, 535-557. Available: <http://www.helsinki.fi/science/networkedlearning/texts/knowledgecreationmetaphor.pdf>.
- [3] (2007) 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]. Available: http://www.kp-lab.org/publications/public-deliverables/documents-of-the-public-deliverables/KP-Lab_D2-2.pdf.