NEESGrid - A Grid Portal Study

Charles Severance
University of Michigan
NEESGrid SI Team
www.neesgrid.org
Acknowledgements

This presentation is based on materials from many members of the NEESGrid System Integration team including: Bill Spencer, Carl Kesselman, Tom Finholt, Lee Liming, Laura Perlman, Paul Hubbard, Joe Futrelle, Kincho Law, Jun Peng, Greg Fenves, Tomasz Haupt, Jim Myers, Doru Marcusiu, Randy Butler, Jim Eng and many others.
NEES Founding

- George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES).
- Funded in 1999 - $100M
- Goal: Transform the nation’s ability to carry out earthquake engineering research, to obtain information vital to develop improved methods for reducing the nation’s vulnerability to catastrophic earthquakes, and to educate new generations of engineers, scientists and other specialists committed to improving seismic safety.
- To be Completed: October 2004
• NEESgrid facilitates research capabilities previously unavailable
• NEESgrid links earthquake researchers across the U.S. with leading-edge computing resources and research equipment and allowing collaborative teams (including remote participants) to plan, perform, and publish their experiments
• NEESgrid is a coordinated and secure architecture/environment
• NEESgrid is a modular and extensible environment with a customizable user interface
• NEESgrid provides common tools that allow leveraging resources and experiences
• Rather than having to worry about the required cyber infrastructure, NEESgrid allows researchers to focus on the earthquake engineering challenges at hand
• The goal of the System Integrator (SI) is to develop NEESgrid as the Cyber Infrastructure that will facilitate this next generation of experimentation/simulation in earthquake engineering
NEES Components

• New experimental facilities (15)
  – Oregon State University, Rensselaer Polytechnic Institute, University of Buffalo, University of Colorado at Boulder, University of Minnesota, University of Nevada at Reno, University of Texas at Austin, and the University of California campuses at Berkeley, Davis and Los Angeles

• Collaborative Software System: NEESGrid
  – Collaboration
  – Data capture and sharing
  – Tele-presense and Tele-operation
  – Simulation
  – Support for Hybrid Simulation and Physical Experiments
Centrifuge: UC Davis
Wave basin: Oregon State
Field structural: UCLA
Field geotechnical: Texas
If we build it, they will collaborate

- Data and access to data represent fundamental barriers to dispersed collaboration
- Efficient movement of vast amounts of data is a prime rationale for cyberinfrastructure
- Federating, visualizing and mining data are principle challenges
NEES Facilities
NEESGrid Software

• Founding NMI Technologies
  – Globus Toolkit
  – OGCE Collaboration Toolkit

• New Work
  – Data and Metadata Repository - NCSA
  – Data Acquisition, Storage, and Visualization
  – Simulation Portal
NEESGrid Partners

• Argonne National Labs
  – Globus toolkit, Data Acquisition, Telepresence

• Information Sciences Institute (USC)
  – Globus toolkit, Teleoperation and Telecontrol

• National Center for Supercomputing Applications (NCSA)
  – System Integration, Data Repository

• University of Michigan
  – Collaborative Grid Portal, Data Modelling, Visualization, Video as Data
NEESGrid Partners

• Stanford  
  – Data Model Design

• Mississippi State University  
  – Simulation Portal

• University of California Berkeley  
  – OpenSEES and FedeasLab

• Pacific Northwest National Laboratory  
  – Scientific Annotation Middleware (SAM), Electronic Notebook
The Grid in NEESgrid

Internet Fabric and Operations

Hub A
- NEESpop A
  - Experimental Equipment
    - Video I/O
    - Audio I/O

Site A: Experimental Data Producer

Hub B
- Telepresence Equipment
  - Active PI
  - Data Cache

Site B: Remote Lead Investigator

Hub C
- Teleobservation Equipment
  - Passive co-PI
  - Data Cache

Site C: Passive Collaborator
NEES Resources

Remote Users:
(Faculty, Students, Practitioners)

Laboratory Equipment

Instrumented Structures and Sites

Simulation Tools Repository

Field Equipment

Leading Edge Computation

Curated Data Repository

Global Connections

Laboratory Equipment

Remote Users:
(K-12 Faculty and Students)
The Main Components of NEESgrid

- Tele-Control Services
- Tele-Observation and Data Visualization
- E-Notebook
- Streaming Data services
- DAQ and related services
- Data and Metadata services
- Remote Collaboration and Visualization tools and services
- Core Grid Services, deployment efforts, packaging
- Computational Simulation component
The NEES Win

- **New engineering capabilities**
  - rapid assembly of virtual teams
  - access to remote facilities and experiments
  - interfaces to distributed data archives/experiment repositories

- National and international cyberinfrastructure leverage
  - corporate and government commitments
  - billions of dollars in leverage
    - commoditization of infrastructure

- Distributed facility and collaboration access
  - NEES equipment sites (ES) and distributed collaborators
  - cooperating institutions and policies

- Strong security features
  - secure experiment control and data sharing policies

- Resource discovery and monitoring services
  - available resource identification and continuous status monitoring
NEES Architecture
The Role of the NEESgrid System Architecture

- Define the core capabilities of NEESgrid
- Facilitate interoperability, extensibility and scalability
- Provide a foundation on which the diverse NEES usage scenarios can be supported
  - Not single point solution
Architecture Approach

• Common infrastructure that can used across all NEES applications
  – Balance generic mechanisms, extensibility for future growth, efficiency for application specific tasks

• Validate against user requirements
  – Input from user requirements analysis
  – MOST, EBD build on proven technology base
NEESgrid and the Grid

• Grid is infrastructure to support
  – Data sharing, numeric simulation, remote observation and control, collaboration
• Maps well into NEES requirements
  – Similarity of problem space and objectives
• Synergistic with many other projects
  – E.G. SCEC, ETF, …
  – Minimizes risk
Open Grid Services Architecture

• Builds on Web Services technology
  – A Grid service is a Web service with extras

• Significant industry buy in
  – IBM, HP, Oracle, SGI, …

• High-quality open source implementation
  – Globus Toolkit®
NEESGrid and NSF Middleware Initiative

- CISE program to harden, test and support national middleware infrastructure
- Significant NMI presence in Grid space
- Plan to eventually fold NEES specific services into NMI releases
Software Components

• Extant software
  – particularly significant elements of the NSF Middleware Initiative (NMI) software system

• Custom software to address general NEESgrid issues
  – Produced by SI team

• Site-specific, and application specific software
  – to be produced by the equipment sites, other NEES participants, or other sources.
Physical Elements

- A moderate number of *equipment sites*,
- A moderate number of *resource sites*,
  - data repositories and/or computer systems
- A potentially large number of *users*
  - including earthquake engineers, students, and others.
- Campus and wide area *networks*
- An *operations center*,
  - provides monitoring and diagnostic facilities for NEESgrid as a whole
NEESgrid Core Capabilities

- Tele-control and tele-observation of experiments
- Data cataloging and sharing
- Remote Collaboration and visualization tools and services
- Simulation execution and integration
NEESgrid High-level Structure

Wide Area Network

Equipment Site

User Site

Resource Site

NEESgrid Operations

NEES Equipment Site

Local computers & storage

Gigabit Ethernet

Edge Router

> Gb/s WAN

Shake table with instrumentation
Centralized NEES-Wide Services

- NFMS
- NMDS
- GridFTP
- MDS
- HTTP
- CHEF web portal
- Information (Index) Server
- Community Authorization Server
- Credential Repository
Non-Centralized NEESgrid Services
Architecture of NEESgrid Equipment site.
Globus Toolkit V3

• High quality open source OGSI implementation
  – Developed by The Globus Alliance
• Commercial support available
• Globus services include:
  – Security
    • Authentication and authorization
  – Status and configuration
  – Resource management
  – Data services
    • Data movement
    • Data access
NEESgrid Software Stack

- Browsers/User Interfaces
- Applications/CHEF
- Programming Interfaces (Java, C APIs, Matlab toolboxes, OpenSEES…)
- NTCP
- GridFTP
- Other Globus Services
- Computational Services
- Widgets
- Plugins
- OGSI Core
- RBNB
Tele-Control Services

• A single, transaction-based protocol and service (NTCP) to control physical experiments and computational simulations.
• OGSI based implementation (GT3.0)
• Plug-ins to interface the NTCP service
  – A computational simulation written in Matlab
  – Shore Western control hardware
  – MTS control hardware (via Matlab and xPC)
  – Labview
  – C
• Security architecture, including GSI authentication and a flexible, plug-in-based authorization model.
Plug-in approach
Programming Interfaces

Applications
- Matlab/Simulink Application
- ICES Application
- OpenSEES Application
- Application

High-level APIs
- Matlab/Simulink Interface
- ICES Interface
- OpenSEES Interface

Low-level APIs
- NTCP APIs
NEES TeleControl Protocol (NTCP)
NEESgrid Core Control Components

• A uniform control interface for both physical and simulation components is achieved through a single control architecture.
  – NTCP Service
  – NTCP Client APIs
  – NTCP Plugin APIs

• Overall, control components are well-defined and available. Equipment sites are installing and configuring their control capabilities through our EBD program.
NTCP Service in Context
High-level NTCP Service Features

• Two-stage control system (propose, execute)
  – Satisfies key equipment site requirements (safety, protection of equipment investments)

• Reliability & robustness features
  – Allow client and server to recover from unusual/failure states

• Plugin architecture
  – Isolates site-specific code from NEESgrid-standard NTCP service code

• OGSI-compliance
  – Ensures that NEESgrid interoperates with other Cyberinfrastructure components (through compatible security and service frameworks)
NTCP Client APIs

• NTCP Client APIs allow software to control a physical or simulation component via the NTCP service.

• NTCP Client APIs are available, are documented, and are in use.
  – Java Client and Java “Helper” APIs are available and were used by Chef in MOST and MiniMOST. These are also used in NEESgrid acceptance testing and will be used in upcoming EBD activities.
  – C/C++ Client API is available for early adopter use. This will be used in upcoming EBD activities.
NTCP Plugins

• An NTCP Plugin links the NTCP Service to the local control system or simulation component.
  – The NTCP Plugin API (available in Java and C/C++) is documented and example Plugins are available for use.
  – Ultimately, it’s the equipment site’s or simulation code developer’s responsibility to “hook up” their components to the NEESgrid core control service.
  – The SI team has developed and tested a number of NTCP Plugins, resulting in many options and examples.
  – Some equipment sites have begun developing their own NTCP Plugins.

• NTCP Plugins have been used in a number of settings.
  – MOST Experiment
  – MiniMOST
  – EBD activities
  – Acceptance Testing and Equipment Site Validation

Equipment Control System or Simulation Code Control Interface

Locally-defined interface

NTCP Plugin

NTCP Plugin API

NEESgrid Standard NTCP Service

NEES Facility

Remote Users

Experiment Control/Coordination

NTCP Client API
NTCP Plugins Developed by SI

- Dummy Plugin
  - Unit testing, Equipment Site Validation
- Mplugin + Matlab NTCP Toolbox
  - Matlab control systems and simulation components (e.g., MOST experiment)
- LabView Plugin
  - LabView control systems, MiniMOST, Still digital camera control
- C Gateway Plugin + C Plugin API
  - Supports Plugins written in C/C++
- ShoreWestern Plugin
  - UIUC components in MOST experiment
NEESGrid Simulation
NEESgrid Simulation Team

- G.L. Fenves, UC Berkeley
- F. McKenna, UC Berkeley
- F.C. Filippou, UC Berkeley
- T. Haupt
  Mississippi State Univ.
- B. Spencer
Simulation Component
Objectives

• Provide capability for modeling and simulation of structural and geotechnical systems within NEESgrid.
• Create NEES open-source community for simulation software for future simulation application development.
• Provide interfaces from simulation software to NEESgrid data repositories using appropriate data models.
• Provide portal access to NEESgrid or other high-end compute resources.
• Provide Matlab framework for research, prototyping, and education in simulation.
NEESgrid Simulation Overview

Simulation Portal

OpenSees NCTP Plugin

Data Repository

Compute Resources
NEESport functionality

- Earthquake Model
  - view data
  - extract data

- Structure Model
  - select or define a structure (set parameters)
  - select location
  - run simulation
  - view data

- Inventory of Structures
  - select or define an inventory
  - run simulation (future)
  - view data
Earthquake model

- select EQ model
- model description
- visualize surface data
Earthquake model (2)

logical→physical; data extraction

Ground Motion Metadata Repository Service

Job Submission Service

NEESgrid Streaming Data Service

EJB

GRAM/MMJFS

GASS

DBMS

File System
Structural Model

- Select structural model
- Model description
- Model instances
- Set model parameters
Structural Model (2)

Structure Models Metadata Repository Service

EJB

DBMS

Grid Job Descriptor (model metadata)

Application Signature and Description

Parameters, arguments, i/o

JSDL-style descriptor (for automatic generation of RSL)

values of parameters, run-time info (for provenance service)
Population Method

- select population method
- population description
- population instances
Individual Structure Response

1. select location of structure
2. see acceleration history for the selected location
3. select structure instance
4. run openSees (structure response simulation)
5. visualize results
Individual Structure Response: Ground Motion Data
Individual Structure Response: Select Location

Ground Motion Metadata Repository Service
Structure Models Metadata Repository Service
Persistence & Provenance Service
Job Submission Service
NEESgrid Streaming Data Service

EJB
DBMS
GRAM/MMJFS
File System
GASS
Individual Structure Response: Select Structure & Run
Individual Structure Response: Replay
Individual Structure Response
SPUR/NEESgrid Grid Solution

select application  create & configure job  run & visualize  run & save

High-Level Job Submission Service

Job Instance

list applications

list application parameters

generate RSL

set values of parameters

Grid Job Metadata Service
Job Status Services
Provenance Service

NEESgrid (NCSA)
myProxy

NEESgrid (MSU)

NEESgrid Streaming Data Service

J2EE/EJB (MSU)

GRAM/MMJFS
GASS
GridFTP/RFT

GT2.4 (NCSA,PSC)
GT3 (MSU)
NEESGrid Data Model Efforts
Overall Data Modeling Efforts

Site Specifications
- Database

Project Description
- Domain Specific models
- Common Elements
- Data / Observations

Equipment
- Experiments
- Trials
- Tsunami Specimen
- Shake Table Specimen
- Geotech Specimen
- Centrifuge Specimen

People
- Units
- Sensors
- Descriptions

Data
- Data
- Data
- Data

NEES
- Site A
- Site B
- Site C
Existing Data Model Representations

- **E-R (Entity Relationship) Diagrams**
  - *Entities*, members of an objects set
  - *Attributes*, values describing some property of an entity
  - *Relationships*, connections among one or more entity sets

- **UML’s ORM (Object Role Models)**

- **XML (Extensible Markup Language) Schema**
  - Encoded in XML to describe document (data) structure
  - Introduces the ideas of data types, cardinality constraints

- **RDF (Resource Description Framework)**
  - Encoded in XML to describe resources with labeled relationships
  - More flexible than hierarchical organizations
  - Extensible: multiple RDF schemes can be combined

- **OWL (Web Ontology Language)**
  - Encoded in XML to describe classes and relations
  - Part of the Semantic Web Activity
Protégé-2000 (http://protege.stanford.edu)

Open Source Ontology Modeling Tool (with many Plugins)

- **A tool** which allows the user to construct a domain ontology
- **A platform** which can be extended with graphical widgets for tables, diagrams, animation components to access other knowledge-based systems
- **A library** which other applications can use to access knowledge bases
- Produces schemas in various data model representations
Prototype Data Model

- Tool: Protégé-2000
- Four groups of classes:
  - ProjectRelated
  - SiteSpecificInformation
  - CommonDataElement
  - CommonExperimentalElement
- Project-centric
- Shake table test (Stanford)
- Geotechnical / centrifuge tests (USC)
- Tsunami (Oregon State)
Observations

- Pre-experiment and post-experiment data could be as valuable as the actual experiment itself.
- Computer simulations play a significant role towards the design of an experiment as well as for post-event investigations.
### Project Entity – OrSt Model

#### Project Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>projectID</td>
<td>Primary key, or unique identifier for this Project.</td>
</tr>
<tr>
<td>specialConditions</td>
<td>Describes any special conditions, limitations, or features associated with the Project as a whole.</td>
</tr>
<tr>
<td>NEESCode</td>
<td>A reference code assigned by the NEES Consortium for the Project.</td>
</tr>
<tr>
<td>labCode</td>
<td>A reference code assigned by the specific lab for the Project.</td>
</tr>
<tr>
<td>title</td>
<td>Brief title for the Project. Since this will be used to identify the Project in lists and menus, we recommend limiting the title to approximately 20 characters.</td>
</tr>
<tr>
<td>shortDescription</td>
<td>Short, “display-style” description of the Project.</td>
</tr>
<tr>
<td>longDescription</td>
<td>More detailed description of the Project, its goals, and other high-level information.</td>
</tr>
<tr>
<td>startDate</td>
<td>Start date for the Project. Will always be on or before the startDate of the earliest Experiment belonging to the Project.</td>
</tr>
<tr>
<td>endDate</td>
<td>End date for the Project. Will always be on or after the endDate of the latest Experiment belonging to the Project.</td>
</tr>
<tr>
<td>createDate</td>
<td>Date when the Project record was added to the database.</td>
</tr>
</tbody>
</table>

#### Table Relationship

<table>
<thead>
<tr>
<th>Table Relationship</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project to PublicationReport</td>
<td>A Project will yield zero or more Publications/Reports (refer to Section 3.11 for information on the PublicationReport table). Publications/Reports that refer to just a particular Experiment, rather than the Project as a whole, are associated with that Experiment instead (refer to Section 3.2 for information on the Experiment table).</td>
</tr>
<tr>
<td>Project to Role</td>
<td>Role of a particular individual with respect to the Project.</td>
</tr>
<tr>
<td>Project to Acknowledgement</td>
<td>A Project will be associated with zero or more Acknowledgements (refer to Section 3.10 for information on the Acknowledgement table).</td>
</tr>
<tr>
<td>Project to PersonPermission</td>
<td>A Project may assign certain permissions to particular individuals, which will last throughout the Project (e.g., the PI may designate who has the authority to modify Project information in the database). That association is made through this relationship (refer to Section 3.9 for information on the PersonPermission table).</td>
</tr>
<tr>
<td>Project to Experiment</td>
<td>A Project includes zero or more Experiments (refer to Section 3.2 for information on the Experiment table).</td>
</tr>
</tbody>
</table>
Project Entity – Revised Model

Key Additions to OrSt Model:

- Project has many events, which categorized in five types
- All the events have trials and versions
- Project deals with certain specimen; but specimen modeling varies widely: domain dependent, project dependent, experiment dependent
Specimen Modeling

- Universal modeling of specimen for all experiments is very difficult if not impossible
- Goal is to provide ways to archive the data and information on the project and the experiment
- Basic formats and desirable features: CAD drawings; scratch drawings and notes; photos; narrative description; electronic notebook; linkage of drawings, sensor locations to data, etc.
Drawings Indicating Sensor Locations

THD-2 Transducer Location

with distributed bars

THD2 Curvature Instrument
THD2 Shear Deformation Instrument
THD2 Hinge Slippage Instrument

UNIVERSITY OF NEVADA RENO

2 - Way Bridge Column Hinge

DESIGNED BY:
CHECKED BY:
DATE:
REVISION

Courtesy of Gokhan Pekcan, Patrick Laplace
<xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:NEESMeta="http://protege.stanford.edu/NEESMeta#"
           xmlns:a="http://protege.stanford.edu/system#"
           xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#AcademicStaff"
               rdfs:label="AcademicStaff">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#Researcher"/>
    </rdf:Class>
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#Acceleration"
               rdfs:label="Acceleration">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#TimeHistorySeries"/>
    </rdf:Class>
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#Accelerometer"
               rdfs:label="Accelerometer">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#Sensor"/>
    </rdf:Class>
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#AdministrativeStaff"
               rdfs:label="AdministrativeStaff">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#Employee"/>
    </rdf:Class>
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#Analysis"
               rdfs:label="Analysis">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#CommonExperimentalElement"/>
    </rdf:Class>
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#Book"
               rdfs:label="Book">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#ScientificDocument"/>
    </rdf:Class>
    <rdf:Class rdf:about="http://protege.stanford.edu/NEESMeta#Boolean"
               rdfs:label="Boolean">;
      <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/NEESMeta#Number"/>
    </rdf:Class>
  </rdf:RDF>
NEESGrid Data Technologies
NEESgrid Data – Core Elements

- Local Repository
- Central Repository
- JAVA APIs – Run locally on the same system as a repository or over OGSA Web Services
  - NEES File Management Services
  - NEES Meta Data Services
- Data Viewers
  - Streaming (numeric, X/Y graph)
  - Stored (X/Y graph, 2-D structure, video)
Core Elements

Data Acquisition

Data/MD Ingest Tools

API

Grid and Web Services

NEESpop

Local Repository

Data Teamlets

API

NEESdata

Central Repository

Data Teamlets

API

Workstation

Data tools

Data viewers
A Simple Experimental Scenario

Developer System

DAQ System

Labview

Glue

Test Specimen

Researcher System
A Simulation Scenario
Boxology

Data Models

Experiment Management

Data Acquisition

Notebook

NEES Grid Data Approach

Experiment Monitoring

Central Repository

Local Repository

Data Analysis
Data Lifecycle

- Data Models
- Experiment Prep
- Experiment Management
- Data Monitoring
- Data Analysis
- Data Publishing
- Data Curation
- Data Discovery and Reuse
Data/Metadata Capture Throughout

- Data Models
- Experiment Prep
- Experiment Management
- Data Monitoring
- Data Analysis
- Data Publishing
- Data Curation
- Data Discovery and Reuse
Data Models

- Data models are developed in RDF
- Local repository supports multiple simultaneous data models with cross-model linkages
- Metadata browser (aka Project browser) becomes the Project Browser, Notebook Browser, Site Specification Database Browser
- Metadata browser can federate multiple sources of Metadata
Multiple Models

Project Model

Site Model

Proj

Exp

Trial

Sensor

Specimen

Element

Element

Site

Person

Facility

Equipment

Notebook

Chapter

Entry
Overall Data Modeling Efforts

Ref. Source: Chuck Severance
<owl:ObjectProperty rdf:ID="hasPublications">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Project"/>
        <owl:Class rdf:about="#Task"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain><rdfs:range rdf:resource="#Publications"/>
</owl:ObjectProperty>
Experiment Preparation

• Notebook
  – Allows the creation of *material* without needing a model
  – The model is pages, chapters, and “stuff”
  – It is all captured with data and metadata
  – A notebook can be attached to any object in the model structure
    (i.e. a project can have a notebook, a trial can have a notebook, etc…)

• Resources
• Discussions
• Project Browser
  – Setup basic structured metadata for the experiment - Trials, descriptions, sensors, etc… This material is captured in accordance to and with the data model
NEESgrid Experiment Data Flow

SiteSpecific

ExperimentalSetup

Data Model

ExperimentalElement

DataElement

NEESGrid Data Repository

Project Browser

Data Ingestion

Experiment Control

DAQ Disk

C

D

Streaming Viewer

Stored Viewer

Data Turbine

DAQ
Data Turbine

- Dynamic data server that provides a unified view of static and streaming data for universal data access
  - Video and multimedia
  - Test data acquisition
  - Telemetry streams
  - Real time monitoring
  - Delay tolerant networking

- Highly scalable by allowing linkage of multiple data turbine servers
- Interfaces to Matlab and Labview
Event: "Oregon Large Tank Test September 8, 2003"

Wave gauge at the front face of the cylinder

Pressure at the front face of the cylinder

Pressure at the front face of the cylinder
Capturing Video and Data

- PTZ/USB
- Still Capture
- DT Client
- Camera Control Gateway
- BT848
- Video Frames
- DT Client
- DT Main System
- DAQ
- Data Capture
- DT Client
- Simulation Coordinator
- Site A
- Site B
Data Monitoring Tools

Camera Control Gateway

DT Main System

Still Image / Camera Control

Still image camera control

Thumbnail

Create viewers
Video and Data Tivo

Thumbnail + Audio + Data
Summary

• As a Grid portal such as NEESGrid is developed, it reveals many requirements that were only vaguely understood before software development started.

• As “things” without user interfaces gain user interfaces, hidden flaws in the underlying “things” are revealed and must be fixed…

• The portal effort is not just a technical job, it becomes one of the major transformative catalysts for the field.

• Be careful assuming that you “know” too much at the beginning of the project.
Overall Summary

- There are many design choices and opportunities when developing a Grid Portal.
- JSR-168 and WSRP have turned to the Portal world upside down and given a chance to re-think many aspects of portals.
- While there is much complexity, the first task is to focus on using JSR-168 to build a set of basic reusable portlets to do the rather generic jobs.
- The Sakai effort is best though of as many portlets written for a particular task rather than a portal technology itself.
- When Sakai is completed it can be blended together with other JSR-168 portlets to produce a collaborative Grid Portal.
Questions

- Thank you for your time.
- On to the JSR-168 tutorial…