Knowledge Resources – A Knowledge Management Approach for Digital Ecosystems

Thomas Kurz, Raimund Eder, and Thomas Heistracher

Salzburg University of Applied Sciences, Austria, Europe
{thomas.kurz,raimund.eder,
thomas.heistracher}@fh-salzburg.ac.at

Abstract. The paper at hand presents an innovative approach for the conception and implementation of knowledge management in Digital Ecosystems. Based on a reflection of Digital Ecosystem research of the past years, an architecture is outlined which utilizes *Knowledge Resources* as the central and simplest entities of knowledge transfer. After the discussion of the related conception, the result of a first prototypical implementation is described that helps the transformation of implicit knowledge to explicit knowledge for wide use.

Keywords: Digital Ecosystems; Knowledge Management, Knowledge Resources.

1 Introduction

Starting with a very ambitious vision of Digital Ecosystems or even Digital Business Ecosystems, a couple of projects were initiated in the past and the respective results were evaluated and utilized by several regions around the world.

The status of the Internet as an information medium is subject to permanent change and evolution. Knowledge is no longer maintained by single experts but it is increasingly provided and supported by communities, which gives everyone the chance to publish knowledge on all kinds of different platforms.

This paper on the conception and prototypical implementation of Knowledge Resources is structured in two main parts. It starts with a reflection of Digital Ecosystems research and deals with the extension of service-focussed research to knowledge-focussed research guided by a general-purpose vision of a knowledge management approach. In addition, the idea of Knowledge Resources (KR) as the central entity in such a system is presented. The second part of the paper explains how such a knowledge management approach could be implemented based on an open source Web Content Management System.

2 Lessons Learned from the Digital Business Ecosystem Project

As the Digital Business Ecosystem (DBE) paradigm was a radically new vision a few years ago, there was no specific background on which the related research could

F.A. Basile Colugnati et al. (Eds.): OPAALS 2010, LNICST 67, pp. 131–145, 2010.

be built upon. Therefore, researchers very often faced the well- known chicken-or-egg causality dilemma. For example, if simulations, models and social analyses were required, a broad adopters community would be necessary but did not yet exist. In addition, real-world data would be needed for designing a proper model of digital ecosystems. On the other hand, a community was initiated in parallel from scratch and this community already was in need of some models and simulation results for setting up the required infrastructure accordingly.

For example it was recognized, that the work on a general-purpose simulation and the work on visualization capabilities of the infrastructure can dramatically improve the mutual understanding between different participating disciplines. Researchers can focus on their field of study and start with approximated data first and then – as the community grows – this data can be replaced by more realistic data from real life scenarios [1],[2].

One of the biggest challenges from the beginning on was the assertion of availability of a critical mass of services and a critical number of SMEs for running a *healthy* Digital Ecosystem [17]. This was accompanied by the discussion about definition of a *service* in a DBE. By widening the term *service* to *real- world services* as well as software services coming from every possible domain, the problem arose to have no benefit for a SME in the software sector finding a service for cleaning hotel rooms, for example. Therefore, the focus of the DBE project (http://www.digital-ecosystem.org) was mainly put onto the tourism sector and onto already built software components and modelling tools in the beginning. Nevertheless, the question of the critical mass for a Digital Ecosystem was left. First and foremost, research on scale-free-networks in general, P2P systems, and search algorithms for finding distributed services produced first results, specifically in the context of huge networks with many participants.

For the search of services, the use of genetic algorithms was suggested by DE researchers (see [3], [4]). However, genetic algorithms only make sense if there is a very large search space where users try to find the best service or service combination available. Optimizing algorithms and developing frameworks for large-scale and distributed networks bases on the assumption that such a network is available and the operational community is willing to contribute to such a network. Moreover, the participants need to publish their services in the Digital Ecosystem infrastructure, which is not a straight-forward procedure compared to other ways of deployment in the IT sector, having in mind the diversity of standards, approaches and technologies, unmentionable the individual solutions SMEs often use.

Publishing services in a Digital Ecosystems also raised the issue of identity and trust (see [5]). The present conception of an open knowledge space (OKS) has to deal with privacy and trust issues in parallel to the goal of an open, collaborative and distributed framework idea. Each user should have the full arbitrament about whom he wants to open access to knowledge. Simulating the benefits for SMEs to open information for others and demonstrating thereby the market benefits out of sharing information to them will be one success criteria for enforcing the idea of a really *open* knowledge space (see http://www.opaals-oks.eu).

3 Shift from IT Services to Knowledge Resources

It is expected from Networks of Excellence, such as the OPAALS project (see http://www.opaals.org) to show new paths for the future rather than to just apply existing technology. Having a totally new field of research like Digital Ecosystems at hand, the challenge herein is to create new knowledge and, in parallel, to set up the infrastructure so that people can get involved in the related community. Consequently, the idea behind an OKS has to be novel, simple to apply and very attractive to become a valid contribution, as there are hundreds of competitive *web 2.0* applications for knowledge management and social interaction.

Services in the DBE were modelled, for example, with the DBE Studio (see [6]) and consequently implemented as services on top of the ServENT infrastructure (see [7]). The ServENT is a P2P application container that abstracts the communication functionality providing a number of methods for lookup and service invocations. Such services usually are similar to standard web services and exhibit a DE-specific XML-based interface description as well as an optional business model for describing the service operations itself.

Following the development of DE research over the last years, it can be recognized that the initially business- and IT-centric *service* focus is now broadening toward Knowledge Services (KS) or Knowledge Resources (KR). We are not longer speaking about pure business or IT but rather about knowledge as such. Consequently, also the types of services need a modification to change from IT services to Knowledge Resources. But what is a Knowledge Resource? We define Knowledge Resources as follows:

A Knowledge Resource is any kind of knowledge a person possesses for private, shared or public use.

The most important point in this definition is the fact that we see knowledge always related to a person. Every Knowledge Resource has a holder or owner which holds the context of the information. The fact that communication with this holder or owner of the service is enabled by the system and he or she can extend the data with his or her know-how, the pure information at one peer qualifies for an upgrade toward a Knowledge Resource.

Hence in the simplest form, Knowledge Resources can be contact information like email addresses, bibliography entries for helpful references or web pages with useful information. More advanced services could be documents or reports, media files or pictures, and the upper end of complexity constitute software components, tools or code archives. Every item which can be stored as a file or archive on a local entity (storage media) can represent a Knowledge Resource.

In order to find services and to have a common format for services in the knowledge space, we suggest that each service is represented by an URI and a meta-description. The URI points to a local hard disc or a web space and the meta-description is provided as plain text. The variants of textual description are dealt with later on in this paper.

4 Where Knowledge Lives – A Vision of an Open Knowledge Space

In the following, the context and the basic requirements for a Knowledge Resource Framework are briefly outlined. In order to make the idea more demonstrative, the first-person narrative is used in the following. From my daily emails, on to my data on local discs and data I can access via the Internet, the amount of electronically accessible data gets bigger and bigger each day. Nevertheless, there is data I want to share with others and there is data I want to use for my private everyday life. For handling this data I have hundreds of possibilities for storing, structuring and retrieving. I personally often wanted to try out tagging but although there are many tools for tagging no one fits all my needs. The idea of Knowledge Resources should enable me to tag whatever I have on my machine or whatever site I have visited remotely. I just want an easy-to- handle web-application which lets me insert the tags or text I want to put as a meta-description and a URI for referring to the Knowledge Resources itself. For searching I want to have a text entry field (like in the Goggle search) for searching my private knowledge space *and* the open knowledge space. In order to fulfil these application requirements, the complexity of the data storage and the search has to be hidden behind a very simplistic user front end.

The minimum requirement for a Knowledge Resource is a tag and an URI. All meta-descriptions of Knowledge Resources are in text-form and are published to the user community. The user can decide how much of his meta-descriptions should be published As the link can point to a private space on a hard disc or a restricted area, access does not need to be limited by the OKS. Consequently, the public meta-information of all services is available and also the URI indicates sometimes which type of information can be found although the direct access is still limited. Nevertheless, there should be the possibility to send a message to the owner of a specific knowledge, to open access for the demanding party. The decision of opening the knowledge is again up to the owner of the knowledge.

In order to avoid ambiguity, the editing tool for the meta-information of the knowledge services needs references to a vocabulary or term definition repository. More detailed information on the editor can be found later on in this paper. Nevertheless, additional complexity like database connections and vocabulary should be hidden from the standard user.

The application areas for such a system are manifold, ranging from the pure private usage for organizing the local music folder up to a fully distributed search for commercial and free software components. We can also think of extending the search with automatic composition capabilities in future but at the moment the implementation described further below is merely for single service search.

What is the reason for setting up another search engine? The point is that the present framework is much more than an information search engine like Google and it is not necessarily governed by one company which reuses and sells this data.

The services can be both, private and public knowledge, and not just information which can be found in the Internet. Furthermore, the type of services outperforms in diversity the kind of information which can be found by a common search engine. Additionally, the introduction of visualisation features for the search results outlined later on will enhance the usability and navigation through search results considerably.

5 Preliminary Conception

Before we start to outline the web-based application for a Knowledge Resource framework, a few notes on the required infrastructure need to be provided. The basis for such an infrastructure system can only be a P2P network. A pure P2P network is governed by the community and it is the only convincing architecture model for setting up a highly collaborative framework for knowledge sharing. Other approaches could easily lead to centralization again. The simplest way of such a P2P node could be an SME with one computer including Internet access. It allows a secured part of its local information to be used by the system and thus shares data. The editor for inserting the URI / meta-description pairs is a web- application. As a company may decide to internally use the Knowledge Resource framework, it can put also an application server as a new node in the network and connect it to the internal network to that application- or web-server node in the OKS network. As long as there is the option of installing a lightweight P2P service for the connections, the decision which application server or which back- end technology is used should be up to the user. URI / meta-description pairs can be stored either file-based on the shared memory space or in a distributed database. Note that the decisions on the technological details are not relevant for this paper.

As already mentioned, the front end of the system should be an easy-to-use editor. The minimal idea is to have just three text edit fields: (1) the editor window where the meta-description of the service can be typed in, (2) a text- field for inserting the URI pointing to the service itself, whether on the local disc or on a shared place in the Internet, and (3) a search window for finding services similar to a common search engine.

The editor window allows entering tags in form of plain text or text-based notations like the Semantics of Business Vocabulary and Business Rules (SBVR) (see [8]). SBVR is an adopted standard of the Object Management Group (OMG) intended to be the basis for formal and detailed natural language declarative description of a complex entity and is chosen here as one possibility of a text-based but logical founded notation. Utility of SBVR was also investigated in connection with DE research, documented for example in [9]. If the user starts to type in a tag, a type assistant should superimpose in order to access already defined terms. That avoids ambiguity and the user can check if there is already a definition of a tag available. In order to store such definitions, we would suggest including a connection to a community repository for example. The repository could include

SBVR statements and definitions which are easy to read and therefore the user gets familiar with the syntax of SBVR even if he is just tagging the services. An alternative here would be a community ontology or set of ontologies. If there is no unambiguous definition available in the repository, the user should have the option to add a new definition. As such definitions can be formal (in SBVR syntax) and informal (plain text for example), it is up to the user to define terms in the way he or she wants.

Besides tagging, SBVR statements can also be inserted for describing services. We can also think of setting up configuration parameters for services as meta-descriptions in SBVR. The concrete software service identified an URI can then be configured automatically by parsing the SBVR meta-description.

The last option for inserting meta-descriptions for Knowledge Resource is natural language. For example, one could implement a small tool which automatically parses papers from the local disc and puts the abstracts as meta description in the editor including the URI path to the paper itself.

As an extra feature for natural language definitions, a transformation from natural language into SBVR could be offered (see [10], [11]). Using tools like the Stanford-parser, SBVR fact-types could be identified in the natural language text and listed for the user. A SBVR fact-type is a concept that conveys the meaning of a verb phrase that involves one or more noun concepts and whose instances are all actualities [8].

Through a simple manual check of the automatically transformed and respectively generated SBVR statements, the user can tell the system which transformation lead to a correct statement and which did not. Introducing a learning system here could feedback the manual corrections and therefore influence and enhance the next transformations.

The second prominent part in the standard user-interface is the search functionality. The first and simplest search could be implemented as a mere keyword search. Understanding whole SBVR expressions or *meanings* of full sentences in the search window would be a very interesting and challenging research question as such and could be an extension of the system for future releases.

Since the descriptions of Knowledge Resources are more meaningful than the contents of plain web pages, the results can be visualised as clouds of clustered results. Here is a short example for better illustration: The user wants to learn more on Genetic Algorithms (GAs) and types in *genetic algorithm* in the search window. The result is a visualization of clustered clouds of links or keywords: One cloud for contacts of people familiar with GAs, one other cloud for useful literature entries, another one for GA implementations and one for tools using GAs. The user has the possibility to click deeper into the implementation cloud, which zooms in and shows sub-categories like implementations in different programming languages for example. Using an intuitive and dynamic navigation, the user can browse through the Knowledge Resources found in the OKS. Then he can either access services directly via the URI or ask for further information by dropping a message to the holder /owner of the Knowledge Resource.

6 Relevance to the OPAALS Network of Excellence

Before we introduce a prototype of a Knowledge Resource based knowledge space, we want to sum up and stress the different ideas in the context of OPAALS. As the idea of Knowledge Resources arose from the discussion about the OPAALS Network of Excellence, it is straight-forward to see connection points to specific DE research domains. Furthermore, we always have in mind the current success of social networking tools of various types and the critical mass of services necessary for the OKS. Amongst others, we see here the following connection points to research questions in OPAALS:

– Tagging - SBVR - natural language
– Natural language evolution
– Evolutionary aspects - Simulations
– Social network analysis
– Advanced visualisation concepts
– Automated code generation
– Tools for automatic input, export or transformation
– Transactions and workflows

In the following we address these points separately.

First, the editor component should process *tags*, SBVR and *natural language* which leads to a natural convergence of the descriptive notations. Utilizing type assistants makes the user implicitly familiar with new notations and helps entering the first meta-descriptions. The community builds up a common vocabulary which becomes more and more advanced and structured in a better way.

The analysis of term definitions as well as the continuous *change in the language* used for describing services has a clear potential for research on the evolution of language. Besides the recognition of specifics for different domains, the open way of including also non-technical services and personal interests like music or photos could lead to a larger community and therefore more data for language analysis.

Data for the performed *simulations* originate from social analyses and questionnaires. Large-scale social network behaviour is most often estimated and extrapolated. With a larger user group, the networks and behaviour of the users can be modelled more precisely and therefore the outcomes of simulations will be more accurate.

Beside the usage in OPAALS and the focus on SMEs, other stakeholders could become interested in the *analysis of the DE social networks* in an Open Knowledge Space. Social science researchers can check their experience with existing social networks with a more structured and evolutionary approach in the OPAALS context. Moreover, the introduction of new volunteers in using the system and in contributing in one way or the other can ease the challenge of sustainability.

The *visualization and animation* of the search results opens a complete field of dynamic visualization options. Not only structure and static dependencies can be shown but the dynamic part of user interaction can make the work for the visualization research more interesting and challenging. It will become a new experience for the users to get structured clouds of community knowledge instead of a simple list of links.

Although, the move from IT-Services to Knowledge Resources means a change in the types of services under observation; Knowledge Resources still can be IT-Services or be represented by software components. The meta-description can either be a summary of the specification or a more detailed description of the business model in SBVR or even a full specification in SBVR. Here we can think of SBVR descriptions which are the basis for a *transformation tool*, transforming the SBVR model into a Grails web-application for example (see [9]). As people become interested in structured specification writing, more of these tools can be developed to add functionality to the Knowledge Resource Frame- work. Analogue to the Grails transformation, tools for *transactions and workflow generation* could be included as well.

To recapitulate, the ideas stated here can provide the basis of a prototypical implementation of a Knowledge Resource application, outlined in the following section. It partially implements the ideas stated so far in a Typo3 open source Web Content Management System (WCMS).

7 Prototypical Implementation

The average information content per page usually depends directly on the number of registered users and their activities. The more active users are, the more information is provided on a page. Therefore, the conception of a web-frontend is a crucial point of the implementation and the envisaged future success. For the first prototype, distributing the whole system is foreseen in the architecture, but it is not implemented in the initial stage. The following points are already carried out in the prototype and described in this section:

- Utilization of web 2.0 technologies and frameworks
- Frontend features like multilingualism and search engine optimization
- Insert, edit, delete and maintenance
- Reward system for placing more knowledge and additional functionalities in the system
- Easy navigation in the web-based system
- Geographical visualization of users and KR

In order to provide these functionalities in a self-explanatory way, Henrik Arndt published a comprehensive approach in [12]. In the centre of consideration is the user of a web page who is surrounded by the context and added-value of the page. In order to reach these surrounding requirements, the page has to fulfill the following aspects:

1. Usability: The two major points here are *traceability* and the ability of *self-description*. A user should have at any time the ability to know, where he/she is and how he/she can access whatever functionality.

2. Utility: Two important points here are relevance and being up-to-date. The page has to fulfil the personal requirements of the user and the user needs to find up-to-date knowledge.
3. Joy of Use: Static pages and complicated navigation reduces the attractiveness of web pages for end-users. The most popular Internet pages stand out because of innovation and the option of personalization, e.g. such has Facebook.

A page might have bug-free source and high-end database connections, but the user experience acts on another level. For the user, these three criteria are important. According to Andt, the decision whether the page will be visited again or not happens within the first 10 seconds. Therefore, even for the prototype, the focus is on easy usability, e.g. the positioning of items and the layout.

These requirements for the first prototype represent a considerable part of the conception of Knowledge Resources that is outlined in this paper. In the following subsections, we document how this implementation was done.

7.1 Technologies

For the implementation of the knowledge management prototype described in this section, we chose Typo3 (see http://typo3.org), an open source Web Content Management System (WCMS). It provides a full editor system, which allows easy maintenance of the page content. Although Typo3 is often cited as the most used WCMS, there are no official numbers available to support these statements. From the viewpoint of a developer, Typo3 is more of a framework than a WCMS. Contrary to other solutions, it needs to go through a few basic steps after installation in order to get a *hello world* page. On the contrary, it allows plugging in an enormous number of already-existing extensions and it supports the development of new extensions, like the knowledge management system for Knowledge Resources. Therefore we facilitated these extension capabilities for the prototype implementation. As there is extensive literature for Typo3, we refer here to the literature such as [13] or for the programming of the backend in PHP to [14] as well as to [15] for more detailed information.

For the implementation of AJAX functionalities, the Typo3 extension xajax was used. It inserts a PHP-AJAX framework. This extension is used to add AJAX functionalities also for other extensions of the prototype [16].

7.2 Frontend Features

For the optimization of the indexing for search engines, the existing Typo3 extension *realurl* is used. Typo3 assigns ID numbers for the single pages in the WCMS, which can be accessed with *realurl* also via text-based URLs. As a side effect, this enhances the ratings in Google searches for example.

The extensions *rgmediaimages* and *rgmediaimagesttnews* serve as additional add-ons for content elements. These allow the playback of videos in content elements. It is implemented by the facilitation of an open source media player.

Additionally, Youtube videos can be viewed directly inside the page. By the utilization of the JavaScript library *SWFObject* in the *rb_flashobject* extension, Flash movies can be included via direct back-end access.

Additionally, the following existing frontend extensions are used for the prototype documented here: *felogin* (blugins), *kontakt* (contact form), *tt_ address* (addresses), *tt_news* (news), *Static info Tables* (countries and currencies), *Static Methods for Extensions* (registration), and *templavoila* (templates).

For better user experience, four Typo3 extensions were adapted in order to fit with the needs of Knowledge Resources.

First, editors should not see passwords in plain text. This is implemented in *kb_md5fepw*. It ciphers the password before it gets inserted in a MD5 hash and is added to the back-end database. Here an adaptation to a W3C conforming code validation was necessary.

Second, the extension *perfectlightbox* was used for zooming of pictures. By darkening the screen and zooming the picture in the middle of the screen, the picture can be viewed much better because of the higher contrast. A small adaptation was needed to be compliant with the newest version of the other plugins.

Third, *sr_feuser_register* was utilised for the registration process of new users. It comes with support for all common registration steps including changing forms for user data. All registered users can be administered in the back-end. The personal factsheet and the assignment of longitude and latitude for geographical information was added to the standard functionalities of this extension.

Forth and finally, the assignment of tags as meta-descriptions for Knowledge Resources should be expressed as an animation of a tag cloud. Therefore, *t3m_cumulus_tagcloud* was extended so that users can not only add tags via the administrator in the back-end, but also in the frontend, when wishing to add a Knowledge Resource.

7.3 Data Insertion, Editing, Deletion and Maintenance

As soon as a user registers in the system, the system provides a personalized factsheet. This user profile information includes at least the registration information username, email address and country of origin. Additionally, a user photo can be added.

For inserting Knowledge Resources, the user gets an inserting form for the Knowledge Resource including title, textual description, a link to a file or web- page, optionally a file on the server and a set of tags. This information represents an extended version of the initial Knowledge Resource idea of textual description and URI. The reason for the additional fields for title, server file and tags is on one hand the familiarity of users with assigning titles and tags as well as the possibility of an easy bootstrap by providing server space for the early adopters. With an additional implementation effort, the title or tags can be incorporated in the textual description.

Both the user profile and the Knowledge Resources can be added in another template. The edit templates are implemented with AJAX in order to provide better usability.

Additionally to the existing Knowledge Resources, users can add so-called KR requests. This allows distributing needs for implementation, knowledge or services. Each KR request can be accessed by other users. When they open the details page of the KR, the system tracks the user ID and acts as observer of this KR.

Finally, also a search for KR is provided. At the moment this search is only implemented as a keyword search. A search can be started by registered and unregistered users. The search provides a free text search. Consequently, the text is searched in title, description and the assigned tags. All hits are listed then as search result. Similar to common search engines, during insertion of the search string, an AJAX-auto-completion helps the user with the search. Additionally to the free text search, a drop-down list with the available tags supports the user in parallel to a direct search functionality based on an auto-generated tag-cloud.

7.4 Reward System

In order to make it more attractive for the users to add more personal data, a reward system was implemented. As people might be deterred by needing to register at the first stage, a first level without registration was implemented too. In the following, the three access levels for the knowledge management prototype are listed:

Level 1 Frontend user without registration:

- Can use search functionality for KRs
- Can view all information, tags, SBVR or language, for every KR
- Cannot search for KR requests
- Cannot download KR attachments

Level 2 Frontend user registered and logged in with at least username, email address and location (city):

- Can search for KR requests
- Can download KR attachments
- Can add a personal profile and add his/her own KRs
- Cannot add files to a KR

Level 3 Frontend user with more than four registered KRs:

- Can add a maximum of 50 files to KRs with a maximum size of 2 MB, each

Level 3 is especially interesting for users without own web space who want to share knowledge with their colleagues or friends. With the restriction in file size and number of file uploads, a misuse is partially prevented for the prototype. Later releases will need more sophisticated mechanisms preventing possible fraud. Moreover, administration of the user data, tags and knowledge resources is possible via the Typo3 backend system.

7.5 Navigation

The navigation of the web-frontend of the prototype is a novelty for web pages. The basic idea is to imitate a keypad of a telephone. Consequently the navigation and access of certain pages can be done by dialing a number. The numbers can be dialed by mouse-click or by a shortcut on the keyboard.

Fig. 1. Web-based frontend for the knowledge management prototype for Knowledge Resources

As can be seen in Fig. 1, the nine pads can be filled with all types of content. Starting with pure text or abstracts for the pages, up to pictures or flash animations can be inserted. The prototype can be navigated by the 3x3 matrix of the keypad OR alternatively also by a traditional WCMS menu on top of the page. The benefit of this navigation approach is the fast and direct access of pages for professional users in parallel with a step-by-step guidance for beginners. With three levels of a 3x3 navigation matrix, 9^3 (729) pages can be accessed with 3 or less clicks or key strokes.

7.6 Geographical Visualisation

The knowledge management prototype for Knowledge Resources includes two visualizations (see Fig. 2). First, a user is able to view all registered users locations on a map. Here a simple click on the position on the map triggers popping up of the basic user data and the registered KRs. This permits a simple search of users in the surrounding area. Second, a single Knowledge Resource can be visualized on a map, including the visualization of all related observers. With this simple feature, a company can for example visualize the potential customers of a resource, i.e. a market for a product or service.

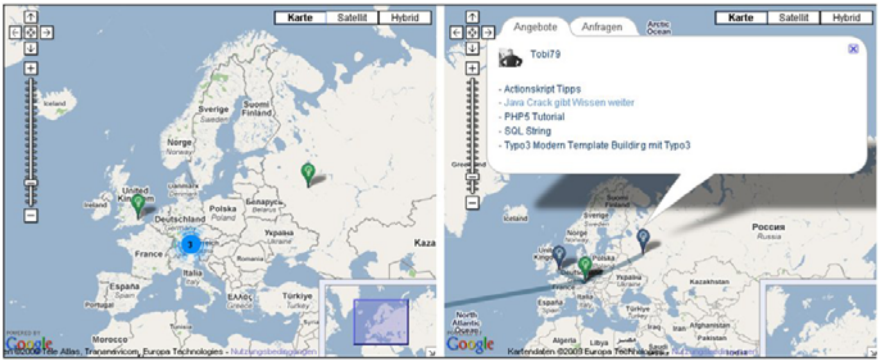

Fig. 2. Visualisation of the Google maps based visualisation of registered users (left) and observers of an item (right)

The necessary latitudes and longitudes of all users are stored in the database of the Typo3 system. For showing connections between the observers and the item for example, polylines are used. After creation each item on the map is added by the system automatically via an overlay. If dense clusters of users are to be visualized within a region or the user zooms out on the map, the single items or users are visualized as clusters. While zooming further in again, the single elements of the cluster appear again as individual items on the map.

8 Conclusion and Outlook

In this paper a knowledge management approach in a Digital Ecosystem context was presented. Beside the broad conception of the system and the recommendation of Knowledge Services as the central entities in the system, a prototype was developed based on the open source Typo3 WCMS.

The implementation of the knowledge management prototype was focused on the transformation of implicit knowledge to explicit knowledge and on the ease of use for a broad user base. Users can easily publish and/or offer their knowledge and take advantage of a web-based system without more requirements than a simple browser.

As the focus at the moment is on a functioning system with easy accessibility for all kinds of potential users, some of the requirements like complete distribution of the system could not be implemented yet. In the following, a few potential future extensions toward the full implementation of a knowledge management system for digital ecosystems are outlined.

First, the distribution of the system is a core need toward a full implementation of the knowledge management approach outlined in the first sections of this paper. The next step for distributing the system would be a fully distributed file-system and database for the Typo3 (sub)system. We assume that at least for the next release some reliable rich web-based user interfaces would be necessary. Later on, the complexity of the user interface could be reduced and could be provided also for smaller clients and even mobile phones.

Moreover, currently a desktop application is developed for automatic upload of Knowledge Resources to the WCMS. It allows users to easily register their resources in the database of the WCMS. This desktop application features an automatic parsing of paper abstracts and an upload option for descriptions. Moreover an automated tagging mechanism will be introduced as well.

According to a request of some medium-sized enterprises, which want to use the prototype for their intranets, also a versioning add-on is currently developed. This add-on will connect each Knowledge Resource with a state-of-the-art versioning system.

Acknowledgments

This work was funded in part by the European Union's 6th Framework Program in the OPAALS Network of Excellence, contract number IST-034744.

References

1. Nachira, F., Nicolai, A., Dini, P., Le Louran, M., Rivera Leon, L.: European Commission - Information Society and Media, Digital Business Ecosystems (2007)
2. Kurz, T., Eder, R., Heistracher, T.J.: Multi-Agent Simulation Framework fr Interdisciplinary Research in Digital Ecosystems. In: Proceedings of International Conference on Digital Ecosystems and Technologies, IEEE DEST (2009)
3. Rowe, J.E.: Foundations of Learning Classifier Systems. Springer, Heidelberg (2005)
4. Briscoe, G.: Digital ecosystems: Evolving service-oriented architectures. In: IEEE First International Conference on Bio Inspired mOdels of NETwork, Information and Computing Systems (2006)
5. McLaughlin, M., Malone, P., Jennings, B.: A Model for identity in digital ecosystems. In: Proceedings of International Conference on Digital Ecosystems and Technologies, IEEE DEST (2009)
6. The DBE Consortium. DBE Studio, `http://dbestudio.sourceforge.net/` (last accessed, 01.10.2009)
7. The DBE Consortium. DBE Servent, `http://swallow.sourceforge.net/` (last accessed, 01.10.2009)
8. OMG. Semantics of Business Vocabulary and Business Rules Specification, Version 1.0, `http://www.omg.org/spec/SBVR/L0/` (January 2008)
9. Eder, R., Filieri, A., Kurz, T., Heistracher, T.J., Pezzuto, M.: Model-transformation-based Software Generation utilizing Natural Language Notations. In: Proceedings of International Conference on Digital Ecosystems and Technologies, IEEE DEST (2008)
10. Eder, R., Kurz, T., Heistracher, T.J., Filieri, A., Russo, M., et al.: D2.2 - Automatic code structure and workflow generation from natural language models, OPAALS Project (April 2008)
11. Eder, R., Kurz, T., Heistracher, T., Filieri, A., Margarito, A.: D2.6 - Prototype of web-based User Interface with interfaces to existing SBVR tools, OPAALS Project (May 2009)
12. Arndt, H.: Integrierte Informationsarchitektur Die erfolgreiche Konzeption professioneller Websites. Springer, Heidelberg (2006)

13. Laborenz, K., Wendt, T., Ertel, A., Dussoye, P., Hinz, E.: Typo3 4.0 - Das Handbuch fr Entwickler Galileo Press (2006)
14. Kannengiesser, M.: Objektorientierte Programmierung mit PHP 5, 1st edn., Franzis Professional Series (2007)
15. Flanagan, D.: Mootools Essentials. Apress (2008)
16. White, J., Wilson, M.: xajax, http://xajaxproject.org (last accessed, 10.12.2009)
17. Kurz, T., Heistracher, T.J.: Simulation of a Self-Optimising Digital Ecosystemp. In: Proceedings of International Conference on Digital Ecosystems and Technologies, IEEE DEST (2007)