



The Institute
for Quantitative Social Science
at Harvard University



JavaOne

Harvard's Dataverse Network: A JavaServer Faces/EJB 3.0 Technology Data Sharing Solution on Java EE 5

Merce Crosas, Ph.D./Robert Treacy
Senior Manager/Architect

Harvard University
<http://thedata.org>

TS-4656



Goal of Our Talk

Interested in developing a full Java™ Platform, Enterprise Edition (Java EE platform) 5 application? Is open-source important to you? Are you working on a data sharing application?

Learn from our experience: The advantages, challenges, and solutions we found building a completely **open-source** Java EE 5 platform data sharing application using JavaServer™ Faces technology/Tiles and Enterprise JavaBeans™ (EJB™) 3.0.

Agenda

What Is the Dataverse Network?

Designing With Open-Source Technologies

Integrating Tiles With JavaServer
Faces Technology

Connecting EJB Specification 3.0
With JavaServer Faces

Final Comments

Q&A

Agenda

What Is the Dataverse Network?

Designing With Open-Source Technologies

Integrating Tiles With JavaServer
Faces Technology

Connecting EJB Specification 3.0
With JavaServer Faces

Final Comments

Q&A

An Introduction to Our Data Sharing Solution

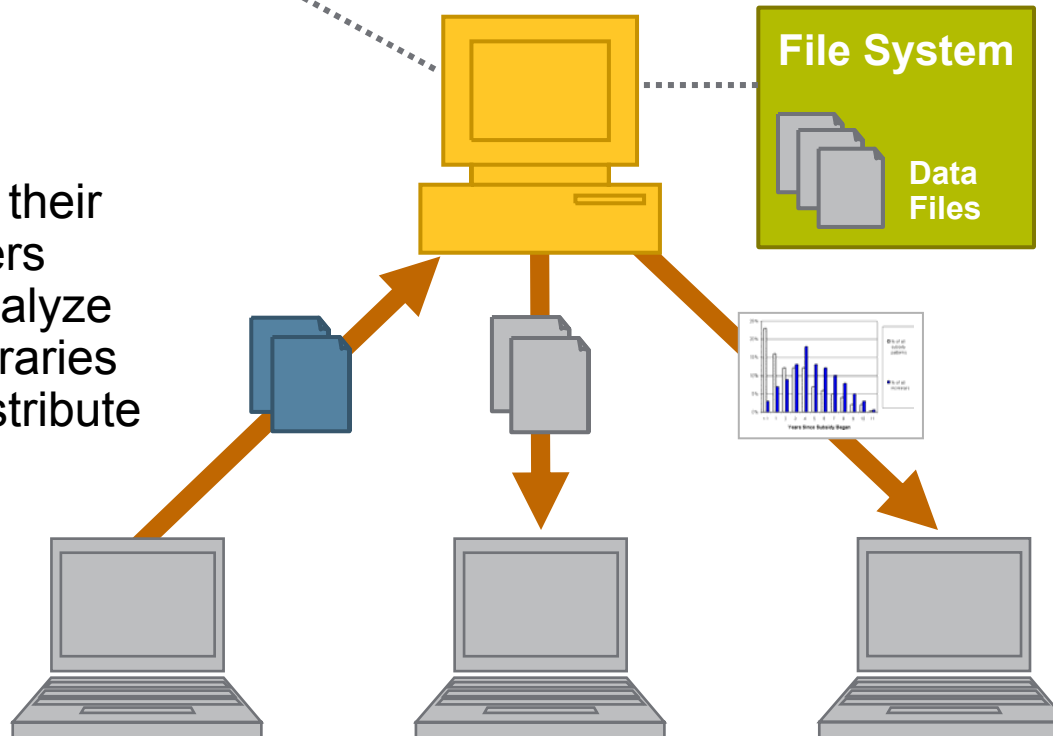
- The Dataverse Network allows researchers, journals and archives to **share, cite, and preserve** research data
- Currently used for Social Science data
- May be extended to other research fields (e.g., bio-medical data from MIT Broad Institute)
- Our development group is at Harvard's Institute for Quantitative Social Science:
<http://www.iq.harvard.edu/>
- The Dataverse Network is the successor of the VDC (Virtual Data Center), also developed at Harvard

Share, Cite, Preserve

We create a new standard for citing quantitative data sets

Gary King; Langche Zeng, 2006, "Replication Data Set for When Can History be Our Guide? The Pitfalls of Counterfactual Inference" [hdl:1902.1/DXRXCFAWPK](https://hdl.handle.net/1902.1/DXRXCFAWPK) UNF:3:DaYIT6QSX9r0D50ye+tXpA== Murray Research Archive [distributor]

Authors upload their data, researchers download or analyze data on-line, libraries and journals distribute their holdings



Our system reformats the data to shield them from dated statistical packages

Search, Browse, Analyze World Data

Search Studies within Harvard-MIT Data Center Dataserve

Advanced Search | Search Tips

Search: For:

- Cataloging Information
- Author
- Title
- Study ID
- Variable Information

Search with Lucene Index Server

Download Subset Recode Descriptive Statistics **Advanced Statistical Analysis**

Selected Variables

Choose Descriptive Statistics output to apply to selected variables:

- Univariate Numeric Summaries
- Univariate Graphic Summaries

Show ▾

Analysis with R and Zelig models

Browse

- Harvard-MIT Data Center
 - Census
 - ICPSR
 - I. Census Enumerations
 - Nations Other than the United States
 - United States
 - II. Community and Urban Studies
 - III. Conflict, Agression, Violence and
 - IV. Economic Behavior and Attitudes
 - V. Education
 - Roper

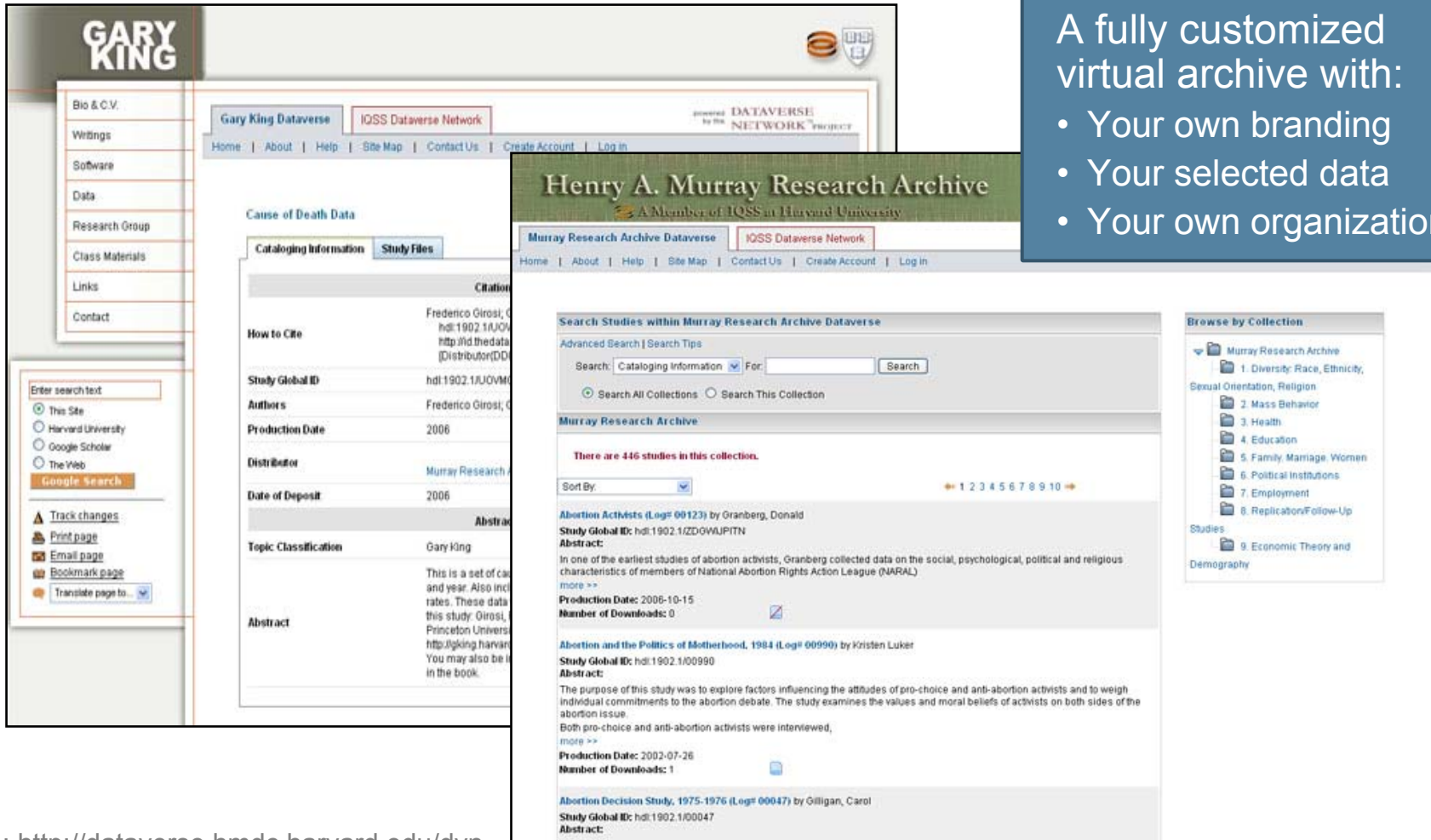
Browse with JavaServer Faces Tree component

Variable ID	Variable Name	Variable Label	Quick Summary
225533	AGE		
225497	EDUC	EDUCATION	
225520	IDENT	ID NUMBER	
5506	MSTAT	MARITAL STATUS	
5549	OCCUP	OCCUPATION	
5532	PARITM1	PARENT LIVING	
5537	PARITM2	PARENTS AGE	

Create Your Own “Dataverse”

A fully customized virtual archive with:

- Your own branding
- Your selected data
- Your own organization

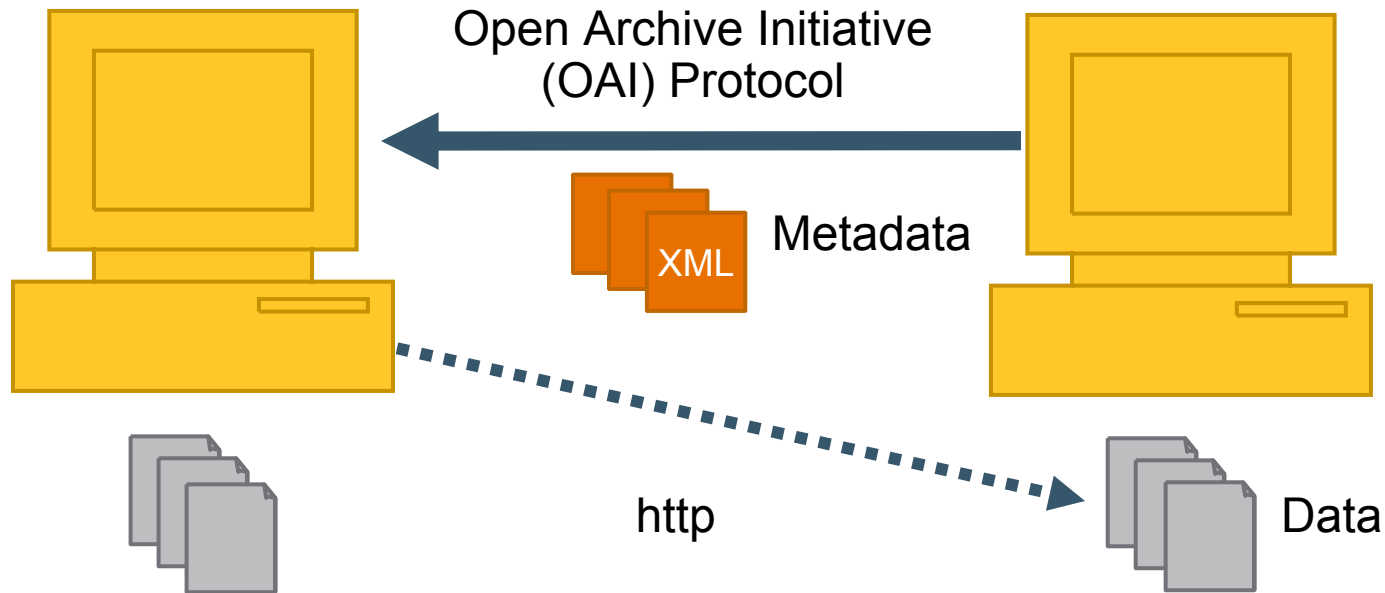


The screenshot displays two overlapping web pages from the Dataverse Network. The top page is the 'Gary King Dataverse', featuring a navigation menu on the left with categories like 'Bio & C.V.', 'Writings', 'Software', 'Data', 'Research Group', 'Class Materials', 'Links', and 'Contact'. The main content area shows 'Cause of Death Data' with a table of 'Cataloging Information' and 'Study Files'. The bottom page is the 'Henry A. Murray Research Archive', which includes a search bar, a list of studies (e.g., 'Abortion Activists (Log# 00123)', 'Abortion and the Politics of Motherhood, 1984 (Log# 00990)', 'Abortion Decision Study, 1975-1976 (Log# 00047)'), and a 'Browse by Collection' sidebar on the right.

Source: <http://dataverse.hmdc.harvard.edu/dvn>

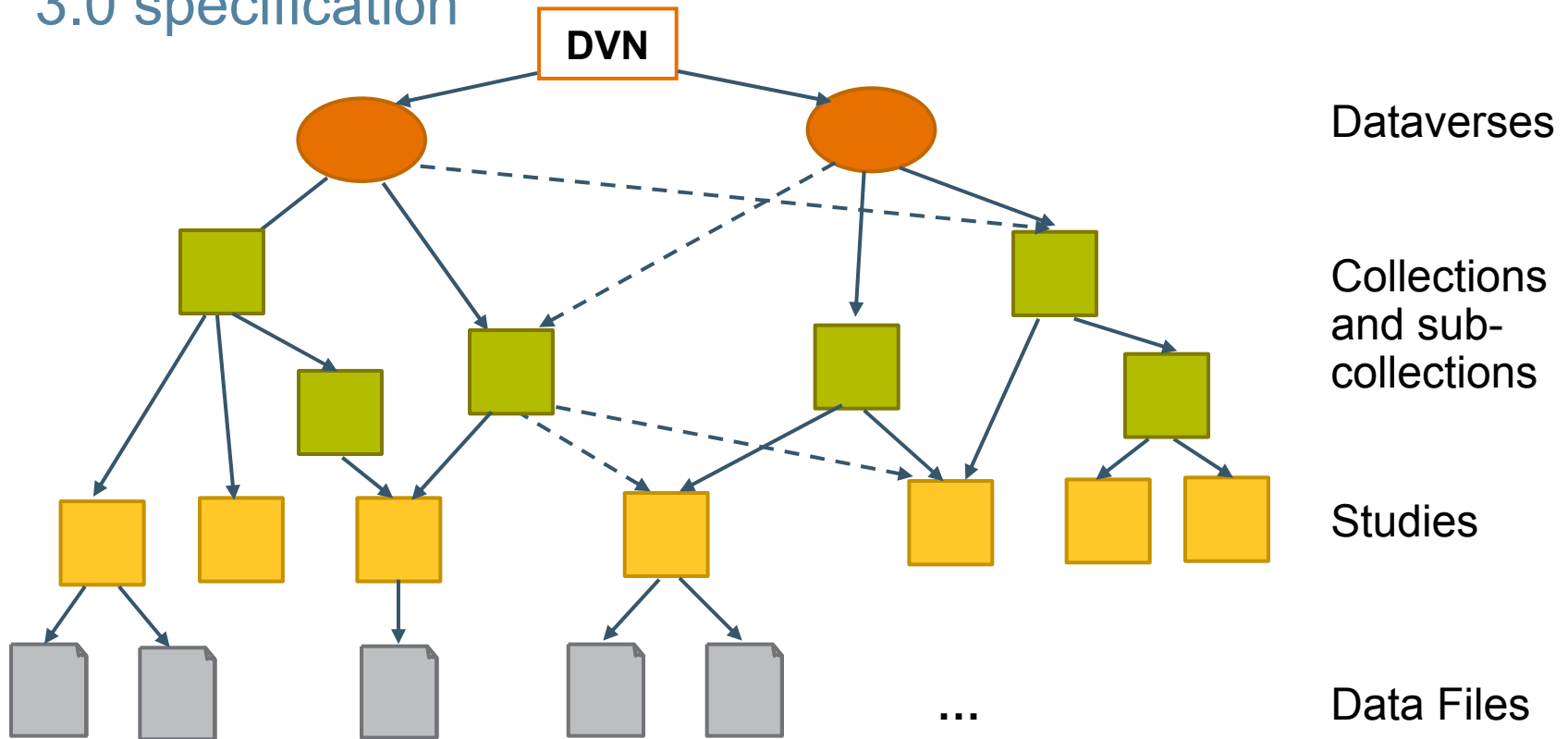
And Share Across Dataverse Networks and Other Archives

A DVN may “harvest” metadata from other installed DVNs (or any OAI server Archive) to allow searching locally for their studies; Data is then retrieved remotely

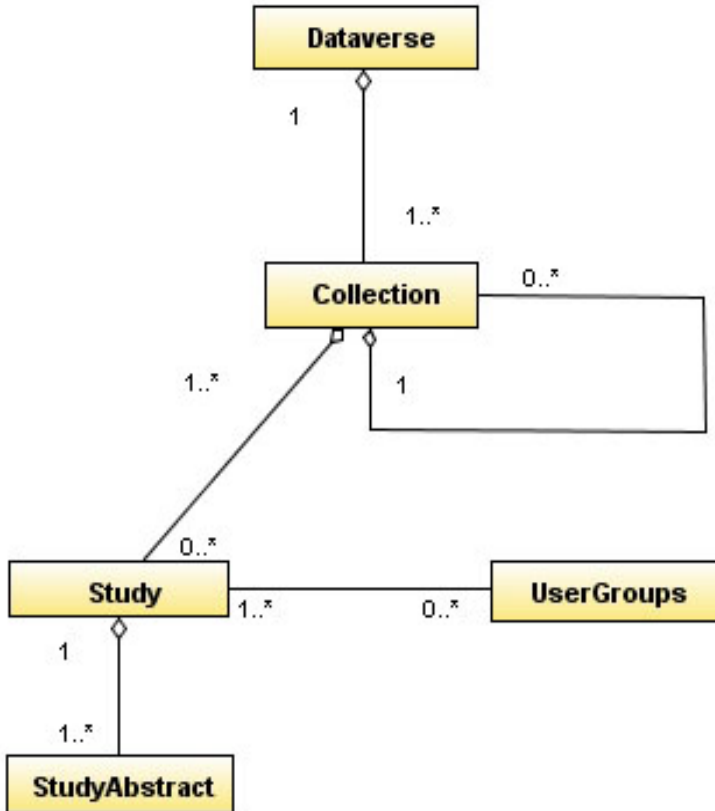


Complex Object Relationships

Our application requires complex relationships between objects which are better modelled with POJO-based EJB 3.0 specification



Many-to-Many Relationships



COLLECTION	
PK	ID
	NAME
	QUERY
	...
	PARENTCOLLECTION_ID

STUDY	
PK	ID
	RESTRICTIONS
	CITATION
	...

Join Table

COLL_STUDIES	
PK, FK1	COLLECTION_ID
PK, FK2	STUDY_ID

One to Many Relationships

STUDYABSTRACT	
PK	ID
	TEXT
	DATE
	...
FK	STUDY_ID

STUDY	
PK	ID
	RESTRICTIONS
	CITATION
	...

Agenda

What Is the Dataverse Network?

Our Use of Open-Source Technologies

Integrating Tiles With JavaServer
Faces Technology

Connecting EJB Specification 3.0
With JavaServer Faces

Final Comments

Q&A

What Technologies Are We Using? And Why?

- About a year ago we decided to **re-write** our data-sharing application (then called VDC) to build a more maintainable and scalable system
- Our requirements
 - Open Source
 - Rapid development
- We chose Java EE 5 platform because:
 - Re-usable JavaServer Faces components make UI development faster
 - EJB 3 specification is much simpler to use than EJB 2 specification (most of our developers had experience with EJB 2 specification)
 - Existence of an open-source Java EE 5 platform compliant app server: Project GlassFish™
- Our development environment: NetBeans™ 5.5.1 software (we used Sun Java Studio Creator software in the beginning to generate all JavaServer Pages™ (JSP™ page) and JavaServer Faces technology-based backing beans)

What Technologies/OSS Are We Using?

- GlassFish V2-b41a
 - <https://glassfish.dev.java.net/>
 - Fully Java EE 5 platform compliant open-source application server
- Shale Tiles and Tiles 2 (with JavaServer Faces technology)
 - <http://struts.apache.org/struts-sandbox/tiles/>
 - Common layout for all pages
 - Customizable header and footer
- PostgreSQL 8.2.3
 - <http://www.postgresql.org/>
 - Reliable and scalable open-source DB
- Lucene 2.0
 - <http://lucene.apache.org/java/docs/index.html>
 - Powerful and fast Java technology-based search engine
- OAICat and OAIHarvester2
 - <http://www.oclc.org/research/software/oai/cat.htm>
 - Java technology-based OAI implementation
- Zelig v2.8-2 and R v2.4
 - <http://gking.harvard.edu/zelig/>
 - Extensive source of statistical models written in R
- Awstats
 - <http://awstats.sourceforge.net/>
 - Web site usage statistics with graphical presentation

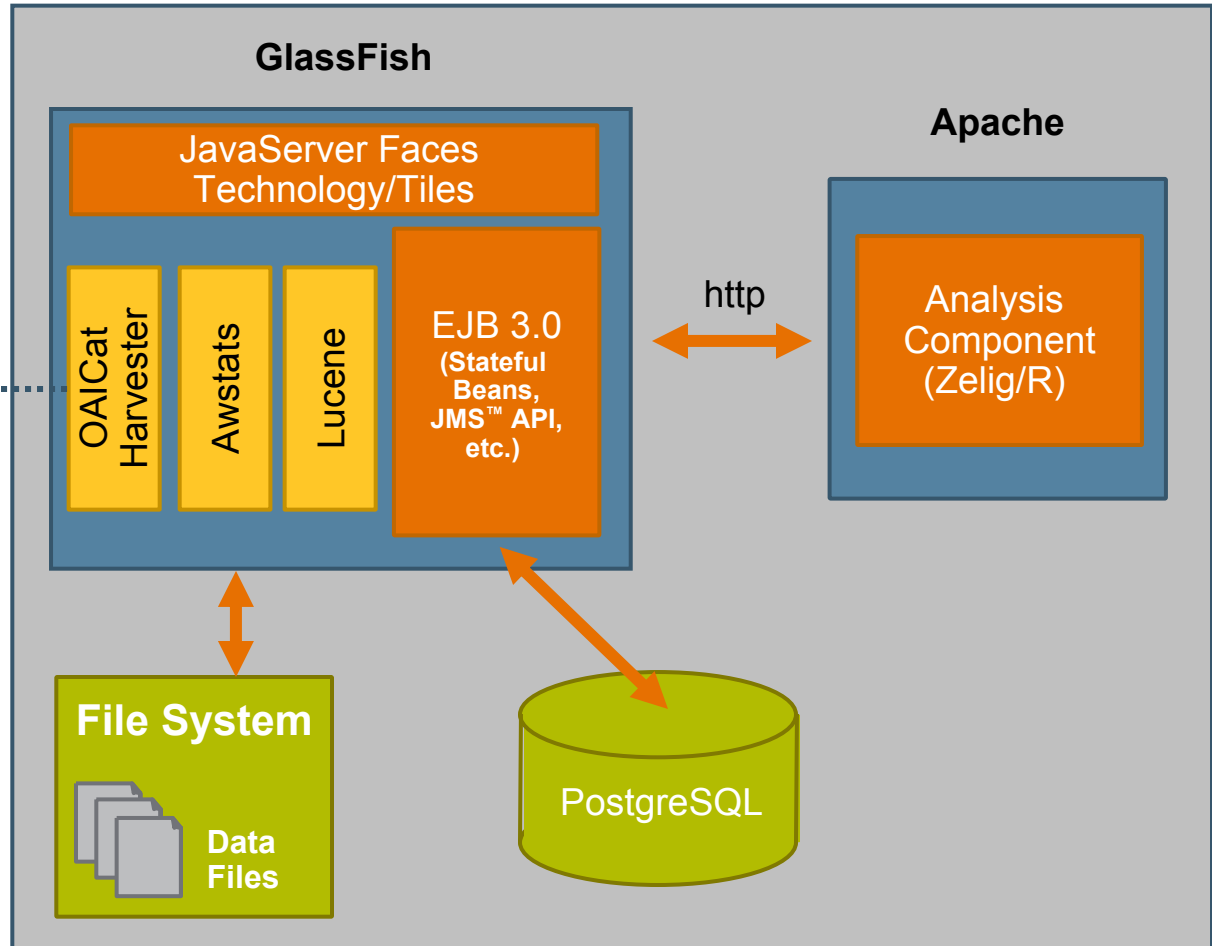
Overall Architecture

Dataverse Network

Other Dataverse Networks
(or any remote OAI Servers)



OAI Protocol



JMS = Java Message Service

Agenda

What Is the Dataverse Network?

Designing With Open-Source Technologies

**Integrating Tiles With JavaServer
Faces Technology**

Connecting EJB Specification 3.0
With JavaServer Faces

Final Comments

Q&A

Tiles/JavaServer Faces 1.2 Integration

Using Shale Tiles With Tiles 2 Standalone
in JavaServer Faces 1.2

- Configure web.xml for the Tiles Servlet
- Set up the Tiles definitions and templates
- Transform Java Studio Creator software pages to tiles
- Modifying Shale Tiles ViewHandler for JavaServer Faces 1.2

Configure web.xml

```
<servlet>
<servlet-name>Tiles Servlet</servlet-name>
<servlet-class>
org.apache.tiles.servlets.TilesServlet</servlet-class>
<init-param>
<param-name>definitions-config</param-name>
<param-value>/WEB-INF/tiles.xml</param-value>
</init-param>
<load-on-startup>2</load-on-startup>
</servlet>
<servlet>
<servlet-name>Faces Servlet</servlet-name>
...
<load-on-startup>1</load-on-startup>...
```



Set Up Tiles Definitions in tiles.xml

```
<tiles-definitions>
<definition name="/main-layout" path="/mainLayout.jsp">
...
</definition>
<definition name="/HomePage" extends="/main-layout">
...
</definition>
<definition name="/AdvSearchPage" extends="/main-layout">
...
</definition>
</tiles-definitions>
```

Layout Definition in tiles.xml

```
<definition name="/main-layout" path="/mainLayout.jsp">
<put name="title" type="string" value=""/>
<put name="banner" type="template"
value="/tiles/CustomBanner.jsp"/>
<put name="connectedbanner" type="template"
value="/tiles/ConnectedBanner.jsp"/>
<put name="menubar" type="template"
value="/tiles/Menubar.jsp"/>
<put name="content" type="template" value=""/>
<put name="footer" type="template"
value="/tiles/CustomFooter.jsp"/>
</definition>
```

Application Pages Extend Layout

```
<definition name="/HomePage" extends="/main-layout">
<put name="title" type="string" value="Home"/>
<put name="content" type="template"
value="/HomePage.jsp"/>
</definition>
<definition name="/AdvSearchPage" extends="/main-layout">
<put name="title" type="string" value="Advanced Search"/>
<put name="content" type="template"
value="/AdvSearchPage.jsp"/>
</definition>
```

JSP Page—mainLayout.jsp

```
...<f:view>
<ui:body>
<h:panelGrid ...><ui:panelLayout id="mainLayout"
panelLayout="flow">
<f:subview id="banner">
<tiles:insert name="banner" flush="false"/></f:subview>...
<f:subview id="content"><tiles:insert name="content" .../>
</f:subview>...
```

Developing JavaServer Faces Technology-Based Pages to Work in Tiles

We used Creator 2.1 to do the initial layout of our pages

- Visual Web Pack was not available when we designed our pages
- Java Studio Creator software can generate bindings for Java 2 Platform, Enterprise Edition (J2EE™ platform) 1.4, but not Java EE 5 platform
- Move pages to NetBeans 5.5 software to code the Java EE 5 platform bindings and edit the pages for Tiles
- Basically, transform views to subviews to make Tiles

Creating a Tile From a Creator Page

Creator generated page

```

<jsp:root version="1.2" ...>                                // change the version
<jsp:directive.page...>                                    // cut this line
<f:view>                                                  // change to subview
<ui:page ...>                                             // cut the next
<ui:html binding...>                                     // few lines
<ui:head binding...>...</ui:head>
<ui:body binding...>                                     // resume at <ui:form ...
<ui:form binding="#{study$MyStudiesPage.form1}"
id="form1"> ...
  
```

Tile

```

<jsp:root version="2.0" ...
xmlns:tiles="http://struts.apache.org/tags-tiles">
<f:subview id="MyStudiesPageView">
<ui:form binding="#{MyStudiesPage.form1}" id="form1"> ...
  
```


Shale Tiles for JavaServer Faces 1.2

JavaServer Faces 1.2 addresses interweaving problems between JavaServer Faces and JSP technology

- Writing of buffers needs to be handled differently
- TilesViewHandler in Shale Tiles needs to be adapted to work in JavaServer Faces 1.2
- Examine ViewHandlerImpl in JavaServer Faces 1.2 RI to see how to rewrite renderView method

Agenda

What Is the Dataverse Network?

Designing With Open-Source Technologies

Integrating Tiles With JavaServer
Faces Technology

**Connecting EJB Specification 3.0
With JavaServer Faces**

Final Comments

Q&A

Collections

A DVN can be represented by a tree

- Collections can have studies and sub-collections
- Studies in a collection can be your own, or can be from other collections
- A DVN can have links to collections in other DVNs

Tree

```
<ui:tree binding="#{HomePage.collectionTree}"  
id="collectionTree" ... />
```

```
public Tree getCollectionTree() {...  
VDCCollectionTree vdcTree = new VDCCollectionTree();  
VDC vdc = getVDCRequestBean().getCurrentVDC();  
collectionTree = vdcTree.populate(vdc); ... }
```

Java Persistence API (JPA) Entities

EJB 3.0 specification Entities are not Entity Beans

- Not coarse-grained objects
- Flexible, as fine-grained as you want
- POJOs enable complex models
- Your application manages relationships
- JPA entities can be used in Java 2 Platform, Standard Edition (J2SE™ platform)

Entity Manager

Concepts

- Managed entities vs. detached entities
- Transaction Scopes vs. extended
- Eager loading vs. lazy loading
- `persist()`, `find()`, `merge()`, `refresh()`

Detached Entities

Entities are not always managed

- Get an entity to the web-tier through a stateless session bean method
- The entity is only managed within the method
- The entity in the web-tier is detached
- If my study has a list of authors, they need to be fetched in the SLSB method to guarantee availability when the entity is detached

Stateless Session Bean

```
// getStudy() can get a Study entity to the web tier
@Stateless
public class StudyServiceBean implements ... {
    @PersistenceContext(unitName="VDCNet-ejbPU")
    EntityManager em;
    public Study getStudy(Long studyId) {
        Study s = em.find(Study.class, studyId);
        for (Iterator<StudyAuthor> it =
            s.getStudyAuthors().iterator(); it.hasNext();) {
            StudyAuthor elem = it.next();
            elem.getId(); // fetch }
        return s;
    }
}
```


Study Entity

```
// what is available after call to getStudy() ?  
@Entity  
@Table(unique constraints= @UniqueConstraint...  
public class Study {  
    private String citationRequirements;  
    String title;  
    @OneToMany(mappedBy="study", cascade=...) ...}  
    private List<StudyAuthor> studyAuthors;  
    @ManyToMany(cascade={CascadeType.REMOVE, ... })  
    private Collection<UserGroup> allowedGroups;  
    @ManyToOne private VDCUser creator;  
    @ManyToMany( cascade={CascadeType.REMOVE,...})  
    private Collection<StudyField> summaryFields;
```

Stateless Session Bean Client

```
public class StudyPage extends VDCBaseBean {
    @EJB
    private StudyServiceLocal studyService;
    Study s = studyService.getStudy();

    // OK (you fetched the authors in getStudy())
    for (Iterator<StudyAuthor> it =
        s.getStudyAuthors().iterator(); it.hasNext();)
    {StudyAuthor author = it.next();... }

    // Not OK (you may or may not have the allowed groups)
    for (Iterator<UserGroup> it =
        s.getAllowedGroups().iterator(); it.hasNext();)
    {UserGroup group = it.next();... }
```

Get Details While Entity Is Managed

```
// if you want all the details, be sure to fetch them
public Study getStudyDetail(Long studyId) {
    Study s = em.find(Study.class, studyId);

    for (Iterator<UserGroup> it =
        s.getAllowedGroups().iterator(); it.hasNext();) {
        UserGroup group = it.next();
        group.getId() // fetch;}

    for (Iterator<StudyAuthor> it =
        s.getStudyAuthors().iterator(); it.hasNext();) {...}

    // Iterate through any other collections of Objects
    // contained in the study if needed in the web-tier

    return study;
}
```

Extended Persistence With Stateful Session Beans

```
// study managed from newStudy() until removal of SFSB
@Stateful
@Transactional(TransactionAttributeType...)
public class EditStudyServiceBean implements ... {
    @PersistenceContext(type =
PersistenceContextType.EXTENDED,unitName="VDCNet-ejbPU")
    EntityManager em;
    public void newStudy(Long vdcId ) {
        study = new Study();
        em.persist(study);...
    }
    @Remove
    @Transactional(TransactionAttributeType.REQUIRED)
    public void save(Long vdcId, Long userId) {...}
```

Study in the Web Tier

```
<ui:panelGroup block="true" id="groupPanel16" >
<h:inputText id="input_title" size="90" maxlength="255"
value="#{EditStudyPage.study.title}"
required="#{EditStudyPage.studyMap[sfc.title].required}"/>
<h:message styleClass="errorMessage" for="input_title"/>
</ui:panelGroup>
```

@EJB

```
(name="editStudy",beanInterface=EditStudyService.class)
public class EditStudyPage extends VDCBaseBean {
    {... //init method
        editStudyService.setStudy(studyId);
    }
    public Study getStudy() {
        return this.study;
    }
}
```

Multi-Tasking (With Multiple Tabs)

```
//keeping the right task in the right tab

Context ctx = new InitialContext();
EditStudyService editStudyService = (EditStudyService)
ctx.lookup("java:comp/env/editStudy");

// sessionPut and sessionGet are backing bean methods
// for managing session scope attributes in our app
sessionPut( editStudyService.getClass().getName()+studyId,
editStudyService);

editStudyService = (EditStudyService)
sessionGet(editStudyService.getClass().getName()
+getStudyIdFromRequest());
```

Agenda

What Is the Dataverse Network?

Designing With Open-Source Technologies

Integrating Tiles With JavaServer
Faces Technology

Connecting EJB Specification 3.0
With JavaServer Faces

Final Comments

Q&A

Final Comments

Lessons learned from writing from scratch a Java EE 5 web application

- Use **Stateless** Session Beans for simple form updates (single object); use **Stateful** Session Beans for more complicated forms (object with collection of dependent objects)
- Use **Java Naming and Directory Interface™ (J.N.D.I.) API** rather than annotations to get a Stateful Session Bean instance from within a backing bean (or other stateless objects)
- If you use **@OrderBy** in a Collection, call `em.refresh()` to maintain the correct order in the cached copy after modification
- Remember, all relationships must be maintained in the application code
- Override **equals()** and **hashCode()** in Entity classes to return equal based on primary keys; (NetBeans does this for you if you choose “New Entity Class”)
- Understand the JavaServer Faces request processing lifecycle

And from a management perspective:

- **JavaServer Faces Technology:** Facilitates rapid development of pages; however, maintaining the conversational state needs to be managed by the developer
- **EJB 3 Specification:** Using annotations and generating the entire schema based on entity class definitions streamline the development of the object relational mapping



Q&A

Merce Crosas, Robert Treacy, Wendy Bossons,
Gustavo Durand, and Ellen Kraffmiller



The Institute
for Quantitative Social Science
at Harvard University



JavaOne

Harvard's Dataverse Network: A JavaServer Faces/EJB 3.0 Technology Data Sharing Solution on Java EE 5

Merce Crosas, Ph.D./Robert Treacy
Senior Manager/Architect

Harvard University
<http://thedata.org>

TS-4656