NEPOMUK
A Social Semantic Desktop

Semantic Desktop Hands-on Session
13/4/2007 Freie Universität Berlin

Stéphane Laurière XWiki / Mandriva
slauriere@xwiki.com

Leo Sauermann DFKI
leo.sauermann@dfki.de

http://nepomuk.semanticdesktop.org
NEPOMUK – A Social Semantic Desktop

- The Social Semantic Desktop, S. Decker – M. Frank 2004
- FP6 Project IST 2006 – 2008, budget 17M€
- 4 big industry actors, 3 SMEs, 8 research centers
- 5 countries
- Open-source
The precursors...

• Vannevar Bush
  • A Memex is “a device in which an individual stores all his books, records, and communications.”
• Doug Engelbart
  • Open Hypertext System “The open hyperdocument system (OHS) is a standards-based, open source framework for developing collaborative, knowledge management applications.”
• Tim Berners Lee
  • Semantic Web
Scenarios

• How to tag all my files with the same tags as my del.icio.us bookmarks? Idem with events, contacts?
• List all my contracts >100k€ signed 2003 -2006?
• From which email is this file from?
• What are all the OSS projects mentioned in this document?
• How to share with my team all our documents annotated with “virtualisation” and “team”?


My desktop is a database. My inbox is a goldmine. My colleagues desktops are a distributed database.
The Social Semantic Desktop

The desktop is a privileged adoption channel for the Semantic Web

**Desktop:** Help individuals in managing information on the Web/their PC

**Semantic:** Make content available to automated processing

**Social:** Enable exchange across individual boundaries

---

**Personal Semantic Web:** a semantically enlarged intimate supplement to memory

**Social protocols and distributed search**

**Social semantic peers**
NEPOMUK

- **APIs** for semantic management of information
- **APIs** for P2P sharing of knowledge
- **Implementation** on top of 3 desktop frameworks

| The Social Semantic Desktop | NEPOMUK APIs
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ontologies</td>
</tr>
<tr>
<td></td>
<td>General guidelines</td>
</tr>
</tbody>
</table>

- **Standardisation** of the APIs
- **SemanticDesktop.org** foundation
Market context

Collaboration
- P2P
- Sharing capabilities
- Synchronisation

Ease of reuse and of integration
- Separation of concerns (SOA)
- Ecosystem size
- Commercial friendly open-source license
- Standardization
- Liveliness
Architecture: Desktop SOA
Some NEPOMUK services

- Service declaration / consuming middleware
- RDF storage service
- WikiModel
- Desktop resource identification service
- Local search (Aperture / Strigi / Beagle++)
- Ranking and recommendation service
- Simple finite state machine
- Task manager (+ task ontology)
- IMapping
- User activity monitoring service
- Ontology alignment
- P2P metadata indexing (P-Grid)
- Text analysis (LanguageWare / Uima / GATE)
NEPOMUK Ontologies

- Representation Language (NRL)
- Annotation Ontology (NAO)
  - nao:hasTag, nao:hasTopic, nao:isRelated
- Graph metadata
  - Named Graphs annotation
  - nrl:subGraphOf, ...
- Information Elements (NIE): extracted metadata ontology
- PIMO desktop ontology

http://www.semanticdesktop.org/ontologies
Involving development communities
Building KDE, Eclipse, Firefox applications on top of NEPOMUK

Case studies

- An electronic laboratory notebook (Cognium Systems)
- A workbench supporting task management (SAP)
- A P2P semantic workbench for TMI consultants (PRC/ICCS)
- A social semantic help-desk (Mandriva)

APIs
Ontologies
General guidelines
Community involvement – Context

• 4 implementations of the Semantic Desktop:
  • Generic middleware services working on MS Windows, Apple, Linux
  • Nepomuk-KDE social semantic desktop
  • Nepomuk-Eclipse social semantic desktop
  • Nepomuk-Mozilla social semantic desktop
• Emulation between the communities
NEPOMUK-KDE

The Social Semantic Desktop

NEPOMUK

Semantic Desktop Hands-On Session 2007 – Freie Universität Berlin

© NEPOMUK Consortium 2006 – 2008
The Nepomuk-KDE Firecracker desktop

- Semantic Pad
- Kaukolu Wiki
- Structure Recommender
- Distributed Index

- Middleware
- PIMO Service
- Data Wrapper
- RDF Repository

- Service registry
- Library KNepomuk
- DBUS

- Class generator: create classes from the ontologies (NAO and NIE)

- C++/QT4/KDE4 service based on Soprano RDF wrapper developed in Nepomuk-KDE

- Strigi(++)
- KMetaData: easy access to metadata via a resource-based API

- KDE4 widgets: tagging, ranking, tag clouds
  - Dolphin
  - Amarok
Classes can then be used in KDE4 applications

• Example

```cpp
File myFile( "/home/jdoha/notes.txt" );
myFile.setTags(["nepomuk", "notes", "review"])  
myFile.setRating( 6 ); 
myFile.setComment( "Notes on NEPOMUK review" );
```
Semantic file annotation in KDE
Nepomuk-KDE Community

- Jos van den Oever
  - Strigi desktop search

- Sebastian Trüg
  - Architect, core developer, coordinator

- NEPOMUK Consortium
  - KTH designers
  - NUIG developers

- nepomuk-kde mailing-list (40)

- KDE reviewers
  - KDE application developers (Dolphin, Amarok, kde-pim, ...)
  - Plasma developers

- KDE users
  - Linux/Windows

- aKademy
  - 2006/2007/2008

- FOSDEM
  - 2007/2008

- Wasabi API
  - (FreeDesktop.org)

- KDE-Research

- KDE-France

NEPOMUK
Semantic Desktop Hands-On Session 2007 – Freie Universität Berlin

© NEPOMUK Consortium 2006 – 2008
Web links

- The NEPOMUK-KDE wiki
  - http://nepomuk-kde.semanticdesktop.org
- NEPOMUK-KDE 1st deliverable
  - http://nepomuk.semanticdesktop.org
- Source code
  - http://websvn.kde.org/trunk/
- FOSDEM 2007 presentation and feedback
  - http://dot.kde.org/1172420257/
Nepomuk-KDE – Next steps

• Metadata management in the KDE core (kdelibs)
  • Metadata management in the KDE core (kdelibs) and applications such as Amarok or Digikam
  • Use of Nepomuk ontologies for metadata classification in KDE
  • Semantic KMail and whole semantic Kontakt suite
• Semantic wiki editor complying with NEPOMUK wiki API
• P2P Components
• Advanced visualization, plasma plugins
NEPOMUK-Eclipse

http://nepomuk-eclipse.semanticdesktop.org
Objective: a set of Nepomuk-Eclipse services

NEPOMUK OSGi bundles
- Middleware
- RDFRepository, Distributed index
- Community manager
- OnNeM ...

NEPOMUK views
- Local search
- Wiki editor
- OnNeM
- Ontology viewer
- Metadata query
- Graph browser/editor
- IMapping

External OSGi bundles
- ECF
- Apogée
- DBin
- Mylar
- Eclipse BitTorrent
Eclipse communities of interest

Eclipse projects related to social semantic technologies

Eclipse Foundation
- ECF
- Higgins
- Mylar
- Corona
- Apogée

IBM
- IBM LanguageWare
- Lotus Hannover

NEPOMUK
- RDF API
- Wiki API
- Social networking
- P2P indexing
- PIMOS, NRL
- Task Model
- ...

Eclipse RCP projects
- WiredReach
- Haystack
- Collaber
- DBin
- TopQuadrant
- SWeDE
- ...

EU Research Projects
- TEAM
- QualiPSo
- Discovery
- EDOS
Need for an Eclipse collaboration umbrella

• Email by Dennis O'Flynn (Eclipse Corona) to corona-dev@eclipse.org (Oct. 2006)

[Some Eclipse projects] are addressing the following collaboration areas:

* Corona - tool collaboration
* ECF – data collaboration
* ALF – process collaboration
* Mylar – task collaboration
* Apogee – content management

We would like to explore any possible synergies our projects may share. We propose that key members from Corona / ECF / ALF / Mylar / Apogee meet and discuss possible ways our projects can work together.

http://dev.eclipse.org/mhonarc/lists/corona-dev/msg00434.html
Combine RCP with web interfaces

- Compromise

- It is possible to use the **rich-client interface** for features that are typically done in a rich client
  - Editing Complex Data. Domain Specific applications like iPad.
  - Annotating files from the filesystem using drag-drop
  - Combining many plugins for complex systems like task management
  - sidebar

- Use **web interface** for features that are typically web
  - semantic wiki
  - searching, information retrieval, faceted browsing
  - debugging interfaces - hacked much quicker in jsp 😊
  - web browsing, web annotations, photo annotations, social software etc

- The golden middle way: Embed web features in the rich client using a web-browser. Example: DBIN’s google maps mashup.
NEPOMUK PSEW

- Establish contacts with existing Eclipse initiatives related to collaboration
- **PSEW**: P2P Semantic Eclipse Workbench
  - (i) NEPOMUK bundles + (ii) external bundles
  - Generic RCP application with views for the NEPOMUK services (configuration, local search, ontology editor, PIMO browser, rich wiki editor, P2P search, etc.)
  - Share code between the case studies RCP
  - [nepomuk-eclipse@semanticdesktop.org](mailto:nepomuk-eclipse@semanticdesktop.org)
  - Possible permanent group within the Eclipse Foundation
Embrace, reuse, adapt, draw inspiration from...

- DBIN.org Brainlets (SeMedia)
- Fresnel: Views and Lenses (MIT)
- Haystack/Hayloft and semantic actions (MIT)
- Data binding components and forms framework (Eclipse)
- “Chinese Framework” data binding (DFKI)
DBin example: the Semantic Web Research Brainlet

- [http://dbin.org/brainlets/swbrainlet/](http://dbin.org/brainlets/swbrainlet/)
- Scripted application:
  - data: three ontologies and one instance data model.
  - a tree showing persons, topics
  - 5 components that are called when the user clicks into the tree:
    - GUEDEditor
    - Annotations
    - Content
    - Gallery
    - Chat
Fresnel: Views and Lenses for RDF display

• Problem: the GUI has to be able to show something nice for any type of resource, be it E-Mail, Person, Task, or Document.

• Solution: A style-language defines how to render a type, there are GUIs that interpret the style and create a user interface that looks good

• Fresnel is such a style language

http://www.w3.org/2005/04/fresnel-info/
PSEW MVC – A proposal

Model
The model is primarily stored in the Nepomuk RDF Repository. To edit data, a temporary model may exist that represents data currently viewed/edited.

- Ontologies (classes, properties)
- Data (instances)
- View-Descriptions (view/lens)
- Formalized application logic (rules, checks, operations)

Manipulation via PIMO-Service
- insertInstance(…)
- deleteInstance(…)
- updateInstance(…changes)
- addOntology(…)

Controller
Gets input from the view, when data changes. Stores model manipulations in the model.
- create a view
- store changes
- manage possible operations on data
- checks validity of input based on business rules/ontologies

View
The visual components showing instance data, interpreting mouse clicks and keyboard events.
- generates User Interface components based on descriptions of user interface selected by controller
- listens to changes in the model
- invokes methods of the controller
- assists user based on ontologies/rules

could build on
- Haystack View/Lens and operation definitions
- DBIN interaction definitions
- Fresnel Lens/Views
- Eclipse Actions, RCP, Views, Plugins

Haystack/Dbin ontologies

this is purely Nepomuk

could build on
- Haystack GUI implementation
- DBIN plugins
- Fresnel implementations
- Eclipse Forms Framework
- Bean Data Binding Components
- WEB guis!
NEPOMUK-Mozilla
Mozilla community

- P2P
  - Contact established with AllPeers.com (MPL) at the FOSDEM
  - Ongoing discussions with Mozilla Europe
- Semantic aspects
  - Being investigated (PiggyBank etc.)
- Desktop integration
  - Linux: over dbus
Getting involved...

- Public mailing-lists
  - https://nepomuk.semanticdesktop.org/wws
    - people@semanticdesktop.org
    - wikimodel@nepomuk.semanticdesktop.org
    - nepomuk-kde@semanticdesktop.org
    - nepomuk-eclipse@semanticdesktop.org
- Public components code
  - Check out the components section of the Nepomuk web site
- Wikis
  - http://www.semanticdesktop.org
  - http://nepomuk-kde.semanticdesktop.org
  - http://nepomuk-eclipse.semanticdesktop.org
  - http://nepomuk-mozilla.semanticdesktop.org (to be announced)
Toward a Semantic Desktop Foundation?

• Objective: sustain research and standardization
• Business model: membership fee, donations, goodies
Hot topics

- Desktop resource identification
- KDE and Eclipse implementations
- Ontologies refinement
- ....
One NEPOMUK Semantic Wiki

Introduction

Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata model using XML but which has come to be used as a general method of modeling knowledge, through a variety of syntax formats (XML and non-XML). The RDF metadata model is based upon the idea of making statements about resources in the form of subject-predicate-object expressions, called triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion “The sky has the color blue” in RDF is as a triple of specially formatted strings: a subject denoting “the sky”, a predicate denoting “has the color”, and an object denoting “blue”. This mechanism for describing resources is a major component in what is proposed by the W3C’s Semantic Web activity; an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF’s simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.
Term disambiguation

- SemanticPad by Cognium Systems (EPL)
  - [http://www.cogniumsystems.com](http://www.cogniumsystems.com)
- IBM LanguageWare integration

IBM term disambiguation technology is used to automatically generate keyword suggestions based on the text in the "Content" field.