

A Possible Relational Schema for the Registry.

Paul Harrison

Motivations

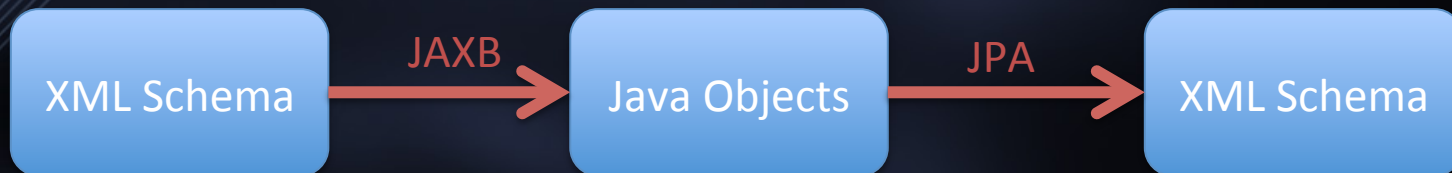
- Registry Interface Standard has standard query language which was a subset of the original ADQL/x 1.0
 - Not expressive enough
 - Only limited parts of the registry data model can be accessed
 - Only limited selection criteria
 - Rather verbose
- AstroGrid chose to implement registry as a native XML database
 - Registry content is defined by a set of XML schema – can be thought of as the Registry Data model (Registry “crown jewels”)
 - Implemented all internal AG inter-tool/service queries on the registry as XQuery
 - Very expressive – can specify exactly what is returned.
 - Standardized.
- At last Interop there was discussion about the limitations of current query language summarized by Ray Plante
 - <http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/RestfulRegistryInterfaceR>
 - Suggestion that query part could reuse TAP
 - Need a relational registry data model

Design Aims

- Reduce the number of Tables to the minimum
 - Makes queries easier
- Still able to express the entire XML schema model
- Be able to do the transform XML \leftrightarrow RDB with no ambiguities

Methodology

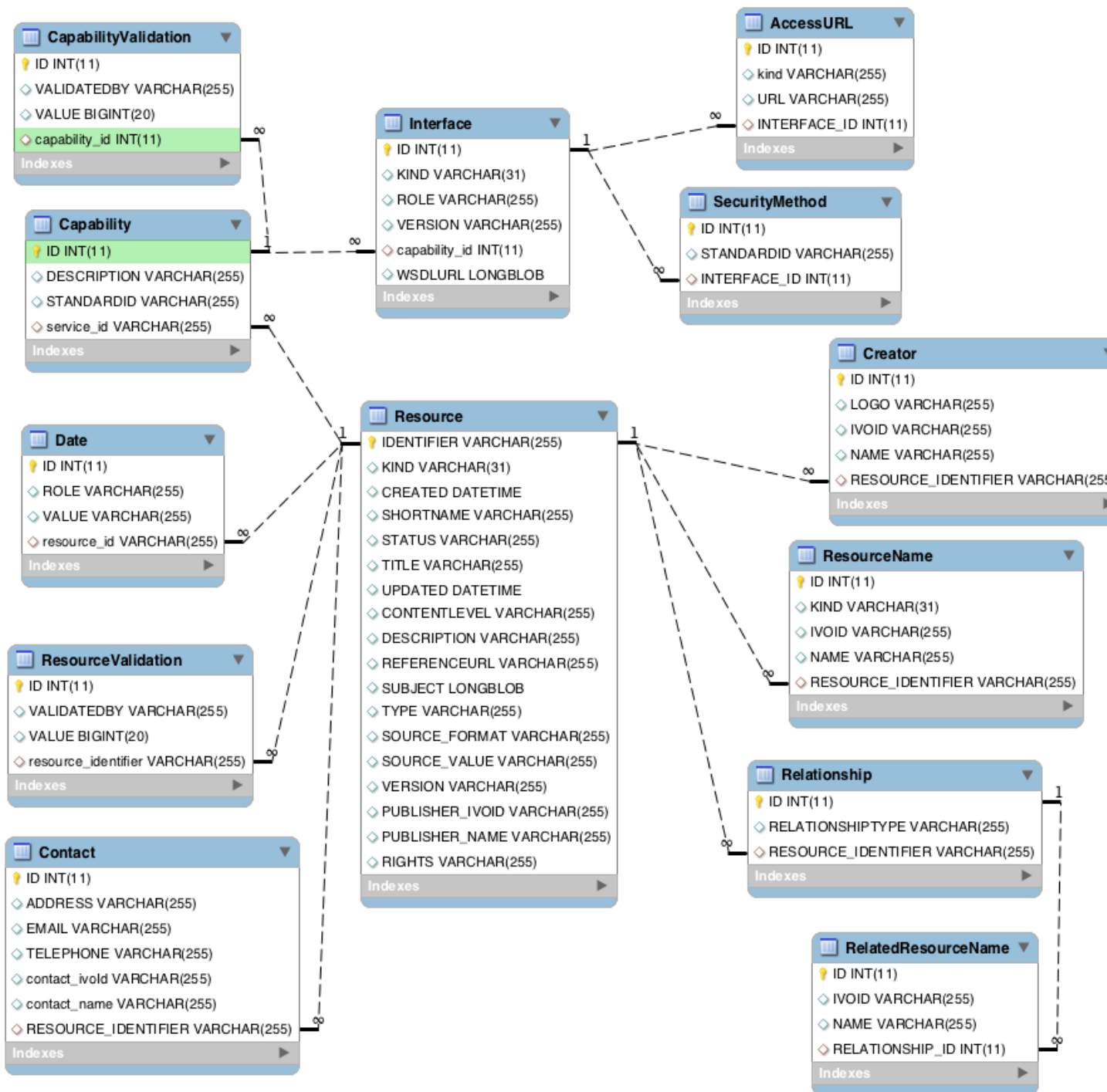
- Initially tried to use the VO-URP framework developed by the Theory WG.
 - Created UML model of Registry schema and then ran tooling
 - Does not regenerate XML schema exactly
- Second approach starting from XML Schema.
 - Use standard Java XML and Relational object mapping technologies
 - Allows flexibility of mapping via Java annotations



Result

VOResource only

- 13 Tables



Mappings – One to Many

- One-to-Many containment relation between types modelled as a separate table with a foreign key relation in the many side.
 - Possible because the registries are in general un-normalized so that there are no many-to-many relations which would require separate join tables.
 - E.g. there might be many “Creator” rows relating to the same actual creator
 - Could be argued that the registries should be more normalized
 - Loose definition of ResourceName (does not have to have a IVORN) responsible

Mappings - Subtypes

- Type hierarchies are collapsed into a single table
 - with a column “KIND” used to distinguish between the types
 - Properties of the sub-type are allowed to appear as nullable columns in the table.
- E.g. Resource type – has sub-types Service, Organization and Authority.

Mappings - Embedding

- If an XML type is a directly embedded member of parent – i.e. has a 1:1 cardinality in the XML – then its properties are added as extra columns in the parent table

E.g. Publisher

Adds two columns to Resource

Publisher_ivoid

Publisher_name

```
<title>NCSA Astronomy Digital Image Library
<shortName>ADIL</shortName>
<identifier>ivo://adil.ncsa/vocone</identifier>

<curation>
  <publisher ivo-id="ivo://adil.ncsa/adil">
    NCSA Astronomy Digital Image Library (A
  </publisher>
  <creator>
    <name>Dr. Raymond Plante</name>
    <logo>http://adil.ncsa.uiuc.edu/gifs/adil
  </creator>
  <date role="created">2002-01-01</date>
  <contact>
    <name>ADIL Librarian</name>
  <email>adil@ncsa.uiuc.edu</email>
  </contact>
</curation>
```


Mappings - Enumerations

- Lists of enumerations have been transformed to a single field with a delimited list of enumeration values
 - Eliminates the need for a separate table
- e.g. contentLevel, type

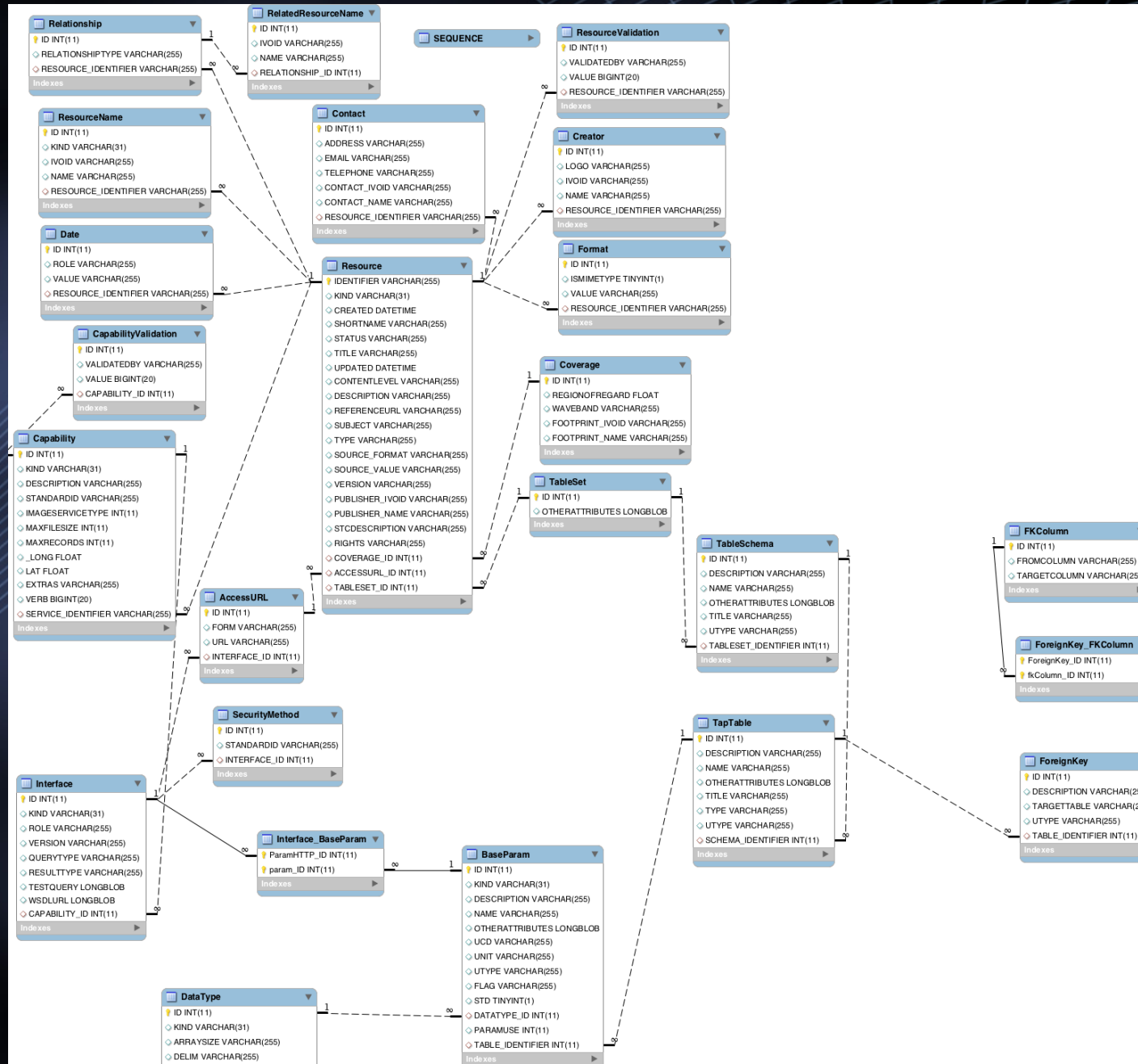
```
</curation>
<content>
  <subject>subject</subject>
  <subject>subject2</subject>
  <description>description</description>
  <source>source1</source>
  <referenceURL>http://tempuri.org</referenceURL>
  <type>Animation</type>
  <type>Education</type>
  <contentLevel>Amateur</contentLevel>
  <contentLevel>General</contentLevel>

```

Changes to existing XML schema.

- Could simplify the relational schema if some changes were made to existing XML Schema
 - reducing cardinalities eliminates table joins
- Allow only one RelatedResource inside a Relationship
 - Multiple relationships are possible already
- Allow only one AccessURL inside an Interface
 - No registry entries make use of this facility
- The above are almost zero impact from operational point of view as long as they are done retrospectively existing schema without changing namespace/version

VODataService Included!



Conclusions

- It is possible to map fully the XML schema to a relational schema
 - However this is not the “three-table-or-so” solution that was talked about – but to avoid the mistakes that were made in the simplified mapping of the 1.0 registry query language we should use the full mapping.
- IVOA should publish XML and RDB schema for all registry extensions.
- I have a Java “reference implementation” already
- The exercise did show some areas of the current XML schema that might be cleaned up.
- See <http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/RelationalRegistryDM> for more information.