Selecting the next database for your project
Exploring RDBMS, OO Databases and Graph Databases ("RDF Triple Stores")

Jans Aasman, Ph.D.
CEO Franz Inc
Ja@Franz.com
This Presentation

- Selecting a database for your next application:
  - RDBMS, OODBs or Triple Store
- *What is a triple store* in 3 minutes
- Some applications where customers choose a triple store
- A simple characterization of various database technologies
- Criteria for choosing a database
- Using a triple store as an event database for “Activity Recognition”
  - Geospatial
  - Temporal
  - Social Network Analysis
Franz Inc.

- Private, founded 1984 - U.C. Berkeley
- Self-funded and profitable

- Software tools for Artificial Intelligence (AI)
- Relational Database Support (last 15 years)
- Object Oriented Database (last 10 years)
- Graph (RDF) Database (last 3 years)

- Major public and industry clients/industry
createTripleStore("seminar.db")

addTriple (Person1 first-name Steve)
addTriple (Person1 isa Organizer)
addTriple (Person1 age 52)
addTriple (Person2 first-name Jans)
addTriple (Person2 isa Psychologist)
addTriple (Person2 age 50)
addTriple (Person3 first-name Craig)
addTriple (Person3 isa SalesPerson)
addTriple (Person3 age 32)

addTriple (Person1 colleague-of Person2)
addTriple (Person1 colleague-of Person3)

addTriple (Person1 likes Pizza)
addTriple ( Person3 neighbour-of Person1)
addTriple ( Person3 neighbour-of Person2)
And now you can query

\[
\text{(select (?xname ?yname)}
\text{ (?x colleague-of ?y)}
\text{ (?y neighbour-of ?x)}
\text{ (?x first-name ?xname)}
\text{ (?y first-name ?yname))}
\]
Use Cases for a Graph Store

- Modeling knowledge and assets
- 1000’s of objects with different feature sets
- Everyday new objects and new features
- You work with rules

Or

- Very regular data but there is a big graph in there

And very often both
Some application areas that require a Triple Store

- Modeling knowledge of assets in an Enterprise
- Modeling an extensive river network
- Representing 1000’s of different types of objects
- Managing biological knowledge
- Multimedia Metadata
- Bug and version tracking
- Collaborative Workspace for Analysts
NASA Constellation project...

- Deals with 1000s of different types of objects
  - Machine parts
  - Processes
  - Software
  - People skills
  - Drawings
  - Documents
- In 100s of distributed databases
- Coordinated through registries
- To provide meaningful search
The graph in a River Network
Regular data with a graph

S1 type stream-segment
S1 upstream S2
S1 upstream S3
S1 left-drainage D1
S1 right-drainage D2
S1 longitude1 12.1
S1 latitude1 -121.2
S1 longitude2 12.12
S1 latitude2 -121.3

Given the polluted segment S1 find all the upstream segments within 50 miles of City1200

Given the polluted drainage D1 find all the schools in the rectangle <x1, y1, x2, y2> that might be influenced
Kodak stores metadata for multimedia as triples
Exploiting Semantics for Personalized Story Creation
Wood, M.D.
Eastman Kodak Co., Rochester, NY;

This paper appears in: Semantic Computing, 2008 IEEE International Conference on
Publication Date: 4-7 Aug. 2008
On page(s): 402-409
Location: Santa Clara, CA,
ISBN: 978-0-7695-3279-0
INSPEC Accession Number: 10131821
Digital Object Identifier: 10.1109/MSC.2008.10
Current Version Published: 2008-08-12

Abstract
The task of creating albums or multimedia output from consumer content is becoming increasingly difficult as the amount of content grows. This work presents a system for using semantic information to automate the process of selecting and combining digital assets into summary presentations or storylines, as well as determining triggers for when to generate such content. The system obtains semantic information from a variety of sources, including the capture metadata, image and video understanding algorithms, user profiles and third party ontologies; all such semantic information is stored in a triple store. Prolog-based rules leverage the triple store to provide a knowledgebase for determining when to create particular types of output and how to select assets for such output. This knowledgebase greatly simplifies the task of creating consumer-grade multimedia content.
Create new services...

Bring family and friends into your world.
Share, create and inspire.

Get 20 FREE prints when you sign up.
- Join now

Save up to 35% on Cards, Calendars and Photo Books.
Learn more >

Hundreds of Card designs
Now you can create sensational holiday cards with your photos using exclusive designs by extraordinary designers. Spread the joy!
Shop Holiday Cards >

Idea Center
Create personalized trick-or-treat totes and learn about more ghoulishly clever projects at the Idea Center.
Visit the Idea Center >
Advancing biology and medicine with tools and methodologies for the structured organization of knowledge.
<table>
<thead>
<tr>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Ontologies: 111</td>
</tr>
<tr>
<td>NCBO Library: 78</td>
</tr>
<tr>
<td>Remote Ontologies: 33</td>
</tr>
<tr>
<td>Number of Classes/Types: 500109*</td>
</tr>
</tbody>
</table>

*from ontologies that have been parsed and indexed*
Steve Jobs

**Steven Paul Jobs** *(born February 24, 1955)* is the co-founder and CEO of Apple and was the CEO of Pixar until its acquisition by Disney. He is currently the largest shareholder at Disney and a member of Disney's Board of Directors. He is considered a leading figure in both the computer and entertainment industries.

Jobs' history in business has contributed greatly to the myths of the quirky, individualistic Silicon Valley entrepreneur, emphasizing the importance of design while understanding the crucial role aesthetics play in public appeal. His work driving forward the development of products that are both functional and elegant has earned him a devoted cult following.

From Wikipedia

---

<table>
<thead>
<tr>
<th><strong>Person</strong></th>
<th><strong>Gender</strong></th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date of birth</strong></td>
<td>Feb 24, 1955</td>
<td></td>
</tr>
<tr>
<td><strong>Place of birth</strong></td>
<td>San Francisco</td>
<td></td>
</tr>
<tr>
<td><strong>Country of nationality</strong></td>
<td>United States</td>
<td></td>
</tr>
<tr>
<td><strong>Profession</strong></td>
<td>Chief Executive Officer, Entrepreneur, Businessperson</td>
<td></td>
</tr>
<tr>
<td><strong>Religion</strong></td>
<td>Atheism</td>
<td></td>
</tr>
<tr>
<td><strong>Ethnicity</strong></td>
<td>White</td>
<td></td>
</tr>
</tbody>
</table>
Just a few triples in Freebase where the subject is Steve Jobs:

<table>
<thead>
<tr>
<th>Outgoing properties:</th>
<th>right</th>
</tr>
</thead>
<tbody>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /common/topic/alias</td>
<td>Steven Paul Jobs (type/text)</td>
</tr>
<tr>
<td>- /people/person/date_of_birth</td>
<td>1955-02-24 (type/datetime)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Стив Джобс (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Стив Джобс (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>스티브 존스 (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Steve Jobs (type/text)</td>
</tr>
<tr>
<td>- /user/mdaconta/human_resources/employee/is_a_supervisor</td>
<td>True (type/boolean)</td>
</tr>
<tr>
<td>- /user/mdaconta/human_resources/employee/title</td>
<td>Chief Executive Officer (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Джобс, Стивен Пол (type/text)</td>
</tr>
<tr>
<td>- /type/object/name</td>
<td>Джобс Стив (type/text)</td>
</tr>
<tr>
<td>- /type/object/permission</td>
<td>Global Write Permission</td>
</tr>
<tr>
<td>- /type/object/type</td>
<td>Topic</td>
</tr>
<tr>
<td>- /common/topic/article</td>
<td>guid/9202a8c04000641f80000000000037481</td>
</tr>
<tr>
<td>- /type/object/type</td>
<td>Person</td>
</tr>
<tr>
<td>- /type/object/type</td>
<td>Film producer</td>
</tr>
</tbody>
</table>
32. Facebook for Spies

Secret agents are people too. They're just very scary people who know lots of classified information. So don't they deserve a social network of their own?

That's why in September, the Federal Government launched A-Space, a highly restricted Facebook-style website that's designed to encourage the sharing of ideas and information among members of the FBI, the CIA, the NSA and the U.S.'s 13 other
So the use cases are

- Modeling knowledge and assets
- 1000’s of objects with different feature sets
- Everyday new objects and new features
- You work with rules

Or

- Very regular data but there is a big graph in there

And very often both
But are they really different?

- Relational Database Systems
- Object Oriented Databases
- Graph databases (Triple Stores)

➔ No, they are all kind of Turing equivalent*

*Turing equivalence is a concept that applies to programming languages, not databases, but you get the point 😊
Relational Databases
Relational Database features

- Efficient table representation to save space
- Efficient joins for simple queries
- Very successful standardized query language (ISO: SQL)
  - Ideal for the enterprise: (relatively) easy to learn, easy for simple queries, easy to find programmers

The biggest problems:
- Inflexible when changing tables on a daily/weekly schedule
- Additional tables for one to many relationships
- Too much worry about what to index (in advance)
- Need $120,000++/year for DBAs for very complex queries
Object Oriented Databases
OO Database features

- Objects (can) resemble real world objects
- The programming world has gone ‘object’
  - No impedance mismatch
- One to many relationships directly encoded
- Changing classes much easier

The biggest problems:
  - **No standards**
  - Very language dependent & programming required
Triple Stores
Triple Store features

- Objects stored as sets of triples
- One to many relationships directly encoded
- Everything is indexed (no choice)
- Designed to facilitate ad hoc queries

- All data structures are **standardized** (W3C: RDF, RDFS, OWL)
- Query language is **standardized** (W3C: SPARQL)

The biggest problem:
  - New kid on the block
Deliberations for DB selection
## Selecting Technologies

<table>
<thead>
<tr>
<th>Tech reasons</th>
<th>RDBMS</th>
<th>OO</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work with 1000s of objects and 1 to many relations, properly indexed?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allow for Pattern Matching and Recursive Graph Search?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change structure of data on a regular basis?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work with rules and reasoning?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can I find the programmers and DBA’s to deal with these new technologies?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will it work with the existing reporting tools?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Will it work with my existing RDBMS?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Selecting Technologies

<table>
<thead>
<tr>
<th>Feature</th>
<th>RDBMS</th>
<th>OO</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work with 1000s of objects and 1 to many relations, properly indexed</td>
<td>-</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Allow for Pattern Matching and recursive Graph Search?</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Change structure of your data on a regular basis?</td>
<td>-</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Work with rules and reasoning?</td>
<td>-</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Can I find the programmers and DBA’s to deal with these new technologies</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Will it work with the existing reporting tools?</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Will it work with my existing RDBMS?</td>
<td>++</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>
AllegroGraph [1]

• Scalable and persistent Triple Store
  – Loads a Billion triples in 8 hours on a 4 processor AMD machine
  – Load 10 Billion triples on EC2 (Amazon) on 10 machines in 10 hours

• Federated
  – Create an abstract store that is a collection of other triple stores. Prolog and SPARQL and Reasoning work transparently against abstract store

• Compliant with standards
  – RDF, RDFS, OWL, SPARQL, Named Graphs, ISO Prolog, OWL-lite reasoning
AllegroGraph [2]

- Relational database efficiency for range queries
  - We support most xml schema types (dates, times, longitudes, latitudes, durations, telephone numbers, etc)
- Spatial database efficiency for geospatial primitives
  - Find elements in bounding boxes as fast as in spatial databases
- Temporal reasoning
  - Reasoning about times and intervals (Allen Logic)
- Social Network Analytics library
  - Find actor degrees and centrality, cliques, group centrality and cohesiveness
AllegroGraph [3]

- Other triple stores:
  - Load the data in bulk
  - precompute all types and other inferences
  - Do queries

- Agraph 3.2 the only dynamic real time triple store
  - Loading triples in linear time
  - Queries and Reasoning can be done at any point in time during the loading
  - 3.2 is done loading the LUBM 8000 benchmark and has done all the queries while the others are still loading.
Activity Recognition

- Our customers use AllegroGraph as an event database with social network analysis and geospatial and temporal reasoning.

Find all meetings that happened in November within 5 miles of Berkeley that was attended by the most important person in Jans’ friends and friends of friends.

```
(select (?x)
 (ego-group person:jans knows ?group 2)
 (actor-centrality-members ?group knows ?x ?num)  SNA
 (q ?event fr:actor ?x)                        SNA
 (qs ?event rdf:type fr:Meeting)               DB Lookup
 (interval-during ?event "2008-11-01" "2008-11-06") RDFS
 (geo-box-around geoname:Berkeley ?event 5 miles) Temporal
 !)                                             Spatial
```
A Simple Event Ontology

- **A type**
  - Meetings, communications event, financial transactions, visit, attack/truce, an insurance claim, a purchase order
  - RDFS++ reasoning
- **A list of actors**
  - Social Network Analysis
- **A place**
  - GeoSpatial Reasoning
- **A Start-time and possible an end-time**
  - Temporal Reasoning
- **Anything else that describes the event**
  - Goods that changed hands
Social Network Analysis
Answers 4 questions

• How far is P1 from P2 (and how strong is the relation?)
• To what groups does this person belong (ego groups, cliques?)
• How important is this person in the group?
• Does this group have a leader, how cohesive are they?
GeoSpatial

• Make the following super efficient
  – Where did something happen?
  – How far was event1 from event2?
  – Find all the events that occurred in a bounding box or radius of M miles?
  – Do these two shapes overlap?
  – Find all the objects in the intersection of two shapes
• On a very large scale
  – when things don’t fit in memory
  – millions of events and polygons
• Adhere to our convention to encode StartTimes and EndTimes and enjoy efficient temporal primitives

• Implementation of Allen’s interval logic primitives
# AllegroGraph Java Edition Tutorial Examples

This Learning Center is designed to facilitate understanding of RDF database technologies and best practices for AllegroGraph. It contains examples for working with RDF triples, Triple Stores and Server Management, Querying with SPARQL, and Reasoning with RDFS++ and Prolog. The software examples are freely downloadable and work with the Java version of AllegroGraph, including the Free Edition.

Please send any comments or suggestions to info@franz.com.

## Preamble and Installation
- Choosing an Edition
- Downloading the software
- Installing the software
- Updating an installation

## Running the Examples
- Starting a server manually
- Building in Eclipse
- Running in Eclipse
- Building from a Command Line
- Running from a Command Line
- Stopping a server manually

## Server Management
- Connecting to a server
- Setting server properties

## Populating a Triple Store
- Loading RDF-XML from a file
- Loading N-Triples from a file
- Adding triples over a socket
- A Note on Duplicates
- Deleting triples

## Indexing a Triple Store
- Overview of Indexing
- Indexing all triples
- Indexing new triples
- Automatic Indexing
- Free Text Indexing

## Basic Retrieval
- Matching Triples
- Working with Queries

## Reasoning over a Triple Store
- Overview of RDFS++
- Reasoning with rdfs:subClassOf
- Reasoning with rdfs:subPropertyOf
- Reasoning with rdfs:domain and rdfs:range
- Reasoning with owl:inverseOf
- Reasoning with owl:sameAs
- Reasoning with owl:TransitiveProperty
- SPARQL with RDFS++ entailment

## Prolog
- Retrieval with Functor q
- RDFS++ reasoning with Functor qg
- Reasoning with Horn Rules
- Disjunction and Recursion

## Federating Triple Stores
Thank You

Jans Aasman
Franz Inc.
www.franz.com