Enhanced Content Models

State and University Library, Denmark

Open Repositories 2010
Duraspace user group

Asger Askov Blekinge
Kåre Fiedler Christiansen

Program

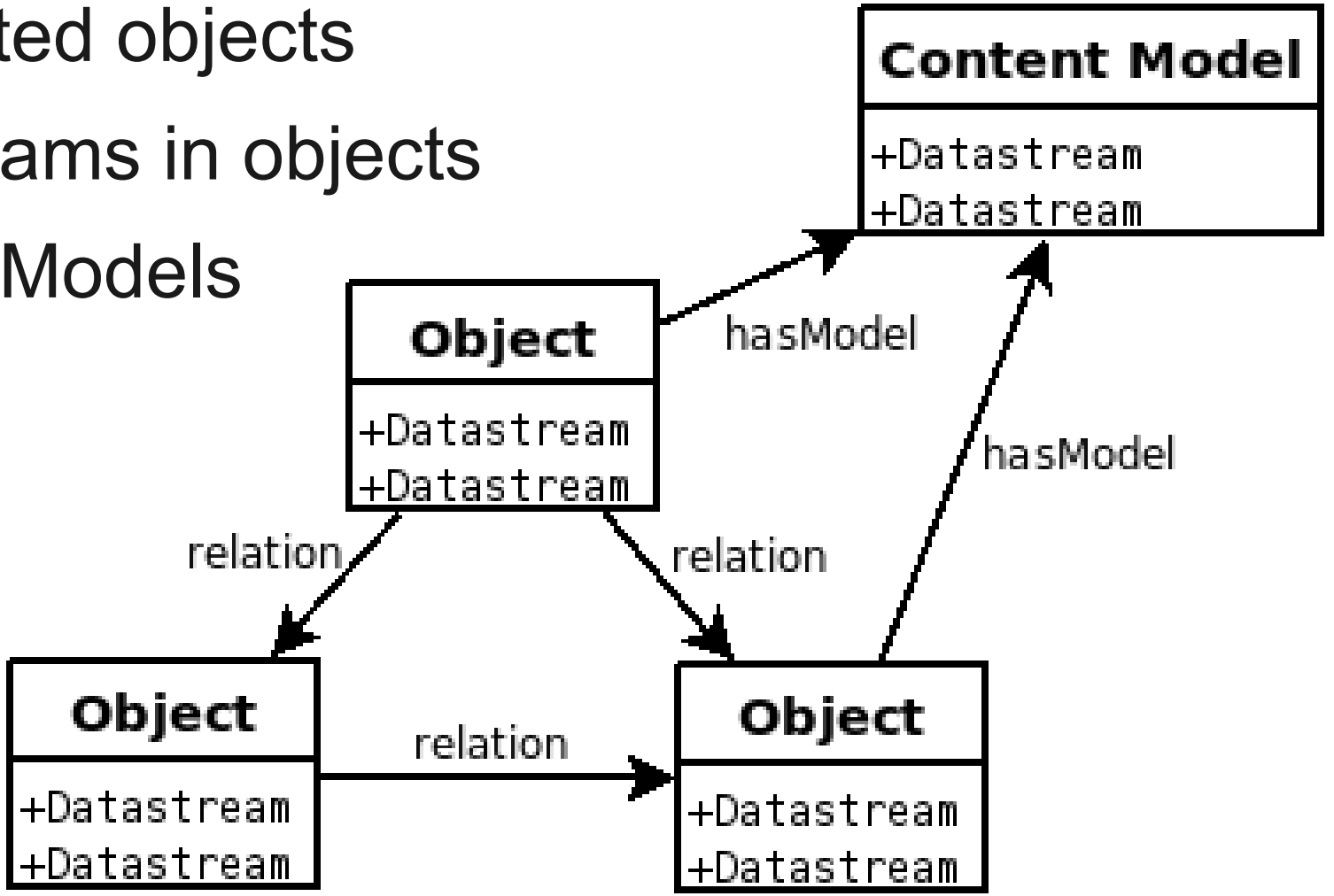
- Introduction
  - Fedora Objects and Content Models
- Enhanced Content Models
  - Optional Datastreams
  - Datastream Schemas
  - Extensibility

Program

  - Object Ontology
  - Datastream Ontology
- Content Model Driven Software
  - Validator

A look at Fedora Objects

- Interrelated objects
- Datastreams in objects
- Content Models



PreECM Content Models

- Content Models declare the classes of data objects
- Content Models declare the existince of datastreams in data objects

PreECM Content Models

- Content Models associate disseminators with data objects
- This is sufficient for many usecases!

Enhanced Content Models

- Extra information in Content Models. Backwards compatible.
- Optional datastreams
- ECMs declare the schemas for xml datastreams

Enhanced Content Models

- ECMs declare cardinality and range for object relations
- ECMs declare cardinality and range for datastream relations

Optional Datastreams

```
<dsCompositeModel>
    <dsTypeModel ID="RELS-INT" optional="true"/>
</dsCompositeModel>
```

Optional datastreams will be validated if present, but it is not an error to leave them out.

Fedora Object reserved datastreams

- DC (magic, required)
- RELS-EXT (optional)
- RELS-INT (optional)
- POLICY (optional)
- AUDIT (magic, optional)

Content Model datastreams

- RELS-EXT (required)
- DS-COMPOSITE-MODEL (optional)
- ONTOLOGY (new, optional)
  - More on this one later

Description Languages - Datastreams

- XMLSchema:
  - There are schemas for most xml metadata formats
  - XMLSchema is reversible.
  - Excellent tool support
  - Fedora is based on XML anyhow

Description Languages - Datastreams

```
<dsCompositeModel>
    <dsTypeModel ID="DC">
        <form MIME="text/xml"/>
    </dsTypeModel>
</dsCompositeModel>
```

Description Languages - Datastreams

```
<dsCompositeModel>
    <dsTypeModel ID="DC">
        <form MIME="text/xml"/>




    </dsTypeModel>
</dsCompositeModel>
```

Description Languages - Datastreams

```
<dsCompositeModel>
    <dsTypeModel ID="DC">
        <form MIME="text/xml"/>
        <extension name="SCHEMA">

        </extension>
    </dsTypeModel>
</dsCompositeModel>
```

Description Languages - Datastreams

```
<dsCompositeModel>
    <dsTypeModel ID="DC">
        <form MIME="text/xml"/>
        <extension name="SCHEMA">
            <reference type=”xsd” datastream=”OAI_DC-SCHEMA”/>
        </extension>
    </dsTypeModel>
</dsCompositeModel>
```

Description Languages - Datastreams

```
<foxml:datastream
    ID="OAI_DC-SCHEMA"
    CONTROL_GROUP="E"
    STATE="A"
    VERSIONABLE="false">
```

Description Languages - Datastreams

```
<foxml:datastreamVersion
    ID="OAI_DC-SCHEMA1.0"
    LABEL="OAI DC xml schema"
    MIMETYPE="text/xml">
        <foxml:contentLocation
            TYPE="URL"
         REF="http://www.openarchives.org/OAI/2.0/oai_dc.xsd"/>
</foxml:datastreamVersion>

</foxml:datastream>
```

Extensibility

```
<dsCompositeModel>
        <dsTypeModel ID="DC">
                <form MIME="text/xml"/>
                <extensions name="MY_EXTENSION">
                        <demoXml>
                                <withStuff/>
                        </demoXml>
                </extensions>
        </dsTypeModel>
</dsCompositeModel>
```

Description Languages - RDF

- OWL Lite:
  - Fedora use RDF.
  - Restrictions on RDF should be in OWL.
  - Lite means that we can still reason about it

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class"/>
</rdf:RDF>
```

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class">
   </owl:Class>
</rdf:RDF>
```

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class">




   </owl:Class>
</rdf:RDF>
```

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class">




   </owl:Class>
   <owl:ObjectProperty
         rdf:about="info:fedora/fedora-system:def/model#hasModel"/>
</rdf:RDF>
```

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class">
     <rdfs:subClassOf>
        <owl:Restriction>
             <owl:onProperty
                  rdf:resource="info:fedora/fedora-system:def/model#hasModel"/>



        </owl:Restriction>
     </rdfs:subClassOf>
   </owl:Class>
   <owl:ObjectProperty
        rdf:about="info:fedora/fedora-system:def/model#hasModel"/>
</rdf:RDF>
```

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class">
     <rdfs:subClassOf>
        <owl:Restriction>
             <owl:onProperty
                  rdf:resource="info:fedora/fedora-system:def/model#hasModel"/>
             <owl:allValuesFrom
                  rdf:resource="info:fedora/fedora-system:ContentModel-
3.0_class"/>
        </owl:Restriction>
     </rdfs:subClassOf>
   </owl:Class>
   <owl:ObjectProperty
        rdf:about="info:fedora/fedora-system:def/model#hasModel"/>
</rdf:RDF>
```

Ontology datastream

```
<rdf:RDF>
   <owl:Class rdf:about="info:fedora/fedora-system:FedoraObject-3.0_class">
     <rdfs:subClassOf>
        <owl:Restriction>
           .....
        </owl:Restriction>
     </rdfs:subClassOf>
     <rdfs:subClassOf>
        <owl:Restriction>
              <owl:onProperty
                    rdf:resource="info:fedora/fedora-system:def/model#hasModel"/>
              <owl:minCardinality>1</owl:minCardinality>
        </owl:Restriction>
     </rdfs:subClassOf>
   </owl:Class>
   <owl:ObjectProperty
         rdf:about="info:fedora/fedora-system:def/model#hasModel"/>
</rdf:RDF>
```

Ontology datastream

There are 5 kinds of Restrictions supported at the moment

- MinCardinality
- MaxCardinality
- Cardinality

Ontology datastream

- AllValuesFrom
- SomeValuesFrom

The ontology is Open, ie. Dataobjects can have more relations than the ones declared in the content model

Datastream Relations

- Datastream relations declared in RELS-INT
- Always have the datastream as the subject
- The ontology is part of the ONTOLOGY datastream, just like the ontology for the RELS-EXT relations.

ONTOLOGY Datastream

```
<rdf:RDF>
    <owl:Class
        rdf:about="info:fedora/fedora-system:FedoraObject-
3.0_class">
        .....
    </owl:Class>
    <owl:Class
        rdf:about="info:fedora/fedora-system:FedoraObject-
3.0/DC_class"/>
</rdf:RDF>
```

Rounding up

- Optional datastreams
- Datastream xsd schemas
- Object Relations ontology
- Datastream Relations ontology
- Extensibility
- The 4 basic content models have been enhanced

Why enhancing the content models

- Precise descriptions of the data objects allow software to reflect upon this ie. Content Model Driven Software
- Encoding of the data model in the content models, not in the surrounding software

Why enhancing the content models

- Enhanced Content Models are now “complete”
- We would like to know about how they are used, and their shortcomings
- What is needed is a Best Practise for content models

Validate method

- A precise description is much more useful, if you can ensure that the object adheres to the description
- Will validate against each of the content models in turn
- validate(pid, asOfDateTime)
- Do not use the Resource Index

Validate Result

- Object PID
- Valid boolean
- AsOfDateTime
- Content Models

Validate Result

- Object Problems
  - List of problems concerning RELS-EXT
- Datastream Problems
  - List of problems for each datastream

Validate Result

```
<validation pid="demo:testObject" valid="false">
    <asOfDateTime>2007-10-26T08:36:28</asOfDateTime>
    <contentModels>
        <model>info:fedora/demo:ContentModelTest</model>
        <model>info:fedora/fedora-system:FedoraObject-3.0</model>
    </contentModels>
```

Validate Result

```
<problems>
   <problem>Relation 'http://demoRelations/next' refers to
            resource 'demo:dataObject2' which, by content
            model 'demo:ContentModelTest'
            should be of the type 'demo:ContentModelTest'
   </problem>
</problems>
```

Validate Result

```
    <datastreamProblems>
        <datastream datastreamID="DC">
            <problem>Datastream 'DC' is required by the content
                        model 'fedora-system:FedoraObject-3.0'
            </problem>
        </datastream>
    </datastreamProblems>
</validation>
```

Validate Result

- Stuff still to do
  - Errorcodes or something like this for machine parsable error handling. We need feedback
  - 3.4 RC 1 validator does not do RELS-INT validation. This will be fixed in 3.4 Final

Building a better CMA

- Basic CMA was the first step
- These enhancements are the second step.
- We do not know how many steps are needed
- For software to be content model driven, we need a general language for content models, a Best Practise.
- Give us feedback on how you use them, and especially how they are insufficient.

Rounding up

- Websites
  - https://wiki.duraspace.org/display/FCREPO/Enhanced+Content+Models
  - http://tinyurl.com/2d537ka
  - Will be moved to the proper location in the Fedora wiki, but this location will forward
  - Email: abr@statsbiblioteket.dk

Rounding up

- This work has been funded by
  - DEFF, Denmark's Electronic Research Library
  - State and University Library, Denmark