Connected Information Management (CoIM)

Axel Rauschmayer

LMU

2010-02-18
Problems of current information mgmt approaches

Example: Writing a paper
Problems of current information mgmt approaches

- Data partitioned by application
Problems of current information mgmt approaches

- Data partitioned by application
- Small notes don’t fit into any application category
Problems of current information mgmt approaches

- Data partitioned by application
- Small notes don’t fit into any application category
- Different ways of organizing the data
  Tags, folders, keyword search, geo-tagging, etc.
Problems of current information mgmt approaches

- Data partitioned by application
- Small notes don’t fit into any application category
- Different ways of organizing the data
  Tags, folders, keyword search, geo-tagging, etc.
- Data available either online or offline
  Even with offline web applications: no backup
Connected information management (CoIM)

Goal: Better information management

Approach: Integrated storage, data-specific editing, versatile unified navigation.
Related work

Existing approaches for managing heterogeneous information (all RDF-based):

- Haystack [QHK03]: personal information management
  - Missing: mixing structured and unstructured data, note taking, synchronization of data
Related work

Existing approaches for managing heterogeneous information (all RDF-based):

- **Haystack [QHK03]:** personal information management
  - Missing: mixing structured and unstructured data, note taking, synchronization of data

- **Nepomuk [BDE+08]:** the Social Semantic Desktop
  - Missing: universal synchronization, no common platform and abstraction for the data (beyond RDF)
Related work

Existing approaches for managing heterogeneous information (all RDF-based):

- **Haystack [QHK03]:** personal information management
  - Missing: mixing structured and unstructured data, note taking, synchronization of data

- **Nepomuk [BDE+08]:** the Social Semantic Desktop
  - Missing: universal synchronization, no common platform and abstraction for the data (beyond RDF)

- **Kiwi [SEG+09]:** Knowledge in a Wiki
  - Different focus: social knowledge-management and querying (versus personal information management and navigation)
  - Missing: First-class file integration, multi-paradigm search, note taking, generic form-based editing
Outline

- Introduction
- Storage
- Editing
- Navigation
- Implementation (including demo, user study)
- Conclusion (summary, contributions, future research)
Storage: Overview

- What to store
- Resource Description Format (RDF) for structured data
- CoIM projects for all data
- Life cycle events for integration
What to store

Structured data
- object data (bookmarks, BibTeX citations)
- meta-data (time of creation of a bookmark)
- configuration data (start wiki page)

Challenge: schema heterogeneity.

Wiki pages (text with references to data)
- notes, outlines

Challenge: make links explicit in RDF.

External data
- files

Challenge: keep in sync.
Summary of the *Resource Description Format* (RDF):

- *Repository with resources* (data entries)
- Semi-structured, extensible schema
- URIs used as
  - attribute names (no name clashes!)
    http://xmlns.com/foaf/0.1/firstName
  - globally unique IDs of resources
    http://rauschma.de/Axel
  - symbols for concepts
    http://downlode.org/rdf/iso-3166/countries#GB
CoIM projects for all data

RDF repository

- data encoded as resources
- link to resource
- embed resource
- annotate with resource
- unstructured text
- structured data
- heterogeneous schema
- proxy resource

External data

- files
- dir
- file
Life cycle events

- Extend RDF to better integrate data
- Ensure consistency of wiki pages and external data
Resource post-processing

Automatically triggered after saving a resource:

- update the date of last modification
- manifest wiki markup references in RDF
URI renaming

Manually invoked operation:

- update URIs inside wiki markup
- efficient due to wiki references in RDF

Resource → Resource2

```
WikiPage
\link{Resource}
```

must change content
Derivation update

Manually invoked operation:
- update proxy resource data
  - File name changes
  - File is removed
  - Meta-data changes
Editing: Overview

Wiki
- Extensible markup language Wikked [RK06]
- Abstract syntax, several output formats
- Handling editing conflicts
- Distributed version history

RDF editing meta-model (REMM, [Rau08])
- Form-based editing for RDF
- Type system
- Batch editing
- Self-hosting

Syntax for notes and meta-data: title tags [Rau10]
Abstract wiki syntax

Wiki page \rightarrow Any resource

abstract syntax

output

HTML (GUI, printing)

LaTeX

translate

translate

embed
RDF editing meta-model (REMM)

- Built on the RDF display vocabulary *Fresnel* [BPKL06].
- Provide form-based **editing** for RDF.
- Main construct: a *lens* specifies a form-based editor.
  - Encoded in RDF
RDF editing meta-model (REMM)

- Built on the RDF display vocabulary *Fresnel* [BPKL06].
- Provide form-based editing for RDF.
- Main construct: a *lens* specifies a form-based editor.
  - Encoded in RDF

Example: a lens for article citations

- `classLensDomain = bibo:Article`
- `showProperties =`
  - `bibo:authorList (sublens = AuthorLens)`
  - `dcterms:title (range=xsd:string)`
RDF editing meta-model (REMM)

- Built on the RDF display vocabulary *Fresnel* [BPKL06].
- Provide form-based **editing** for RDF.
- Main construct: a *lens* specifies a form-based editor.
  - Encoded in RDF

Example: a lens for article citations

- `classLensDomain = bibo:Article`
- `showProperties =`
  - `bibo:authorList (sublens = AuthorLens)`
  - `dcterms:title (range=xsd:string)`

<table>
<thead>
<tr>
<th>authorList</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td></td>
</tr>
</tbody>
</table>
Title tags

Syntax to quickly enter notes and meta-data.

Write paper #todo #due=2011-01-01

http://www.amazon.com/dp/1847064140/ #read
Navigation: Overview

- Faceted navigation
- Meta-faceted navigation
- Tagging
- Broad facets
- Multi-paradigm search
- Search calculus (query language)
- Inspectors
Faceted navigation

- Common view-based navigation (iTunes, shopping web sites).
- CoIM: adapt to RDF.
- Facet: basically a synonym for “attribute”.
### Faceted navigation

<table>
<thead>
<tr>
<th>Genre</th>
<th>Artist</th>
<th>Album</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>Beatles</td>
<td>Revolver</td>
<td>Good Day Sunshine</td>
</tr>
<tr>
<td>Funk</td>
<td>James Brown</td>
<td>Cold Sweat</td>
<td>Cold Sweat</td>
</tr>
<tr>
<td>Rock</td>
<td>Beatles</td>
<td>White Album</td>
<td>I’m So Tired</td>
</tr>
<tr>
<td>Funk</td>
<td>Prince</td>
<td>Parade</td>
<td>Kiss</td>
</tr>
</tbody>
</table>
Faceted navigation

<table>
<thead>
<tr>
<th>Genre</th>
<th>Artist</th>
<th>Album</th>
</tr>
</thead>
<tbody>
<tr>
<td>Funk</td>
<td>James Brown</td>
<td>Cold Sweat</td>
</tr>
<tr>
<td></td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>Funk</td>
<td>Prince</td>
<td>Parade</td>
</tr>
<tr>
<td></td>
<td>(1)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Genre</th>
<th>Artist</th>
<th>Album</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Funk</td>
<td>James Brown</td>
<td>Cold Sweat</td>
<td>Cold Sweat</td>
</tr>
<tr>
<td>Funk</td>
<td>Prince</td>
<td>Parade</td>
<td>Kiss</td>
</tr>
</tbody>
</table>
Meta-faceted navigation

Meta-facet: time

- “Show me all time-related information”

<table>
<thead>
<tr>
<th>Entity</th>
<th>Occurrence</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmark2</td>
<td>modified</td>
<td>2010-01-21</td>
</tr>
<tr>
<td>Citation7</td>
<td>created</td>
<td>2009-10-04</td>
</tr>
<tr>
<td>Person3</td>
<td>born</td>
<td>1970-03-05</td>
</tr>
</tbody>
</table>

- Occurrence: show **what** was found and **where** it was found.
- Meta-facet = set of facets (born, located-at, created, modified, ...)
- Often special-purpose visualization, e.g. a calendar.
- Restrict by occurrence key and/or value
- Used in addition to faceted navigation
Multi-paradigm search

The CoIM multi-paradigm search integrates
- text search
- faceted navigation
- meta-faceted navigation
- ....

The *search calculus* is a query-like calculus for multi-paradigm search.
Implementation: Overview

- Extensible architecture [Rau05]
- Import (CSV, bookmarks, BibTeX)
- Export (LaTeX, printable HTML, RSS, JSON)
- Synchronize projects between installations
- Demo
- User study
Placcius note cabinet (1689)
Placcius note cabinet (1689)

- Invented by Vincentius Placcius
- 3000 hooks for notes
- Leibniz managed about a million notes with it
Hyena

Hyena is an implementation of the CoIM ideas.

- About 180,000 LoC
- Hyena/Eclipse: Eclipse-based desktop application
- Hyena/Web: GWT-based web application
Hyena has been downloaded in these countries:

Argentina, Australia, Austria, Belgium, Canada, Chile, China, France, Germany, Hungary, India, Iran, Ireland, Italy, Japan, Luxembourg, Malaysia, Nepal, Netherlands, Norway, Poland, Portugal, Russian Federation, Serbia, Spain, South Africa, Switzerland, Taiwan, Thailand, Turkey, United States
Extensible architecture: Hyena/Web

- Web service API
- GUI
- Core API
  - RDF data
  - Components
- RDF engine
- Web server
- Web browser
- Ajax, GWT
Demo
User study

- Small ongoing study
- Participants used Hyena for a week (via an online account)
- Evaluation: log user interface events, examine created RDF
User study

- Small ongoing study
- Participants used Hyena for a week (via an online account)
- Evaluation: log user interface events, examine created RDF
- Types of resources: 64% wiki pages, 21% tags, 4% bookmarks
User study

- Small ongoing study
- Participants used Hyena for a week (via an online account)
- Evaluation: log user interface events, examine created RDF
- Types of resources: 64% wiki pages, 21% tags, 4% bookmarks
- Per wiki page: 0.33 outgoing links, 0.48 incoming links, 0.5 tags.
  ⇒ traditional wiki use and note taking
- Small ongoing study
- Participants used Hyena for a week (via an online account)
- Evaluation: log user interface events, examine created RDF
- Types of resources: 64% wiki pages, 21% tags, 4% bookmarks
- Per wiki page: 0.33 outgoing links, 0.48 incoming links, 0.5 tags.
  ⇒ traditional wiki use and note taking
- Search (percent of time): 98% facets, 0% text search
User study

- Small ongoing study
- Participants used Hyena for a week (via an online account)
- Evaluation: log user interface events, examine created RDF
- Types of resources: 64% wiki pages, 21% tags, 4% bookmarks
- Per wiki page: 0.33 outgoing links, 0.48 incoming links, 0.5 tags.
  ⇒ traditional wiki use and note taking
- Search (percent of time): 98% facets, 0% text search
- Resource sets displayed as (average per user):
  - List: 1h 48min 57s
  - Month: 14min 39s
  - Day: 7min 3
Conclusion: Overview

- Summary
- Contributions
- Future research
Summary

The following contributions have been shown:

- **Storage:** projects, life cycle events
- **Editing:** abstract wiki syntax with multiple output formats, form-based editing, title tags for notes
- **Navigation:** faceted navigation, meta-faceted navigation, multi-paradigm search
- **Implementation:** extensible architecture, user study
Problems solved?

- Data partitioned by application
  ⇒ Common foundation
Problems solved?

- Data partitioned by application  
  ⇒ Common foundation

- Small notes don’t fit into any application category  
  ⇒ Support for note taking
Problems solved?

- Data partitioned by application
  → Common foundation
- Small notes don’t fit into any application category
  → Support for note taking
- Different ways of organizing the data
  → Uniform multi-paradigm search
Problems solved?

- Data partitioned by application
  ⇒ Common foundation
- Small notes don’t fit into any application category
  ⇒ Support for note taking
- Different ways of organizing the data
  ⇒ Uniform multi-paradigm search
- Data available either online or offline
  ⇒ Synchronization of projects
Contributions:

- **A new kind of information management**
  - Extending known ingredients so that they fit together and on top of RDF.
  - New synergies: wiki pages for notes, structured meta-data, meta-facets, ... 

- **A new kind of application architecture**
  - RDF core = data driven
  - Unified models for data storage and navigation.

- **At the intersection of information management and software engineering**
Future research

- Move from generic interface towards specialized applications on top of a common foundation.
- Collaboration:
  - distributed version control (with peer-to-peer synchronization) of all data (not just wiki pages).
  - Live collaborative editing.
Thank you!
Ansgar Bernardi, Stefan Decker, Ludger van Elst, Gunnar Aastrand Grimnes, Tudor Groza, Siegfried Handschuh, Mehdi Jazayeri, Cédric Mesnage, Knud Müller, Gerald Reif, Michael Sintek, and Leo Sauermann.
The social semantic desktop: A new paradigm towards deploying the semantic web on the desktop.
In Jorge Cardoso and Miltiadis D. Lytras, editors, Semantic Web Engineering in the Knowledge Society, chapter XII, pages 290–312.
IGI Global, 2008.

Christian Bizer, Emmanuel Pietriga, David Karger, and Ryan Lee.
Fresnel: A browser-independent presentation vocabulary for RDF.

Dennis Quan, David Huynh, and David R. Karger.
Haystack: A platform for authoring end user semantic web applications.
Axel Rauschmayer.
An RDF editing platform for software engineering.

Axel Rauschmayer.
Lightweight data modeling in RDF.

Axel Rauschmayer.
Structure your wiki: Improving support for structured data in wikis.
Axel Rauschmayer and Walter Christian Kammergruber.  
A wiki as an extensible RDF presentation engine.  
In *ESWC Wsh. Semantic Wikis—From Wiki to Semantics*. M. Jeusfeld  
c/o Redaktion Sun SITE, Informatik V, RWTH Aachen, June 2006.

Sebastian Schaffert, Julia Eder, Szaby Grünewald, Thomas Kurz, Mihai Radulescu, Rolf Sint, and Stephanie Stroka.  
Kiwi – a platform for semantic social software.  
Hybrid between the *Creole* wiki syntax standard and LaTeX:

<table>
<thead>
<tr>
<th>this</th>
<th>is</th>
<th>//italics// and <strong>bold</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>table</td>
<td></td>
</tr>
</tbody>
</table>

* A list with
* two bullets
* and a \href{http://nytimes.com}{link}

<table>
<thead>
<tr>
<th>this</th>
<th>is</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>table</td>
</tr>
</tbody>
</table>

*italics* and *bold*

- A list with
- two bullets
- and a \href{http://nytimes.com}{link}