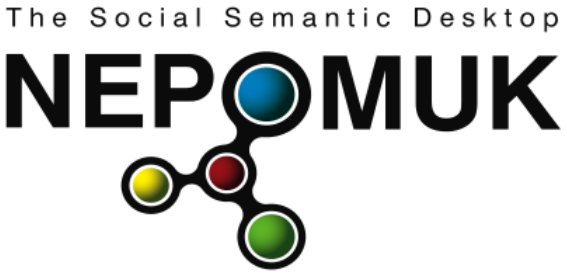
Integrated Project

Priority 2.4.7

Semantic based knowledge systems



Mandriva Community Case Study
First Prototype of
A Social Semantic Help Desk

Deliverable D11.2

Version 1.0

15.10.2007

Dissemination level: PU

Nature	P
Due date	30.09.2007
Lead contractor	EDGE-IT S.A.R.L
Start date of project	01.01.2006
Duration	36 months

Authors

David Barth, EDGE-IT S.A.R.L
Arnaud Laprévote, EDGE-IT S.A.R.L
Stéphane Laurière, XPertNet
Fabio Mancinelli, XPertNet

Mentors

Ansgar Bernardi, DFKI
Robert Jaeschke, L3S

Project Co-ordinator

Dr. Ansgar Bernardi
German Research Center for Artificial Intelligence (DFKI) GmbH
Trippstadter Str. 122
D 67663 Kaiserslautern
Germany
Email: bernardi@dfki.uni-kl.de, phone: +49 631 205 75 105

Partners

DEUTSCHES FORSCHUNGSZENTRUM F. KUENSTLICHE INTELLIGENZ GMBH
IBM IRELAND PRODUCT DISTRIBUTION LIMITED
SAP AG
HEWLETT PACKARD GALWAY LTD
THALES S.A.
PRC GROUP - THE MANAGEMENT HOUSE S.A.
EDGE-IT S.A.R.L
COGNIUM SYSTEMS S.A.
NATIONAL UNIVERSITY OF IRELAND, GALWAY
ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE
FORSCHUNGSZENTRUM INFORMATIK AN DER UNIVERSITAET KARLSRUHE
UNIVERSITAET HANNOVER
INSTITUTE OF COMMUNICATION AND COMPUTER SYSTEMS
KUNGLIGA TEKNISKA HOEGSKOLAN
UNIVERSITA DELLA SVIZZERA ITALIANA
IRION MANAGEMENT CONSULTING GMBH

Versions

Version	Date	Reason
0.1	04.08.2007	First draft by S. Laurière
0.2	04.09.2007	Input from A. Laprévote
0.3	01.10.2007	Version 0.3 by S. Laurière and F. Mancinelli
1.0	15.10.2007	Remarks from the mentors taken into account; final streamlining by IMC

Explanations of abbreviations on front page

Nature
R: Report
P: Prototype
R/P: Report and Prototype
O: Other

Dissemination level
PU: Public
PP: Restricted to other FP6 participants
RE: Restricted to specified group
CO: Confidential, only for Nepomuk partners

Executive summary

The objective of WP11000 is the adoption, application and validation of the services provided by the NEPOMUK[1] platform in the context of the on-line community of Mandriva Linux users[2]. WP11000 is designing a prototype providing the members of this community with a new generation tool for sharing knowledge related to the Mandriva Linux project.

Large open-source projects use and produce huge amounts of semi-structured data exchanged through numerous channels such as wiki pages, mailing-lists, forums, issue tracking reports, requirement specifications, task descriptions, commented source code, instant messages etc. This massive amount of information is constantly activated as a resource for knowledge which helps the project partakers to solve existing problems collectively, and which sustains the collective innovation process at stake in the design of the next versions of the projects. This collective process has led to impressive results with relatively simple knowledge management tools, i.e. wiki engines, mailing-list servers, source control patch/diff systems and instant messengers. The Linux eco-system is one of the result of this process, consisting in 2007 of millions of users worldwide, thousands of contributors and of companies doing business around the related platform. A typical Linux user or developer has to deal with a huge amount of information in order to get the most of his system and of the project he's interested in.

By providing customizable tools for linking the mental model of the user with the collective knowledge of the community, the NEPOMUK platform applied to the context of a Linux user's activities makes the process of learning, getting help and contributing to existing knowledge easier and more efficient.

This report describes the first version of a community help desk system combined with a knowledge base engine that is being experimented in particular with the Mandriva Linux community through the Mandriva Club platform[3]. Even though the prototype focuses on help desk aspects, the approach, if successful, could be widened to all the processes involved in the collective design, use and maintenance of a software system by allowing contributors to annotate not only the questions and answers about the system, but all artefacts of the software forge: source code, bugs, specifications, tests etc. In that sense, the NEPOMUK platform is likely to be of interest in the emerging domain of Semantic Web Software Engineering.

As of October 2007, a first version of the WP11000 prototype has been designed and brought on-line as an additional service in alpha stage of the Mandriva Club at the following URL: `http://club.mandriva.com/xwiki/bin/helpdesk/`. This prototype features a tight integration with the NEPOMUK-Eclipse desktop prototype[4]. The system builds on the one hand on NEPOMUK components and on the other hand on XWiki components. XWiki[5] is an open-source extended wiki engine and a generic platform for collaborative Web applications. The prototype's code name is currently `XWITS`, standing for `XWiki Issue Tracking System`: help desks and issue trackers share many common concerns indeed, and the designed system may be experimented later on as a semantic issue tracker.

The first version of the WP11000 community help desk lets the users publish questions and answers and add detailed metadata to them so that the information search and discovery processes get significantly improved compared

[1] `http://nepomuk.semanticdesktop.org`
[2] `http://wiki.mandriva.com`
[3] `http://club.mandriva.com`
[4] `http://nepomuk-eclipse.semanticdesktop.org`
[5] `http://www.xwiki.org`

to traditional help desk systems. Metadata can be added both collectively on the server side or individually on local desktops for the purpose of learning and memorizing. Users can for instance link a question and the set of related answers with some personal files, emails and bookmarks, all together related to a technical task. The established semantic graph is then meant to let him find again information efficiently when faced to a similar task or issue.

The current help desk prototype shows a solid basis which will be extended in the next months according to a development roadmap outlined in this report. Now that the assembling of NEPOMUK components with the XWiki ones has been carried out, it is expected that the next design and implementation steps will occur at a fast pace, with approximately one release per month until the end of the project. The next steps will consist of adding a P2P and a social layer to the system in addition to the semantic one through the integration of the NEPOMUK distributed index and of the NEPOMUK community manager. The semantic layer itself will be enhanced by the integration of the NEPOMUK text analysis service as well as the capability of dealing with the user's context. The help desk will be improved both on the backend and on the desktop. Beside the use of NEPOMUK-Eclipse, NEPOMUK-KDE[6] and NEPOMUK-Mozilla libraries will be put into practice for drawing further the links between a NEPOMUK desktop and the help desk backend, with the objective to create a large scale Semantic Web application tightly integrated with the users' desktops.

[6] `http://nepomuk-kde.semanticdesktop.org`

Table of contents

1 Introduction

The objective of WP11000 is the adoption, application and validation of the services provided by the NEPOMUK platform in the context of the on-line community of Mandriva Linux users. WP11000 is designing a prototype providing the members of this community with a new generation tool for sharing knowledge related to the Mandriva Linux project.

Large open-source projects use and produce huge amounts of semi-structured data exchanged through numerous channels such as wiki pages, mailing-lists, forums, issue tracking reports, requirement specifications, task descriptions, commented source code, instant messages etc. This massive amount of information is constantly activated as a resource for knowledge which helps the project partakers to solve existing problems collectively, and which sustains the collective innovation process at stake in the design of the next versions of the projects. This collective process has led to impressive results with relatively simple knowledge management tools, i.e. wiki engines, mailing-list servers, source control patch/diff systems and instant messengers. The Linux eco-system is one of the result of this process, consisting in 2007 of millions of users worldwide, thousands of contributors and of companies doing business around the related platform. A typical Linux user or developer has to deal with a huge amount of information in order to get the most of his system and of the project he's interested in.

By providing customizable tools for linking the mental model of the user with the collective knowledge of the community behind the tools he's interested in, the NEPOMUK platform applied to the context of a Linux user's activities makes the process of learning, getting help and contributing to existing knowledge easier and more efficient.

This report describes the first version of a help desk system combined with a knowledge base engine that is being experimented in particular with the Mandriva Linux community[7] through the Mandriva Club platform[8]. Even though the prototype focuses on help desk aspects, the approach, if successful, could be widened to all the processes involved in the collective design, use and maintenance of a software system by allowing contributors to annotate not only the question and answers related to the system, but all artefacts of the software forge: source code, bugs, specifications, tests etc. In that sense, the NEPOMUK platform is likely to be of interest in the emerging domain of Semantic Web Software Engineering.

As of October 2007, a first version of the WP11000 prototype has been designed and brought on-line as an additional service in alpha stage of the Mandriva Club platform at the URL `http://club.mandriva.com/xwiki/bin/helpdesk/`. This prototype features a tight integration with the NEPOMUK-Eclipse desktop prototype[9]. The system builds on the one hand on NEPOMUK components and on the other hand on XWiki components. XWiki[10] is an open-source extended wiki engine and a generic platform for collaborative web applications. The prototype's code name is currently `XWITS`, standing for `XWiki Issue Tracking System`: help desks and issue trackers share many common concerns indeed, and the designed system may be experimented later on as a semantic issue tracker.

The first version of the NEPOMUK community help desk has semantic capabilities in the sense that it takes advantage of several ontologies for structuring the contents written by community members, and has social features to the

[7] `http://wiki.mandriva.com`
[8] `http://club.mandriva.com`
[9] `http://nepomuk-eclipse.semanticdesktop.org`
[10] `http://www.xwiki.org`

extent that it provides support for establishing and harnessing a trust network across the participants. As of October 2007, the desktop integration capability focuses mainly on NEPOMUK-Eclipse, and will be extended to the other flavours of the NEPOMUK desktop – (i) general desktop instrumented by NEPOMUK technologies, (ii) NEPOMUK-KDE and (iii) NEPOMUK-Mozilla desktops – in the upcoming releases of the platform.

The report consists of the following parts:

- Section 2 describes the specification of the current prototype, its architecture and its data model;
- Section 3 depicts the main functionalities of the prototype and illustrates the implementation of some of the use cases introduced in the deliverable D11.1 [17];
- Section 4 sets out the way the first evaluation of the prototype will be conducted, and presents the assessment of the system capabilities completeness level;
- Section 5 describes the commercial opportunities that are being pursued for deriving an enterprise solution from the prototype;
- Section 6 concludes the report and introduces the actions that will be carried out in the next months for adding functionalities to the prototype through the integration of other NEPOMUK components, and for assessing the next releases of the system.

2 Prototype specification

This section presents the help desk architecture and the ontologies used for its data model.

2.1 Architecture

The cornerstones of the XWITS system are comprised of (i) the NEPOMUK components, (ii) the XWiki workspace engine, (iii) the NEPOMUK ontologies completed by domain specific ontologies. These three pieces are described in the following paragraphs.

2.1.1 Overview

XWITS consists of a set of components assembled using an OSGi [7] container. The components assembled for the first version of XWITS are represented in figure 1 with a grey background, while the components whose title is in italics will be integrated in the upcoming releases of XWITS.

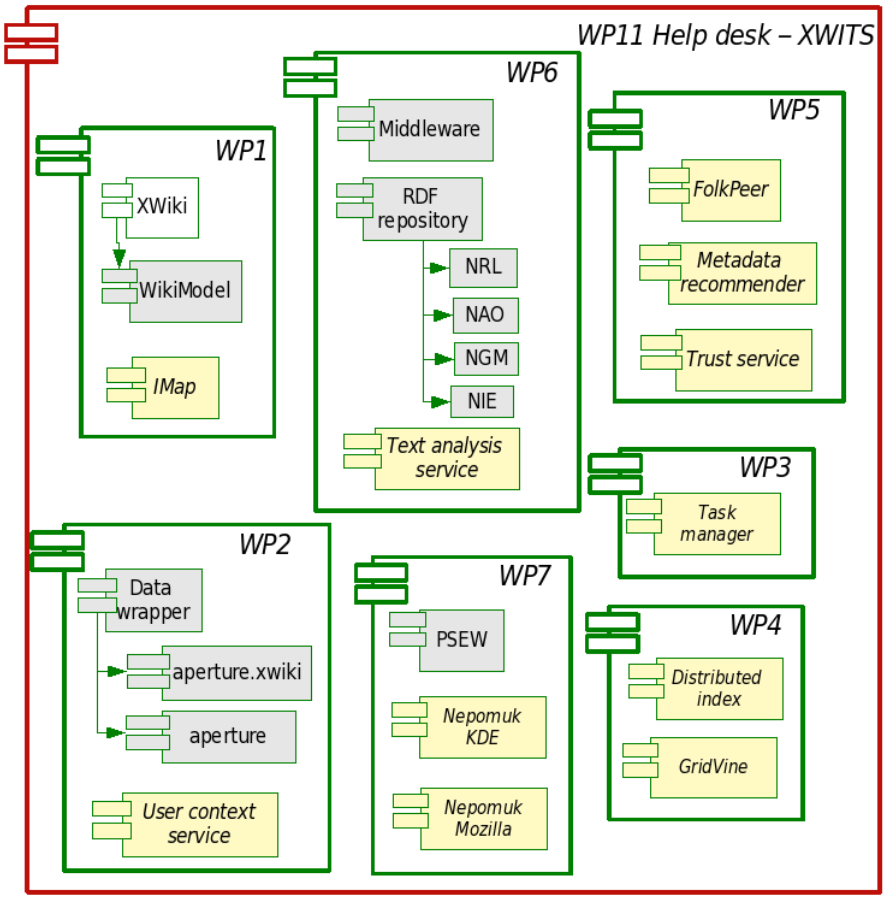


Figure 1: XWITS component diagram

2.1.2 Integration with a third-party system: XWiki

The XWiki engine is one of the assembled components. XWiki[11] is an extended wiki engine featuring a collaborative application development framework. It has been used as the Mandriva Club application engine since 2004 and was hence a natural choice for building the Mandriva NEPOMUK help desk together with the NEPOMUK components.

In order to get XWiki integrated with the NEPOMUK components, an `OSGi` wrapper has been developed for XWiki. By doing so, the NEPOMUK services together with XWiki services can run server side or client side in a single `Java` virtual machine. Figure 2 represents the internal components of the XWiki engine: they take over the management of users, documents, data classes, relational storage, remote procedure calls through the `XML-RPC` protocol, and basic P2P infrastructure. Those components are being migrated toward the Plexus [8] framework. An `OSGi-Plexus` adapter is being worked on in order to get all NEPOMUK `OSGi` services visible from Plexus components, as illustrated by figure 3.

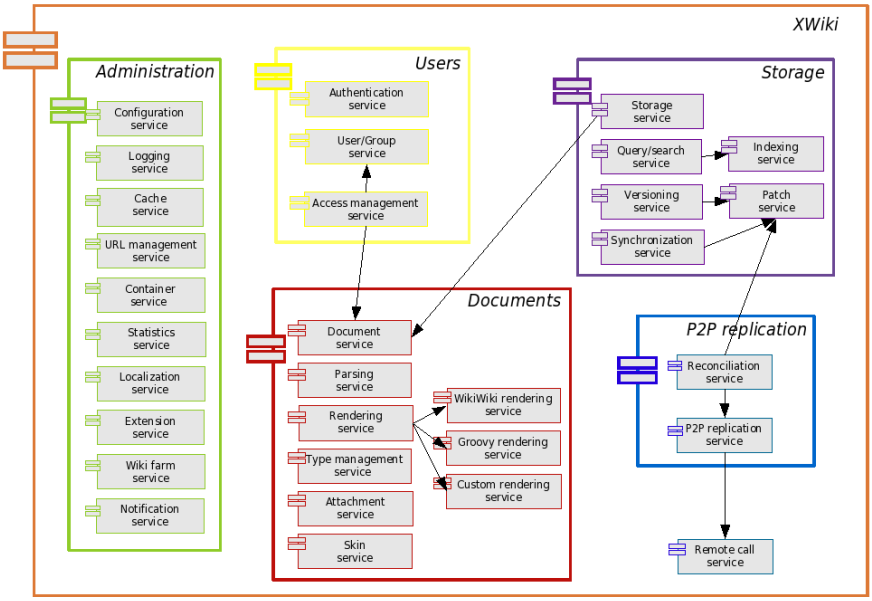


Figure 2: XWiki component diagram

2.1.3 Usage of the NEPOMUK components

The following paragraphs describe more specifically how each NEPOMUK component is used in the context of `XWITS`.

NEPOMUK middleware The NEPOMUK middleware is used as a registry of services [21].

NEPOMUK RDF repository The RDF repository is used for storing all the annotations added to help desk resources, using the NAO [22] and NRL [23] ontologies, as well as the domain ontologies presented below.

NEPOMUK DataWrapper The `DataWrapper` [18] is used both server side and client side. An Aperture crawler has been implemented for XWiki resources for converting the resources metadata into RDF statements. This conversion combines the meta-

[11] `http://www.xwiki.org`

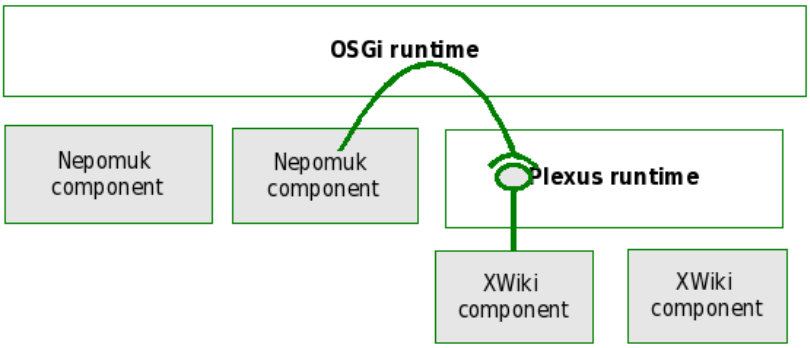


Figure 3: Exposing NEPOMUK components to the XWiki components

data that is available from an XWiki database and the metadata directly stored into the NEPOMUK RDF repository. Later on, a native use of the NEPOMUK RDF repository directly from XWiki will be considered.

On the client side, the `DataWrapper` component is used for crawling the user's personal resources and storing their metadata in the RDF repository.

NEPOMUK-Eclipse PSEW

PSEW is an Eclipse RCP application for manipulating desktop metadata [12]. It lets the user browse his personal semantic web, annotate existing resources and create new resources. PSEW can be used for annotating any URI, either local or remote, using NAO. The annotations are then stored in the local RDF repository. By linking local URIs with remote ones, PSEW lets the user bridge his personal desktop model with the ones available from the Web such as the semantic graph of help desk and knowledge base resources in the context of the WP11000 case study. By doing so, the user extends the public knowledge with personal views and links. This activity supports him in his learning process. In the next implementation, users will be able to share their personal semantic webs with each other, and to query the ones of their network of friends through the NEPOMUK distributed index.

WikiModel

`WikiModel` [19] is a component that generates SAX events or a DOM tree from wiki documents. `WikiModel` supports inline annotations. As such, it is a very handy tool for annotating resources using plain text. This type of annotations is only for advanced users though. Beginners are invited to use the PSEW annotation interfaces which let them enter annotations visually through wizards.

2.2 Data model and ontologies

`XWITS` data model consists of a set of classes described below. Those classes represent the help desk core data model. However, in order to capture the statements expressed in the question and answers brought by the community, it is important to use domain specific ontologies so that an accurate metadata enhancement of the resources can be carried out by the community of users in order to perform efficient semantic queries against the corpus of documents.

2.2.1 XWITS application data model

The help desk data model was designed on the one hand by using the requirements expressed by Mandriva and on the other hand using as inspirational sources the data models of existing applications such as Request Tracker [9], Launchpad Answers [6], JIRA [5], Bugzilla [1].

The data model of the application consists of the following classes, whose fields are described in the tables below:

- Issue class: represents a question in the help desk context, or an issue in the more general context of issue tracking;
- Issue log class: represents an answer brought to a question or an issue;
- User annotation class: encapsulates the annotations brought by a user to a question, an answer or another user. As of October 2007, those annotations consist of a rating and of a set of tags.
- Statement class: represents statements than can be added to a question, an answer or another user using one of the ontologies used by XWITS.

During the implementation phase, the model classes have been mapped to XWiki documents since several fields are available natively as XWiki document properties.

Field	Description
Title	Title of the question
Language	Text language
Reporter	Person who reported the question or the issue
Description	Text of the question/issue itself
Components	Related components (Linux packages etc.)
Status	Status of the question (can be one of the following: new, assigned, verified, resolved or reopened)
Priority	Priority of the question
Resolution	Type of resolution brought (can be one of the following: fixed, duplicate, works for me, won't fix or invalid)
Tags	Topics covered by the question
Creation date	Date the question was submitted at

Table 1: Issue class

Field	Description
Language	Answer's language
Author	Person who reported the question or the issue
Creation date	Date the answer was submitted at
Content	Answer's text

Table 2: Issue log class

Field	Description
User	User identifier
Rating	Rating given by the user to a question, an answer or another user. Rating can have the following values: -1 (meaning uninteresting), 1 (interesting), 2 (very interesting)
Tags	Tags given by the user to the related resource (question, answer or user).

Table 3: User annotation class

Field	Description
Predicate	Statement predicate: one of the predicates available from the ontologies loaded into XWITS. The statement's subject is the document the statement is being added to: a question, an answer or a user.
Object	Value of the predicate.

Table 4: Statement class

2.2.2 Ontologies

The following ontologies are used in addition to NAO for describing the metadata managed by XWITS. This metadata is stored in the NEPOMUK RDF repository.

LIO

LIO stands for "Linux Issue Ontology". It was introduced for describing the relations between an issue and the related problematic hardware or software, other similar issues, the context of the issue, the software packages involved in the problem etc. The current version of LIO consists of the properties listed in table 5.

HWO

HWO stands for "Computer Hardware Ontology". This ontology is based on the XML schema created within the Hardware Lister project [4]. Since the ontology originates from the ezix.org project, the corresponding domain name was used for the first version of this ontology namespace: `http://ezix.org/2007/09/30/hwo`. The main elements of this ontology are summarized in table 6. Hardware configurations are of significant importance when trying to solve technical issues related to a Linux based operating system, since many issues relate to device drivers. The HWO ontology is used for describing the user configurations and for identifying specific devices. Table 6 lists the main classes and properties of the HWO ontology.

DOAP

DOAP [2] stands for "Description of a project". DOAP was created in 2002 by Edd Dumbill. The ontology is used within XWITS for identifying software projects and their successive releases. When questions are being submitted, it's important indeed to identify precisely the related version of the software that raises an issue, and possibly to link the issue with a bug that is referenced within the project's bug database. The DOAP ontology is used for that purpose.

Baetle

Baetle [16] stands for "Bug And Enhancement Tracking LanguagE". It may be used in the future for linking XWITS questions to bugs. An alternative could be the Evoont Bug ontology from the Zurich University [20].

FOAF

The FOAF ontology [3] will be used for describing the users of the help desk. The conversion of user data in the XWiki internal representation toward a FOAF representation is managed by the Aperture XWiki data wrapper. The FOAF representations of users are then stored in the NEPOMUK RDF repository.

SIOC

SIOC [10] will be used for describing the questions and answers brought by

Predicate	Description and range
problematicHardware	This Links to hardware components or systems involved in the issue: complete Laptop system or individual component such as a given sound card. Hardware components are identified and described using the hardware ontology HWO presented below.
problematicSoftware	Links to software involved in the issue: either a complete software system such as "Mandriva Linux" or a specific software release. The software and the versions are identified using the DOAP ontology.
hasRunContext	This property indicates the execution context of the issue. The context can have for instance the following values: installation, run, uprade, boot, newhardware
hasTopics	This property relates to the functional categories the issue relates to, like for instance Security, Networking, Office applications etc.
hasRelatedBugs	Links to bug identifiers referenced in a bug tracking system.
hasRelatedIssues	Links to related issues.
hasInvolvedPackages	Links to specific software packages involved in the issue.
hasTechnicalLevel	Gives an indication on the technical level required to solve the issue.
hasRelatedResources	Links to resources of interest for this issue. This property is a sub-property of the rdfs:seeAlso property.
isFixedByPackage	Links to packages fixing the issue.

Table 5: LIO ontology

the community. The conversion of XWITS discussion threads to a SIOC representation will be taken over by a specific data converter.

Class or Predicate	Description and range
Node class	Represents a node in a tree of hardware components.
Product class	Represents a product.
Vendor class	Represents a vendor.
vendor	A node's vendor.
type	A node's type.
description	Description of a node.
vendor	A node's vendor.
product	A node's product.
physid	A node's physical ID.
version	A node's version.
serial	A node's serial number.
version	A node's version.

Table 6: HWO ontology, derived from the HardwareLister data model

3 Prototype description

This section presents XWITS in action. The first part gives an overview of the application. The second part focuses on specific functionalities directly related to the use cases and the requirements presented in the deliverable D11.1 "Scenario report" [17].

3.1 Overview

Figure 4 is a reminder of the case study context. Mandriva Club members have a NEPOMUK enabled desktop, which includes rich user interfaces for manipulating metadata and for communicating over a P2P network. These components are based on the NEPOMUK-KDE, NEPOMUK-Eclipse and NEPOMUK-Mozilla frameworks.

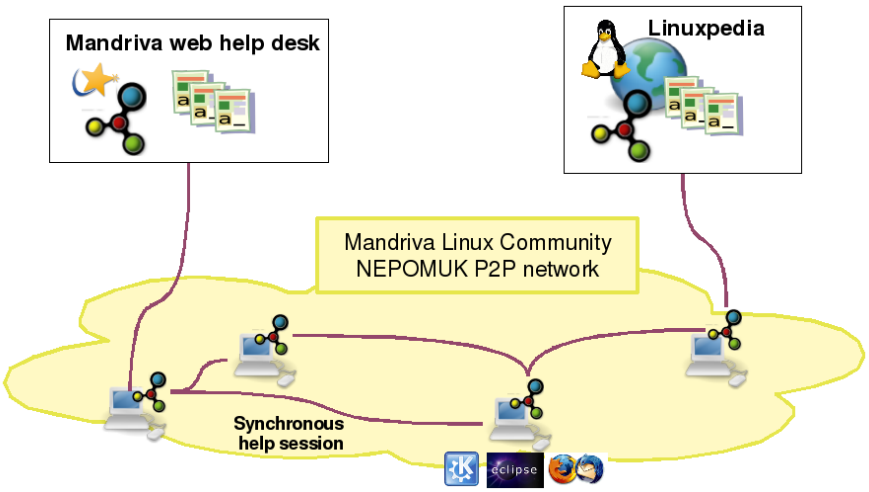


Figure 4: Big picture of the Mandriva Linux community case study

The users can share information across a P2P network connected to two main knowledge bases: Linuxpedia and Mandriva Club knowledge bases. The P2P network serves three purposes: first, users can share metadata directly between one desktop and the one of a peer connected to the network; second, it lets collaborative groups work on a set of documents that are shared on a private network of desktops without the need to transfer their private documents to a centralized area in the first place; third it is used for downloading shared files and their annotations through a BitTorrent like protocol.

The need for a "Linuxpedia", i.e. an encyclopaedia focusing on annotations related to existing Linux related material and having the same organizational model as Wikipedia, has emerged from the user requirements analysis. Experimenting with a wide Linux-related knowledge base and cross Linux-distribution help desk building on the NEPOMUK technologies is being worked on in parallel of the case study for maximizing the case study impact.

Figure 5 presents the home page of the XWITS application. The application contains a set of panels for submitting questions, bringing answers, annotating existing resources, rating answers or experts and searching within the set of resources using the indexed metadata.

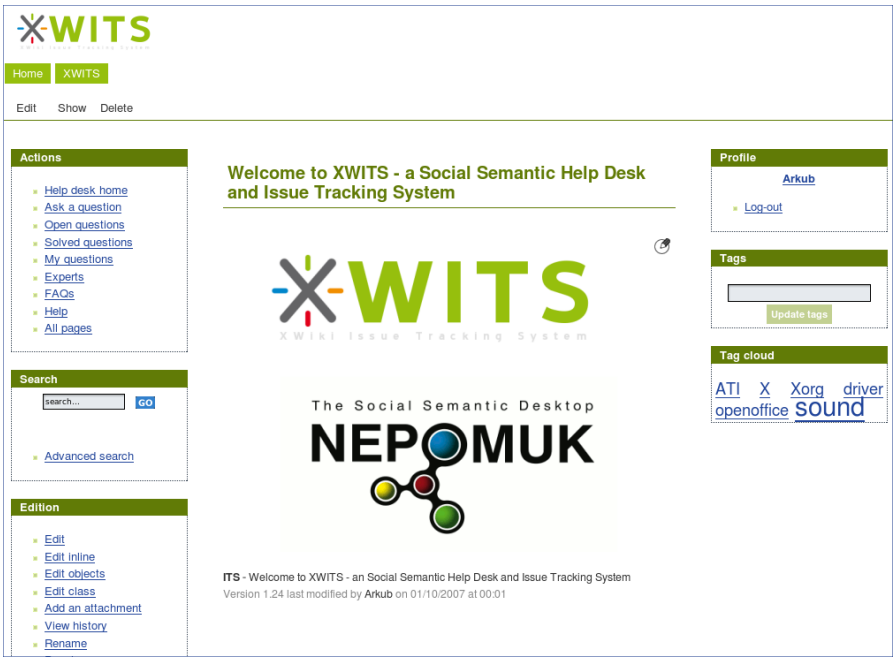


Figure 5: XWITS home page

3.2 Implemented functionalities

This section describes the main functionalities of the XWITS help desk system as of October 2007.

3.2.1 Submitting an issue and adding metadata to it

Submitting an issue from XWITS consists of two steps: first, the user enters the question text using wiki markup through a Web interface. Second, the user has the possibility to add metadata to the question using a dedicated form, and to attach files to the question.

Figure 6 illustrates the form for submitting a question. Once the question is submitted, the author or any other member of the community can add metadata to it by using predicates from the LIO ontology, as illustrated by the figure 7, where the user mentions that the submitted question (i) relates to the software "Shorewall" and "Maradns", (ii) has a related issue, and (iii) requires a high technical expertise to get understood and solved.

Once the metadata has been added, it is displayed contextually as illustrated by the figures 8 9.

In the upcoming versions of the system, Web based metadata management will be simplified through the availability of a Web based visual annotator and of a semantic wiki editor. Users will also have the possibility to submit questions directly from desktop applications.

3.2.2 Bringing an answer to an issue

Help desk users can browse the submitted questions by their status from the help desk home page. Figure 10 presents a list of open questions.

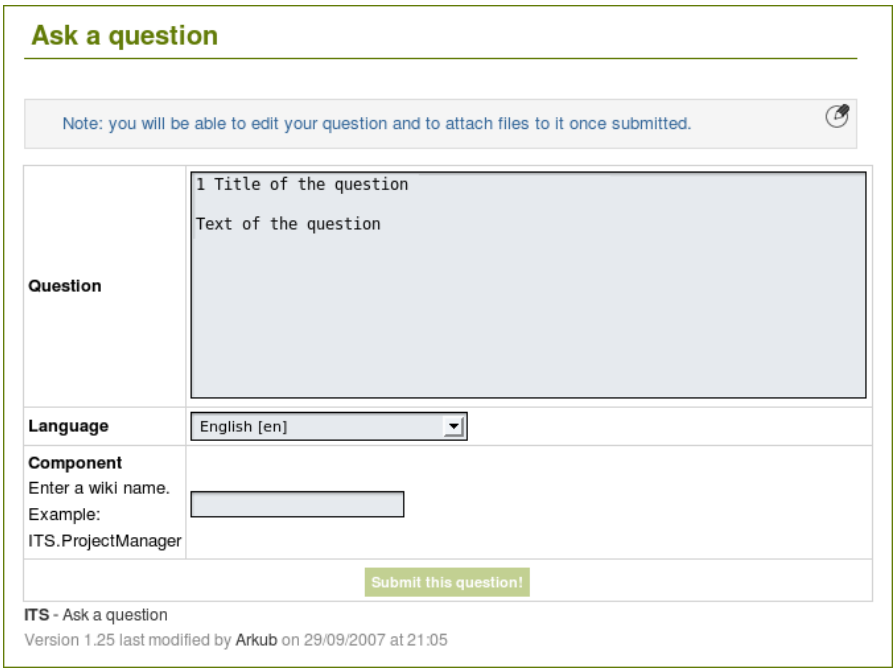


Figure 6: Submitting a question in XWITS

Once a given question is clicked, it lets the user bring an answer, as illustrated by the figure 11.

3.2.3 Rate an expert or an answer

A key aspect of the help desk will rely on its capacities of harnessing the social network drawn between the users, with the objective to let them get help more efficiently and discover new relevant material based on the input provided by people they trust. The current help desk prototype integrates the capacity of adding a personal rating to a resource representing a person, a question or an answer.

3.2.4 Creating a personal help assistant by enhancing existing knowledge

This functionality illustrates a typical interaction between a semantic desktop and the help desk. The help desk consists of general resources aiming at helping users to achieve specific tasks or to solve technical problems. Users willing to customize these resources for their own use, typically to create their own documented task list for achieving a given complex activity are able to do so by using the NEPOMUK semantic desktop for deriving local semantic graphs and personal semantic task lists from the help desk resources.

Let's imagine Kim wants to use his semantic desktop for remembering the actions to be carried out in order to create a curved path on an image using the graphical image editor software GIMP. He has submitted a question on this topic a few days ago, to which several useful answers have been brought. In the meantime he also exchanged a couple of emails with his friend Hellen who masters the topic, and he browsed the ebook "GIMP from novice to professionals", which contains a section covering the management of paths. These various sources contain complementary information and useful exam-

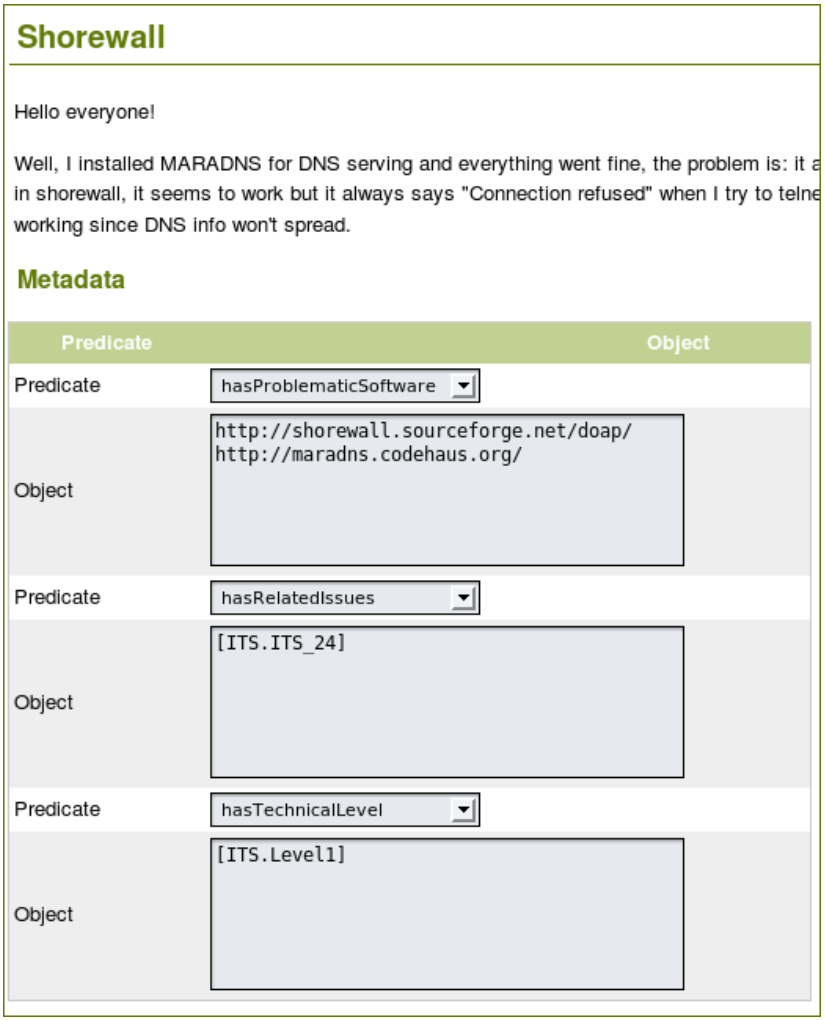


Figure 7: Adding LIO metadata to a question

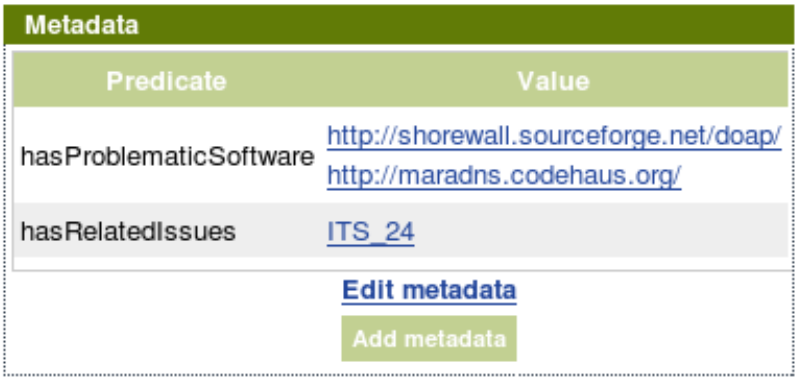


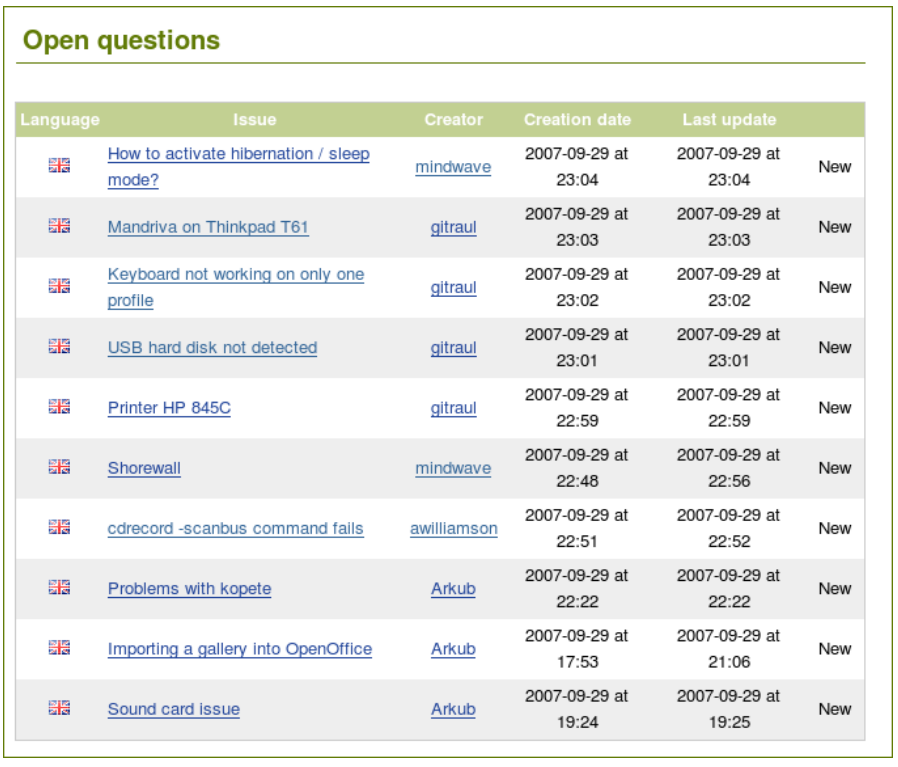
Figure 8: Visualizing LIO statements linked to a question

Lifecycle

Property	Value
Status	New
Severity	Enhancement
Resolution	Invalid
Environment	Mandriva 2008.0
Reporter	Arkub
Components	Shorewall
Project	
Due date	07/10/2007 16:55:14
Priority	
Assignee	

Edit properties

Figure 9: Visualizing the status of a question



Open questions

Language	Issue	Creator	Creation date	Last update	
	How to activate hibernation / sleep mode?	mindwave	2007-09-29 at 23:04	2007-09-29 at 23:04	New
	Mandriva on Thinkpad T61	gitraul	2007-09-29 at 23:03	2007-09-29 at 23:03	New
	Keyboard not working on only one profile	gitraul	2007-09-29 at 23:02	2007-09-29 at 23:02	New
	USB hard disk not detected	gitraul	2007-09-29 at 23:01	2007-09-29 at 23:01	New
	Printer HP 845C	gitraul	2007-09-29 at 22:59	2007-09-29 at 22:59	New
	Shorewall	mindwave	2007-09-29 at 22:48	2007-09-29 at 22:56	New
	cdrecord -scanbus command fails	awilliamson	2007-09-29 at 22:51	2007-09-29 at 22:52	New
	Problems with kopete	Arkub	2007-09-29 at 22:22	2007-09-29 at 22:22	New
	Importing a gallery into OpenOffice	Arkub	2007-09-29 at 17:53	2007-09-29 at 21:06	New
	Sound card issue	Arkub	2007-09-29 at 19:24	2007-09-29 at 19:25	New

Figure 10: List of questions awaiting an answer

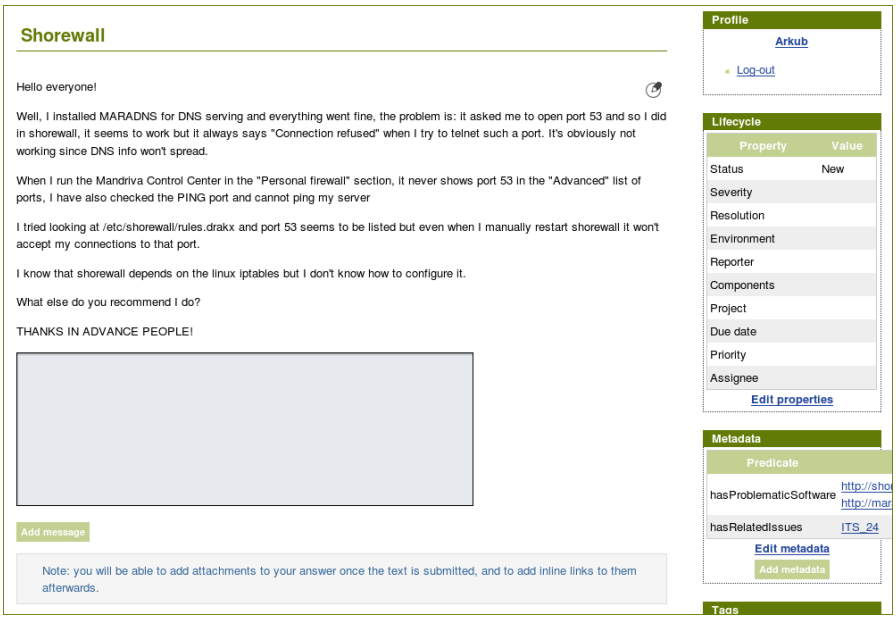


Figure 11: Answer form

ples. Kim would like to store these sources as information elements that relate to the task of creating a curved path in the GIMP. He may consider later on to summarize all these sources in the form of a personal memo, but for now it's faster to simply draw relations between those information sources, so that he can find them easily again later on, and keep track of the persons who authored the information in case he has further questions. In order to draw the links, Kim uses the PSEW[12] workbench: from PSEW, Kim can indeed carry out the following operations:

- create a task object "Creating curved paths in GIMP",
- annotate this task by indicating that it relates to the software "GIMP",
- annotate this task by giving it a rating reflecting the importance it has to him,
- annotate the question submitted previously to the help desk and link it to the created task in his personal meta-data repository on his desktop,
- add an annotation linking the file "GIMP from novice to professionals" to the task,
- add an annotation linking the email discussions he had with Hellen to the task,
- link the task to a set of bookmarks referencing articles covering the topic.

Even though a large part of this use case is available by combining several components of the prototype version of October 2007 as depicted in the following paragraphs, it is not entirely implemented yet. It is expected that the december 2007 milestone will fulfill the complete set of requirements for such a use case.

Figure 12 illustrates the use of PSEW in combination with the help desk. Help desk resources are available from the XWiki view and can be annotated using

[12] Instructions for downloading and running PSEW are available from the following page: `http://nepomuk-eclipse.semanticdesktop.org`

the PSEW annotator, using the NAO [22] ontology. Local resources can be annotated as well. PSEW makes it possible to draw semantic links between local and remote resources. The use of PSEW in this context is explained further below.

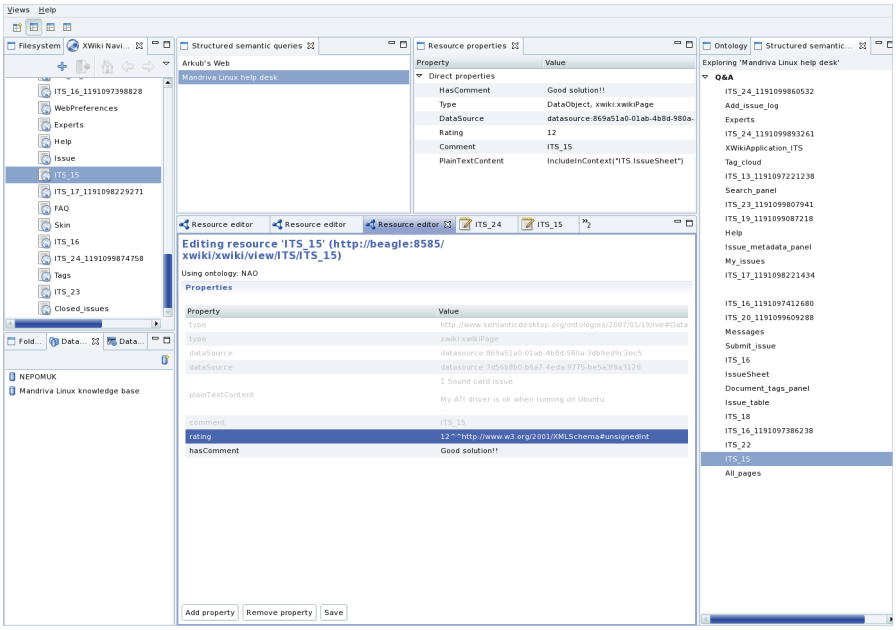


Figure 12: Annotating help desk resources from PSEW

For annotating help desk resources locally, the user goes through the following steps: first, he creates a data source referencing the help desk, as illustrated by figure 13. Once configured, this data source can be explored using a dedicated tree browser, as represented in the top left view of the figure 14, and it can be indexed using a crawler complying with the Aperture indexing framework. The indexing process can be launched locally from a contextual menu, as illustrated by figure 14. In a future version of the prototype, the indexing process of the resources that are hosted on the help desk server will take place on the server itself and the index will be made available to the peers connecting to the help desk, so that they can enhance it locally and possibly share the added meta-data with other peers, using the NEPOMUK social services.

The user can then edit the help desk resources from PSEW, as illustrated by figure 15. The resource properties are made available contextually, above the editor.

More importantly, the user can add local annotations to the resources, lending them a private comment or rating, as represented by figure 16 which represents the Eclipse annotation editor in its current state. From this editor, the user can choose the ontologies he wants to use for annotation. By default, only NAO (NEPOMUK Annotation Ontology) is available from this editor. The public properties and their values are represented in a gray color and are disabled in edition, while the properties added by the user are represented in black, and can be modified.

Figure 16 illustrates the dialogue that gets prompted from the annotation editor when the user adds a new property: the list of NAO properties gets displayed. A similar dialogue appears for assisting the user in choosing the properties' value: the editor selects resources among the range of the ones that comply with NAO range constraints. The prompted dialogue lets the user refine its search by harnessing the Lucene full-text index through the RDF

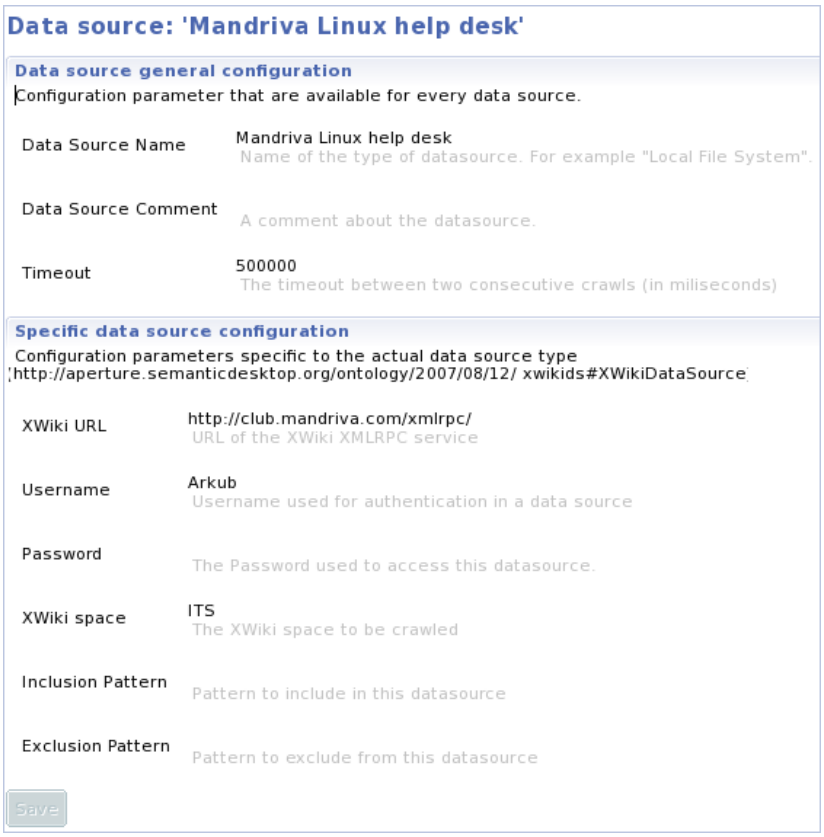


Figure 13: Configuring a data source for the help desk

repository component's API (cf figure 17). Note that the `uuid` values present on the figure will be replaced by more explicit labels in future versions of the prototype.

Complementary annotation ontologies can be loaded to the resource editor, selecting from the ones loaded by PSEW and which can be browsed in the ontology browser displaying the available classes and their properties, as represented on figure 18.

In future versions of the help desk and of the NEPOMUK clients, each user will be able to import not only individual resources, but complete semantic graphs from the help desk meta-data repository into their individual workspace, and to enhance the public graphs locally by adding concepts and relations that make sense to them, involving local desktop resources for the purpose of learning and memorizing. This means that the NEPOMUK semantic desktop infrastructure will let the user associate a set of help desk resources and local notes with specific tasks he's doing regularly on her computer.

3.2.5 Semantic search

Semantic search can be performed either against the public repository of meta-data or locally on one desktop, using a combination of server-side metadata and personal annotations.

On the server, semantic search can be performed in the form of SPARQL queries through a dedicated form. Later versions will include more easy to use forms. On the client side, the semantic query interface consists of several components: a SPARQL query editor, and also a semantic tree which consists

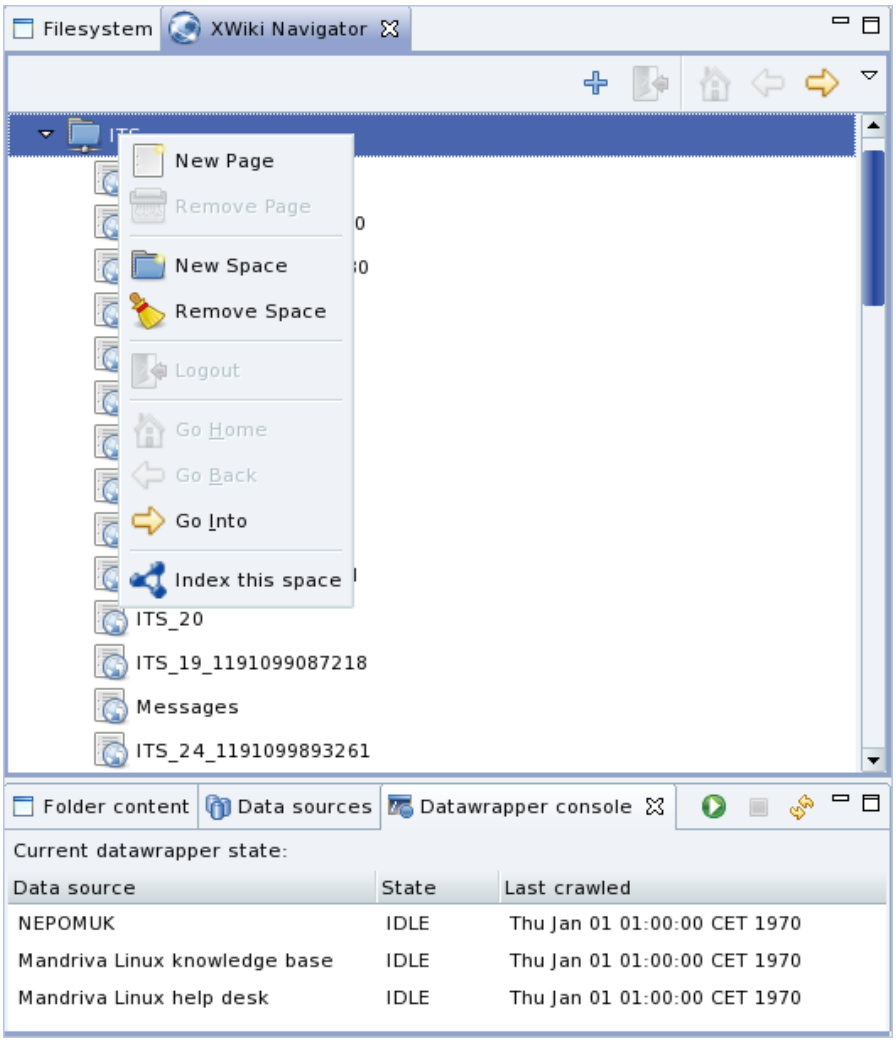


Figure 14: Launching the indexing of help desk resources

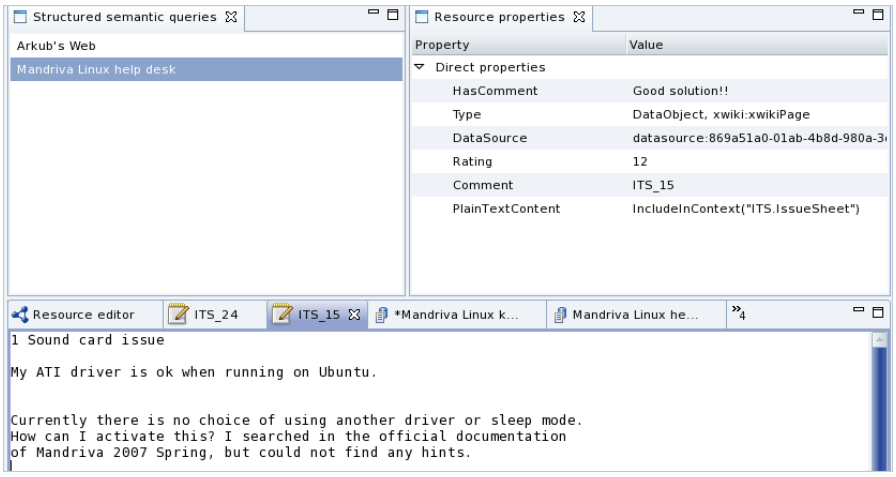


Figure 15: Editing a help desk resource from PSEW

Figure 16: Adding a private annotation to a public resource

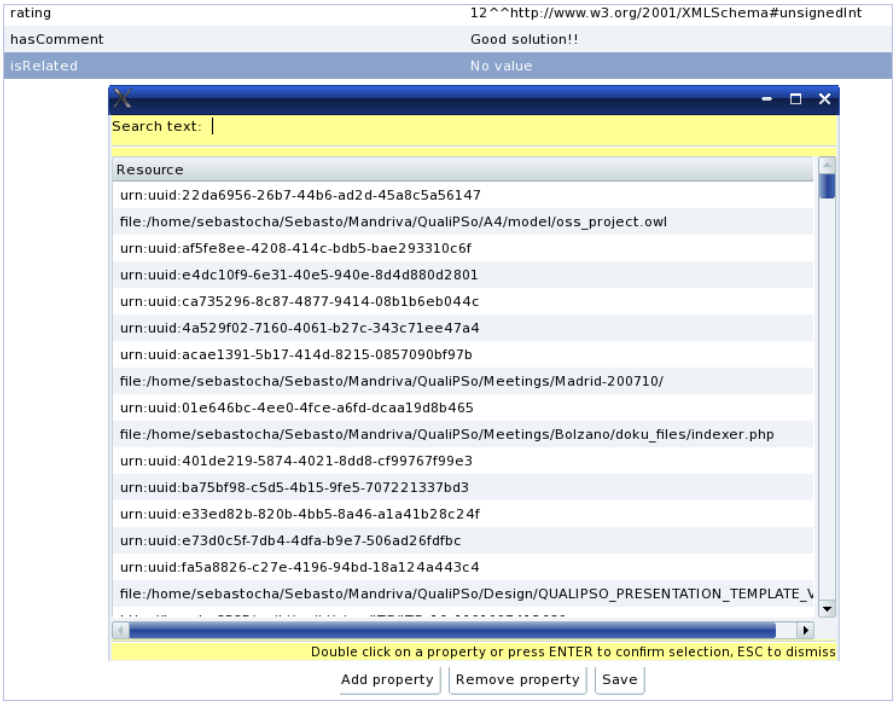


Figure 17: Assistant for choosing a property's value

Figure 18: Ontology browser

of nodes containing one or several SPARQL queries.
Figure 19 represents the SPARQL editor of the help desk Eclipse client.

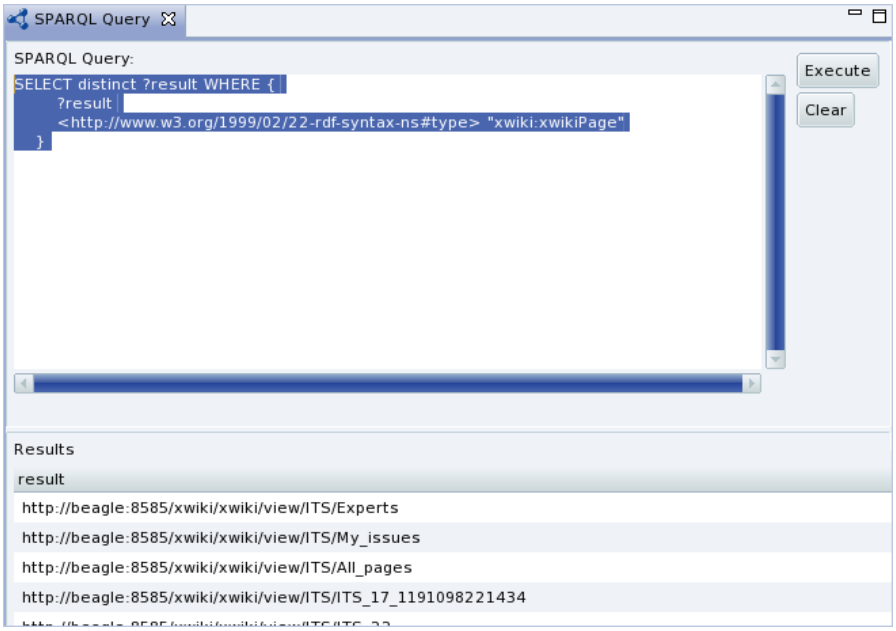


Figure 19: SPARQL editor

Figure 20 illustrates a "Semantic tree", i.e. a tree that is made of a hierarchy of SPARQL queries, where the queries at a given node level can reference a property of the node that results from the parent query execution.

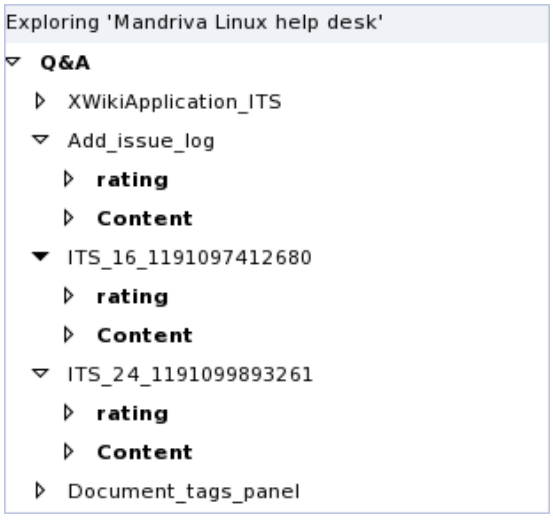


Figure 20: Exploring the resources using a SPARQL tree

The tree on figure 20 consists of the following SPARQL queries:

```
Level 1:
  SELECT distinct ?result WHERE {
                ?result
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> "xwiki:
xwikiPage"
  }
```

```
Level 2:
  SELECT ?result WHERE {
                ##resource##
                <http://www.semanticdesktop.org/ontologies/2007/03/31/nao#
rating>
                ?result
  }

  SELECT ?result WHERE {
                ##resource##
                <http://www.semanticdesktop.org/ontologies/2007/01/19/nie#
plainTextContent>
                ?result
  }
```

3.3 Future functionalities

The current help desk prototype shows a solid basis which will be extended in the next months according to the development roadmap outlined in the following. Now that the assembling of NEPOMUK components with the XWiki ones has been carried out, it is expected that the next design and implementation steps will occur at a fast pace, with approximately one release per month until the end of the project. The functionalities planned for the future are:

Solving an issue using the help desk: this functionality consists of identifying similarities between the submitted questions, with the objective to facilitate the retrieval of existing appropriate answers. It will be implemented by combining the NEPOMUK community manager [14] with the NEPOMUK text analysis service.

Collaborative enhancement of existing knowledge: this functionality makes it possible for a group of co-workers to share their personal semantic webs using a P2P protocol for issuing distributed queries across the network of connected peers and for producing new material collectively. It will be implemented by using the NEPOMUK distributed index [13].

Sharing personal knowledge / social search: this functionality consists of putting into practice the NEPOMUK community management components [14] for taking into account in the search process the resource annotations brought by members of the user's social network. The integration of the NEPOMUK community related components has started and a first integrated prototype will be released by the end of 2007.

Acquiring knowledge related to an issue: the NEPOMUK text analysis service will be put into practice for converting questions into semantic graphs. Several implementations of the service will be experimented, based both on open-source frameworks and on libraries with restricted use including the IBM Galaxy socio-semantic tooling.

Ad hoc learning material creation: this functionality consists of deriving learning material from a semantic query against the help desk knowledge base, taking into account the profile of the target audience (technical level of the participants, duration of the learning session, language etc.). The help desk interface will allow to combine documents using portions of different knowledge base resources.

Email interaction: email interaction is crucial to efficient question and answers exchange since it significantly lowers the entry level for using the system and brings a handy notification of events. `XWITS` extensions will

consist of plug-ins for widely used email clients such as Mozilla Thunderbird and KMail. These extensions will allow to submit questions or answers, to add metadata to questions, to query the available wiki pages and to suggest automatically relevant annotations.

3.4 Mandriva Linux knowledge base

The help desk system is meant to be tightly integrated with the Mandriva Linux knowledge base: one of the objectives of the designed system is to lower indeed the costs of converting technical discussions into reusable knowledge by using semantic and social technologies. Help desk partakers are encouraged to provide semantic links with the Mandriva Linux knowledge base when they answer questions, and to constantly enhance the knowledge base resources by adding metadata.

The knowledge base builds upon the same main components as the help desk, i.e the NEPOMUK and XWiki components. It comprises a wide range of contents that will be considerably broadened by the end of 2007 through the development of dedicated extraction tools which will target in particular the following data sources: the Linux manual pages, software packages repositories and the official Mandriva Linux documentation. Figure 21 represents the home page of the Mandriva Linux semantic knowledge base prototype, which will be brought online during the month of November 2007, featuring a semantic wiki editor.

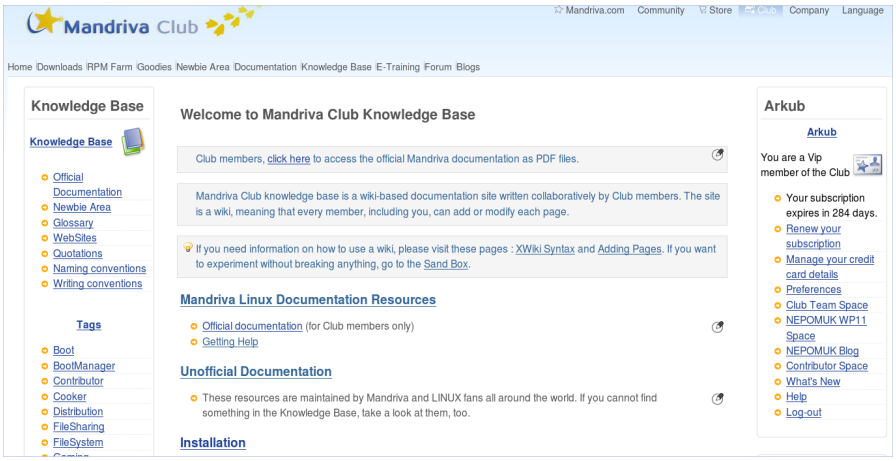


Figure 21: Mandriva Linux knowledge base

4 Prototype evaluation

This section describes the methodology of the assessment that will be performed on the first version of the XWITS prototype. The evaluation will consider the data recorded within October/November 2007 and will be concluded until end of 2007.

The prototype has been brought online on the Mandriva Club in October 2007. Figure 22 illustrates the integration of the prototype in the Club Web site.

Figure 22: XWITS on the Mandriva Club

4.1 User feedback

The feedback from users will be gathered by asking the users to go through a set of evaluation scenarios, and then to answer a set of questions online for gathering their impressions after running the prototype.

4.1.1 Evaluation scenarios

The following scenarios will be proposed to a panel of approximately 20 users for the purpose of evaluation:

Scenario 1: question annotation

- 1. Submit a question to the help desk.
- 2. Add public metadata to it through the XWITS Web user interface.
- 3. Add private metadata to the question using the PSEW help desk client.

Scenario 2: perform a search using metadata criteria from the Web application

- 1. Perform an advanced search across the existing set of annotated questions using the Web interface.

Scenario 3: perform a search using metadata criteria from a desktop application

- 1. Perform an advanced search across the existing set of annotated questions using the NEPOMUK-Eclipse interface.

4.1.2 Questionnaire

The user satisfaction assessment will be conducted through the questionnaire consisting of the questions presented in table 7.

User assessment questionnaire
On a descent scale from A to E, what grade would you give to the whole help desk system?
How would you grade following modules of the system? Eclipse client, Web based semantic editor module, Metadata browsing module
Is it possible and easy for you to articulate your ideas and personal knowledge related to online resources using the system?
Is it possible and easy for you to visualize your personal knowledge using the system?
How relevant are the resources recommended by the system?
Would you say you get answers to your question more efficiently than when using other systems (Mandriva forum, mailing-lists etc)?
Which features are you missing?

Table 7: User assessment questionnaire

4.1.3 Results

The results will be presented in an amendment to this deliverable.

4.2 Completeness level

The objective of the first XWITS implementation milestone was to carry out a sound integration of the NEPOMUK components for the design of a domain specific application. This objective has been met. Several features introduced in the deliverable D11.1 [17] have not been implemented yet, but sound foundations have been laid down for an efficient design of advanced functionalities on top of the first prototype. Table 8 lists those features together with a note on the way they will be implemented.

In terms of non-functional requirements, as of October 2007, the help desk prototype is integrated with the Eclipse version of the NEPOMUK semantic desktop infrastructure only; as mentioned earlier, the integration with the KDE desktop and the NEPOMUK Mozilla extensions will be carried out in the upcoming versions of the prototype.

Upcoming feature	Usage and implementation note
Ontology refactoring	This will let the users evolve the LIO and HWO ontologies directly from the help desk knowledge base. This will be made possible by enhancing the RDF repository.
Embedded semantic documents and support for combined documents	Users will be able to combine help desk documents for creating ad hoc elearning material for their own use. This feature will use the capabilities of the WP1 semantic wiki editor.
Conversion of natural language to statements	Users will get assisted in their production of metadata by putting into practice the NEPOMUK text analysis service.
Text completion based on the domain ontologies	A set of text editors implemented in the Eclipse, KDE, Mozilla frameworks will assist the users in issuing semantic statements.
Zoomable semantic exploration of the help desk resources	iMapping interfaces will be integrated into the help desk GUI from WP1 components.
Faceted browsing	The browsing across the help desk resources will be improved using dedicated UI components, both Web based and desktop based.
Search refinement along the semantic and social axis, and search results ranking	The search engine will integrate social features by integrating the NEPOMUK community manager services.
Trust and reputation management	Trust capabilities will be implemented on top of the NEPOMUK community services.
P2P search across a group of peers	P2P aspects will build upon the NEPOMUK distributed index and the GridVine [11] layer.
Ontological representation of desktop events and contextual recommendation	Contextual recommendation will harness the NEPOMUK context service.
Personal workflow and task pattern support	Task support will be introduced by the integration of the NEPOMUK WP3000 outcome [15]
Email integration	Email integration will focus on the KMail and Mozilla Thunderbird applications. The integration will be based on the NEPOMUK-KDE and NEPOMUK-Mozilla libraries.
Help desk API availability	The help desk will provide a set of REST APIs for being seamlessly integrated in other applications.

Table 8: Upcoming features and the related NEPOMUK components

4.3 Quantitative assessment

Upcoming versions of the prototype will include services for computing daily the following indicators, as introduced in the deliverable D11.1 [17] and reminded below.

Content indicators
Number and size of documents available in the whole distributed system of peers
Number and size of documents produced in the knowledge base
Number of questions submitted to the help desk
Number of answers submitted to the help desk
Percentage of questions finding an accepted answer

Table 9: Content indicators

Metadata indicators
Number of RDF statements available in the system in total or grouped by most relevant RDF types or properties
Number of RDF statements resulting from automatic processing (text-analysis, automatic tagging, etc.
Number of RDF statements entered manually by users
Number of statements created over a period of time
Percentage of questions finding an accepted answer

Table 10: Metadata indicators

Social effects indicators
Number of groups formed
Percentage of accepted answers receiving reviews
Number of resource ratings produced
Number of person ratings produced
Percentage of questions finding an accepted answer

Table 11: Social effects indicators

System use indicators
Number of users using the system for producing contents
Number of users using the system for reading contents
Number of top level resources viewed per period of time
Number of top level resources produced per period of time
Number of search issued per period of time using the search form (distinguishing real searches from browsing by clicking tags or ontology elements that refine the search)
Number of users using exclusively the web based version of the system
Number of users using the rich client desktop version of the system
Number of new users registering to the service per period of time
Number of peers that are online in average

Table 12: System usage indicators

5 Commercial exploitation opportunity

The market of help desks with social capabilities able to harness an in depth integration with a semantic knowledge base is emerging. XPertNet[13] will consider deriving a commercial product from the XWITS prototype. As of October 2007, the prototype has been packaged as an XWiki application; it will continue its lifecycle as a standalone XWiki product enhancing the XWiki product family consisting of the social feed reader XWiki Watch[14], the project management solution Chronopolys[15] and other applications available from the following page: http://www.xwiki.com/xwiki/bin/Solutions/.

Mandriva expects that the combination of Web 2.0 community driven approach with Semantic Web principles tightly integrated with the Linux desktop will lead to a rich ecosystem of content providers and consumers around the Mandriva platform, thus easing the use and the dissemination of the Mandriva products, and paving the way for the use of the NEPOMUK Semantic Desktop by other open-source project communities. Mandriva will also consider using the XWITS solution for its professional support platform.

[13] http://www.xwiki.com
[14] http://www.xwiki.com/xwiki/bin/Solutions/XWikiWatch
[15] http://www.xwiki.com/xwiki/bin/Solutions/Chronopolys

6 Conclusion

The work achieved for the first release of the WP11000 social semantic help desk has consisted on the one hand of assembling together several NEPOMUK components with the XWiki engine, and on the other hand of designing an application layer fulfilling a first set of the targeted help desk specific requirements. The first `XWITS` prototype lets the users combine a local metadata repository with a remote one and to issue semantic queries. Metadata is first extracted by an Aperture compliant crawler and is then manually enhanced. A first set of experimentations is about to be conducted with the community of Mandriva users.

The next steps will consist in adding a P2P and a social layer to the system in addition to the semantic one through the integration of the NEPOMUK distributed index and the community manager. The semantic layer itself will be enhanced through the integration of the NEPOMUK text analysis service as well as the capability of dealing with the user's context. The help desk will be improved both on the server side and on the client side. Beside the PSEW Eclipse user interface, NEPOMUK-KDE and NEPOMUK-Mozilla libraries will be put into practice for drawing further the links between a NEPOMUK enabled desktop and the help desk backend.

In order to expose the system to the widest audience and also for designing the future versions of the domain ontologies used (mostly LIO and HWO), the help desk will also be experimented in the context of an open metadata reference site about Linux under the `Linuxpedia.org` umbrella.

Promising opportunities for commercial exploitation of the prototype are pursued.

References

[1] Bugzilla. http://www.bugzilla.org.

[2] DOAP – Description Of A Project. http://www.usefulinc.com/doap.

[3] FOAF – Friend Of A Friend. http://xmlns.com/foaf/spec/.

[4] Hardware Lister. http://ezix.org/.

[5] JIRA. http://www.atlassian.com/software/jira/.

[6] Launchpad Answers. http://answers.launchpad.net.

[7] OSGi. http://www.osgi.org.

[8] Plexus. http://plexus.codehaus.org/.

[9] Request Tracker. http://bestpractical.com/rt/.

[10] SIOC – Semantically-Interlinked Online Communities. http://rdfs.org/sioc/spec/.

[11] K. Aberer, P. Cudre-Mauroux, M. Hauswirth, and T. van Pelt. GridVine: Building internet-scale semantic overlay networks. In International Semantic Web Conference (ISWC), volume 3298 of LNCS, pages 107–121, 2004.

[12] David Barth, Costas Christidis, Stéphane Laurière, Fabio Mancinelli, and Leo Sauermann. Nepomuk Deliverable D7.3 – Eclipse Community Involvement. http://nepomuk.semanticdesktop.org/xwiki/bin/Main1/D7-3, 2007.

[13] Vasilios Darlagiannis, Roman Schmidt, Renault John, and Ekaterini Ioannou. Nepomuk Deliverable D4.1 – Distributed Search System - Basic Infrastructure. Technical report, EPFL and L3S, 2006.

[14] Gianluca Demartini, Parisa Haghani, Robert Jäschke, Ann Johnston, Malte Kiesel, and Raluca Paiu. Nepomuk Deliverable D5.1 - Community Support Software First Version. http://nepomuk.semanticdesktop.org/xwiki/bin/Main1/D5-1, 2006.

[15] Olaf Grebner, Ernie Ong, Uwe Riss, Marko Brunzel, Ansgar Bernardi, and Thomas Roth-Berghofer. Nepomuk Deliverable D3.1: Task management model. http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/D3-1.

[16] Henry Story. Baetle – Bug And Enhancement Tracking LanguagE. http://code.google.com/p/baetle/.

[17] Stéphane Laurière, Alexandre Solleiro, Sebastian Trüg, Cristian Bogdan, Kristina Groth, and Pär Lannero. Nepomuk Deliverable D11.1 – Mandriva Community Nepomuk Scenario Report. http://nepomuk.semanticdesktop.org/xwiki/bin/Main1/D11-1, 2006.

[18] Leo Sauermann and Enrico Minack. Nepomuk Deliverable D2.1 - Adapters, Extractors, and Knowledge Structure Services. http://nepomuk.semanticdesktop.org, 2006.

[19] Mikhail Kotelnikov. WikiModel. http://www.wikimodel.org.

[20] University of Zurich. EvoOnt – A Software Evolution Ontology. http://www.ifi.uzh.ch/ddis/msr/.

[21] Gerald Reif, Tudor Groza, Siegfried Handschuh, Medhi Jazayeri, and Cédric Mesnage. Nepomuk Deliverable D6.2A – Intermediate Nepomuk Architecture. http://nepomuk.semanticdesktop.org/xwiki/bin/Main1/D6-2A, 2007.

[22] Ludger van Elst Simon Scerri, Michael Sintek and Siegfried Handschuh. Nepomuk Annotation Ontology Specification. http://www.semanticdesktop.org/ontologies/nao/, 2007.

[23] Ludger van Elst Simon Scerri, Michael Sintek and Siegfried Handschuh. Nepomuk Rrepresentation Language. http://www.semanticdesktop.org/ontologies/nrl/, 2007.