Hafslund SESAM

2013-05-29, ESWC 2013

Lars Marius Garshol, larsga@bouvet.no, http://twitter.com/larsga
About this project

• A fully commercial project
  – intended to fill a business need
  – implementation of new archive for commercial company
• Won “Archive of the year 2012”
  – for functionality and architecture
• Customer says project has already paid for itself
  – through cost savings at document centre
• Would have been very hard to build with traditional technology
  – we think the architecture is as interesting as the project itself
Hafslund

• A conglomerate of companies
  – energy production/trading
  – electricity grid around Oslo
  – remote heating
  – ...

• Huge documentation collection
  – documentation of the entire electricity grid
  – contracts and agreements with land owners
  – ...

• Wanted a new archive solution
  – with better metadata quality
  – and more documents archived
  – a user-friendly way to find documents in the archive
The curse of NOARK

• All Norwegian public-sector organizations must follow NOARK archive standard
  – very strict, rigid model of archive internals
  – applies to Hafslund, as grid company is a monopolist
• Must also used approved NOARK software
  – not many vendors to choose from
• NOARK solutions widely hated
  – generally low quality software, both UI and technically
  – surprisingly difficult to integrate with
• Users commonly refuse to use the systems
  – documents often wind up not being archived
  – metadata generally is very poor
But does the archive matter?

• It’s the official record of everything the organization has done
  – all correspondence should be here
  – also all important internal documents

• Generally not important at all, until it suddenly becomes all-important
  – not finding the agreement about land use rights from 1935 can cost many millions
  – not having archived a key document can turn into a big PR problem
  – leaks from the archive can cause all sorts of difficulties
Project vision

• Make it easy to file documents in archive
  – do it from the application you’re working in
  – reuse metadata from that app
  – enrich that metadata automatically

• Make archived documents useful
  – build a user-friendly search solution on top
  – connect documents with business context

• Where possible, display archive content *inside* business applications
The system
As seen by customer
High-level architecture

- ERP
- CRM
- SharePoint
- Search engine
- CMIS
- Archive

Triple store (Virtuoso 6)

SDShare
Main principle of data extraction

• No canonical model!
  – instead, data reflects model of source system
• One ontology per source system
  – subtyped from core ontology where possible
• Vastly simplifies data extraction
  – for search purposes it loses us nothing
  – and translation is easier once the data is in the triple store
Simplified core ontology
Data structure in triple store

- ERP
- Archive
- CRM
- Sharepoint

sameAs
Duplicate suppression

<table>
<thead>
<tr>
<th>Field</th>
<th>Record 1</th>
<th>Record 2</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>acme inc</td>
<td>acme inc</td>
<td>0.9</td>
</tr>
<tr>
<td>Assoc no</td>
<td>177477707</td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>Zip code</td>
<td>9161</td>
<td>9161</td>
<td>0.6</td>
</tr>
<tr>
<td>Country</td>
<td>norway</td>
<td>norway</td>
<td>0.51</td>
</tr>
<tr>
<td>Address 1</td>
<td>mb 113</td>
<td>mailbox 113</td>
<td>0.49</td>
</tr>
<tr>
<td>Address 2</td>
<td></td>
<td></td>
<td>0.5</td>
</tr>
</tbody>
</table>

http://code.google.com/p/duke/
Auto-tagging

Annotations on RDF properties tell us what statements to traverse to gather tags.
Access control

• Users only see objects they’re allowed to see
• Implemented by search engine
  – all objects have lists of users/groups allowed to see them
  – on login a SPARQL query lists user’s access control group memberships
  – search engine uses this to filter search results
• In some cases, complex access rules are run to resolve ACLs before loading into triple store
  – e.g: archive system
## Data volumes

<table>
<thead>
<tr>
<th>Graph</th>
<th>Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFS data</td>
<td>5,417,260</td>
</tr>
<tr>
<td>Public 360 data</td>
<td>3,725,963</td>
</tr>
<tr>
<td>GeoNIS data</td>
<td>44,242</td>
</tr>
<tr>
<td>Tieto CAB data</td>
<td>138,521,810</td>
</tr>
<tr>
<td>Hummingbird 1</td>
<td>32,619,140</td>
</tr>
<tr>
<td>Hummingbird 2</td>
<td>165,671,179</td>
</tr>
<tr>
<td>Hummingbird 3</td>
<td>192,930,188</td>
</tr>
<tr>
<td>Hummingbird 4</td>
<td>48,623,178</td>
</tr>
<tr>
<td>Address data</td>
<td>2,415,315</td>
</tr>
<tr>
<td>Siebel data</td>
<td>36,117,786</td>
</tr>
<tr>
<td>Duke links</td>
<td>4,858</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>626,090,919</strong></td>
</tr>
</tbody>
</table>
Viser 10 typer

Arbeidsordre fra IFS (316871)
Fil fra 360 (77809)
Dokumentkort fra 360 (77039)
Anlegg fra IFS (33809)
Kunde fra 360 (26799)
Kunde fra IFS (26791)
Leverandør fra 360 (15541)
Leverandør fra IFS (15540)
Anlegg fra CAB (6340)
Kunde fra CAB (4763)

Mer ▼
Viser 3 filterverdier

<table>
<thead>
<tr>
<th>Jordstjerneveien 15B, Anlegg fra IFS (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>066020015998 - Jordstjerneveien 15B, Anlegg fra CAB (1)</td>
</tr>
<tr>
<td>66020015998 - Jordstjerneveien 15B, Anlegg fra CAB (4)</td>
</tr>
</tbody>
</table>
Jordstjerneveien 15B
Anlegg fra IFS
MCH 66020015998
innig i Mortensrudhøyden (Jordstjerneveien)
Lengdegrad 59.8457151

Åpne i IFS  Mer info

Søk videre eller start nytt søk

Resultat 1 - 2 av 2

066020015998 - Jordstjerneveien 15B
Anlegg fra CAB
Adresse Jordstjerneveien 15B
Postnummer 1283
Poststed OSLO

66020015998 - Jordstjerneveien 15B
Anlegg fra CAB
Adresse Jordstjerneveien 15B
Postnummer 1283
Poststed OSLO
Sekundærledninger Mortensrudhøyden
Arbeidsordre fra IFS
utført på Mortensrudhøyden (Jordstjerneveien)
innrapportert av Thormod Kvarme

Kundesentral for Mortensrudhøyden boligfelt
Arbeidsordre fra IFS
utført på Mortensrudhøyden (Jordstjerneveien)
innrapportert av Thormod Kvarme

Pumpe stoppet
Arbeidsordre fra IFS
utført på Mortensrudhøyden (Jordstjerneveien)
innrapportert av Olaf Nilsen

Kundesentraler sekundærsiden og villavekstere
Arbeidsordre fra IFS
utført på Mortensrudhøyden (Jordstjerneveien)
innrapportert av Thormod Kvarme

Jordstjerneveien 52
Anlegg fra IFS
MCH KLUM410W107H
ingår i Mortensrudhøyden (Jordstjerneveien)

Jordstjerneveien 113
Anlegg fra IFS
MCH 6602013999
ingår i Mortensrudhøyden (Jordstjerneveien)

Jordstjerneveien 84
Anlegg fra IFS
MCH 6602013999
ingår i Mortensrudhøyden (Jordstjerneveien)

Jordstjerneveien 101
Anlegg fra IFS
MCH 66020101999
ingår i Mortensrudhøyden (Jordstjerneveien)

Jordstjerneveien 29C og 29D
Anlegg fra IFS
MCH KLUM410W107B
ingår i Mortensrudhøyden (Jordstjerneveien)
Viser 4 typer

Anlegg fra IFS (102)
Arbeidsordre fra IFS (27)
Fil fra 360 (11)
Dokumentkort fra 360 (10)
The data integration

• All data transport done by SDShare
• A simple Atom-based specification for synchronizing RDF data
  – http://www.sdshare.org
• Provides two main features
  – snapshot of the data
  – fragments for each updated resource
Basics of SDShare

• Source offers
  – a dump of the entire data set
  – a list of resources changed since time $t$
  – a dump of each resource

• Completely generic solution
  – always the same protocol
  – always the same data format (RDF/XML)
SDShare service structure
Implementing the fragment feed

```
select objid, objtype, change_time
from history_log
where change_time > :since:
order by change_time asc

<atom>
  <title>Fragments for ...</title>
  ...

  <entry>
    <title>Change to 34121</title>
    <link rel=fragment href="..."/>
    <sdshare:resource>http://...</sdshare:resource>
    <updated>2012-09-06T08:22:23</updated>
  </entry>
  ...

  <entry>
    <title>Change to 94857</title>
    <link rel=fragment href="..."/>
    <sdshare:resource>http://...</sdshare:resource>
    <updated>2012-09-06T08:22:24</updated>
  </entry>
  ...
```
The SDShare client

![Diagram of SDShare client architecture]

**Frontend**

**Core**

**SPARQL-backend**

**POST-backend**

---

<table>
<thead>
<tr>
<th>Source</th>
<th>Fragments</th>
<th>Msecs pr frag</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://haf66dok03/sparql/">http://haf66dok03/sparql/</a></td>
<td>301</td>
<td>444.1661</td>
</tr>
<tr>
<td><a href="http://haf66dok04:9090/sdshare/collections/IFSD">http://haf66dok04:9090/sdshare/collections/IFSD</a></td>
<td>230000</td>
<td>418.77567</td>
</tr>
<tr>
<td><a href="http://haf66dok04:9090/sdshare/collections/SiebelCustomer">http://haf66dok04:9090/sdshare/collections/SiebelCustomer</a></td>
<td>132000</td>
<td>103.94148</td>
</tr>
<tr>
<td><a href="http://haf66dok04:9090/sdshare/collections/SiebelAddress">http://haf66dok04:9090/sdshare/collections/SiebelAddress</a></td>
<td>1990000</td>
<td>87.86205</td>
</tr>
<tr>
<td><a href="http://haf66dok04:9090/sdshare/collections/SiebelAsset">http://haf66dok04:9090/sdshare/collections/SiebelAsset</a></td>
<td>1590000</td>
<td>116.458565</td>
</tr>
<tr>
<td><a href="http://haf66dok04:9090/sdshare/collections/SiebelServicepoint">http://haf66dok04:9090/sdshare/collections/SiebelServicepoint</a></td>
<td>763805</td>
<td>119.147026</td>
</tr>
<tr>
<td><a href="http://haf66dok01:8090/collection.aspx?collectionId=cease">http://haf66dok01:8090/collection.aspx?collectionId=cease</a></td>
<td>0</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

**jdbc:oracle:thin:@172.19.4.174:1521:WLSPRD**

<table>
<thead>
<tr>
<th>Source</th>
<th>Fragments</th>
<th>Msecs pr frag</th>
</tr>
</thead>
</table>

---

http://code.google.com/p/sdshare-client/
Getting data out of the triple store

- Set up SPARQL queries to extract the data
- Server does the rest
- Queries can be configured to produce
  - any subset of data
  - data in any shape
Properties of the system

• Uniform integration approach
  – everything is done the same way

• Really simple integration
  – setting up a data source is generally very easy

• Loose bindings
  – components can easily be replaced

• Very little state
  – most components are stateless (or have little state)

• Idempotent
  – applying a fragment 1 or many times: same result

• Clear and reload
  – can delete everything and reload at any time
Winding up
The Sesam architecture

• Much more widely applicable
  – good for all kinds of data integration
• Used it very successfully to build an intranet
  – for DSS (Departmental Service Centre, dss.dep.no)
  – not in production, unfortunately
• Anticipate many other uses for it
  – the basic architecture and components are reusable
  – many (but not all) are open source

http://www.slideshare.net/larsga/hafslund-sesam-semantic-integration-in-practice/57
Why we used RDF

• Schemaless
  – makes it easy to accommodate new data sources
• A generic syntax
  – can transport different types of data unchanged
• First-class support for identity
  – URIs and owl:sameAs
• Standardized
  – can choose between lots of different tools
• Support for schema annotation
  – used for metadata enrichment as well as other things
People complaining that it’s hard to do things the usual way with RDF don’t get it.

RDF is great for not doing things the usual way.