XNAT Customizations

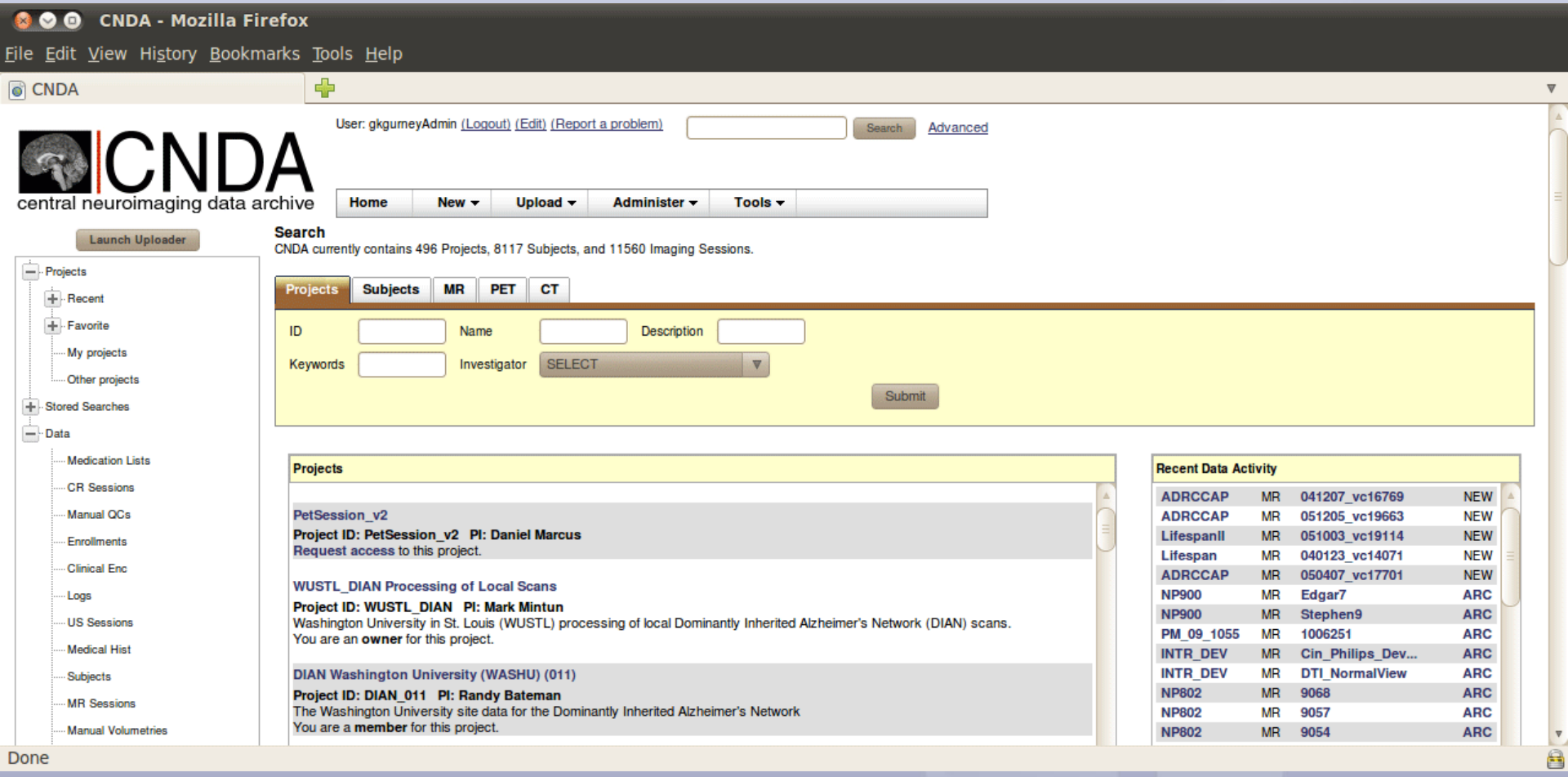
The ability to store all your data the way you want

XNAT

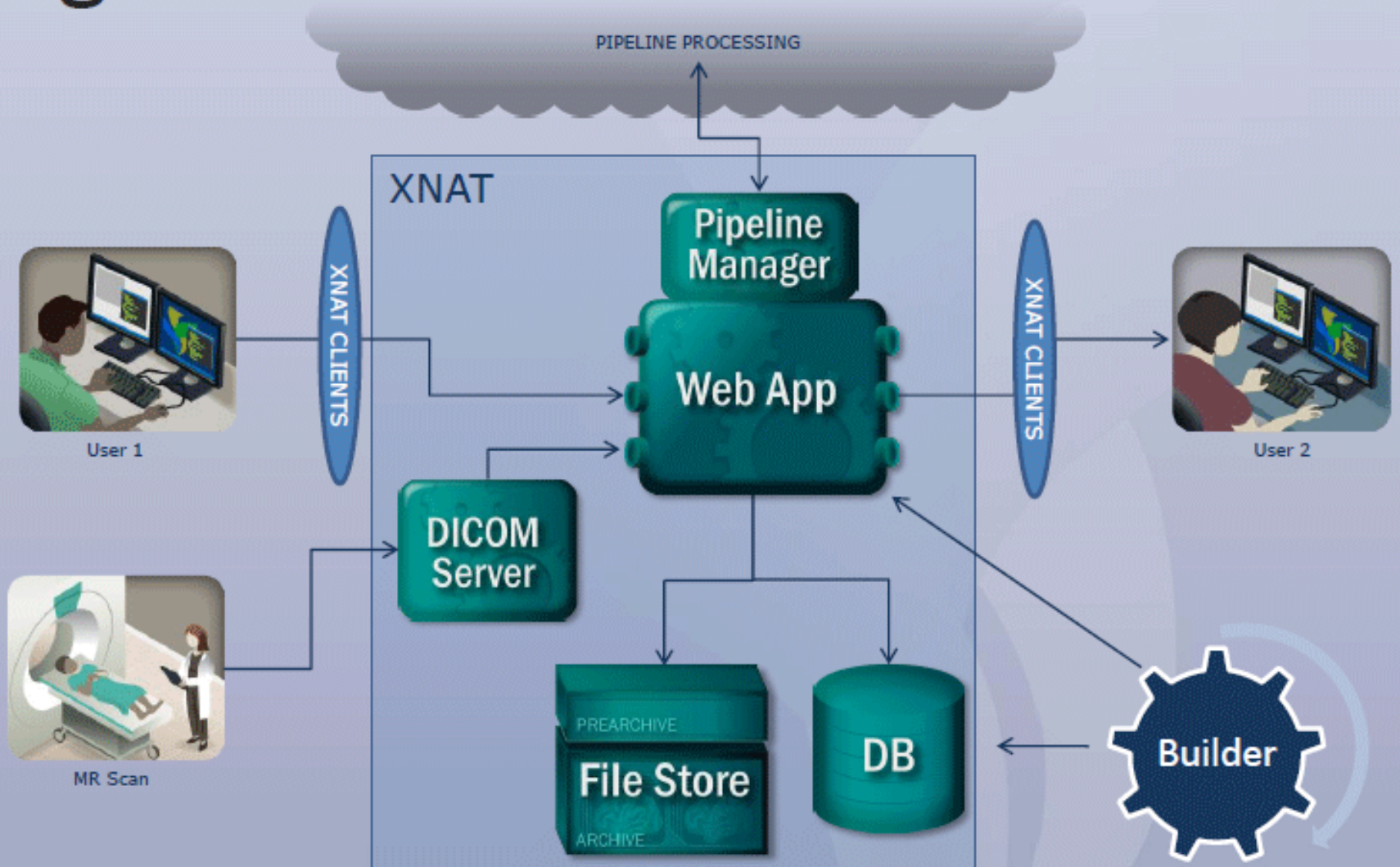
Contact info

- Jenny Gurney
- gurneyj@wustl.edu

XNAT


CNDA - Mozilla Firefox
File Edit View History Bookmarks Tools Help
CNDA
User: gkgurneyAdmin (Logout) (Edit) (Report a problem)
Search
Advanced
CNDA
central neuroimaging data archive
Home
New
Upload
Administer
Tools
Launch Uploader
Projects
Recent
Favorite
My projects
Other projects
Stored Searches
Data
Medication Lists
CR Sessions
Manual QCs
Enrollments
Clinical Enc
Logs
US Sessions
Medical Hist
Subjects
MR Sessions
Manual Volumetries
Search
CNDA currently contains 496 Projects, 8117 Subjects, and 11560 Imaging Sessions.
Projects
Subjects
MR
PET
CT
ID
Name
Description
Keywords
Investigator
SELECT
Submit
Projects
PetSession_v2
Project ID: PetSession_v2 PI: Daniel Marcus
Request access to this project.
WUSTL_DIAN Processing of Local Scans
Project ID: WUSTL_DIAN PI: Mark Mintun
Washington University in St. Louis (WUSTL) processing of local Dominantly Inherited Alzheimer's Network (DIAN) scans.
You are an owner for this project.
DIAN Washington University (WASHU) (011)
Project ID: DIAN_011 PI: Randy Bateman
The Washington University site data for the Dominantly Inherited Alzheimer's Network
You are a member for this project.
Recent Data Activity
ADRCCAP MR 041207_vc16769 NEW
ADRCCAP MR 051205_vc19663 NEW
LifespanII MR 051003_vc19114 NEW
Lifespan MR 040123_vc14071 NEW
ADRCCAP MR 050407_vc17701 NEW
NP900 MR Edgar7 ARC
NP900 MR Stephen9 ARC
PM_09_1055 MR 1006251 ARC
INTR_DEV MR Cin_Philips_Dev... ARC
INTR_DEV MR DTI_NormalView ARC
NP802 MR 9068 ARC
NP802 MR 9057 ARC
NP802 MR 9054 ARC
Done

Big Picture


PIPELINE PROCESSING
XNAT
Pipeline Manager
Web App
DICOM Server
PREARCHIVE
File Store
ARCHIVE
DB
XNAT CLIENTS
XNAT CLIENTS
User 1
User 2
MR Scan
Builder

In this talk

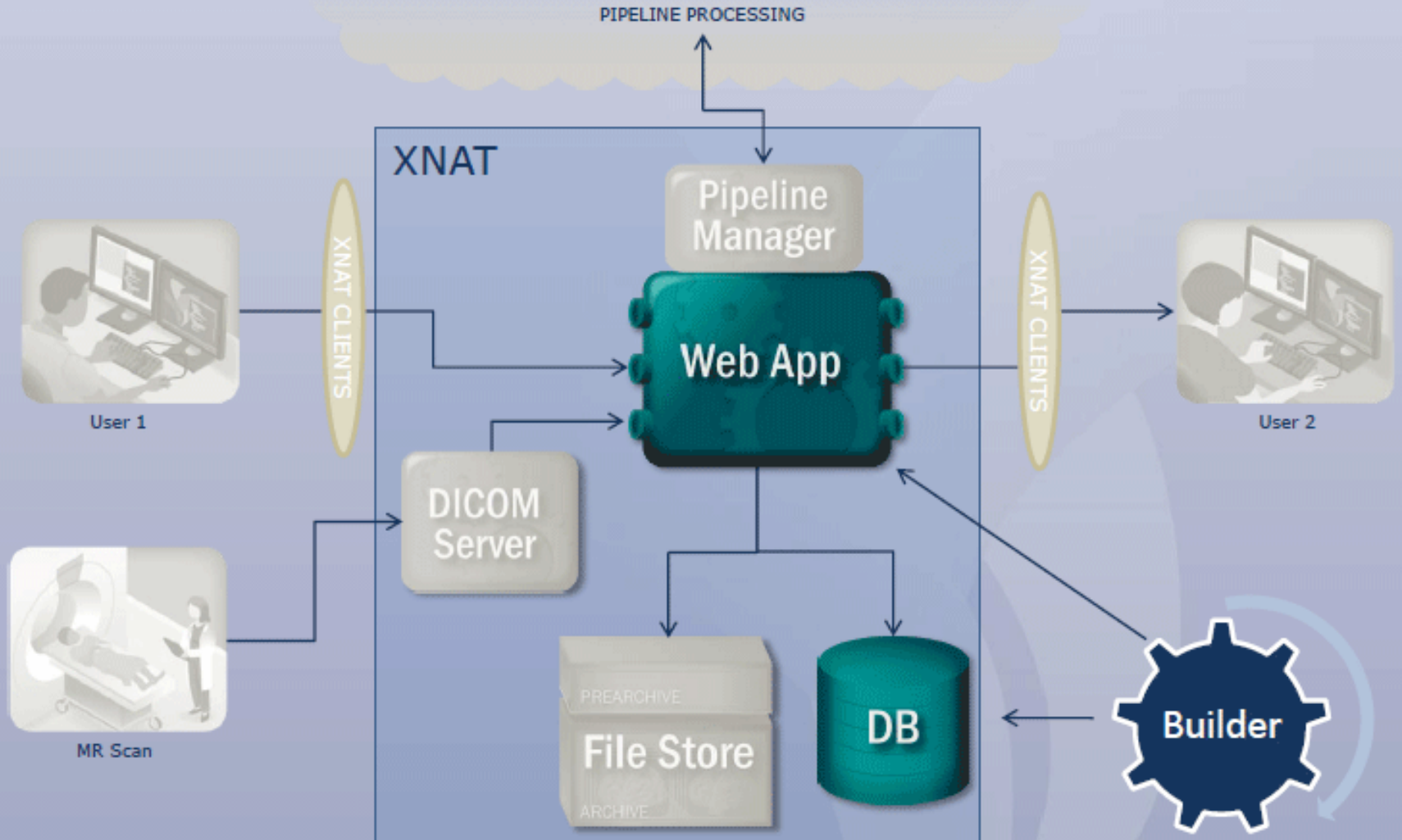
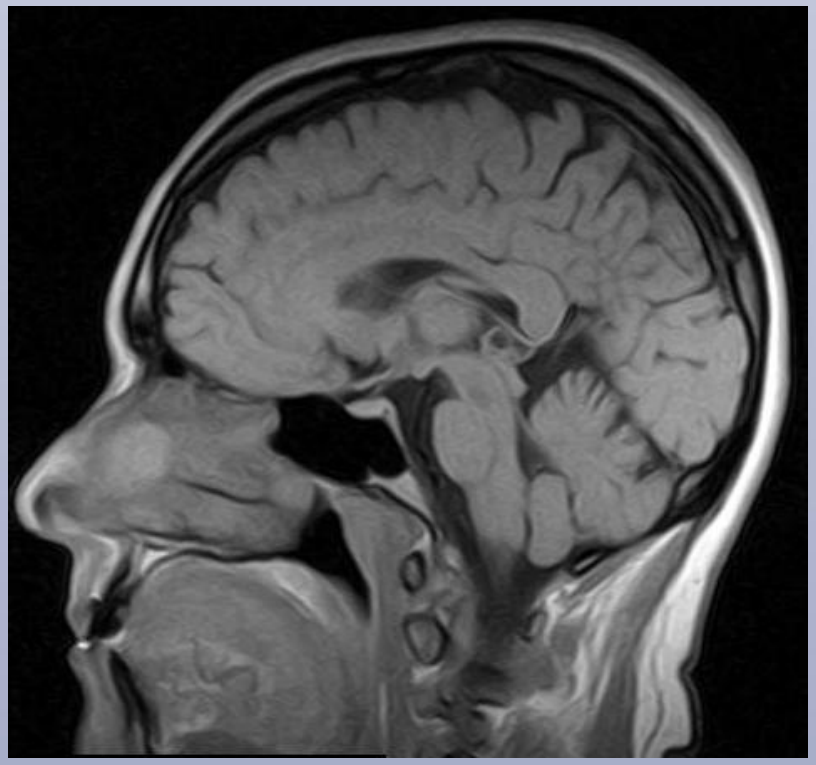

PIPELINE PROCESSING
XNAT
Pipeline Manager
Web App
XNAT CLIENTS
XNAT CLIENTS
User 1
User 2
DICOM Server
MR Scan
PREARCHIVE
File Store
ARCHIVE
DB
Builder

Image Assessor Data-Type Example: Radiology Read



Research Radiological Assessment

XNATDemo_E00037_RAD_1277750461318 / 000002_RAD_1277750461318

Subject:	TOSHIBA_TARO
Date of Birth:	
Age at Scan:	--
Cohort:	
Session Id:	000002
Date of scan:	
Type:	
Scanner:	000000000
Date of assessment:	2010-06-28
Reader:	A. Admin
Exam:	Brain with contrast
History:	
Technique:	
Comparison:	
Finding:	There are scattered nonspecific subcortical white matter changes in both cerebral hemispheres. The white matter changes comprise less than 25% of the total white matter volume.
Diagnosis/Impression:	Minor nonspecific subcortical white matter changes
Status:	
Recommend Further Evaluation:	false

XNAT

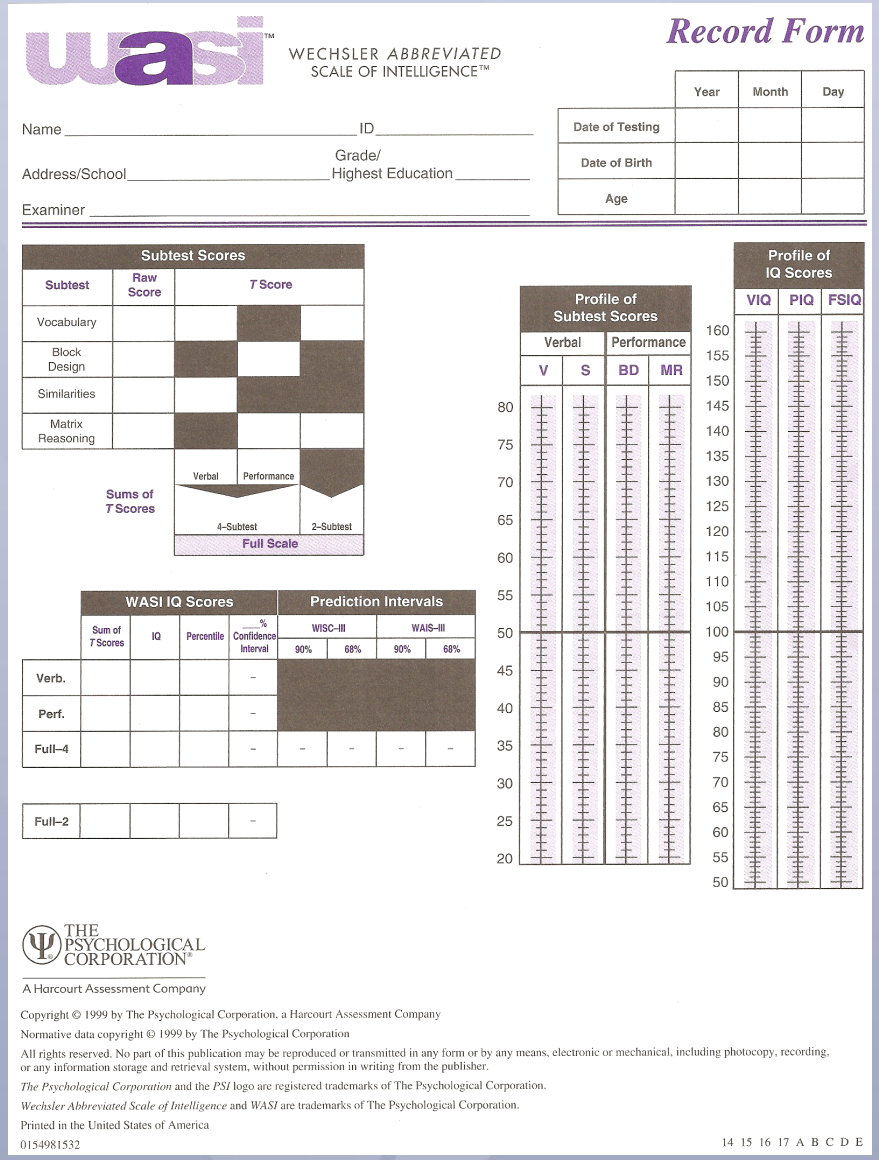
Subject Assessor Data-Type Example: IQ Assessment

IQ Assessment

Derived Full Scale IQ (FSIQ)	140
Verbal IQ	
Performance IQ	
IQ Source	WASI

WASI Assessment

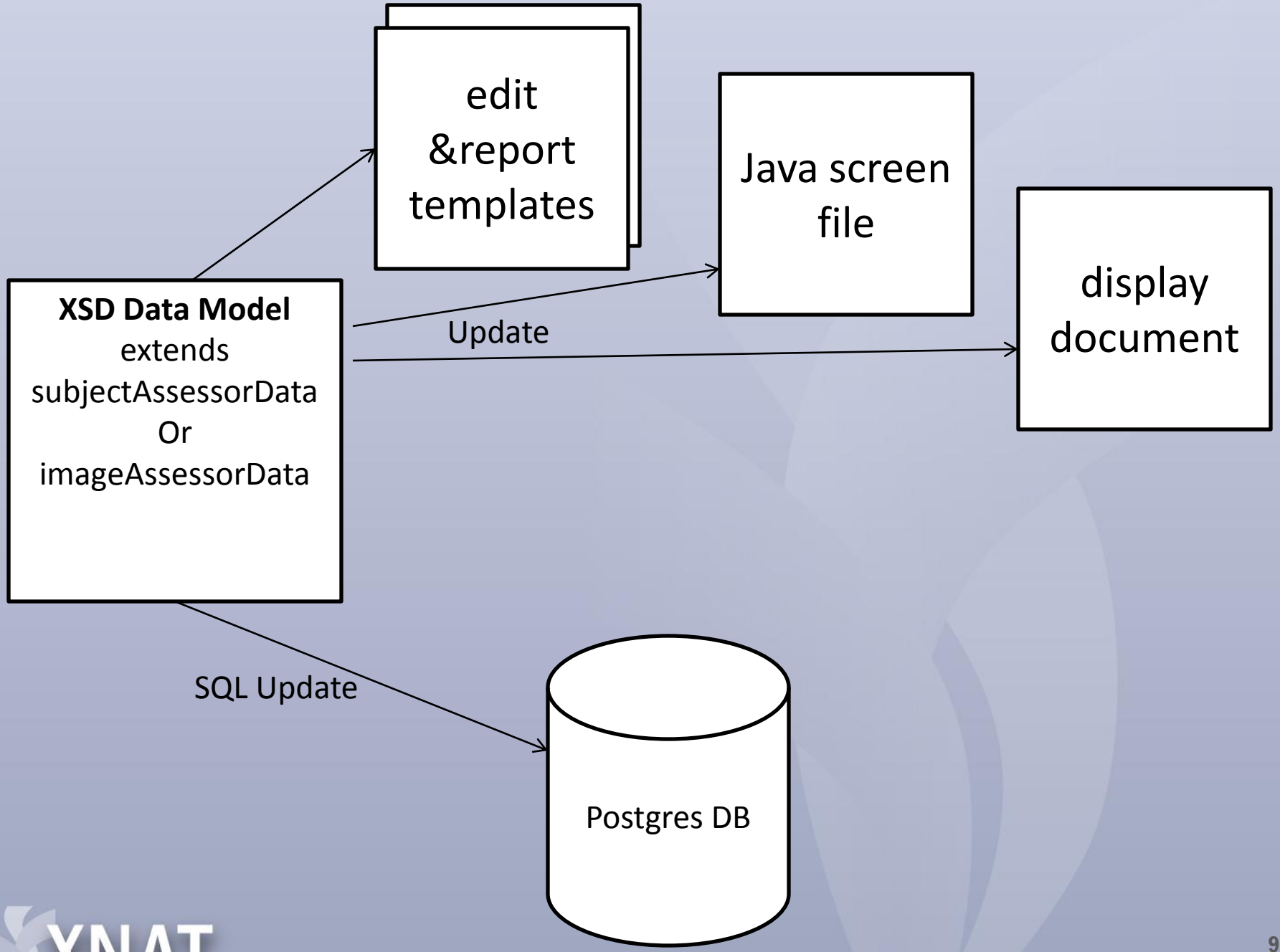
Description	Raw Score	T-Score	Notation
Vocabulary	42	64	
Block	48	76	
Similarities	31	69	
Matrix Reasoning	28	68	

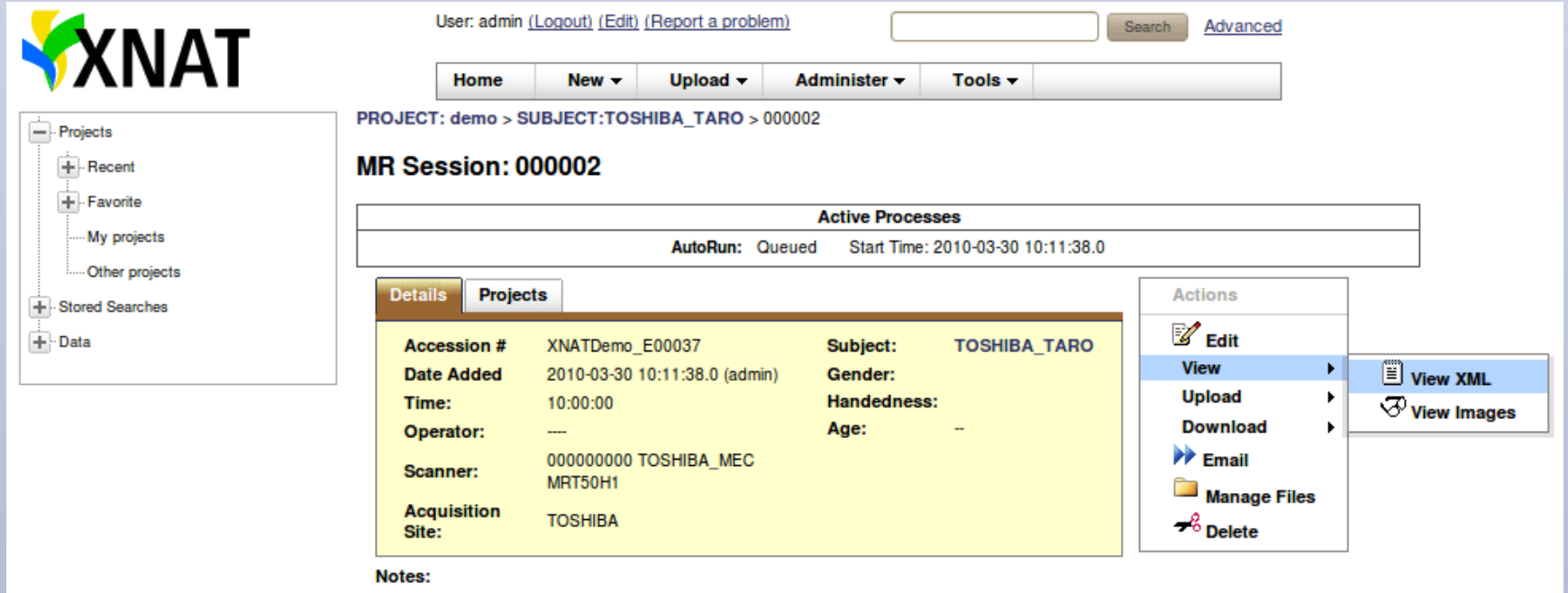


3 Common XNAT Customizations

- Extend XNAT data model with new data-types
- Modify the graphical user interface
- Customize data listings

XNAT


edit
&report
templates
Java screen
file
display
document
XSD Data Model
extends
subjectAssessorData
Or
imageAssessorData
Update
SQL Update
Postgres DB



http://localhost:8080/xnat/schemas/iq/iq.xsd http://nrg.wustl.edu/catalog http://localhost:8080/xnat/schemas/catalog/catalog.xsd http://nrg.wustl.edu/pipe
http://localhost:8080/xnat/schemas/pipeline/repository.xsd http://nrg.wustl.edu/arc http://localhost:8080/xnat/schemas/project/project.xsd
http://nrg.wustl.edu/xnat http://localhost:8080/xnat/schemas/xnat/xnat.xsd http://nrg.wustl.edu/uds http://localhost:8080/xnat/schemas/uds/uds.xsd
http://nrg.wustl.edu/xnat_assessments http://localhost:8080/xnat/schemas/assessments/assessments.xsd http://www.nbirn.net/prov http://localhost:8080
/xnat/schemas/birn/birnprov.xsd http://nrg.wustl.edu/imgassr http://localhost:8080/xnat/schemas/imgassr/imgassr.xsd http://nrg.wustl.edu/security
http://localhost:8080/xnat/schemas/security/security.xsd">
<xnat:time>10:00:00</xnat:time>
<xnat:acquisition_site>TOSHIBA</xnat:acquisition_site>
<xnat:subject_ID>XNATDemo_S00032</xnat:subject_ID>
<xnat:scanner manufacturer="TOSHIBA_MEC" model="MRT50H1">000000000</xnat:scanner>
<xnat:operator>----</xnat:operator>
−<xnat:scans>
−<xnat:scan ID="3" UID="1.2.840.113654.2.45.257.31197" xsi:type="xnat:mrScanData">
<xnat:image_session_ID>XNATDemo_E00037</xnat:image_session_ID>
<xnat:quality>usable</xnat:quality>
<xnat:scanner manufacturer="TOSHIBA_MEC" model="MRT50H1">000000000</xnat:scanner>
<xnat:frames>1</xnat:frames>
<xnat:file label="DICOM" URI="/home/xnat/xnat_data/archive/demo/arc001/000002/SCANS/3/DICOM/scan_3_catalog.xml" format="DICOM"
content="RAW" xsi:type="xnat:resourceCatalog"/>
−<xnat:parameters>
<xnat:voxelRes x="1.1719" y="1.1719" z="5.0"/>
<xnat:orientation>Cor</xnat:orientation>

XML – Extensible Markup Language

```
<?xml version="1.0"?>
<note>
    <to>John</to>
    <from>Jane</from>
    <heading>Reminder</heading>
    <body>Don't forget the meeting!</body>
</note>
```

XNAT

XSD – XML Schema Definition

```
<?xml version="1.0"?>
  <xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  </xs:schema>
```

XNAT

Prefix

URI

```
<?xml version="1.0"?>
  <xs:schema targetNamespace="http://nrg.wustl.edu/msg"
   xmlns:msg="http://nrg.wustl.edu/msg"
   xmlns:xs="http://www.w3.org/2001/XMLSchema">

   <xs:element name="note">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>

   </xs:schema>
```

XNAT

```
<?xml version="1.0"?>
<msg:note xsi:schemaLocation="
   http://nrg.wustl.edu/msg"
   https://cnda.wustl.edu/schemas/msg/msg.xsd>
   <msg:to>John</msg:to>
   <msg:from>Jane</msg:from>
   <msg:heading>Reminder</msg:heading>
   <msg:body>Don't forget the meeting!</msg:body>
</msg:note>
```

XNAT

Where to Store New XSDs

- projects
  - xnat
    - src
      - schemas
        - imgassr
          - imgassr.xsd
        - iq
          - iq.xsd

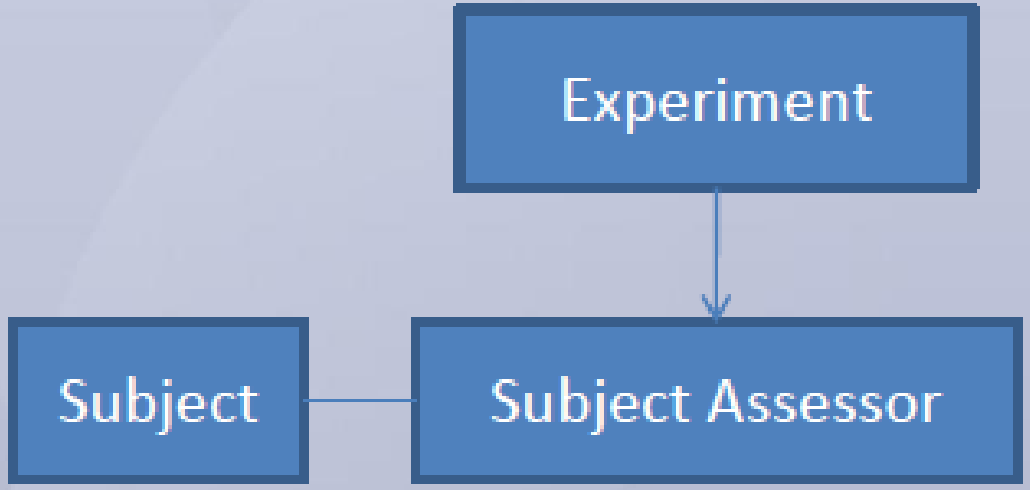
Benefits of extending subjectAssessorData or imageAssessorData

1. Template for your data-type
2. Automatic generation through the update process of edit and report screens, display documents (to help with search) and a helper Java screen class.
3. Data type automatically accessible through REST API
4. Easier to get support from XNAT group

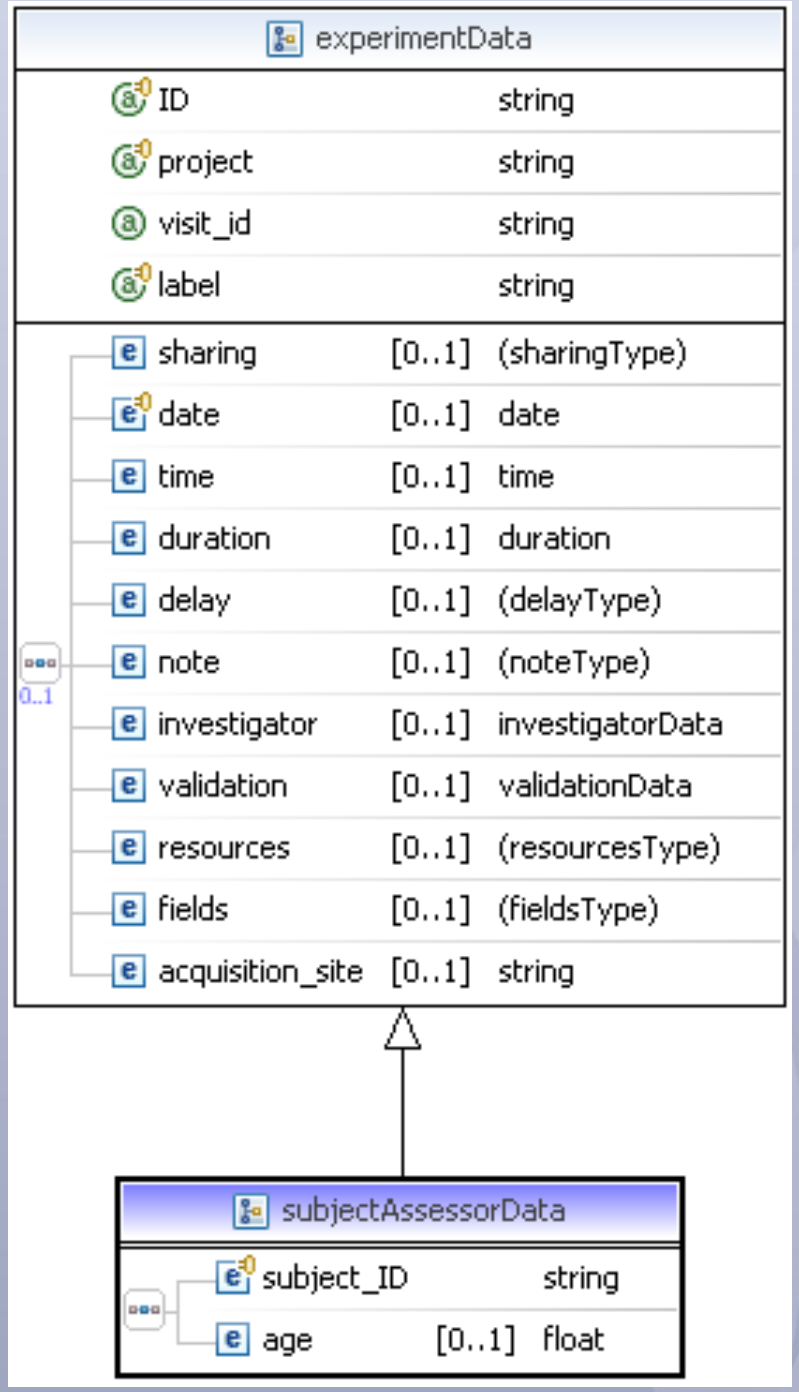
Extending the subjectAssessorData or imageAssessorData

1. Choose a namespace
2. Create a new XSD file for the new namespace, under the projects directory (if necessary)
3. Define the type extending subjectAssessorData or imageAssessorData
4. Create global/root element for your new type

Extending the Subject Assessor



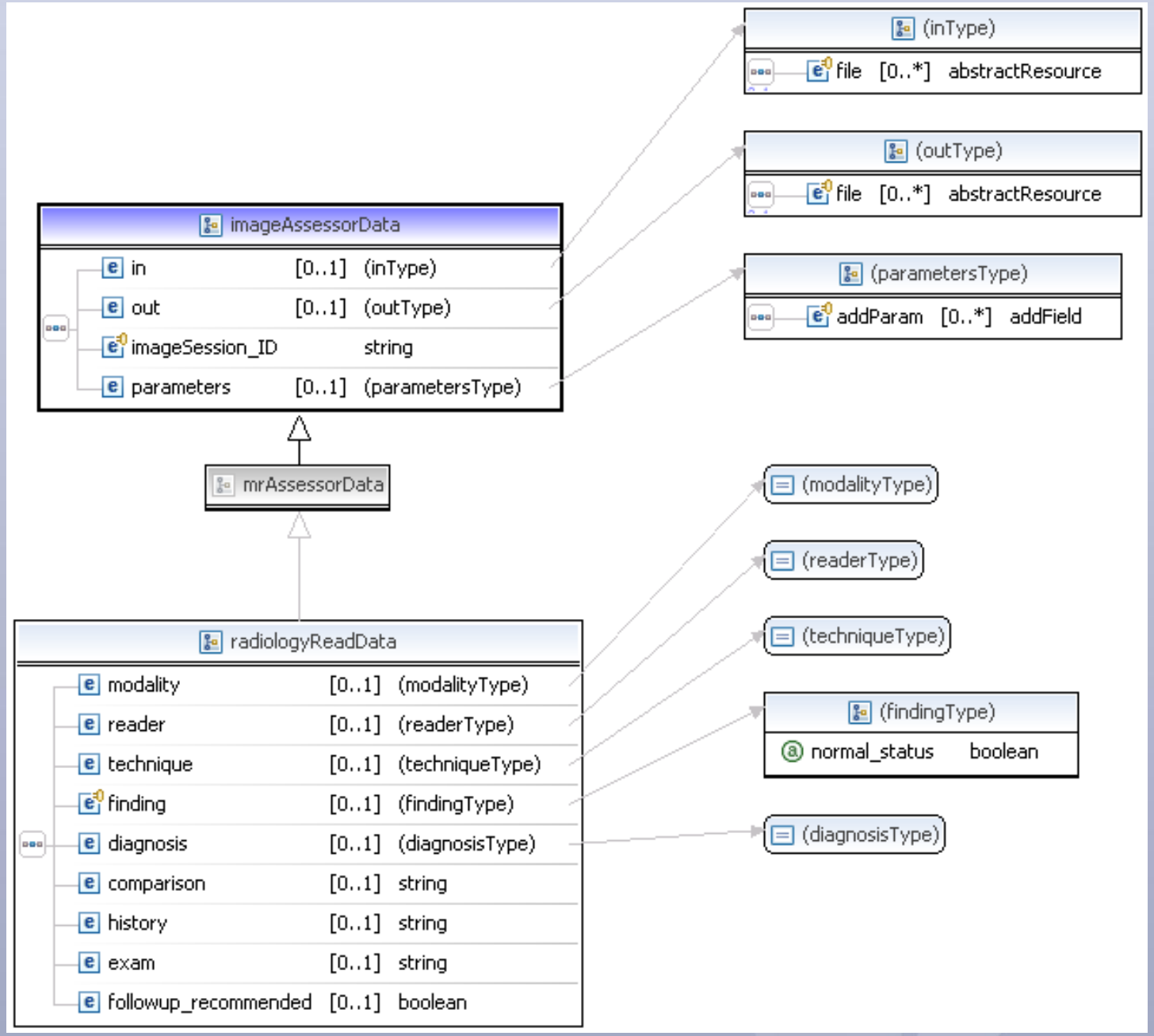
XNAT

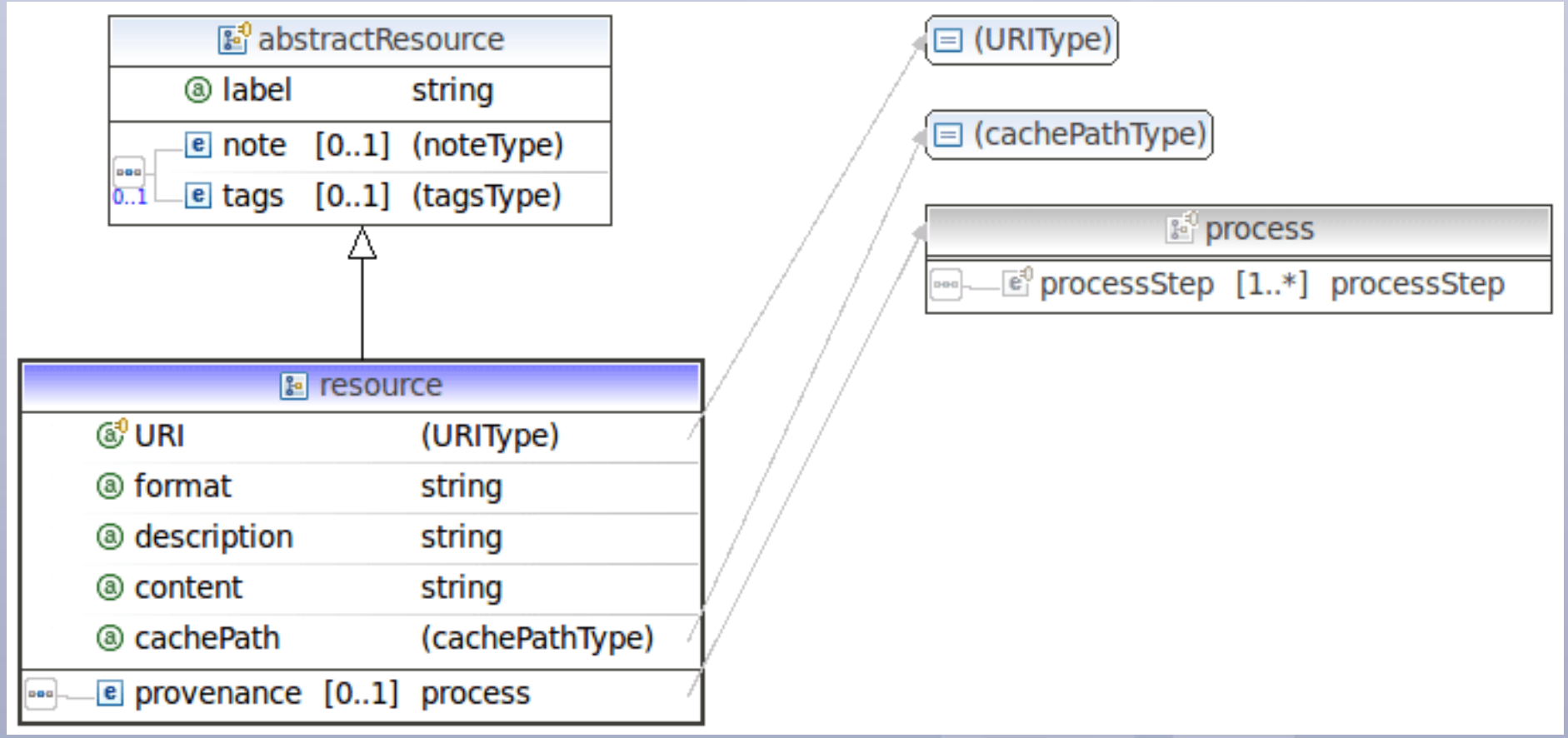

experimentData
ID string
project string
visit_id string
label string
sharing [0..1] (sharingType)
date [0..1] date
time [0..1] time
duration [0..1] duration
delay [0..1] (delayType)
note [0..1] (noteType)
0..1
investigator [0..1] investigatorData
validation [0..1] validationData
resources [0..1] (resourcesType)
fields [0..1] (fieldsType)
acquisition_site [0..1] string
subjectAssessorData
subject_ID string
age [0..1] float

Demonstration

XML Graphical Design Tools

- Eclipse IDE
  - By Eclipse, http://www.eclipse.org/downloads
  - Java or Java EE Developers version
  - Free
- XMLSpy
  - By Altova, http://www.altova.com
  - Professional or Enterprise Edition
  - Licensed
- Oxygen
  - By <oXygen/>, http://www.oxygenxml.com
  - XML Editor
  - Licensed


(inType)
file [0..*] abstractResource
(outType)
file [0..*] abstractResource
(parametersType)
addParam [0..*] addField
imageAssessorData
in [0..1] (inType)
out [0..1] (outType)
imageSession_ID string
parameters [0..1] (parametersType)
mrAssessorData
(modalityType)
(readerType)
(techniqueType)
(findingType)
normal_status boolean
(diagnosisType)
radiologyReadData
modality [0..1] (modalityType)
reader [0..1] (readerType)
technique [0..1] (techniqueType)
finding [0..1] (findingType)
diagnosis [0..1] (diagnosisType)
comparison [0..1] string
history [0..1] string
exam [0..1] string
followup_recommended [0..1] boolean


abstractResource
label string
note [0..1] (noteType)
tags [0..1] (tagsType)
0..1
resource
URI (URIType)
format string
description string
content string
cachePath (cachePathType)
provenance [0..1] process
(URIType)
(cachePathType)
process
processStep [1..*] processStep

Unbounded Elements

- Avoid unbounded elements
- Use <appinfo> element to set unique identifier
- Wrap in parent element of maxoccurs=1

<appinfo>

```
<xs:element name="fields" minOccurs="0">
    <xs:complexType>
        <xs:sequence minOccurs="0">
            <xs:element name="field" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:appinfo>
                        <xdat:field>
                            <xdat:relation uniqueComposite="NAME"/>
                        </xdat:field>
                    </xs:appinfo>
                </xs:annotation>
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attribute name="name" type="xs:string">
                                <xs:annotation>
                                    <xs:appinfo>
                                        <xdat:field uniqueComposite="NAME"/>
                                    </xs:appinfo>
                                </xs:annotation>
                            </xs:attribute>
```

Unbounded Elements

- Avoid unbounded elements
- Use <appinfo> element to set unique identifier
- Wrap in parent element of maxoccurs=1

XNAT

XNAT Schema Best Practices Summary

- Keep names short
- Limit unbounded elements
- Use annotation tags to specify unique field(s) for database entries
- Wrap unbounded elements in a parent element with a maxoccurs = 1
- Limit enumerations
- Use namespace prefix for directory, filename, and prefix.

Enable New Schemas in XNAT

projects/xnat/InstanceSettings.xml

```
<Instance_Settings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="%XDAT_PROJECT%/schemas/xdat/
    <Databases>
        <Database Type="POSTGRESQL" Id="%DB_NAME%" Driver="%DB_DRIVER%" Url="%DB_URL%" User="%USER%" Pass="%PASSWORD%" MaxConnection
    </Databases>
    <Models>
        <Data_Model File_Name="security.xsd" File_Location="%XDAT_PROJECT%/schemas/security" DB="%DB_NAME%"/>
        <Data_Model File_Name="birnprov.xsd" File_Location="%XDAT_PROJECT%/schemas/birn" DB="%DB_NAME%"/>
        <Data_Model File_Name="xnat.xsd" File_Location="%XDAT_PROJECT%/schemas/xnat" DB="%DB_NAME%"/>
        <Data_Model File_Name="workflow.xsd" File_Location="%XDAT_PROJECT%/schemas/pipeline" DB="%DB_NAME%"/>
        <Data_Model File_Name="repository.xsd" File_Location="%XDAT_PROJECT%/schemas/pipeline" DB="%DB_NAME%"/>
        <Data_Model File_Name="project.xsd" File_Location="%XDAT_PROJECT%/schemas/project" DB="%DB_NAME%"/>
        <Data_Model File_Name="assessments.xsd" File_Location="%XDAT_PROJECT%/schemas/assessments" DB="%DB_NAME%"/>
        <Data_Model File_Name="catalog.xsd" File_Location="%XDAT_PROJECT%/schemas/catalog" DB="%DB_NAME%"/>
        <Data_Model File_Name="uds.xsd" File_Location="%XDAT_PROJECT%/schemas/uds" DB="%DB_NAME%"/>
        <Data_Model File_Name="iq.xsd" File_Location="%XDAT_PROJECT%/schemas/iq" DB="%DB_NAME%"/>
        <Data_Model File_Name="imgassr.xsd" File_Location="%XDAT_PROJECT%/schemas/imgassr" DB="%DB_NAME%"/>
        <!--<Data_Model File_Name="ext.xsd" File_Location="%XDAT_PROJECT%/schemas/ext" DB="%DB_NAME%"/>-->
    </Models>
</Instance_Settings>
```

XNAT

Run Update

bin/update.sh –Ddeploy=true

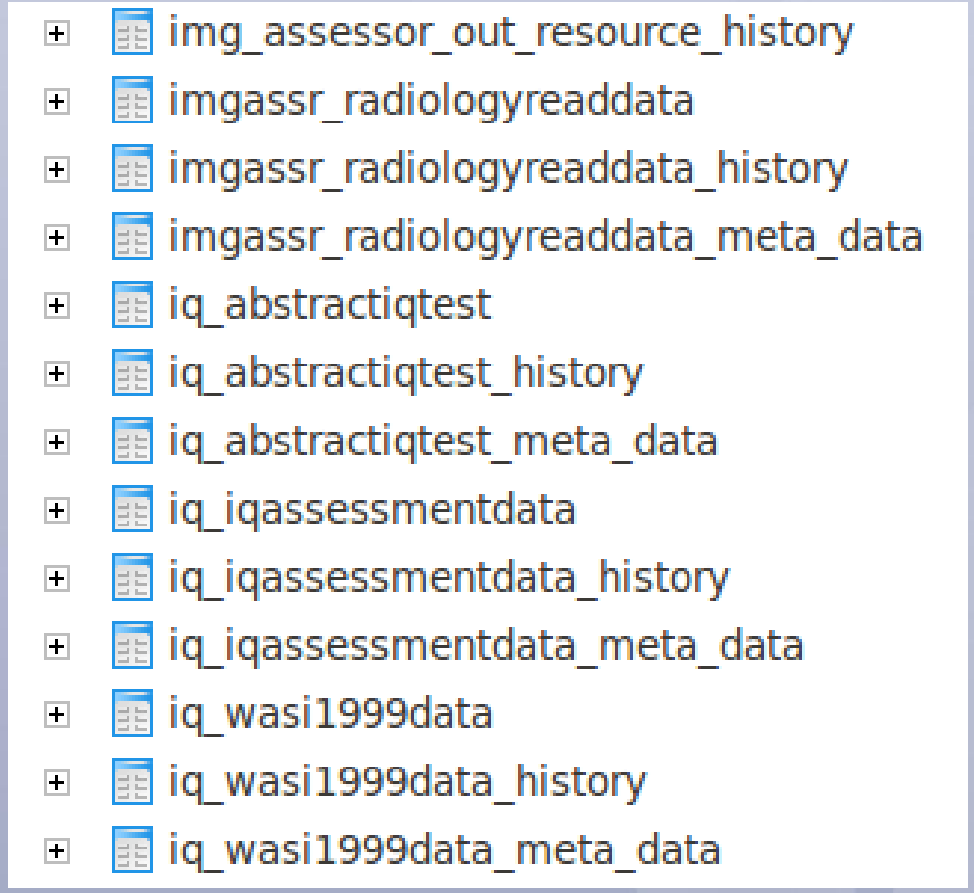
```
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat/WEB-INF/lib
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat/WEB-INF/lib
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat/WEB-INF/lib
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat/WEB-INF/lib
    [exec]     [copy] Copying 1157 files to /home/xnat/tomcat/webapps/xnat/WEB-INF/classes
    [exec]     [copy] Copying 195 files to /home/xnat/tomcat/webapps/xnat/scripts
    [exec]     [copy] Copying 249 files to /home/xnat/tomcat/webapps/xnat/base-templates
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat/WEB-INF/conf
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat
    [exec]     [echo] Copying /home/xnat/xnat/lib/xdat-1.jar To /home/xnat/tomcat//webapps/xnat/WEB-INF/lib/xdat-1.jar
    [exec]     [copy] Copying 1 file to /home/xnat/tomcat/webapps/xnat/applet
    [exec] BUILD SUCCESSFUL
    [exec] Total time: 41 seconds
    [exec] Finished at: Mon Jun 28 14:46:12 CDT 2010
    [exec]
BUILD SUCCESSFUL
Total time: 3 minutes 6 seconds
Finished at: Mon Jun 28 14:46:13 CDT 2010
```

Run SQL Update

- From deployments/xnat, run:

psql –U <userid> –f sql/xnat-update.sql <dbname>

Check for Generated Database Tables



Check for Generated Files

- Edit and Report Pages

/projects/xnat/src/base-templates/screens/

XDATScreen_*_<schema>_<datatype>.vm

- Java Screen Page

/projects/xnat/src/java/org/nrg/xdat/turbine/modules/screens/

XDATScreen_*_<schema>_<datatype>.java

- Display Documents

/projects/xnat/src/schemas/<schema>/display/

<schema>_<datatype>.xml

Check Logs

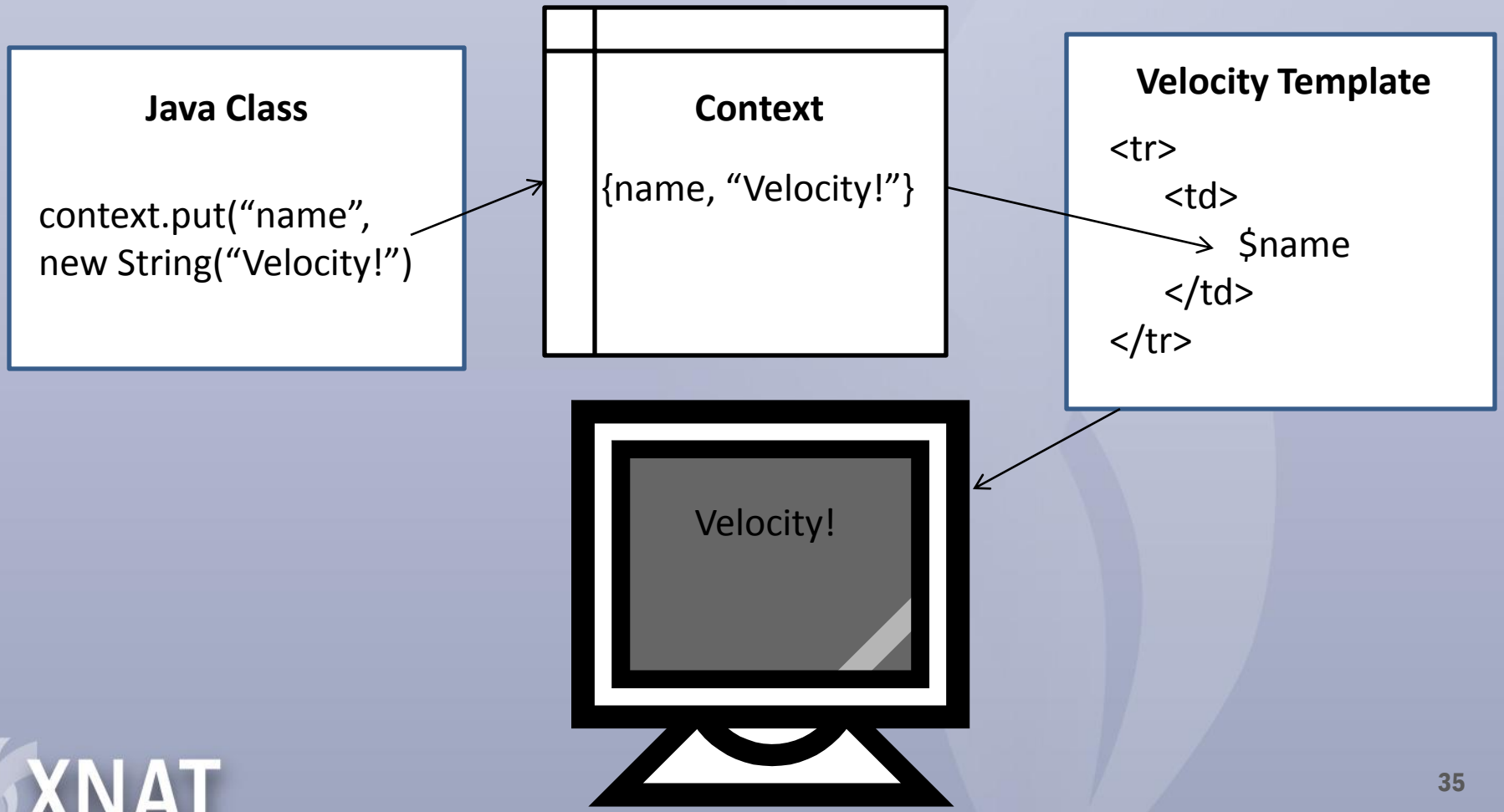
- <tomcat dir>/xnat/logs/turbine.log

Apache Turbine and Velocity

- XNAT Uses Apache Turbine Web Application Framework
- Uses Model-View-Controller (MVC) model, similar to Struts or PHP
- Velocity templates (.vm files) contain normally contain a mix of HTML, JavaScript, and Velocity Template Language (VTL)

Velocity Overview

- Data Objects are stored to and retrieved from the context



Velocity Overview

- Velocity Template Language Syntax

- Variable reference $
- Comments ##
- Arithmetic Operators +, -, *, /, %
- Relational == ## not just equivalency, can be used to compare objects
- Logic Operators &&, ||, !
- Range Operator [n..m]
- Escape Character \

Velocity Overview

- Velocity Template Language Syntax

  - Assign a value to a variable

```
#set( $foo = "Velocity" )
#set( $foo = 6)
```

  - Reference a variable

```
$foo
or
$!foo   ## if value is null, print nothing
```

Velocity Overview

- Velocity Template Language Syntax

  - Reference a method

$customer.getAddress()
$customer.getAddress(“current”)

  - Index notation

$foo[0] ## $foo takes in an Integer look up to an ArrayList

Velocity Overview

- Velocity Template Language Syntax

  - if/elseif/else

example 1: single argument
#if($foo) Velocity! #end

example 2: relational operator
#if($name==“apples”) apples
  #elseif ($name==“oranges”) oranges
  #else bananas
#end

Velocity Overview

- Velocity Template Language Syntax

  - loops

```
#foreach( $criterion in $criteria )
   Current value:  $!criterion
#end

#foreach( $i in [n..m])
   Name $i:  $criterion.getName($i)
#end
```

Velocimacros

```
#macro (formXDATCheckBox $label $field $value)
  <td bgcolor="$ui.formFieldColor">
     <input type="checkbox" name="$field" value="true"
        #if ($value) checked #end />
      <b>
         $label
      </b>
</td>
#end
```

- Velocimacros provide ways to repeat segments of text
- XNAT provides a library of macros already available from:
  plugin-resources/webapp/xdat-templates/macros/TurbineMacros.vm

Velocity Overview

- Velocity Template Language Syntax

  - Parse
    - Allows the template designer to import a local file that contains Velocity Template Language (VTL)
    - Velocity will parse the VTL and render the template specified

```
#parse( "parsefoo.vm" )
```

Site Description Customization

```
<tr>
  <td valign="top" colspan="2">
      #parse("/screens/site_description.vm")
  </td>
 </tr>
```

Adding Analytics Code

- A good place to add analytics code is in projects/xnat/src/templates/navigations/DefaultBottom.vm
- Occurs on almost every page (except login)

XNAT Data Objects in the Context

- $turbineUtils
  - XDAT Core library containing various utilities
  - In particular,

$turbineUtils.getTemplateName (String module, String dataType, String project)

is used quite frequently to enable customization by project

Many other useful methods in TurbineUtils such as:

formatDate(date, pattern)

getArrayIndex(Object[],int)

Project Specific Customization

#parse($turbineUtils.getTemplateName(
"_report",
$om.getXSIType(),
$project))

XNAT will first look in:
projects/xnat/src/templates/screens/<schema>_<datatype>
/<schema>_<datatype> _<project>.vm

If not found, XNAT will try:
projects/xnat/src/templates/screens/<schema>_<datatype>
/<schema>_<datatype>.vm

XNAT Data Objects in the Context

- $om
  - Object inserted by default into the context
  - Provides getter and setter methods for all elements in the child and parent objects
  - A corresponding Java class (automatically generated) can be further edited

projects/xnat/src/java/org/nrg/xdat/om/base/auto/

Auto<schema><datatype>.java

Demonstration

XNAT

Deploying the Updated Templates

- Create new copies of the XDATScreen*.vm files in the projects directory.
  - From: projects/xnat/src/base-templates/screens
  - To: projects/xnat/src/templates/screens
- Customize *.vm files
- Run bin/quick-deploy-templates.sh to deploy the templates to Tomcat. (No restart necessary).

Placing Your Own Objects in the Context

- Create auto-generated Java code
  - From: projects/xnat/src/java/org/nrg/xdat/turbine/modules/screens
  - To: projects/xnat/src/java/org/apache/turbine/app/xnat/modules/screens
- Customize the Java file: context.put("name", object)
- If you're using an IDE run:

bin/quick-deploy.sh -Dclass.dir=<class dir>

Tomcat restart required for Java

Adding Action Classes

- Action classes are not auto-generated, but can be added
- Create action Java files in:
  projects/xnat/src/java/org/apache/turbine/app/xnat/modules/actions

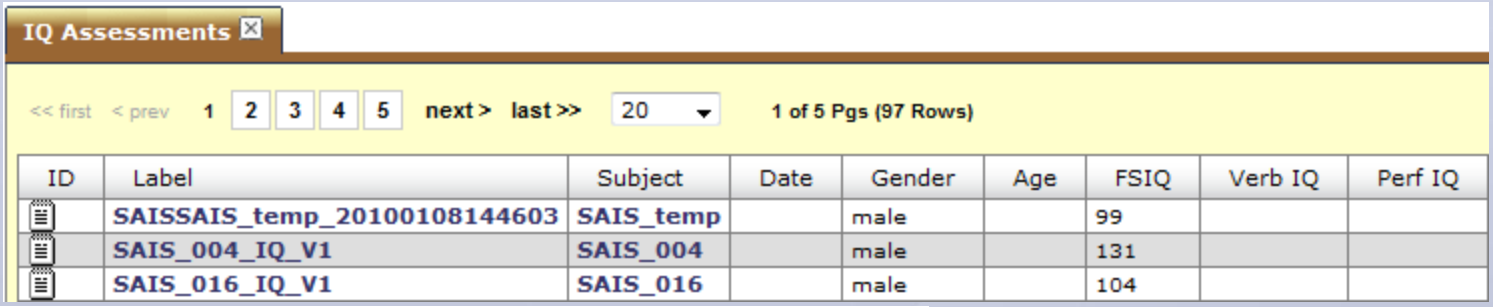
Demonstration

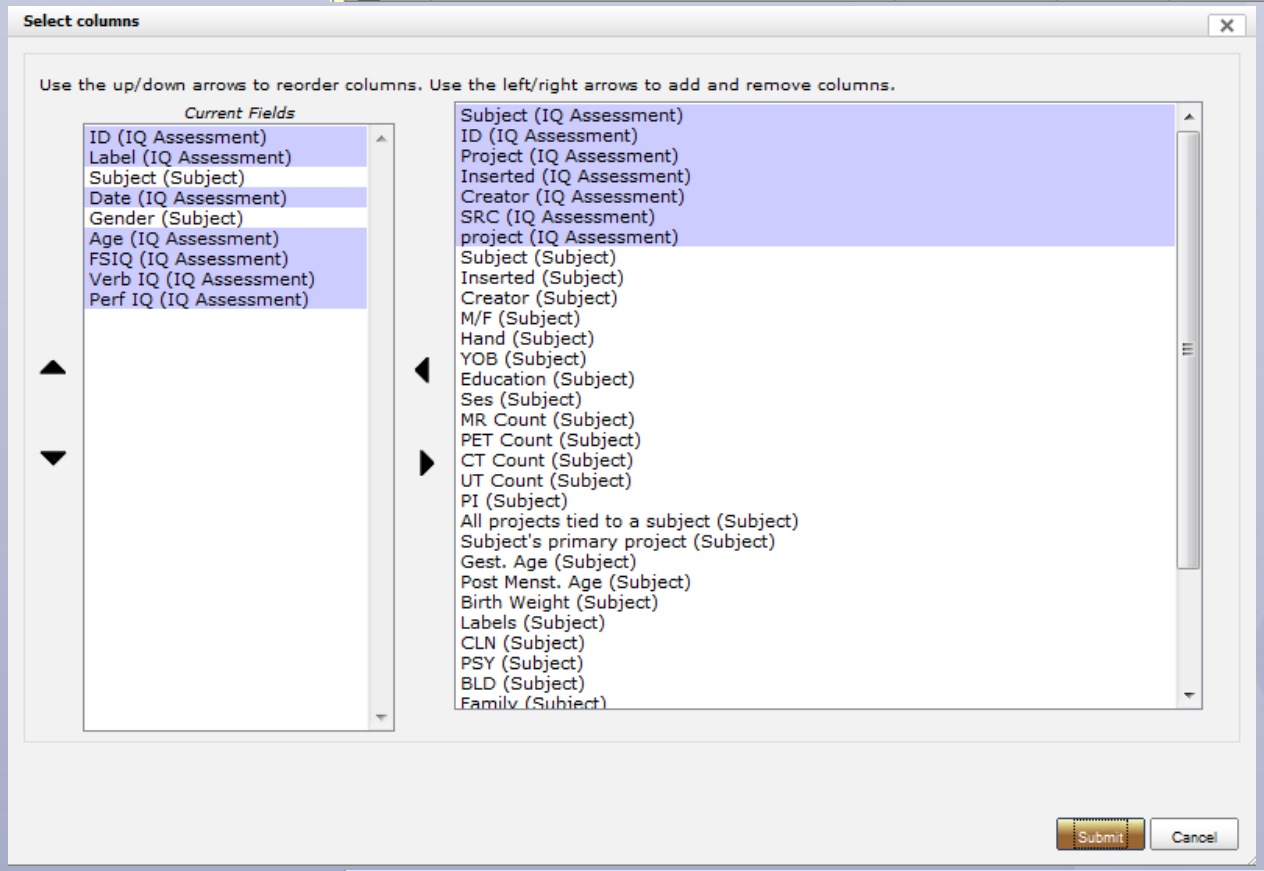
XNAT Display Documents

- Auto-generated and located in:

  projects/xnat/src/schemas/<schema>/

  display/<schema>_<datatype>.xml
- Customize the files in place, no need to move
- Run: bin/quick-deploy-templates.sh
- Reload Display Documents on the Administer page





	A	B	C	D	E	F	G	H	I
1	id	Label	Subject	Date	Gender	Age	FSIQ	Verb IQ	Perf IQ
2	/@WEBAF	SAISSAIS_	SAIS_temp		male		99		
3	/@WEBAF	SAIS_004_	SAIS_004		male		131		

<DisplayField>

```
<DisplayField id="EXPT_ID" header="ID" visible="true" searchable="true">
    <DisplayFieldElement name="Field1" schema-element="iq:iqAssessmentData.ID"/>
    <HTML-Link>
        <Property name="HREF" value="none"/>
        <Property name="ONCLICK" value="return rpt('@Field1','iq:iqAssessmentData','iq:iqAssessmentData.ID');">
            <InsertValue id="Field1" field="EXPT_ID"/>
        </Property>
    </HTML-Link>
</DisplayField>
```

- Define element you may see in a listing or search
- Must be defined to be included in the Add Columns list
- Element can be defined as text or link

XNAT

<DisplayVersion>

```
<DisplayVersion versionName="listing" default-order-by="DATE" default-sort-order="DESC" brief-description="IQAssessment" dark-c
    <DisplayFieldRef id="RPT"/>
    <DisplayFieldRef id="LABEL"/>
    <DisplayFieldRef id="SUBJECT_LABEL" element_name="xnat:subjectData"/>
    <DisplayFieldRef id="DATE"/>
    <DisplayFieldRef id="GENDER" element_name="xnat:subjectData"/>
    <DisplayFieldRef id="AGE"/>
    <DisplayFieldRef id="FSIQ"/>
    <DisplayFieldRef id="VERBALIQ"/>
    <DisplayFieldRef id="PERFORMANCEIQ"/>
</DisplayVersion>
```

- Groups DisplayFields into listing by ID
- Some default listings:
  - listing
  - listing_csv
  - brief
  - detailed
  - project bundle

<SQLView>

```
<SQLView name="IQ_IQASSESSMENTDATA_PROJECTS"
sql="SELECT id, '&lt;' || expt.project || '&gt;' || xs_a_concat(', ' || shared.project)
AS projects FROM xnat_experimentData expt LEFT JOIN xnat_experimentData_share shared
ON expt.id=shared.sharing_share_xnat_experimentda_id LEFT JOIN xdat_meta_element xme
ON expt.extension = xme.xdat_meta_element_id WHERE element_name='iq:iqAssessmentData'
GROUP BY expt.id,expt.project"/>
```

```
<DisplayField header="Projects" id="PROJECTS" data-type="string">
    <DisplayFieldElement name="Field1" viewName="IQ_IQASSESSMENTDATA_PROJECTS" viewColumn="PROJECTS"/>
</DisplayField>
```

```
<ViewLink alias="IQ_IQASSESSMENTDATA_PROJECTS">
    <Mapping TableName="IQ_IQASSESSMENTDATA_PROJECTS">
        <MappingColumn rootElement="iq:iqAssessmentData" fieldElement="iq:iqAssessmentData.ID" mapsTo="id"/>
    </Mapping>
</ViewLink>
```

- Way to display unbounded elements in listings
- Allows just about any SQL command: concatenate text, do math operations

<Arc-Definition> and <Arc>

```
<Arc-Definition Id="PARTICIPANT_EXPERIMENT">
    <CommonField id="DATE" type="DATE"/>
    <CommonField id="PART_ID" type="STRING"/>
    <CommonField id="EXPT_ID" type="STRING"/>
    <Bridge-Element name="xnat:subjectData" field="SUBJECT_ID"/>
    <Filter field="EXPT_ID" filterType="distinct"/>
    <Filter field="DATE" filterType="closest"/>
    <Filter field="PART_ID" filterType="equals"/>
</Arc-Definition>
```

```
<Arc name="PARTICIPANT_EXPERIMENT">
    <CommonField id="PART_ID" local-field="SUBJECT_ID"/>
    <CommonField id="DATE" local-field="MR_DATE"/>
    <CommonField id="EXPT_ID" local-field="EXPT_ID"/>
</Arc>
```

- Allows joins between displayVersions
- Arc-Definition is only defined once
- Every member element must include an ARC and is expected to include all CommonFields

<SchemaLink>

```
<SchemaLink element="xnat:visitData" type="mapping" alias="xnat:visitData">
<Mapping TableName="SUBJECT_VISIT_DISTINCT">
<MappingColumn rootElement="xnat:subjectType" fieldElement="xnat:subjectType.xnat_subjecttype_id" mapsTo="xnat_subjecttype_id"/>
<MappingColumn rootElement="xnat:visitData" fieldElement="xnat:visitData.id" mapsTo="xnat_visitdata_id"/>
</Mapping>
</SchemaLink>
```

- Allows developer to include fields from another Schema Element
- Element can then be used in <DisplayField> elements as if it is the displayable element

Additional Customizations Notes

- Thanks to those who have already provided additional customizations
- XNAT customizations are being cataloged and will be made available as soon as possible at http://www.xnat.org/Customizing+XNAT
- Many types of customizations we didn't see today
  - Pipelines
  - Many more data types
  - Many more GUI customizations
- Request for customizations: gurneyj@wustl.edu

Links

- http://xnat.wikispaces.com/XNAT+2010+Workshop+-+Customizations
- http://www.xnat.org/Customizing+XNAT
- http://xnat.wikispaces.com/XNAT+Display+Documents

Questions