Large-scale Reasoning with a Complex Cultural Heritage Ontology (CIDOC CRM)

Vladimir Alexiev, Dimitar Manov, Jana Parvanova, Svetoslav Petrov

Practical Experiences with CIDOC CRM and its Extensions (CRMEX 2013)
TPDL 2012, 26 Sep 2013, Malta
• ResearchSpace project
• RS Semantic Search
• Fundamental Relation (FR) search
• Implemented FRs
• OWLIM Rules
• Example: FR92i_created_by
• Sub-FRs and Dependency Graph
• Complexity: Classes (Type statements)
• Complexity: Properties
• Comparison to Other Repositories
• Performance of Straight SPARQL Implementation
• Performance of Our Implementation
Funded by Mellon Foundation, run by the British Museum, sw dev by Ontotext
  - Stage 4: expected to start in 2013, with more development and more museums/galleries on board

Support collaborative research projects for CH scholars
  - Open source framework and hosted environment for web-based research, knowledge sharing and web publishing

Intends to provide:
  - Data conversion and aggregation (LIDO/CDWA/similar to CIDOC CRM)
  - Semantic search based on Fundamental Relations
  - Collaboration tools, such as forums, tags, data baskets, sharing, dashboards
  - Research tools, such as Image Annotation, Image Compare, Timeline and Geographical Mapping...
  - Web Publication

Semantic technology is at the core of RS because it provides effective data integration across different organizations and projects.
  - Uses Ontotext's OWLIM repository: powerful reasoning (equivalent to OWL2 RL), fast performance, efficient multi-user access, full SPARQL 1.1 support, incremental assert and retract
• Allows a user that is not familiar with CRM or the BM data to perform simple and intuitive searches.

• Features:
  – Intuitive "sentence-based" UI
  – Searches can be saved, bookmarked (put in a "data basket"), edited, shared between users
  – Auto-completion across all searchable thesauri. Available search relations and appropriate Thesauri are coordinated
  – Search across datasets. E.g. once the entity "Rembrandt" is co-referenced between the BM People and RKD Artists thesauri, paintings by Rembrandt can be found across the BM and RKD datasets
  – Faceting of search results
  – Details, thumbnails (lightbox), list, timeline view
  – Put search result to data basket, invoke RS tool

RS Semantic Search

Large-scale Reasoning with CIDOC CRM
RS Search: Example 1

29 Results

Object Type
1 box
1 broadside
7 calligraphy
1 document
4 invitation
1 textile

Creator
0 Middle East and North Africa Modern Art
1 Mughal Style
1 Osman Waziri
1 Syed Tajammul Hussain
0 The British Museum
1 Thomas Arna

Places
1 Asia
1 South Asia
1 India pre-1947
28 Europe
28 British Isles
28 England

Created
1 (missing this field)
1927-1965
1959

sorted by: Title then by...

RFM1619 Calligraphic composition. Silkscreen print...

RFM1620 Print. Calligraphy. Silkscreen print.

RFM1621 Print. Calligraphy. Silkscreen print.

RFM1622 Print. Calligraphy. Silkscreen print.

RFM2064 Arabic calligraphy; ink and gold on vellum...
calligraphy; print: RFM2064 Arabic calligraphy; ink and gold on vellum... Created: Osman Waziri; Middle East and North Africa Modern Art. London England. 1990. Material: paper.
Finds narrower terms

RS Video by Dominic Oldman (RS PI and BM IT dev manager)
http://www.youtube.com/watch?v=HCnwgq6ebA5
• How does a user search through a large CRM network?

• An answer: Fundamental Relations.
  – Aggregate a large number of paths through CRM data into a smaller number of searchable relations.
  – Provide a "search index" over the CRM relations

• E.g.: FR "Thing from Place"

• Initial implementation presented at SDA 2012 (TPDL 2012), Sep 2012, Cyprus (CEUR WS Vol.912)
<table>
<thead>
<tr>
<th>N</th>
<th>FR</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FR92i_created_by</td>
<td>Thing (or part/inscription thereof) was created or modified/repaired by Actor (or group it is member of, e.g. Nationality)</td>
</tr>
<tr>
<td>2</td>
<td>FR15_influenced_by</td>
<td>Thing's production was influenced/motivated by Actor (or group it is member of). E.g.: Manner/ School/ Style of; or Issuer, Ruler, Magistrate who authorised, patronised, ordered the production.</td>
</tr>
<tr>
<td>3</td>
<td>FR52_current_owner_keeper</td>
<td>Thing has current owner or keeper Actor</td>
</tr>
<tr>
<td>4</td>
<td>FR51_former_or_current_owner_keeper</td>
<td>Thing has former or current owner or keeper Actor, or ownership/custody was transferred from/to actor in Acquisition/Transfer of Custody event</td>
</tr>
<tr>
<td>5</td>
<td>FR67_about_actor</td>
<td>Thing depicts or refers to Actor, or carries an information object that is about Actor, or bears similarity with a thing that is about Actor</td>
</tr>
<tr>
<td>6</td>
<td>FR12_has_met</td>
<td>Thing (or another thing it is part of) has met actor in the same event (or event that is part of it)</td>
</tr>
<tr>
<td>7</td>
<td>FR67_about_period</td>
<td>Thing depicts or refers to Event/Period, or carries an information object that is about Event, or bears similarity with a thing that is about Event</td>
</tr>
<tr>
<td>8</td>
<td>FR12_was_present_at</td>
<td>Thing was present at Event (eg exhibition) or is from Period</td>
</tr>
<tr>
<td>9</td>
<td>FR92i_created_in</td>
<td>Thing (or part/inscription thereof) created or modified/repaired at/in place (or a broader containing place)</td>
</tr>
<tr>
<td>10</td>
<td>FR55_located_in</td>
<td>Thing has current or permanent location in Place (or a broader containing place)</td>
</tr>
<tr>
<td>11</td>
<td>FR12_found_at</td>
<td>Thing was found (discovered, excavated) at Place (or a broader containing place)</td>
</tr>
<tr>
<td>12</td>
<td>FR7_from_place</td>
<td>Thing has former, current or permanent location at place, or was created/found at place, or moved to/from place, or changed ownership/custody at place (or a broader containing place)</td>
</tr>
<tr>
<td>13</td>
<td>FR67_about_place</td>
<td>Thing depicts or refers to a place or feature located in place, or is similar in features or composed of or carries an information object that depicts or refers to a place</td>
</tr>
<tr>
<td>14</td>
<td>FR2_has_type</td>
<td>Thing is of Type, or has Shape, or is of Kind, or is about or depicts a type (e.g. IconClass or subject heading)</td>
</tr>
<tr>
<td>15</td>
<td>FR45_is_made_of</td>
<td>Thing (or part thereof) consists of material</td>
</tr>
<tr>
<td>16</td>
<td>FR32_used_technique</td>
<td>The production of Thing (or part thereof) used general technique</td>
</tr>
<tr>
<td>17</td>
<td>luc:myIndex</td>
<td>The full text of the thing's description (including the-saurus terms and textual descriptions) matches the given keyword. FTS using Lucene built into OWLIM.</td>
</tr>
<tr>
<td>18</td>
<td>FR108i_82_produced_within</td>
<td>Thing was created within an interval that intersects the given interval or year.</td>
</tr>
<tr>
<td>19</td>
<td>FR1_identified_by</td>
<td>Thing (or part thereof) has Identifier. Exact-match string</td>
</tr>
<tr>
<td>20</td>
<td>FR138i_has_representation</td>
<td>Thing has at least one image representation. Used to select objects that have images</td>
</tr>
<tr>
<td>21</td>
<td>FR138i_representation</td>
<td>Thing has image representation. Used to fetch all images of an object</td>
</tr>
<tr>
<td>22</td>
<td>FR_main_representation</td>
<td>Thing has main image representation. Used to display object thumbnail in search results</td>
</tr>
<tr>
<td>23</td>
<td>FR_dataset</td>
<td>Thing belongs to indicated dataset. Used for faceting by dataset</td>
</tr>
</tbody>
</table>
• **OWLIM reasoning features:**
  - Custom rule-sets. The standard semantics that OWLIM supports (RDFS, RDFS Horst, OWL RL, QL and DL) are also implemented as rulesets.
  - Fully-materializing forward-chaining reasoning. Rule consequences are stored in the repository and query answering is very fast.
  - `sameAs` optimization that allows fast cross-collection search using coreferenced values.
  - Incremental retraction: when a triple is deleted, OWLIM removes all inferred consequences that are left without support (recursively).
  - Incremental insert: when a triple is inserted (even an ontology triple), all rules are checked. If a rule fires, the new conclusion is also checked against the rules, etc.
  - Efficient rule execution: rules are compiled to Java and executed quickly.

• **120 OWLIM Rules** to implement 23 FRs:
  - 14 rules implement RDFS reasoning, `owl:TransitiveProperty`, `owl:inverseOf` (OWL) and `ptop:transitiveOver` (PROTON ).
  - 106 rules implement FRs. Used a method of decomposing an FR to sub-FR: conjunctive (e.g. checking the type of a node), disjunctive (parallel), serial (property path), transitive.
Example: FR92i_created_by

- **Thing created by Actor**
  - Thing (or part/inscription thereof) was created or modified/repaired by Actor (or a group it is a member of)

- **Source properties:**
  - P46_is_composed_of, P106_is_composed_of, P148_has_component: navigates object part hierarchy
  - P128_carries: to transition from object to Inscription carried by it
  - P31i_was_modified_by (includes P108i_was_produced_by), P94i_was_created_by: Modification/Production of physical thing, Creation of conceptual thing (Inscription)
  - P9_consists_of: navigates event part hierarchy (BM models uncorrelated production facts as sub-events)
  - P14_carried_out_by, P107i_is_current_or_former_member_of: agent and groups he's member of

- **Sub-FRs**
  - FRT_46_106_148_128 := (P46|P106|P148|P128)+
  - FRX92i_created := (FC70_Thing) FRT_46_106_148_128* / (P31i | P94i) / P9*
  - FR92i_created_by := FRX92i_created / P14 / P107i*
Use a simple shortcut notation
- Script translates ";" to newline and "=>" to "----------"
- Also weaves from wiki
- Checks variable linearity
- Generates dependency graph (see next)

10 rules for FRT_46_106_148_128
7 rules for FR92i_created_by:

```
x <rdf:type> <rso:FC70_Thing>; x <crm:P31i_was_modified_by> y => x <rso:FRX92i_created> y
x <rdf:type> <rso:FC70_Thing>; x <crm:P94i_was_created_by> y => x <rso:FRX92i_created> y
x <rso:FRT_46_106_148_128> y; y <crm:P31i_was_modified_by> z => x <rso:FRX92i_created> z
x <rso:FRT_46_106_148_128> y; y <crm:P94i_was_created_by> z => x <rso:FRX92i_created> z
x <rso:FRX92i_created> y; y <crm:P9_consists_of> z => x <rso:FRX92i_created> z
x <rso:FRX92i_created> y; y <crm:P14_carried_out_by> z => x <rso:FR92i_created_by> z
x <rso:FRX92i_created> y; y <crm:P14_carried_out_by> z; z <rso:FRT107i_member_of> t => x <rso:FR92i_created_by> t
```
Sub-FRs and Dependency Graph

- 51 source classes/properties, shown as plain text
- 13 intermediate sub-FRs, shown as filled rectangles. Used by several FRs to simplify the implementation
- 19 target FRs, shown as rectangles

Large-scale Reasoning with CIDOC CRM
• Museum objects: 2,051,797 (most from the British Museum)
  – Currently completing the ingest of Yale Center for British Art objects to RS (50k)

• Thesaurus entries: 415,509 (skos:Concept)
  – All kinds of "fixed" values that are used for search: object types, materials, techniques, people, places, ... (a total of 90 ConceptSchemes)

• Explicit statements: 195,208,156. We estimate that of these:
  – 185M are for objects (90 statements/object)
  – 9M are for thesaurus entries (22 statements/term)

• Total statements: 916,735,486.
  – Expansion ratio is 4.7x (i.e. for each statement, 3.7 more are inferred)
  – Considerably higher compared to the typical expansion for general datasets

• Nodes (unique URLs and literals): 53,803,189 (don't use blank nodes)

• Repository size: 42 Gb
  – Object full-text index: 2.5 Gb, thesaurus full-text index (used for search auto-complete): 22Mb.

• Loading time (including all inferencing):
  – 22.2h on RAM drive
  – 32.9h on hard-disks
### Complexity: Classes (Type statements)

- 238 classes, some of the top are summarizes in the table
- 415k skos:Concept (terms)
- 2M FC70_Thing (museum objects)
- Hierarchy is 10 levels deep: E1>E77>E70>E71>E28>E90>E73>E36>E37>E34
- For each Inscription, 12 type statements are inferred
- 6.3M E12_Production, repeated as the super-class E11_Modification, plus a few hundred Repairs
- Each E12 also repeated as E63_Beginning_of_Existence; plus 100k Birth and Formation
- Each E7 repeated as E5_Event, which is repeated as E4_Period (plus 19k historic Periods) and E2_Temporal_Entity
- 37% of all statements are type statements!

#### Class Statement

<table>
<thead>
<tr>
<th>Class</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>owl:Thing</td>
<td>36485904</td>
</tr>
<tr>
<td>E1_CRM_Entity</td>
<td>36485903</td>
</tr>
<tr>
<td>E77_Persistent_Item</td>
<td>17408450</td>
</tr>
<tr>
<td>E70_Thing</td>
<td>17339714</td>
</tr>
<tr>
<td>E71_Man-Made_Thing</td>
<td>17216212</td>
</tr>
<tr>
<td>E72_Legal_Object</td>
<td>17192518</td>
</tr>
<tr>
<td>E28_Conceptual_Object</td>
<td>14776488</td>
</tr>
<tr>
<td>E90_Symbolic_Object</td>
<td>14629292</td>
</tr>
<tr>
<td>E2_Temporal_Entity</td>
<td>11924877</td>
</tr>
<tr>
<td>E4_Period</td>
<td>11924877</td>
</tr>
<tr>
<td>E5_Event</td>
<td>11922986</td>
</tr>
<tr>
<td>E7_Activity</td>
<td>11796470</td>
</tr>
<tr>
<td>E63_Beginning_of_Existence</td>
<td>6377421</td>
</tr>
<tr>
<td>E11_Modification</td>
<td>6296015</td>
</tr>
<tr>
<td>E12_Production</td>
<td>6295825</td>
</tr>
<tr>
<td>rso:FC70_Thing</td>
<td>2051797</td>
</tr>
<tr>
<td>skos:Concept</td>
<td>415509</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>302149587</strong></td>
</tr>
</tbody>
</table>

*Lawyers of the world, rejoice!*
Reduce types: owl:Restriction vs Mixins

• Erlangen CRM states owl:Restrictions, e.g.:
  E72_Legal_Object SubClassOf: E70_Thing,
  P104_is_subject_to some E30_Right,
  P105_right_held_by some E39_Actor
  – M.Doerr has criticized this for ontological over-commitment
  – We don't need them so we cut them with XQuery tool deriving simpler profiles

• E72_Legal_Object:
  – Scope note: "material or immaterial items to which instances of E30 Right, such as the right of ownership or use, can be applied"
  – Do we really need it in the main hierarchy?

• Just state P104 domain, and E72 will be inferred as needed
  – Akin to Common Lisp mixins or Ruby traits

• PSNC gives up rdfs:subClassOf inference
  – Using OWLIM custom rules (flexibility is good!)
  – For one node, all classes can be found with SPARQL 1.1 Path queries
  – May be a bit drastic...
Complexity: Properties

Properties | Statements | Percent |
---|---|---|
rdf:type | 302149587 | 37.50% |
Objects: CRM, rdfs:label | 365430152 | 45.35% |
Extensions: BMO, RSO | 35903831 | 4.46% |
FRs (70M=9%) and sub-FRs (26M=3%) | 96526377 | 11.98% |
Thesauri: BIBO, DC, DCT, FOAF, SKOS, QUDT, VAEM | 5715250 | 0.71% |
Ontology: RDF, RDFS, OWL | 4159 | 0.00% |
**Total** | **805729356** | **100.00%** |

- Total 339 properties, grouped above
- Type statements take 37%: too much (see prev slides)
- Inverses (79) are convenient, but take 18% (duplicates)
- Sub-properties: max depth is 4 (e.g.: P12>P11>P14>P22). No estimate of the sub-property inference, sorry
- Objects take the majority: 45%
- Thesauri and ontologies are negligible: 0.7%
- FRs take only 12%, which doesn't slow OWLIM perceptibly
## Comparison to Other Repositories

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CRM</td>
<td>2.0 1</td>
<td>195 1</td>
<td>90 1</td>
<td>916 1</td>
<td>4.7 1</td>
<td>54 1</td>
<td>17.0 1</td>
<td>rdfs+tran+FR</td>
</tr>
<tr>
<td>PSNC</td>
<td>3.1 1.5</td>
<td>234 1.2</td>
<td>75 0.83</td>
<td>535 0.58</td>
<td>2.3 0.49</td>
<td>60 1.1</td>
<td>8.9 0.52</td>
<td>rdfs-subClass</td>
</tr>
<tr>
<td>EDM</td>
<td>20.3 9.8</td>
<td>998 5.1</td>
<td>50 0.56</td>
<td>3798 4.1</td>
<td>3.8 0.8</td>
<td>266 4.9</td>
<td>14.3 0.84</td>
<td>owl-horst</td>
</tr>
<tr>
<td>FF</td>
<td>1673 8.6</td>
<td>3211 3.5</td>
<td></td>
<td>10192 11</td>
<td>1.9 0.4</td>
<td>456 8.4</td>
<td>7.0 0.41</td>
<td>owl-horst</td>
</tr>
<tr>
<td>LLD</td>
<td>6706 34</td>
<td></td>
<td></td>
<td>1554 29</td>
<td>1.5 0.3</td>
<td></td>
<td>6.6 0.38</td>
<td>rdfs+tran</td>
</tr>
</tbody>
</table>

- **Repos:**
  - PSNC Polish Digital Library: [http://dl.psnc.pl](http://dl.psnc.pl)
  - Europeana EDM: [http://europeana.ontotext.com](http://europeana.ontotext.com)
  - FactForge: [http://www.factforge.net](http://www.factforge.net)
  - LinkedLifeData: [http://linkedlifedata.com](http://linkedlifedata.com)

- **First** col is Million triples (exc. Expansion/Density), **second** col is ratio to CRM
- **Expansion**=Total statements/Explicit statements: intensity of inference
- **Nodes**=unique URIs and literals
- **Density**=Statements/Nodes: relative density of the graph
• Straight SPARQL 1.1 for "FR92i_created_by rkd-artist:Rembrandt":

```
select distinct ?obj {
  ?obj a rso:FC70_Thing;
  (crm:P46_is_composed_of|crm:P106_is_composed_of|crm:P148_has_component|crm:P128_carries)*/
  (crm:P31i_was_modified_by|crm:P94i_was_created_by) / crm:P9_consists_of* /
  crm:P14_carried_out_by / crm:P107i_is_current_or_former_member_of*
  rkd-artist:Rembrandt
} limit 20
```

• RS endpoint takes over 15 minutes to answer. If you add more FRs, even worse. The reflexive * really kills it.

• The query can be optimized a bit by using intermediate variables instead of property paths, but the performance is still untenable.
• **Objects by Rembrandt:** sub-second response time:
  
  ```sql
  select distinct ?obj {?obj rso:FR92i_created_by rkd-artist:Rembrandt} limit 500
  ```

• **Find terms "drawing" and "mammal":**
  
  ```sql
  select * {?s rdfs:label "drawing"} → thes:x6544  
  select * {?s rdfs:label "mammal"} → thes:x12965
  ```

• **Drawings by Rembrandt about **mammals:** still sub-second response time, and the query is simple:
  
  ```sql
  select distinct ?obj {  
    ?obj rso:FR92i_created_by rkd-artist:Rembrandt;  
    rso:FR2_has_type thes:x6544, thes:x12965} limit 500
  ```

• **RS search takes 4.5s (significantly longer than the query alone)** because after obtaining up to 500 objects, it executes several more queries to fetch their display fields, facets, and images

• **Facets are loaded into the browser using Exhibit,** so subsequent facet restrictions are immediate
Thanks for listening!

Questions? vladimir.alexiev@ontotext.com