ARCHITECTURE OF KNOWLEDGE ARTIFACT TOOL AIMING TO SUPPORT COLLABORATIVE KNOWLEDGE CREATION

[1] Vassiliy Tchoumatchenko, [2] A. M. Scapolla, [3] Tania Vasileva,
[4] Ivan Furnadhziev, [5] A. Poggi

[1, 3, 4] Department of Electronics, Technical University of Sofia
8, Kliment Ohridski St., 1000 Sofia, Bulgaria, phone: +359888966871, vpt@tu-sofia.bg

[2, 5] Dipartimento di Inggeneria Biofisica ed Elettronica Universita' di Genova, Italy

In recent years, semantic web based applications have been developed in numerous fields, including those of knowledge management and education/training. EU project KP-Lab is intended to productively utilize emerging technologies of the semantic web infrastructure. The paper considers the functional requirements and the architecture of Knowledge artifact tools developed to support collaborative knowledge creation. These tools are responsible for Knowledge Artifact (KA) management and annotations. KA annotations taxonomy and relationship between KA tools, repositories and artifact are also discussed.

Keywords: Knowledge practices, Semantic Web, Knowledge artifact

1. INTRODUCTION

The EU project Knowledge-practices Laboratory (KP-Lab) [1] aims at facilitating innovative practices of working with knowledge in education and workplaces. The overall objective is to propose a new model for higher education. This new model recognizes that higher education should simultaneously concentrate on the participation to activities in universities as well as activities at workplace. In order to facilitate knowledge creation in educational and professional contexts, it is necessary to achieve deeper understanding of concrete processes and mechanisms of knowledge creation. Relating on *trialogical* technologies the project refers to applications that help groups to create, modify, discuss about, comment on, build on and rise above shared objects of activity.

The project focuses on developing collaborative tools that allow the users to use visual representations, whether graphics or video, for making, organizing and restructuring their contributions. In this regard the Semantic Web that is considered to be the next, revolutionary stage of web technology promises to provide novel resources. KP-Lab is intended to productively utilize emerging technologies of the semantic web infrastructure.

The user environment is organized as a virtual shared space that enables collaborative knowledge practices. It provides users a personalized and temporal view that organizes participants' collaborative activity around shared knowledge artifacts. Knowledge practices are supported by comprehensive set of tools to manage knowledge processes, to work with knowledge artifacts, and to collaborate with peers.

The paper discusses the functional requirements and the architecture of KA tools developed to support collaborative knowledge creation. These tools are responsible for Knowledge Artifact management and annotations.

2. SYSTEM ARCHITECTURE

Technologically KP-Lab is a ubiquitous collaborative environment that allows individual agents and their communities to work for developing shared knowledge artifacts using cross-media communication devices. Wide range of innovative tools is required to support KP-Labs will be developed, including those for: personal productivity; collaborative working with digital knowledge artifacts; organizing KP-Labs processes; capturing tacit knowledge; storing and managing knowledge about cognitive efforts; communication tools based on presence information. All tools will operate in the framework of the Shared Space.

The overall Shared Space architecture is shown on Figure 1.

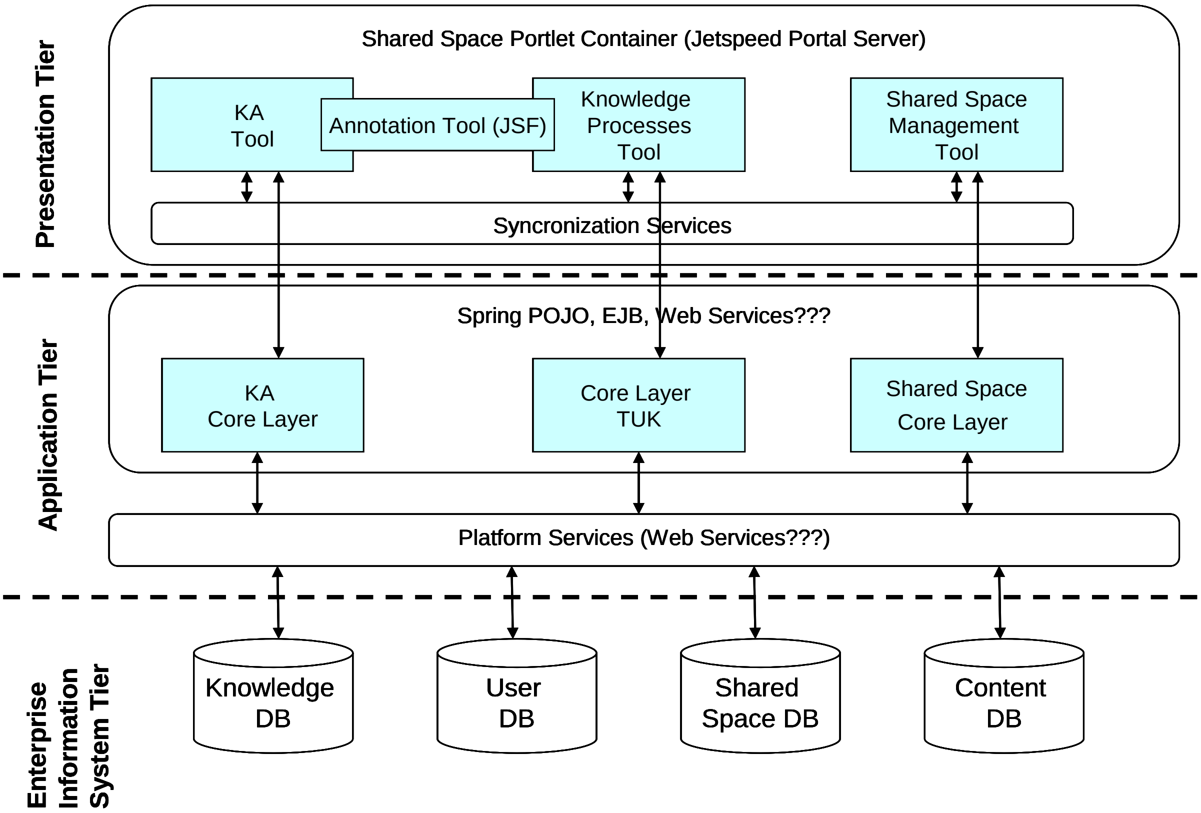

Figure 1. Shared space architecture

The KP-Lab tools rely on the knowledge middleware system to be developed in the project. The development of the knowledge middleware is based on existing semantic web related technologies available from the on-going research in the IST program and in the open-source community. In order to make the knowledge middleware operational, the project will develop metadata and ontology descriptions of the KP-Lab processes for the various user contexts addressed in the project.

3. Knowledge Artifact Annotations Taxonomy

Figure 2 defines a very simple KA annotations taxonomy [2], [3]. The main goal of introducing this classification is to determine what kind of annotations will be handled by the KA annotation tool and where these annotation will be stored.

Formal (semantic) annotations are based on ontological terms and definitions. They are stored in the knowledge repository. Informal annotations are not restricted to a particular limited vocabulary. As minimum textual annotations should be supported, but voice, graphics or other formats should not be excluded. The informal annotations are further divided in two categories - internal (or embedded) annotations, which are stored within the content and external annotations.

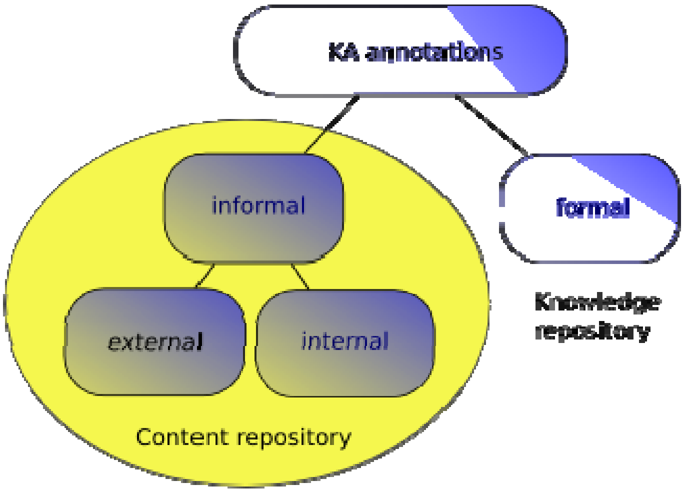

Figure 2 Knowledge Artifact annotations taxonomy

Internal annotations relate to items inside the artifact - e.g. particular paragraph in text document, or particular image area. These annotations are stored inside the digital object (content item) in an application-specific format. Examples of internal annotations are the notes inserted in the documents by the word processors like OpenOffice.org Write or MS Word. The external annotations relate to the entire content item (digital object). They will be stored as a meta-data in the content repository.

4. Knowledge Artifact Tools

The KA tool accesses the content repository and the ontology-based knowledge repository (directly or indirectly) in order to perform two sets of functions: management and annotation.

KA management functions

- create, retrieve, update, delete knowledge artifacts
- visualize artifacts (both content and annotations)

KA annotation functions

- annotate tasks
- annotate artifacts
- support various annotation types (i.e. semantic and informal)

Other functions

- Versioning support for the content repository
- KA to background image mapping
- Collaborate with knowledge middle-ware to provide notifications to the users when the KA is changed
- Collaborate with the logging and profiling tools

It seems beneficial to think about the KA tool as a set of tools, focused on specific groups of tasks. The first two members of the KA tools set are the Knowledge Artifact management tool and Knowledge Artifact annotation tool.

The major KA tools and their relations with the artifacts and repositories are shown in Fig. 3.

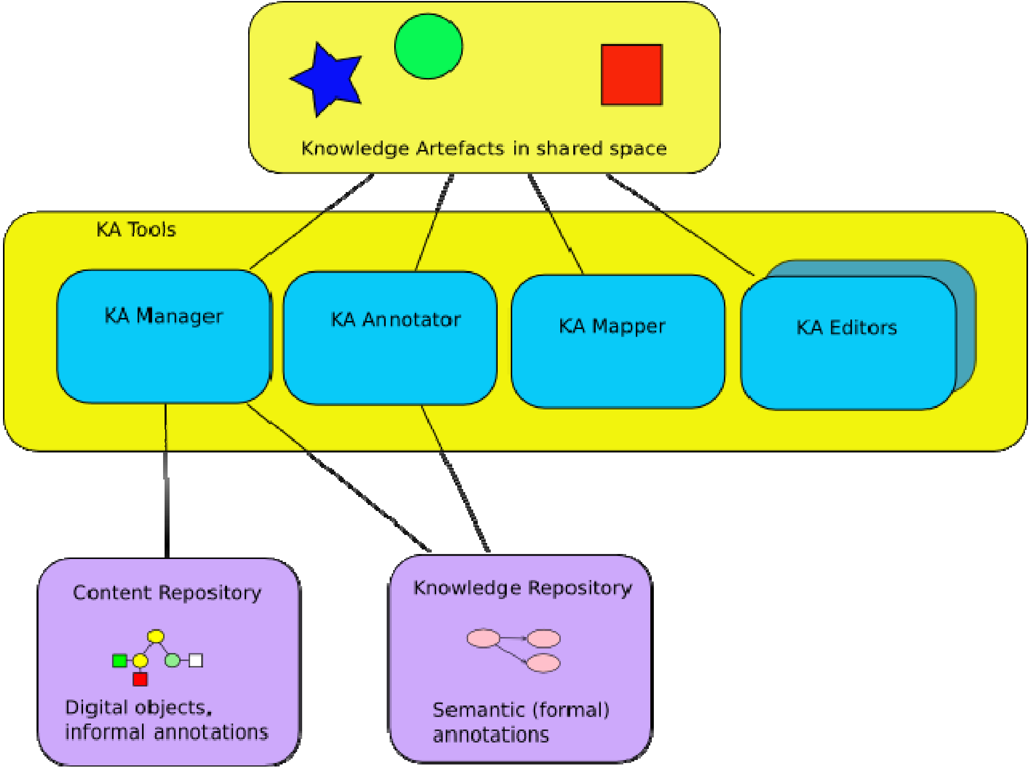
Knowledge Artefacts in shared space
KA Tools
KA Manager
KA Annotator
KA Mapper
KA Editors
Content Repository
Digital objects, informal annotations
Knowledge Repository
Semantic (formal) annotations

Figure 3. Relationship between KA tools, repositories and artifact

4.1. Knowledge artifact management tool

The Knowledge Artifact Management tool should perform the following functions:

- Create new KA into the shared space. Import an existing file into the shared space or invoke a content editor. Add formal (semantic) annotation(s) to the content.
- Save the new KA into the persistent storage.
- Search for KA into the persistent storage by given criteria
- Retrieve a KA from the persistent storage and transfer to the shared space
- Modify the KA in the shared space
- Save/update the modified KA into the persistent storage while optionally creating a new version of the digital object stored in the Content Repository

The principal role of the Knowledge Artefact Management tool (KA Manager) is to help the user to transfer knowledge artefacts between the content repository and the shared space. Therefore it seems appropriate to adopt the familiar “file manager” user interface metaphor – two windows showing the local (shared space) and remote (content repository) KAs (Figure 4). The user can select an icon representing a KA and drag it to the other window, therefore transferring the KA content.

When looking for KAs in the content repository the user will be able to browse the repository or to search by specifying keywords. The KA manager will translate the user’s search criteria into XPath and send a query to the content repository. The search results will be listed in a separate window. After selecting an item from the search results list, the corresponding node will be shown in the Content Repository window and the user can start browsing from here. Similar search functionality will be provided for the knowledge repository as well.

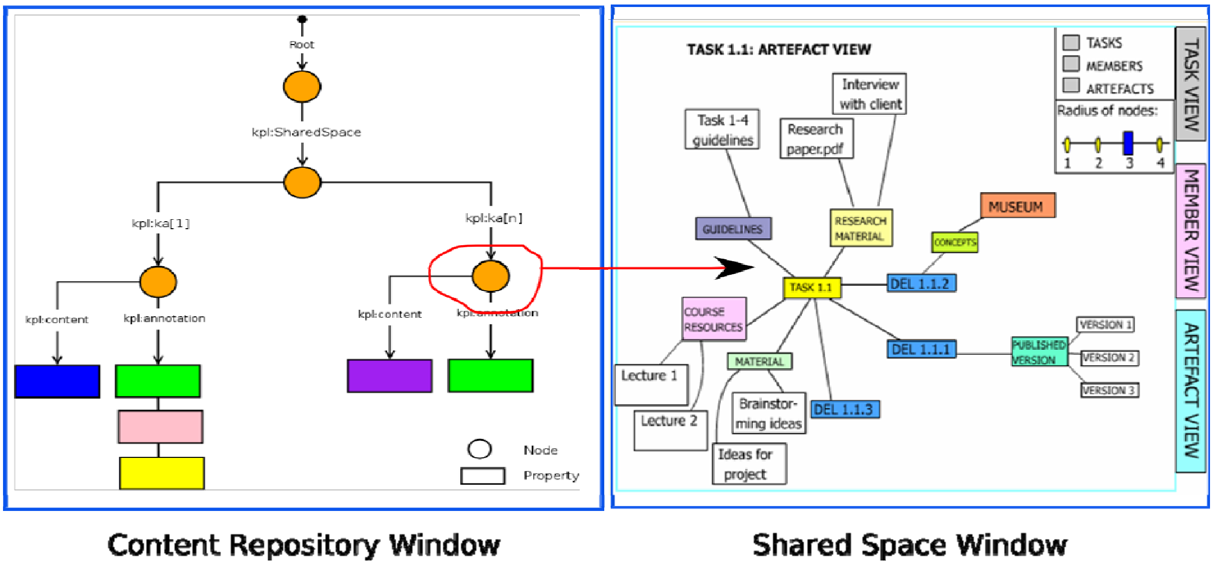

Figure 4. KA manager user interface (the shared space part of the picture is from [WP6-6]).

When transferring artefacts from shared space to the content repository the user will be able to choose where in the repository tree the artefact will be placed.

An “update” function, which will synchronize the content of the KAs between the repository and the shared space, will be provided. If the KA is modified in the shared space, than a new version of the KA will be created in the repository.

4.2 Knowledge artifact annotation tool

The Knowledge Artifact Annotation tool should perform the following functions:

- Add/update/delete formal annotations to KAs
- Add/update/delete informal, external annotations to KAs
- Add/update/delete formal annotations to tasks
- Add/update/delete informal, external annotations to tasks
- Add/update/delete informal, internal annotations to KA

4.3 Other KA tools

KA mapper is intended for mapping artifacts to background images.

Approaches for storing the mapping:

- The background image URI and the artefact coordinates (#x,y,z,t) are represented as properties in the content repository.
- The background image and artefact(s) coordinates is part of the shared space profile of the team.

The mapping between the KA and the background image will be represented as a property of the KA node. The value of this property will include the UUID of the image node and the coordinates of the KA on the image (x, y) or (x, y, z, t) at later stage.

The proposed mapping mechanism allow for multiple KAs to be mapped on the same background image, if deemed necessary.

A natural evolution of the idea of mapping to images would be to integrate Google Map services into the mapper tool for cases when the artifacts need to be geographically correlated.

KA Editors – specific tools for creating and manipulation editing content with different file formats and MIME types.

4.4 Content repository

The Content Repository can be implemented in many ways: relational database, object-oriented database, files in a directory structure, custom-made content storage facility or JSR-168 compliant Java Content Repository (JCR).

It seems that using JSR-168 compliant repository will be a good match to the requirements of the Shared Space Content Repository:

- hierarchical n-ary tree model
- metadata is supported via properties (metadata should be saved in the content or in the knowledge repository?)
- versioning
- locking
- the nodes can be made referenceable via UUIDs
- notification on changes
- searching witch XPath
- open source implementations are available (Jackrabbit, eXo JCR, Jeiceira)

Technologies

The KA tools will be implemented as JSR-168 portlets and deployed on the same portal / portlet container together with the rest of Shared Space tools.

JSF and JSP (or facelets) will be used to implement the portlets.

5. Conclusion

This paper provides a preliminary overview on the architecture of the KP-Lab shared space and on the Knowledge Artefact tools in particular. The architecture is based on emerging technologies, such as semantic web, real-time multimedia communication, ubiquitous access using wireless devices, and inter organizational computing. When ready, hopefully the KP-Lab shared space will provide users with a useful knowledge-creation tool.

6. References

[1] KP-Lab web site http://www.kp-lab.org/

[2] Eyal Oren1 et al., Annotation and Navigation in Semantic Wikis, http://eyaloren.org/pubs/semwiki2006.pdf

[3] Catherine C Marshall., Toward an ecology of hypertext annotation, Proceedings of the 9th ACM conference on Hypertext and Hypermedia 1998, http://www.csdl.tamu.edu/~marshall/ht98-final.pdf