

# Numerical Simulations For Active Tectonic Processes: Increasing Interoperability And Performance

---

## Interoperability Milestone I

2/27/2003

Interoperability Prototype – Complete prototype described in Milestone "H" and test with improved codes. Review board approves.

- Demonstration of interface of Gateway and GeoFEST with simple visualization satisfying preliminary design requirements.
- Mesh generation - Demonstrate ingesting fault geometry and rheology from federated DB, to generate a starting mesh.
- Functional fault DB and documentation for Southern California.
- Riva: Produce movies of the strain, stress, and displacement data generated from Virtual California and GeoFEST of 1 km resolution for S. California in an integrated way through the grid framework.

---

## Team

**Geoffrey Fox:**  
**Information Architect**

Community Grid Computing  
Laboratory  
Indiana University  
501 N. Morton, Suite 224  
Bloomington, IN 47404-3730  
[gcf@indiana.edu](mailto:gcf@indiana.edu)  
812-856-7977

**Andrea Donnellan:**  
**Database Design and  
Implementation**

Jet Propulsion Laboratory  
Mail Stop 183-335  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099  
[donnellan@jpl.nasa.gov](mailto:donnellan@jpl.nasa.gov)  
818-354-4737

**Jay Parker:**  
**Overall Software Engineer**

Jet Propulsion Laboratory  
Mail Stop 238-600  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099  
[Jay.W.Parker@jpl.nasa.gov](mailto:Jay.W.Parker@jpl.nasa.gov)  
818-354-6790

**Marlon Pierce:**  
**Code Interoperability Software  
Engineer**

Community Grid Computing  
Laboratory  
Indiana University  
501 N. Morton, Suite 224  
Bloomington, IN 47404-3730  
[marpierc@indiana.edu](mailto:marpierc@indiana.edu)  
812-856-1212

**Dennis McLeod:**  
***Database Interoperability***

Professor  
Computer Science Department  
University of Southern California  
Los Angeles, CA 90089-0781  
[mcleod@usc.edu](mailto:mcleod@usc.edu)  
213-740-4504

**Lisa Grant:**  
***Fault Database Architect***

University of California, Irvine  
Environmental Analysis and Design  
Irvine, CA 92697-7070  
[lgrant@uci.edu](mailto:lgrant@uci.edu)  
949-824-5491

**Anne Chen:**  
***Database Implementation***

University of Southern California  
Mail Code 0781  
3651 Trousdale Parkway  
Los Angeles, CA 90089-0742  
[yunanche@usc.edu](mailto:yunanche@usc.edu)  
213-740-7285

**Miryha Gould:**  
***Population of Fault Database***

University of California, Irvine  
Environmental Analysis and Design  
Irvine, CA 92697-7070  
[miryha@uci.edu](mailto:miryha@uci.edu)  
949-824-5491

---

## **Overview**

---

The QuakeSim Problem Solving Environment has been developed according to the design document posted at:

<http://quakesim.jpl.nasa.gov/DesignDocument2.1.pdf>.

This milestone report documents the completion of a prototype system using this interoperable framework for linking processes on diverse remote host machines. We will describe posted supporting documents, the status of the QuakeSim Gateway framework, the resources and processes that have been integrated with this framework, and a demonstration example (“Running GeoFEST with visualization”) with available variations.

---

## **Supporting Documents**

---

The project software engineering plan is located at:

[http://www-aig.jpl.nasa.gov/public/dus/gem/CT/SW\\_Eng\\_plan.html](http://www-aig.jpl.nasa.gov/public/dus/gem/CT/SW_Eng_plan.html)

Requirements for the project have been defined and can be found at:

[http://quakesim.jpl.nasa.gov/CT\\_Requirements.doc](http://quakesim.jpl.nasa.gov/CT_Requirements.doc)

The initial Portal User's Guide (currently focused on the demonstration example, and including an appendix with the final report on guiVISCO) is posted at:

<http://quakesim.jpl.nasa.gov/PortalExample.pdf>

The Fault database is described here:

<http://quakesim.jpl.nasa.gov/FaultDatabase.doc>

RIVA examples are at:

<http://pat.jpl.nasa.gov/public/RIVA>

A complete description is pending JPL Document clearance.

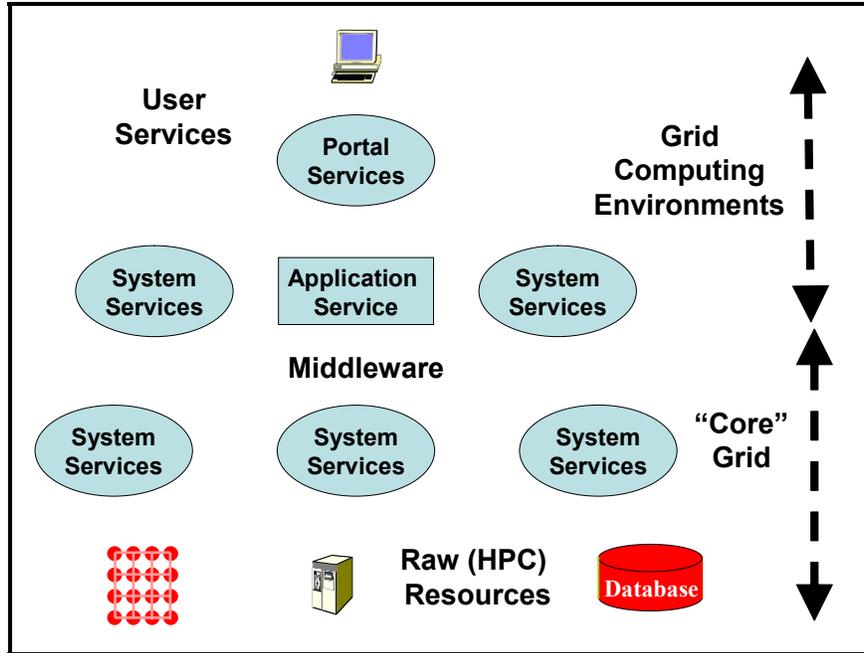
---

## **Portal Prototype**

---

We are building a new Problem Solving Environment (QuakeSim) for use by the seismological, crustal deformation, and tectonics communities for developing an understanding of active tectonic and earthquake processes. The top-level operational architecture of our proposed solid earth research virtual observatory (SERVO) shows science users interacting with interface programs as well as modeling, simulation, and analysis tools. The general architecture follows the "Web Services" model being developed by business interests, but is applied to scientific applications and supporting software resources (such as databases).

The system is divided into three tiers: a user interface layer (implemented as a browser interface), a system resource layer, and a middle control layer that maintains proxies (or brokers) to the system resources (Figure 1). The middle tier provides a uniform interface to the resource layer. Following the Web Services approach, we define XML interface abstractions (in WSDL) for basic services (such as File Management) and implement the interface with appropriate technologies (such as with a relational database). Communication between the services is done with an XML messaging architecture (SOAP).



**Figure 1.** High level architecture of planned system showing grids, portals, and grid computing environments.

One of the most critical aspects of our proposed system is supporting interoperability given the heterogeneous nature of data sources as well as the variety of application programs, tools, and simulation packages that must operate with data from our system. Interoperability will be implemented by using distributed object technology combined with development of object API's that conform to emerging standards. We will define our object API's in XML and dynamically map this specification into the chosen object model. This strategy was successfully used in the Gateway portal, which currently uses a CORBA middle tier but has used a pure Java solution with the same objects.

Our objective is to develop a system with the following specific components.

1. A database system for handling both real and simulated data.
2. Fully three-dimensional finite element code with adaptive mesh generator capable of running on workstations and supercomputers for carrying out earthquake simulations.
3. Inversion algorithms and assimilation codes for constraining the models and simulations with data.
4. A collaborative portal (Object Grid Framework) allowing for seamless communication between codes, reference models, and data.
5. Pattern recognizers capable of running on workstations and supercomputers for analyzing data and simulations.

For our development we will follow the software engineering plan but may add adaptations in cases where particular unique requirements emerge.

---

### ***Portal Status***

---

The GEM portal and testbed consists of the following pieces:

1. A portlet-based portal environment that allows users to customize their displays and service interfaces.
2. Support for the GEM codes Disloc, Simplex, and GeoFEST.
3. XML based Web services that allow the user to
  - Transfer files from desktop to backend and between backend.
  - Submit jobs
  - Monitor job progress on backend resources
  - Archive and resubmit old jobs
  - Visualize GeoFEST output
  - Exchange fault and dislocation information between Disloc and Simplex.
  - Services for managing portal applications.

The current testbed, uses following resources:

- Grids, a Sun Ultra 60 acts as the main web server, runs the portal, and manages connections to remote services through client stubs.
- The fault database is hosted at USC.
- Services for job submission, file management, and job monitoring are hosted on a range of Linux and Solaris servers (danube, noahsark, solar, and grids).
- Grids also hosts several local services for session management.

The portal is based primarily on the following technologies:

- Jetspeed is used to build the portal framework (for which we have added some extensions).
- JavaServer Pages are used to create the web pages.
- Remote invocations and communications between various machines and databases in the testbed use Web services running in the Apache Axis implementation of WSDL and SOAP.
- We use apache-ant (with various custom extensions) to coordinate related tasks (such as the steps needed to create the initial mesh or to generate the movie).

Details on the use of the portal for a GeoFEST simulation may be found in <http://quakesim.jpl.nasa.gov/PortalExample.pdf>.

---

### ***Fault database***

---

The database has a web interface that allows experts to enter fault and layer data. It also allows anyone to select for printing all the faults (or layers), or a subset that matches some of the search fields. There is also access directly to the underlying items by remote SQL queries. The Portal interacts with the database by this second method.

The database itself is designed to assist the documentation of entries through the interpretation fields such as Publication, Author, Year and Comments.

Faults may be made of many segments. The database supports this association. A segment may have multiple interpretations, as authors rely on different assumptions and field data.

Permitted users may operate on the data through an Internet connection. The user interface is browser-based. The interface is written in HTML and JavaScript for reliability and wide access. The user input is done by web forms written in HTML; a click on “submit” (here means “insert”, “select”, or “delete”),” invokes JSP processing that performs the interaction with the database itself.

The essential functions are “insert,” which includes data validity checking and creation or modification of a database entry, “select” which retrieves data based on a matching query, and “delete” which removes an entry.

Currently supported items are Interpretations (including literature reference material), Faults, (which are made up of) Segments, and Layers. Layers indicate the rheology, including quantitative physical values such as elasticity constants.

Currently the database is populated with a small number of examples. Many more, including all entries from the California Geological Survey database will be entered this summer.

Complete descriptions may be found in:

[http://quakesim.jpl.nasa.gov/Fault\\_Da.doc](http://quakesim.jpl.nasa.gov/Fault_Da.doc)

---

***Model assembly and mesh generation (with iterative smoothing)***

---

We use a Linux-based toolset (originally collected within guiVISCO as reflected in the appendix of <http://quakesim.jpl.nasa.gov/PortalExample.pdf> ) developed by Jin-fa Lee of Ohio State University in part from a NASA SENH sponsored collaboration. The Portal does not currently produce the kinds of graphical feedback described in the guide, but does support the geometry and meshing functions described there. This set of tools is designed with the current problem type in mind, where a problem is assembled as a stack of layers in which are embedded rectangular fault segments. Currently the high-level interface supports only such problems, but nonrectangular faults and layers can be added in the future at this high level. All meshing and refinement tools are already compatible with block-wise materials and complex faulting.

The system develops a tetrahedral mesh in two kinds of steps. The first simply fills the domain with a minimal set of tetrahedra that fill the volume and are consistent with all surfaces including faults. The second is a refinement step that adds more points (and tetrahedra) in a region indicated as deficient. This indication is by a simple list of elements so far as the mesh generator is concerned. The generation of such a list may use any criterion designed by the higher-level programmer.

One type that is supported within the portal is a maximum limit. By specifying in a form a size no larger than 30 (km), a loop is executed that performs refinement until this limit is reached for all elements.

A second type compares element size with a scale that is a mathematical function of the geometry. For Milestone E, we chose a function that seems reasonable (imposed by a process written in Perl, invoked between each refinement cycle). This function requires elements to be small close to the edges of the fault (where high stress develops). The function chosen is that size be smaller than the distance to the midpoint of the element, down to a minimum size (so the process terminates). This minimum size is also controllable from the Portal. For the Milestone E case the mesh had elements near the fault with size about 20 times smaller than the largest elements at the domain boundary.

A third type would feed back properties of an initial finite element solution (such as the strain energy density or an error estimate) back to the mesh refinement. This has not been explored for faulting problems so far.

For the sake of controllability and smoothness of the mesh, only a limited number of elements are added at each refinement step. A Portal user should expect to repeat the refinement step 5-10 times to obtain a high quality mesh.

---

### ***Submission to GeoFEST***

---

The mesh created by the tools in the previous step consists of geometric entities (points, tetrahedral). In order to submit this to GeoFEST a set of boundary conditions (including slip at faults), time step and process controls must be added. This is performed by a quick process written in Perl. This process was adapted from the GeoFEST Milestone E benchmark case, and has not been fully generalized. This leads to an arbitrary (and temporary) restriction on the type of problem available within the current Portal example: the domain must have three layers (of any thickness), and one embedded fault. The preparation process also creates a file of surface triangles that are used for interpolation by RIVA.

---

### ***Remote Interactive Visualization and Analysis System (RIVA)***

---

RIVA (Remote Interactive Visualization and Analysis System) is a parallel terrain rendering system begun in 1994. RIVA can be used as an interactive system to explore and visualize large terrain dataset in 3-D perspective views. It can also be used as an animation tool to generate fly-by movies using high-resolution images and digital elevation. The software will be available through the JPL HPCC Earth and Space Sciences Project Software Catalog.

For the demonstration, the RIVA visualization component takes two input files, the surface triangle mesh generated by GeoFEST mesh generator, and the output file produced by the GeoFEST simulation. It contains three programs:

1. The interpolator. This extracts the surface deformation data from the GeoFEST output file, interpolates them into a regular mesh, and converts the deformation data into RGB colors similar to InSAR fringes. The interpolator outputs a multiple timestep image file in a tiled format used by RIVA, the parallel terrain renderer.
2. The parallel rendering program, RIVA. This is used to render the GeoFEST output overlaying on top of a LandSAT image with digital elevation map. RIVA produces movie frames.
3. The mpeg encoder. This converts the movie frames into an mpeg movie.

The computation is done on an 8 processor SGI Origin 300 machine located at JPL. It was invoked by the GEM Web Portal. When the movie is generated, a message is sent back to the user with a URL pointing to the mpeg movie for retrieval.

For ease of early implementation, the simulation size was fixed at 100 time steps. Fixed region is the biggest limitation right now. We need more information about the faults and the region of the simulation in order to carve out the corresponding region from the LandSat data. The second limitation is no interaction. There is no way to change the flight path and other rendering parameters via the portal.

---

### ***Portal example demonstration***

---

In order to speed evaluation of portal capabilities and address the specific details of this milestone, a demonstration has been constructed. This demonstration has specific steps described in the PortalUserGuide.doc. The example uses all the parts of the system described in these milestone goals:

Demonstration of interface of Gateway and GeoFEST with simple visualization satisfying preliminary design requirements

Mesh generation - Demonstrate ingesting fault geometry and rheology from federated DB, to generate a starting mesh.

Riva: Produce movies of the strain, stress, and displacement data generated from Virtual California and GeoFEST of 1 km resolution for S. California in an integrated way through the grid framework.

In the Project Input step, the user may select faults and layers from the federated database. The parameters of these entities appear on forms where the user may approve or edit them.

At the Create Initial Mesh step the user may specify a maximum “Mesh size” and “Mesh Refine Limit” (sizes of largest and smallest elements). “Generate Mesh” creates the initial mesh. “Refine Mesh” allows insertion of additional mesh detail. “Execute GeoFEST” and “Launch GeoFEST” complete the input file and submit the 100 time step GeoFEST job for (currently) a sequential workstation solution. The solution file is automatically transferred to the RIVA host at JPL, where the parallel visualization is completed (exceeding the requirement for “simple visualization:” additional geometry and mesh visualization is in progress). The user is sent a URL of the animation when the job is completed.