

# Introducing a New Paradigm for Computational Earth Science— A web-object-based approach to Earthquake Simulations

Geoffrey C. Fox

*School for Computational Science and Information Technology and Department of Computer Science,  
Florida State University, Dirac Science Library, Tallahassee, Florida*

Ken Hurst, Andrea Donnellan, and Jay Parker

*Jet Propulsion Laboratory/California Institute of Technology, Pasadena California*

Computer simulations will be key to substantial gains in understanding the earthquake process. Emerging information technologies make possible a major change in the way computers are used and data is accessed. An outline of a realizable computational infrastructure includes standardization of data accessibility, harnessing high-performance computing algorithms, and packaging simulation elements as distributed objects across wide networks. These advances promise to reduce dramatically the frustration and cost of doing earthquake science as they transform the fragmentary nature of the field into one of integration and community.

## 1: INTRODUCTION

Earthquakes in urban centers are capable of causing enormous damage. The recent January 16, 1995 Kobe, Japan earthquake was only a magnitude 6.9 event and yet produced an estimated \$200 billion loss. Despite an active earthquake prediction program in Japan, this event was a complete surprise. The 1989 Loma Prieta and 1994 Northridge earthquakes were also unexpected and caused billions of dollars of damage as well as loss of lives. Partly as a result of these events the volume of earthquake related data being collected is rapidly increasing. Simulations are the necessary next step in order to understand earthquakes and examine the broad parameter space related to them.

The overarching goal of earthquake physics is to characterize and predict the behavior of systems of earthquake faults. One way to address this question is to take advantage of the currently available computational power and ever increasing wealth of data to construct realistic models of the earthquake process. These models can then be run for hundreds of thousands of virtual years to span several cycles of the entire system, and a statistical mechanics approach applied to look for patterns in both the synthetic and real seismicity and crustal deformation catalogues.

Recent work in earthquake physics has focused largely on understanding dynamic rupture processes, such as how ruptures grow into large earthquakes and how faults heal themselves. Other work has focused on analyzing observed seismicity in an attempt to look for precursory activity. Much more recently investigators have begun studying earthquakes using a systems approach in which individual faults interact with an entire system of faults.

In parallel efforts, a great wealth of geophysical data pertaining directly to the earthquake problem is being collected. The traditional seismic networks are being improved with broadband and strong ground motion instruments. GPS networks have been expanding globally and have virtually replaced the more traditional trilateration and triangulation networks. Concerted efforts in paleoseismology have added a wealth of data on the surface characteristics of major faults, particularly in southern California.

Rapidly expanding datasets and the advent of object broker technologies make it possible to pursue the development of complex, sophisticated models for predicting the behaviors of fault systems. Surface geodetic, seismicity, strong motion, and other data provide the necessary constraints for carrying out realistic simulations of fault interactions. Information technology is providing the means for defined and accessible data formats and code protocols that allow for communicating distributed components, the basis for the next wave of advanced simulations. The emerging technologies of IT enable frameworks for documentation of codes and standards as well as visualization and data analysis tools that will make for true community-based efforts. Without such tools it will be impossible to construct the more complex models and simulations necessary to further our understanding of earthquake physics.

Simulations are critical to understanding the behavior of fault systems, a major problem in earth science, because earthquakes occur in the real earth at irregular intervals on timescales of hundreds to thousands of years. Simulations generate arbitrarily long seismicity catalogs and provide a numerical laboratory in which the physics of earthquakes can be investigated from a systems viewpoint.

Simulations also provide invaluable feedback for the planning and design of future data collection efforts. Emerging information technology applied to the data and analysis will result in a revolutionary change in the manner in which earth scientists explore and analyze data. The cost of data access will plummet while the usefulness of data will multiply. Forging the feedback loop between the simulations and design of data collection efforts will further enhance the value and utility of the data.

Recent research indicates that the phenomena associated with earthquakes occur over many scales of space and time. Understanding the dynamic processes responsible for these events will require not only a commitment to develop the necessary observational datasets, but also the technology required to use these data in the development of sophisticated, state-of-the-art numerical simulations and models. The models can then be used to develop an analytical and predictive understanding of these large and damaging events, thus moving beyond the current, more descriptive approaches routinely employed. Future approaches emphasizing the development of predictive models and simulations for earthquakes will be similar to methods now used to understand global climate change, the onset of the El Niño-Southern Oscillation events, and the evolution of the polar ozone depletion zones.

### *1.1: Current Problems in Earthquake Physics*

The goal of the earthquake physics working group of the Southern California Earthquake Center (SCEC) is to model the evolution of seismic histories and space-time evolution of the stress field in southern California in order to understand space-time clustering of earthquakes. The focus has been on modeling two-dimensional networks of faults and on simulating the occurrence of large and intermediate-magnitude earthquakes to compare with contemporary earthquake catalogs.

It is not surprising that until recently the major focus in earthquake physics has been on the rupture process since seismic data represent the largest data set and the catalogue extends approximately half a century back from present. Since seismographs record earthquakes they give us insight into the elastic response of the earth's crust and into the rupture process. In contrast, new data types such as crustal deformation data provide us with information about the processes leading up to failure as well as information on the static response from earthquakes. Surface deformation data can be used as boundary conditions to continuum models of the earthquake cycle.

The current state of the science of earthquake physics is rather disjoint. Several investigators have constructed complex and realistic models of a single facet of earthquake processes, while others have performed statistical analysis of seismicity. Computer performance is now such that these facets can be joined together into comprehensive models of the entire earthquake process.

Simulations are the only comprehensive means to study earthquake fault systems because earthquakes occur on timescales of decades to thousands of years. Creating physically realistic simulations requires an understanding of the mechanical properties of faults and the bulk material surrounding the faults, perhaps inclusive of the entire lithosphere. Building such understanding requires a concentrated modeling effort. Thus earthquake simulations of increasing realism will help aid planning efforts by predicting the level of shaking from a hypothetical event, and will also help focus attention on areas and phenomena that need additional measurements or theoretical development, leading to yet more realism in the simulations.

### *1.2: Computational Overview*

There is substantial international interest in the use of large-scale computation in the earthquake field including an activity in Japan where major computational resources are being deployed and an effort among several Asia-Pacific nations including USA (the so-called APEC initiative). Here we will focus on an American activity known as GEM for its goal to produce a "General Earthquake Model" (<http://www.npac.syr.edu/projects/gem>, <http://milhouse.jp-l.nasa.gov/gem>).

There are currently no approaches to earthquake forecasting that are uniformly reliable. The field uses phenomenological approaches, which attempt to forecast individual events or more reliably statistical analyses giving probabilistic predictions. The development of these methods has been complicated by the fact that large events responsible for the greatest damage repeat at irregular intervals of hundreds to thousands of years, and so the limited historical record has frustrated phenomenological studies. Direct numerical simulation has not been extensively pursued due to the complexity of the problem and the (presumed) sensitivity of the occurrence of large events to detailed understanding of earth constituent make up, myriad initial conditions, and the relevant micro-scale physics which determines the underlying friction laws.

This field is different from many other physical sciences such as climate and weather, as it so far has made little use of parallel computing and only now is starting its own "Grand Challenges". It is thus not known how important large-scale simulations will be in Earthquake science. Nevertheless it is essentially certain that they can provide a numerical laboratory of semi-

realistic earthquakes which will enable development and testing of other more phenomenological methods based on pattern recognition. Also, simulations provide a powerful way of integrating data into statistical and other such forecasting methods as has been demonstrated in the use of data assimilation techniques in other fields.

This field has some very challenging individual simulations but it has only just started to use high performance computers. Therefore the most promising computations at this stage involve either scaling up existing simulations to large system sizes with modern algorithms or integrating several computational components with assimilated data, thus creating prototype full fault-system simulations. The latter has important real-world applications in the area of crisis response and planning. For example, one may carry the computations through from initial sensing of stress build up through the structural simulation of building and civil infrastructure responding to propagating waves. This is discussed briefly in Sections 5.3 and 5.4.

Earthquake fault modeling exhibits many different types of codes discussed in Section 3, which eventually could be linked together to support either real time response to a crisis or fundamental scientific studies. The computational framework GEMCI introduced in Section 2 has been carefully designed to support multiple types of component-component linkage including rich user front ends, termed problem solving environments.

Indeed, such a comprehensive modeling environment must be established to enable full exploitation of the new and expanding types of geophysical data relevant to both earthquake science and engineering. Codes must be able to communicate to allow for cross-validation between models. The wealth of geophysical data and modeling codes now being collected and developed needs to be standardized to allow rapid and easy sharing of information, retrieval of data, and model development and validation. At present, researchers laboriously transform the data formats into ones useable by their own individual programs. Current practices that are frustrating and wasteful include FTP of data sets larger than needed, parsing and reformatting of data, and transcription of information directly from printed papers.

Existing modeling and simulation programs are fragmented; however, most of the pieces are in place to construct powerful integrated simulations. We can envision this approach providing the key to crustal data assimilation, and successfully addressing the self-consistent use of physics at multiple scales. For example, kinematic models use surface deformation data, and digital elevation models (DEMs) to estimate plate and microplate motions. Elastic models use surface observations to estimate fault geometries and slip rates. Viscoelastic calculations incorporate surface deformation, DEMs, and lab, heat-flow, stress, seismic and geologic data into models that produce estimates of crustal rheology and structure, fault geometry, fault slip and stress rates, heat flow, gravity, and refined estimates of surface deformation. These provide the inputs for quasi-static dynamic models of single faults and systems of faults. The quasi-static dynamic models provide space-time patterns, correlations, and information about fault interactions.

As the models become more complex and as the data volume continues to increase we must harness high-performance techniques that allow fast data incorporation and running more and more complex models in shorter and shorter times. Some promising new techniques include fast multipole methods (described in Section 4), pattern recognition, statistical mechanics, and adaptive meshing techniques. Tasks easily carried out for simple problems need to be automated for complex problems such as southern California.

A comprehensive computational infrastructure would result in numerous investigators utilizing the solid earth data and models produced. Rather than current practices which rarely go beyond sharing results, these technologies will make it easy (when desired) to share and probe mathematical assumptions and cross-check the simulations and analysis methods while a particular topic is hot (for example during the data collection and analysis phase that always follows major earthquakes). The cost per scientific output would drop dramatically. Instead of spending long hours manipulating data, investigators will be exploring and interpreting the data.

The value will further increase because more sophisticated models and tools will be created and tested speedily within this comprehensive environment. For example, data assimilation tools will allow ingestion and comparison of large volumes of data rather than small subsets; three-dimensional adaptive meshing technology for constructing finite element meshes makes three-dimensional finite element modeling of complex interacting fault systems practicable.

### *1.3: Geoscience Overview*

Earthquake science embodies a richness present in many physical sciences as there are effects present spread over more than ten orders of magnitude in spatial and temporal scales (Figure 1).

Success requires linking numerical expertise with the physical insight needed to coarse grain or average the science at a fine scale to be used phenomenologically in simulations at a given resolution of relevance to the questions to be addressed. Again the nonlinear fault systems exhibit a wealth of emergent, dynamical phenomena over a large range of spatial and temporal scales, including space-time clustering of events, self-organization and scaling. An earthquake can be modeled as a clustering of slipped fault segments as seen in studies of critical phenomena. As in the latter field, one finds (empirically) scaling laws that include the well-known Gutenberg-Richter magnitude-frequency relation, and the Omori law for aftershocks (and foreshocks). Some of the spatial scales for physical fault geometries include:

**The microscopic scale** ( $\sim 10^{-6}$  m to  $10^{-1}$  m) associated with static and dynamic friction (the primary nonlinearities associated with the earthquake process).

**The fault-zone scale** ( $\sim 10^{-1}$  m to  $10^2$  m) that features complex structures containing multiple fractures and crushed rock.

**The fault-system scale** ( $10^2$  m to  $10^4$  m), in which faults are seen to be neither straight nor simply connected, but in which bends, offsetting jogs and sub-parallel strands are common and known to have important mechanical consequences during fault slip.

**The regional fault-network scale** ( $10^4$  m to  $10^5$  m), where seismicity on an individual fault cannot be understood in isolation from the seismicity on the entire regional network of surrounding faults. Here concepts such as "correlation length" and "critical state" borrowed from statistical physics have led to new approaches to understanding regional seismicity.

**The tectonic plate-boundary scale** ( $10^5$  m to  $10^7$  m), at which Planetary Scale boundaries between plates can be approximated as thin shear zones and the motion is uniform at long time scales.

## 2: GEM COMPUTATIONAL INFRASTRUCTURE (GEMCI)

### 2.1: Introduction

The components of GEMCI can be divided into eight areas.

1) Overall Framework including agreement to use appropriate "commodity industry standards" such as XML (a language for metadata) and CORBA (a distributed object access standard and broker), as well as more specialized high performance computing standards like MPI (Message Passing Interface).

2) Use of GEMCI to construct multiple Problem Solving Environments(PSE's) to address different scenarios.

3) Web-based User Interface to each PSE

4) Simulation engines built in terms of the GEMCI framework

5) Geophysical-specific libraries such as modules to estimate local physics and friction. These would also use the GEMCI framework which would already include generic libraries

6) Data analysis and Visualization

7) Data Storage, indexing and access for experimental and computational information

8) Complex Systems and Pattern Dynamics Interactive Rapid Prototyping Environment (RPE) for developing new phenomenological models -- RPE includes analysis and visualization aspects and would be largely on the client (the local light-weight workstation). In contrast, the large simulations in 4) above, are naturally thought of as distributed server side computational objects.

In the remainder of this Section, we describe the overall GEMCI framework and how it can be constructed in terms of components built according to emerging distributed object and Web standards and technologies. This describes the "coarse grain" (program level) structure of the GEMCI environment. There are myriad important details inside each module (or grain), which could be a finite element simulation code, data streaming from a sensor, a visualization subsystem, a Java eigensolver used in the client RPE or field data archived in a web-linked database. In Section 3, we describe some of the existing simulation modules available to the GEM collaboration while Section 4 goes into one case in detail. This is the use of fast multipole methods in large-scale Green's function computations. Section 5 illustrates how these ideas can be integrated together into a variety of different scenarios. These essentially correspond to different problem solving environments that can be built by using the same GEMCI framework to link GEM components in various ways. In Section 6, we make brief remarks on other parts of the GEMCI framework and speculate a little on the future.

There are several important activities which have pioneered the use of object based techniques in computational science. *Legion* has developed a sophisticated object model optimized for computing (<http://www.cs.virginia.edu/~legion/>) and such a framework could be integrated into GEMCI. Currently we are focussing on broad capabilities available in all important distributed object approaches and we can refine this later as we develop more infrastructure. *Nile* developed the use of CORBA for experimental data analysis (<http://www.nile.utexas.edu/>) but we need a broader functionality. *POOMA* is an interesting technology developed at Los Alamos (<http://www.acl.lanl.gov/Poo-maFramework/>) aimed at object oriented methods for finer grain objects used to build libraries as discussed in Section 2.5. GEMCI could use modules produced by POOMA as part of its repository of coarse grain distributed components.

### 2.2: Distributed Objects and the Web

The nature of the web demands that any computing environment based in it must be based on distributed object models. In the following sections we discuss several aspects of the GEMCI in light of the demands of the web environment.

*2.2.1: Multi-Tier Architectures.* Modern information systems are built with a multi-tier architecture that generalizes the traditional client-server to become a client-broker-service model. This is seen in its simplest realization with the classic web access which involves 3 tiers – the Browser runs on a client; the middle-tier is a Web server; the final tier or backend is the file system containing the Web page (Figure 2). One could combine the Web server and file system into a single entity and return to the client-server model. However the 3-tier view is better as it also captures the increasingly common cases where the Web Server does not access a file system but rather the Web Page is generated dynamically from a backend database or from a computer program invoked by a CGI script. More generally the middle tier can act as an intermediary or broker that allows many clients to share and choose between many different backend resources. This is illustrated in Figure 3, which shows this architecture with the two interfaces separating the layers. As we will discuss later the specific needs and resources of the Earthquake community will be expressed by metadata at these interfaces using the new XML technology.

This 3-tier architecture (often generalized to a multi-tier system with several server layers) captures several information systems which generalize both the web-page access model of Figure 2 and the remote computer program invocation model of Figure 4.

The architecture builds on modern distributed object technology and this approach underlies the “Object Web” approach to building distributed systems. Before describing this, let us define and discuss several computing concepts.

*2.2.2: Clients, Servers and Objects.* A *server* is a free standing computer program that is typically multi-user and in its most simplistic definition, accepts one or more inputs and produces one or more outputs. This capability could be implemented completely by software on the server machine or require access to one or more (super)computers, databases or other information resources such as (seismic) instruments.

A *client* is typically single-user and provides the interface for user input and output. In a distributed system, multiple servers and clients, which are in general geographically distributed, are linked together. The clients and servers communicate with *messages* for which there are several different standard formats.

An (electronic) *object* is essentially any artifact in our computer system. Two examples among many are displayed in Figure 4. One kind is simply a computer program. The best-known distributed object is a web page.

In the example of Figure 2, a *Web Server* accepts an HTTP request and returns a web page. HTTP is a simple but universal protocol (i.e. format for control information) for messages specifying how Web Clients can access a particular distributed object – the web page. A Database Server accepts an SQL request and returns records selected from the database. SQL defines a particular “Request for Service” in the interface of Figure 3 but usually the messages containing these requests use a proprietary format. New standards such as JDBC (The Java Database Connectivity) imply a universal message format for database access with vendor dependent bridges converting between universal and proprietary formats. An Object Broker as in Figure 4 uses the industry CORBA standard IIOP message protocol to control the invocation of methods of a distributed object (e.g. run a program). IIOP and HTTP are two standard but different protocols for communicating between distributed objects.

*2.2.3: Services in a Distributed System.* Distributed objects are the units of information in our architecture and we need to provide certain critical operating system *services* to support them. Services include general capabilities such as “persistence” (store information unit as a disk file, in a database or otherwise), and “security” as well as more specialized capabilities for science such as visualization. One critical set of services is associated with the unique labeling of objects and their look-up. We are familiar with this with Domain Name Servers and Yellow Page services for computers on the Internet. Web pages with URL’s build on this technology and Web search engines like Alta Vista provide a sophisticated look-up service. More general objects can use natural variants of this approach with a possibly arcane URL linking to a database, supercomputer or similar resource. The “resource” interface in Figure 3 defines the properties of back-end resources and how to access them. In particular it defines the equivalent of a URL for each object. The set of these resource specifications forms a database, which defines a distributed object repository.

*2.2.4: XML Extended Markup Language.* The new XML technology is used to specify all resources in the GEMCI. A good overview of the use and importance of XML in Science can be found in *Bosak and Bray* (1999, <http://www.sciam.com/1999/0599issue/0599bosak.html>) and we illustrate it in Figure 5, which specifies a computer program used in a prototype GEM problem solving environment described later and shown in Figure 15.

Readers who are familiar with HTML will recognize that XML has a similar syntax with elements defined by nested tags such as `<application > .... </application>`. This information is further refined by attribute specification as in the string `id=disloc`, which naturally indicates that this is the disloc application code. XML is a very intuitive way of specifying the structure of digital objects as simple ASCII byte streams. One could equally well specify the same information by appropriate tables in an object-relational database like Oracle, and indeed XML files can easily be stored in such a database if you require their powerful data access and management capabilities. Correspondingly relational database contents can easily be exported to XML format. XML is obviously more easily written and read than a complex database. Further there are growing set of powerful tools which can process XML – these include parsers, editors and the Version 5 browsers from Microsoft and Netscape which can render XML under the control of a powerful language XSL which specifies the mapping of tags to the display device. In the multi-tier generalization of Figure 3, one has multiple linked servers in the middle tier. These servers exchange

information, which is expressed as a stream of objects. These objects need to be “serialized” so as to be transportable between servers. There are many ways of converting complex data-structures into a stream of bytes but XML is one of the most general and often the best choice.

*2.2.5: Dynamic Resources.* Traditionally one labels a computer with an IP address that reflects a particular physical domain, or addresses a web page with a URL, which reflects a particular server and file system. This approach is appropriate for a fixed resource but not well designed for mobile or dynamic resources such as palm top devices and the growing number of Internet enabled consumer products. These resources are often transient and one cannot assume their continuous availability. There is substantial academic and commercial interest in new object registration, look-up and connection approaches supporting fault tolerance and dynamic clients and servers. Objects must be self defining and able to announce themselves universally to a network of registration servers. Jini from Sun Microsystems (<http://www.sun.com/jini/>) and the Ninja system from UC Berkeley (<http://ninja.cs.berkeley.edu/>) are well known examples of new approaches to dynamic objects. This technique can be used for both programs and data sets as objects. Thereby either new compute capabilities or new information (raw from sensors or processed) can be dynamically incorporated into simulation systems.

*2.2.6: The Object Web.* The Object Web signifies the merger of distributed object and web technologies, which is implicitly embodied in the discussion above. There are four rather distinct but important object standards.

*CORBA* is the Industry Standard supporting objects in any language on any platform. New features in CORBA tend to be deployed relatively slowly as they have a cumbersome approval process and must satisfy complex constraints.

*COM* is the Microsoft standard, which is confined to PC’s but broadly deployed there and high performance.

*Java* is the software infrastructure of the web and although single language, the same object can be used on any computer supporting the Java VM.

*XML* comes from the Web Consortium and has been briefly described above. It can be used to define general objects in an intuitive format illustrated in Figure 5.

The *Pragmatic Object Web* implies that there is no outright winner in the distributed object field and one should mix and match approaches as needed. For instance, CORBA Objects can use interfaces (as in Figure 3) defined in XML, Clients and Servers programmed in Java, with rendered displays using COM.

### *2.3: Architecture of the GEMCI Problem Solving Environment*

A Problem Solving Environment or PSE is an application that integrates access to the data, computers and tools needed for a particular computational science area. There is general agreement that Object Web technology is the natural software infrastructure for building PSE’s, for the entities one needs to integrate can be considered as distributed objects. With this choice, there is a growing trend to term web-based PSE’s as portals in analogy to the term used to describe environments built commercially to allow access to personal or business information. Commercial portals allow both administrative and user customizability from a suite of objects and services supporting them.

GEMCI illustrated in Figure 6, is an Object Web PSE and has the classic three-tier structure discussed above.

In GEMCI, everything is a “distributed object” whether it be a simulation on a supercomputer, the basic GEM Web pages, the notes from a field trip entered on a palm top, CNN real-time coverage of the latest earthquake or the data streaming in from sensors. GEMCI provides an integrated view of these diverse resources with XML definitions for the raw objects themselves and the data they produce. The services shown in Figure 6, from collaboration, security, object discovery, visualization and computer access, are generic to all computing portals. Building GEMCI using the same approach and tools as other portals ensures the availability of these services. They will require customization as for instance there are many different visualization packages and each requires non-trivial work to include in such a portal. Again, collaboration corresponds to sharing distributed objects, and this can currently only be automated for some objects. Many web pages can be shared using generic techniques illustrated in Figure 5 but sharing say the control and output of a general simulation can require quite a lot of custom modifications.

Most importantly, to use GEMCI shown in Figure 6, one needs to define the entities in the GEM environment as distributed objects. For computer programs this implies a rather arcane process termed “wrapping the program as a distributed object”. Operationally this implies allowing a middle-tier server (the CORBA object broker or Java application Server) to be able to run the program on one or more machines, specify the input files and either specify output files or access them as streams of data in the fashion of UNIX pipes. Each distributed object technology has a rather different approach to this using what is termed an IDL or Interface Definition Language and specialized Java and C++ code to implement the wrapping.

We can use the concept of the Pragmatic Object Web to simplify this process. Our strategy is to define all relevant properties of computer programs in XML as illustrated in Figure 5. These properties are used to generate the needed object wrappers, either statically or dynamically. This approach requires the user specify what they know – the properties of their program – while the filter copes with the obscure syntax of each object model. Obviously this also allows one to support all object models

– COM CORBA, and Java – by changing the filter. In this way one can adapt to changes in the commercial infrastructure used in the middle tier.

One must apply the XML object definition strategy to all entities in GEMCI; programs, instruments and other data sources and repositories. This gives the metadata defining macroscopically the object structure. In addition, one needs to look at the data stored in, produced by or exchanged between these objects. This data is itself a typically a stream of objects, each an array, a table or more complex data structure. One could choose to treat the data at some level as an unspecified (binary) “blob” with XML defining the overall structure but detailed input and output filters used for the data blobs. As an example, consider the approach that an electronic news organization could take for their data. The text of news flashes would be defined in XML but the high volume multimedia data (JPEG images and MPEG movies) would be stored in binary fashion with XML used to specify <IMAGEOBJECT> or <MOVIEOBJECT> metadata.

As explained earlier, systematic use of XML allows use of a growing number of tools to search for, manipulate, persistently store and render the information. It facilitates the linkage of general and specific tools/data sources/programs with clearly defined interfaces. This will help the distributed GEM collaborators to separately develop programs or generate data, which will be easily able to interoperate.

More generally XML standards will be defined hierarchically starting with distributed information systems, then general scientific computing and finally application specific object specifications.

For example GEMCI would develop its own syntax for seismic data sensors but could build on general frameworks like the XSIL scientific data framework developed by Roy Williams at Caltech (<http://www.cacr.caltech.edu/SDA/x-sil/index.html>). XSIL supports natural scientific data structures like arrays and the necessary multi-level storage specification.

Another example is MathML which provides XML support for the display and formulation of Mathematics. We can expect MathML to be supported by tools like Web Browsers and white boards in collaborative scientific notebooks and allow one to enhance theoretical collaboration in GEM. There will for instance be modules that can be inserted into applications for parsing MathML or providing graphical user specification of mathematical formulae. This could be used in sophisticated implementations of the Complex Systems and Pattern Dynamics Interactive Rapid Prototyping Environment with scripted client side specification of new analysis methods. One can also use MathML in high level tools allowing specification of basic differential equations that are translated into numerical code. This has been demonstrated in prototype problem solving environments like PDELab but have so far not had much practical application (*Houstis et al.*, 1998 and <http://www.cs.purdue.edu/research/cse/pdelab/pdelab.html>). Greater availability of standards like MathML should eventually allow more powerful interchangeable tools of this type.

Finally we can mention a set of graphical XML standards such as X3D (3 dimensional objects) and VML which is a vector graphics standard, which can be expected to be important as basis of application specific plot and drawing systems.

#### 2.4: Building the GEMCI Problem Solving Environment

We have described above some aspects of the base strategy of defining GEM components as distributed objects. Here we discuss some existing experience from NPAC at Syracuse in integrating such objects and tools that manipulate them into an overall environment. The NPAC team has built several exemplar problem solving environments for both the NSF and DoD HPCMO (High Performance Computing Modernization Office) supercomputer centers. In Figure 7, we show some useful tools including a first cut at a “wizard” that helps produce the distributed object wrappers described above.

This AAD (Abstract Application Descriptor) can be extended to allow specification of all needed input parameters of an application, with an automatic generation of input forms respecting default values and allowed value ranges. The object wrappers of an application should not only invoke the code and allow parameter specification but have built in description/help systems.

One major NPAC Problem Solving Environment was built for the Landscape Management System (LMS) project at the U.S. Army Corps of Engineers Waterways Experiment Station (ERDC) Major Shared Resource Center (MSRC) at Vicksburg, MS, under the DoD HPC Modernization Program, Programming Environment and Training (PET). The application can be idealized as follows.

A decision maker (the end user of the system) wants to evaluate changes in vegetation in some geographical region over a long time period caused by some short term disturbances such as a fire or human activities. One of the critical parameters of the vegetation model is soil condition at the time of the disturbance. This in turn is dominated by rainfall that possibly occurs at that time. Consequently as shown in Figure 8, the implementation of this project requires:

- 1) Data retrieval from remote sources including DEM (data elevation models) data, land use maps, soil textures, dominating flora species, and their growing characteristics, to name a few. The data are available from many different sources, for example from public services such as USGS web servers, or from proprietary databases. The data come in different formats, and with different spatial resolutions.

- 2) Data preprocessing to prune and convert the raw data to a format expected by the simulation software. This preprocessing is performed interactively using WMS (Watershed Modeling System) package.

3) Execution of two simulation programs: EDYS for vegetation simulation including the disturbances and CASC2D for watershed simulations during rainfalls. The latter results in generating maps of the soil condition after the rainfall. The initial conditions for CASC2D are set by EDYS just before the rainfall event, and the output of CASC2D after the event is used to update parameters of EDYS and the data transfer between the two codes had to be performed several times during one simulation. EDYS is not CPU demanding, and it is implemented only for Windows95/98/NT systems. On the other hand, CASC2D is very computationally intensive and typically is run on powerful backend supercomputer systems.

4) Visualization of the results of the simulation. Again, WMS is used for this purpose.

One requirement of this project was to demonstrate the feasibility of implementing a system that would allow launching and controlling the complete simulation from a networked laptop. We successfully implemented it using WebFlow middle-tier servers with WMS and EDYS encapsulated as WebFlow modules running locally on the laptop and CASC2D executed by WebFlow on remote hosts. Further the applications involved showed a typical mix of supercomputer and computationally less demanding personal computer codes. LMS was originally built using specialized Java Servers but these are now being replaced by commercial CORBA object brokers but in either case the architecture of Figure 8 is consistent with the general structure of Figures 6 and 3.

For this project we developed a custom front-end shown in Figure 9, that allows the user to interactively select the region of interest by drawing a rectangle on a map. Then one could select the data type to be retrieved, launch WMS to preprocess the data and make visualizations, and finally launch the simulation with CASC2D running on a host of choice.

Our use of XML standards at the two interfaces in Figure 3, allows us to change front end and middle-tier independently. This allowed us the middle tier upgrade described above which will bring security and “seamless access” capability to our PSE’s.

Seamless access is an important service, which is aimed at allowing applications to be run on an arbitrary appropriate backend. Most users wish their job to just run and usually do not mind what machine is used. Such seamless capability is rather natural in the architecture of Figure 3. Essentially the front end defines the “abstract” job passing its XML specification to a middle tier server. This acts as a broker and instantiates the abstract job on one of the available backend computers. The middle-tier backend XML interface is used to specify both the available machines sent to the servers and the “instantiated job” sent from the servers. We build the back end mapping of jobs to machine using the important Globus technology (<http://www.glo-bus.org>). Globus is a distributed or meta-computing toolkit providing important services such as resource look-up, security, message passing etc.

One can view WebFlow (Figures 7 and 8) as a “kit” of services, which can be used in different ways in different applications. In Figure 10, we show another capability of WebFlow, which supports the ability to compose complex problems by linking different applications together with data flowing between different modules chosen from a palette. This WebFlow service is supported by a Java front end and a middle tier service matching abstract modules to back end computers and supporting the piping of data between the modules.

### 2.5: Libraries or Distributed Components?

Since the beginning of (computing) time, large programs have been built from components that are assembled together into a complete job. One can do this in many ways and perhaps the best known is the use of libraries linked together. Modern “object-oriented” approaches can be used in this fashion with method calls. Design frameworks can be used to establish a systematic methodology with say specific interfaces (“calling sequences”) to be followed by for instance different implementations of friction laws. The discussion in Section 2.4 above describes a rather different component model where possibly distributed modules are linked together in a dynamic fashion. This is more flexible than the library approach and further as explained above, this supports distributed components. The component approach is not surprisingly less efficient than the library approach as component linkage is typically obtained through explicit exchange of messages rather than efficient compiled generated parameter passing as for library methods. The inefficiency is particularly serious for small components as the component mechanisms come with large start up (latency) overheads. Thus we adopt a hybrid approach with small objects using the traditional library mechanism within a set of agreed interfaces and design frameworks to promote easier interchange of modules. On the other hand, large objects (roughly “complete programs”) use a component approach within a distributed object framework, which is defined in XML.

## 3: CURRENT GEM COMPUTATIONAL COMPONENTS

The first step in building the GEMCI has been to inventory the computer codes currently available in the community. A summary of the results of this inventory is maintained at <http://milhouse.jpl.nasa.gov/gem/gemcodes.html> and the status as of the time of writing is given below.



The categories covered by the existing codes include codes for understanding dislocations: elastic models - both half-space and layered, viscoelastic models, finite element and boundary element codes, and inversion codes. There are a set of codes for understanding how faults interact: cellular automata, finite element and boundary element codes, fault friction models. Some codes are designed to understand the dynamics of the earthquake rupture itself. Finally there are codes for displaying the results of various simulations. We follow with short descriptions of the codes grouped by type.

### *Elastic*

**3D-DEF** -- Performs elastic dislocation boundary-element calculations  
**coulomb** -- Computes 3D elastic dislocation & 2D boundary element stress and strain  
**DYNELF** -- Models 3D elastodynamic finite difference with frictional faults  
**faultpatch** -- Generates earthquake sequences, given fault geometries and loading rates using  
**FLTSLP** -- Inverts groups of focal mechanism solutions or slickenline data for orientation and relative magnitudes of principal strain rates and for relative micropolar vorticity.  
**GNStress** -- Model stresses induced by faulting, for studying fault interaction.  
**layer** -- Calculates surface displacements and strains for vertical strike-slip point source in horizontal layer above half-space.  
**RNGCHN** -- Calculate surface displacements and strains in elastic half space.  
**scoot** -- 2D elastodynamic finite difference with frictional fault  
**simplex** -- 3D inversion of geodetic data for displacement on faults.

### *Viscoelastic*

**DYNELF** -- Models 3D elastodynamic finite difference with frictional faults  
**FLTGRV** and **FLTGRH** -- Compute 3 vector components of surface displacement from slip on a dipping thrust fault contained within an elastic layer overlying a viscoelastic-gravitational half space.  
**STRGRH** and **STRGRV** -- Computes 3 vector components of surface displacement from slip on a dipping strike slip fault contained within an elastic layer overlying a viscoelastic-gravitational half space.  
**Virtual\_California** -- Realistic cellular automata (CA) viscoelastic earthquake simulator.  
**VISCOID** -- Computes viscoelastic spherical deformation due to faulting or dike emplacement.

### *Cellular Automata*

**faultpatch** -- Generates earthquake sequences, given fault geometry's and loading rates using Cellular automata methods.  
**Virtual\_California** -- Realistic cellular automata (CA) viscoelastic earthquake simulator.

### *Dynamic*

**DYNELF** -- Models 3D elastodynamic finite difference with frictional faults  
**scoot** -- 2D elastodynamic finite difference with frictional fault

### *Inverse*

**FLTSLP** -- Inverts groups of focal mechanism solutions or slickenline data for orientation and relative magnitudes of principal strain rates and for relative micropolar vorticity.  
**simplex** -- 3D inversion of geodetic data for displacement on faults.  
**qoca** -- optimal combining of various geodetic measurements resulting in a consistent deformation field.

### *Visualization*

**VelMap** -- Aid for visualizing displacement and velocity of crust.

## 4: LARGE-SCALE SIMULATIONS IN GEMCI

## 4.1: Overview of Earthquake Fault System Simulations

With reasonable approximation, one may model the long-term evolution of stresses and strains on interacting fault segments with a Green's function approach. As in other fields this method leads to a boundary (the faults) value formulation, which looks numerically, like the long-range force problem. The faults are paneled with segments (with area of some 100 m<sup>2</sup> in definitive computations) which interact as though they were dipoles. The original calculations of this model used the basic  $O(N^2)$  algorithm but a new set of codes will be using the "fast multipole" method adapted from gravitational-particle astrophysics. There are interesting differences between the earthquake and gravitational applications. In gravity we get wide ranges in density and dynamical effects from the natural clustering of the gravitating particles. Earthquake "particles" are essentially fixed on complex fault geometries and their interactions fall off faster than those in the astrophysical problem.

Several variants of this model have been explored including approximations, which only keep interactions between nearby fault segments. These "cellular automata" or "slider-block" models look very like statistical physics with an earthquake corresponding to clusters of particles slipping together when the correlation length gets long near a critical point. The full Green's function approach should parallelize straightforwardly in either  $O(N^2)$  or multipole formulation. However cellular automata models will be harder to parallelize, as we know from experience with the corresponding statistical physics case where clustering models have been extensively studied.

An interesting aspect of these simulations is that they give the "numerical laboratory" for the study of space-time patterns in seismicity information. This type of analysis was used successfully in the climate field to aid the prediction of "El Nino" phenomena. These pattern analyses may or may not need large computational resources although they can involve determination of eigensolutions of large matrices which is potentially time consuming.

## 4.2: Green's Function Formulation and Approximations

4.2.1: *Basic Equations.* Let us consider in more detail the problem addressed by the *Virtual\_California* code described in Section 3 and introduced in *Rundle* 1988a. If one is given a network of faults embedded in an Earth with a given rheology, subject to loading by distant stresses, and neglecting elastic waves (see discussion below), the evolution of the state of slip  $\mathbf{s}(\mathbf{x}, t)$  on a fault at  $(\mathbf{x}, t)$  is determined from the equilibrium of stresses according to Newton's Laws:

$$\partial \mathbf{s}(\mathbf{x}, t) / \partial t = \Phi \{ \sum_i (\boldsymbol{\sigma}_i) \} \quad (1)$$

where  $\Phi \{ \}$  is a nonlinear functional, and  $\sum_i (\boldsymbol{\sigma}_i)$  represents the sum of all stresses acting within the system. These stresses include

- 1) The interaction stress  $\boldsymbol{\sigma}_{\text{int}}[\mathbf{x}, t; \mathbf{s}(\mathbf{x}', t'); \mathbf{p}]$  provided by transmission of stress through the Earth's crust arising from background tractions  $\mathbf{p}$ , as well as stresses due to slip on other faults at other sites  $\mathbf{x}'$  at times  $t'$ ;
- 2) The cohesive fault frictional stress  $\boldsymbol{\sigma}_f[\mathbf{x}, t; \mathbf{s}(\mathbf{x}, t)]$  at the site  $(\mathbf{x}, t)$  associated with the state of slip  $\mathbf{s}(\mathbf{x}, t)$ ; and
- 3) Other stresses such as those due to dynamic stress transmission and inertia.

The transmission of stress through the Earth's crust involves both dynamic effects arising from the transient propagation of seismic waves, and from static effects that persist after wave motion has ceased. Rheologic models typically used for the Earth's crust between faults are all linear (e.g., *Rundle and Turcotte*, 1993) and include

- 1) A purely elastic material on both long and short time scales;
- 2) A material whose instantaneous response is elastic but whose long term deformation involves bulk flow (viscoelastic); and
- 3) A material that is again elastic over short times, but whose long term response involves stress changes due to the flow of pore fluids through the rock matrix (poroelastic).

In Figure 11, we show the basic conceptual "wiring diagram" for the model, which indicates the interplay between loading stresses, rupture, interactions with other faults, and relaxation processes following a major earthquake.

4.2.2: *The Green's Function Approximation.* Considering GEM models that assume a linear interaction rheology between the faults implies that the interaction stress can be expressed as a spatial and temporal convolution of a stress Green's function  $\mathbf{T}_{ij}^{kl}(\mathbf{x} - \mathbf{x}', t - t')$  with the slip deficit variable  $\phi(\mathbf{x}, t) = \mathbf{s}(\mathbf{x}, t) - \mathbf{V}t$ , where  $\mathbf{V}$  is the long term rate of offset on the fault. Once the slip deficit is known, the displacement Green's function  $\mathbf{G}_{ij}^k(\mathbf{x} - \mathbf{x}', t - t')$  can be used to compute, again by convolution, the deformation anywhere in the surrounding medium exterior to the fault surfaces (e.g. *Rundle* 1988a). In the first implementation of GEM models, we expect to specialize in the case of quasistatic interactions, even during the slip events. Observational

evidence supports the hypothesis that simulations carried out without including inertia and waves will have substantial physical meaning. *Kanamori and Anderson* (1975) estimated that the seismic efficiency  $\eta$ , which measures the fraction of energy in the earthquake lost to seismic radiation, is less than 5%-10%. This implies that inertial effects in the dynamical evolution of slip in studying large populations of earthquakes will be of lesser importance for initial calculations. Elastic waves will be included in later simulations when errors arising from other effects are reduced to the 5%-10% level. At present, inclusion of these effects is severely limited by available computational capability, so we anticipate that it may be practical to include only the longest wavelengths or largest spatial scales.

**4.2.3: Inelastic Rheologies.** In quasistatic interactions, the time dependence of the Green's function typically enters only implicitly through time dependence of the elastic moduli (e.g., *Lee*, 1955). Because of linearity, the fundamental problem is reduced to that of calculating the stress and deformation Green's function for the rheology of interest. For materials that are homogeneous within horizontal layers, numerical methods to compute these Green's functions are well known (e.g., *Okada*, 1985, 1992; *Rundle*, 1982a,b, 1988; *Rice and Cleary*, 1976; *Cleary*, 1977; *Burridge and Varga*, 1979; *Maruyama*, 1994). Problems in heterogeneous media, especially media with a distribution of cracks too small and too numerous to model individually, are often solved by using effective medium approaches, self-consistency assumptions (*Hill*, 1965; *Berryman and Milton*, 1985; *Ivins*, 1995a,b), or damage models *Lyakovskiy et al.* (1997). Suffice to say that a considerable amount of effort has gone into constructing quasistatic Green's functions for these types of media, and while the computational problems present certain challenges, the methods are straightforward as long as the problems are linear. In the initial work, we expect to focus on elastic (with possible incorporation of damage parameters) and layered viscoelastic models only.

### 4.3: Friction Models:

Friction models to determine the slip condition must augment the overall differential equations. At the present time, six basic classes of friction laws have been incorporated into computational models. They fall into three families as follows.

1) Two basic classes of friction models arise from **laboratory experiments**:

**Slip Weakening.** This friction law (*Rabinowicz*, 1965; *Bowdon and Tabor*, 1950; *Beeler et al.*, 1996; *Li*, 1987; *Rice*, 1993; *Stuart and Tullis*, 1995) assumes that the frictional stress at a site on the fault  $\sigma_f = \sigma_f[\mathbf{s}(\mathbf{x}, t)]$  is a functional of the state of slip. In general,  $\sigma_f[\mathbf{s}(\mathbf{x}, t)]$  is peaked at regular intervals. The current state of the system is found from enforcing the equality  $\sigma_f[\mathbf{s}(\mathbf{x}, t)] = \sigma_{int}[\mathbf{x}, t; \mathbf{s}(\mathbf{x}', t'); \mathbf{p}]$  prior to, and just after, a sliding event.

**Rate and State.** These friction laws are based on laboratory sliding experiments in which two frictional surfaces are slid over each other at varying velocities, usually without experiencing arrest (*Dieterich*, 1972; 1978; 1981; *Ruina*, 1983; *Rice and Ruina*, 1983; *Ben Zion and Rice*, 1993; 1995; 1997; *Rice*, 1993; *Rice and Ben Zion*, 1996). In these experiments, the laboratory apparatus is arranged so as to be much "stiffer" than the experimental "fault" surfaces. The rate dependence of these friction laws refers to a dependence on logarithm of sliding velocity, and the state dependence to one or more state variables  $\theta_i(t)$ , each of which follows an independent relaxation equation.

2) Two classes of models have been developed and used that are based on laboratory observations, but are **computationally simpler**.

**Coulomb-Automata.** These are widely used because they are so simple (e.g., *Rundle and Jackson*, 1977; *Nakanishi*, 1991; *Brown et al.*, 1991; *Rundle and Brown*, 1991; *Rundle and Klein*, 1992; *Ben Zion and Rice*, 1993, 1995, 1997). A static failure threshold, or equivalently a coefficient of static friction  $\mu^S$  is prescribed, along with a residual strength, or equivalently a dynamic coefficient of friction  $\mu^D$ . When the stress at a site increases, either gradually or suddenly, to equal or exceed the static value, a sudden jump in slip (change of state) occurs, that takes the local stress down to the residual value. These models naturally lend themselves to a Cellular Automaton (CA) method of implementation.

**Velocity Weakening.** This model (*Burridge and Knopoff*, 1967; *Carlson and Langer*, 1989) is based on the observation that frictional strength diminishes as sliding proceeds. A constant static strength  $\sigma_f = \sigma^F$  is used as above, after which the assumption is made that during sliding, frictional resistance must be inversely proportional to sliding velocity.

3) Two classes of models are based on the use of **statistical mechanics** involving the physical variables that characterize stress accumulation and failure. Their basic goal is to construct a series of nonlinear stochastic equations whose solutions can be approached by numerical means:

**Traveling Density Wave.** These models (*Rundle et al.*, 1996) are based on the slip weakening model. The principle of evolution towards maximum stability is used to obtain a kinetic equation in which the rate of change of slip depends on the functional derivative of a Lyapunov functional potential. This model can be expected only to apply in the mean field regime of long range interactions, which is the regime of interest for elasticity in the Earth. Other models in this class include those of *Fisher et al.* (1997) and *Dahmen et al.* (1997).

**Hierarchical Statistical Models.** Examples include the models by *Allègre et al.* (1982, 1996); *Smalley et al.* (1985); *Blanter et al.* (1996); *Allègre and Le Mouel* (1994); *Heimpel* (1996); *Newman et al.* (1996); and *Gross* (1996). These are probabilistic

models in which hierarchies of blocks or asperity sites are assigned probabilities of failure. As the level of external stress rises, probabilities of failure increase, and as a site fails, it influences the probability of failure of nearby sites.

#### 4.4: Multipole Methods and Fast Numerical Simulation

*4.4.1: Introduction.* The GEM group has started to investigate the use of advanced parallel solvers for the codes described in Section 3. The Green's function approach can be formulated numerically as a long-range all-pairs interaction problem and this can be straightforwardly parallelized using well-known algorithms. However one cannot reach the required level of resolution without switching from an order  $O(N^2)$  to one of the  $O(N)$  or  $O(N \log N)$  approaches. As in other fields, this can be achieved by dropping or approximating the long-range components and implementing a neighbor-list based algorithm. However it is more attractive to formulate the problem as interacting dipoles and adapt existing fast-multipole technology developed for particle dynamics problems. We have already produced a prototype general purpose "fast multipole template code" by adapting the very successful work of *Salmon and Warren* (1997). These codes have already simulated over 300 million gravitating bodies on a large distributed memory system (a 4500-processor subset of the ASCI "Red" machine), so we expect these parallel algorithms to scale efficiently up to the problem sizes needed by GEM.

*4.4.2: Multipolar Representation of Fault Systems.* A primary key to a successful implementation of GEM models of faults systems will be to utilize computationally efficient algorithms for updating the interactions between fault segments. Converting the Green's function integrals to sums, without truncation or approximation, would require  $O(N^2)$  operations between earthquakes, and possibly more for segments of faults experiencing an earthquake. For quasistatic interactions, the Green's functions  $\mathbf{T}_{ij}^{kl}$  and  $\mathbf{G}_{ij}^k$  for linear elasticity have a simple time dependence. Moreover, the Green's functions for linear viscoelasticity and for linear poroelasticity can be obtained from the elastic Green's functions using the correspondence principle (e.g., *Lee*, 1955; *Rundle* 1982a,b). These simplifications strongly suggest that multipole expansions (*Goil*, 1994; *Goil and Ranka*, 1995) will be computationally efficient algorithms.

The stress and displacement Green's functions  $\mathbf{T}_{ij}^{kl}$  and  $\mathbf{G}_{ij}^k$  represent the tensor stress and vector displacement at  $\mathbf{x}$  due to a point double couple located at  $\mathbf{x}'$  (*Steketee*, 1958). The orientation at  $\mathbf{x}'$  of the equivalent fault surface normal vector, and of the vector displacement on that fault surface, are described by the indices  $i$  and  $j$ . Displacement and stress indices at the field point  $\mathbf{x}$  are described by indices  $k$  and  $l$ . Integration of  $\mathbf{T}_{ij}^{kl}$  and  $\mathbf{G}_{ij}^k$  over the fault surface then corresponds to a distribution of double couples. For that reason, representation of the stress over segments of fault in terms of a multipole expansion is the natural basis to use for the GEM computational problem. In fact, the use of multipolar expansions to represent source fields in earthquake and explosion seismology was introduced by *Archambeau* (1968) and *Archambeau and Minster* (1978), and later revisited from a different perspective by *Backus and Mulcahy* (1976). *Minster* (1985) gives a review of these early representations.

*4.4.3: Application of Fast Multipole Methods to GEM.* In the gravitational N-body problem, each body interacts with every other one in the system according to the familiar law of gravitational attraction. Simply computing all pairs of interactions requires  $N(N-1)/2$  separate evaluations of the interaction law. This formulation of the problem has some important advantages: it is easy to code, it is easy to vectorize and parallelize, it is readily expressible in High Performance Fortran, and it is even amenable to special-purpose hardware [e.g. GRAPE]. Nevertheless, even today's fastest special-purpose systems, running in a dedicated mode for extended times at rates of nearly 1 TERAFLOP, cannot simulate systems larger than about 100,000 bodies.

Tremendous computational savings may be realized by combining bodies into "cells" and approximating their external field with a truncated multipole expansion. When this idea is applied systematically, the number of interactions may be reduced to  $O(N \log N)$  (*Appel*, 1985; *Barnes and Hut*, 1986) or  $O(N)$  (*Greengard and Rokhlin*, 1987; *Anderson*, 1992). The cells are generally arranged in a tree, with the root of the tree representing the entire system, and descendants representing successively smaller regions of space. *Salmon and Warren* (1997) have demonstrated that such codes can run in parallel on thousands of processors and have simulated highly irregular cosmological systems of over 300 million bodies using ASCI facilities.

The basic idea of fast multipole methods is shown in the three versions of Figure 12. In 12(a), we show the simple  $O(N^2)$  approach with a single particle on the left interacting with all the individual particles in a box on the right. The naïve computational simplification shown in Figure 12(b) is to replace (in the gravity case) the particles by their center of mass and use a single force calculation. However this approximation may not be accurate enough and so in Figure 12(c) we show the original box broken into 8 sub-boxes with the total summed over the individual sub-box contributions. This idea can be automated and implemented in a clever way so that the boxes used get smaller as you move closer to a given particle or if there is significant variation in particle density. Further there are subtle but very efficient parallel versions of the method to build the set of hierarchical sub-boxes (which form a tree so that these are often termed tree codes) which in the case of gravity is rebuilt each time step. In Figure 13 below, we illustrate the adaptive division generated for 10,000 particles clustered in a two-dimensional disk.

There is a direct analogy between the bodies in an astrophysical  $N$ -body system and the fault segments in a GEM. In both cases, there exists a pair-wise interaction that seems to require  $O(N^2)$  interactions. But if we represent the distribution of

sources in a region by a multipole expansion, the external field generated by a large number of bodies can be computed to any desired degree of accuracy in constant time. Thus, the GEM problem can also be reduced to  $O(N \log N)$  or  $O(N)$  total interactions, so that large calculations are tractable. On the other hand, although multipole methods can deliver large performance gains, they also require a considerable infrastructure. This is especially true of efficient parallel implementations. We intend to develop the multipole version of GEM using a library that has been abstracted from Salmon and Warren's successful astrophysical  $N$ -body codes. This new library is:

**Modular** - The “physics” is cleanly separated from the “computer science”, so that in principle, alternative physics modules such as the evaluation of the GEM Green's functions, can simply be “plugged in”. The first non-gravitational demonstration was a vortex dynamics code written by *Winckelmans et al.* (1995). The interface to the physics modules is extremely flexible. A general decision-making function tells the treecode whether or not a multipole, or any other approximation, is adequate for a given field evaluation. Short-range interactions, which vanish outside a given radius, can be handled as well.

**Tunable** - Careful attention to analytical error bounds has led to significant speed-ups of the astrophysical codes, while retaining the same level of accuracy. Analytic error bounds may be characterized as quantifying the fact that the multipole formalism is more accurate when the interaction is weak: when the analytic form of the fundamental interaction is well-approximated by its lower derivatives; when the sources are distributed over a small region; when the field is evaluated near the center of a “local expansion”; when more terms in the multipole expansion are used, and when the truncated multipole moments are small. These issues are primarily the concern of the “physics” modules, but the library provides a sufficiently powerful interface to make these parameters adjustable. The formulation is general enough that the same library can be used to support evaluation of  $O(N)$ ,  $O(N \log N)$  and  $O(N^2)$  approximation strategies, simply by changing the decision criteria and interaction functions.

**Adaptive** - The tree automatically adapts to local variations in the density of sources. This can be important for GEM as it is expected that large earthquakes are the result of phenomena occurring over a wide range of length and time scales.

**Scalable** - The library has been successfully used on thousands of processors, and has sustained 170 Gflops aggregate performance on a distributed system of 4096 200Mhz PentiumPro processors.

**Out of core** - The library can construct trees, and facilitates use of data sets that do not fit in primary storage. This can allow one to invest hardware resources into processing rather than memory, resulting in more computations at constant resources.

**Dynamically load balanced** - The tree data structure can be dynamically load-balanced extremely rapidly by sorting bodies and cells according to an easily computed key.

**Portable** - The library uses a minimal set of MPI primitives and is written entirely in ANSI C. It has been ported to a wide variety of distributed memory systems - both 32-bit and 64-bit. Shared memory systems are, of course, also supported simply by use of an MPI library tuned to the shared memory environment.

**Versatile** - Early versions of the library have already been applied outside the astrophysics and molecular dynamics area. In particular the Caltech and Los Alamos groups have successfully used it for the vortex method in Computational Fluid Dynamics.

In the full GEM implementation, we have a situation similar to the conventional  $O(N^2)$   $N$ -body problem but there are many important differences. As the most obvious difference, the GEM case corresponds to double-couple interactions, which correspond to a different force law between “particles” from the gravitational case. Further the critical dynamics -- namely earthquakes -- are found by examining the stresses at each time step to see if the friction law implies that a slip event will occur. As discussed above, many different versions of the friction law have been proposed, and the computational system needs be flexible so we can compare results from different laws. Noting that earthquakes correspond to large-scale space-time correlations including up to perhaps a million 10-to-100 meter segments slipping together shows analogies with statistical physics. As in critical phenomena, clustering occurs at all length scales and we need to examine this effect computationally. However, we find differences with the classical molecular dynamics  $N$ -body problems not only in the dynamical criteria of importance but also in the dependence of the Green's function (i.e. “force” potential) on the independent variables. Another area of importance, which is still not well understood in current applications, will include use of spatially dependent time steps (with smaller values needed in active earthquake regions). An important difference between true particles and GEM is that in the latter case, fault positions are essentially fixed in space. Thus the  $N$ -body gravitational problem involves particles whose properties are time-invariant but whose positions change with time, while GEM involves “particles” whose positions are fixed in time, but whose properties change with the surrounding environment. Of course a major challenge in both cases is the issue of time-dependent “clustering” of “particles.” It may be possible to exploit this in the case of GEM -- for example by incrementally improving parallel load balancing.

#### 4.5: Computational Complexity

Current evidence suggests that forecasting earthquakes of magnitude  $\sim 6$  and greater will depend upon understanding the space-time patterns displayed by smaller events; i.e., the magnitude 3's, 4's and 5's. With at least 40,000 km<sup>2</sup> of fault area in southern California, as many as  $10^8$  grid sites (10-meter segment size) will be needed to accommodate events down to magnitude 3. Extrapolations based upon existing calculations indicate that using time steps of  $\sim 100$  seconds implies  $\sim 10^8$  time steps will be required to simulate several earthquake cycles. This leads to the need for teraflop class computers in this as in many physical simulations.

Here we make the conservative assumption that the GEM dipole-dipole Green's function evaluations are ten times as computationally expensive as the Newtonian Green's functions evaluated in Salmon and Warren's code. At this stage, we cannot guess how far the teraflop class of computer will take us and the systems needed to support research, crisis managers or insurance companies assessing possible earthquake risk, may require much higher performance.

#### 4.6: Integration of Simulation Modules into GEMCI

As we described in Section 2.5, we will adopt a hierarchical model for the simulation components used in GEMCI. The full Greens function programs would be macroscopic distributed objects which can be manipulated by systems like WebFlow and linked with other coarse grain components such as sensor data, other programs such as finite-element solvers and visualization systems. These components are made up of smaller objects used in library fashion. Here we have the friction modules and different solver engines such as the conventional  $O(N^2)$  and fast multipole methods with each having sequential and parallel versions. The latter would often exist in multiple forms including an openMP version for shared memory and MPI version for distributed memory.

### 5: TYPICAL COMPUTATIONAL PROBLEMS

Here we give three sample scenarios, which would correspond to distinct problem solving environments built using GEMCI.

#### 5.1: Seismicity Models and Data Assimilation

Our first example of how the GEMCI might be used is drawn from an attempt to create a computer model of the seismicity of California or other seismically active region. Such a model, to the extent that it is realistic, could be quite useful in guiding intuition about the earthquake process, and suggesting new measurements or lines of inquiry.

Making such a model realistic requires many different types of data and there are for instance already substantial archives accumulated by the Southern California Earthquake Center (SCEC), as well as the Seismological Laboratory of the California Institute of Technology, and the Pasadena field office of the United States Geological Survey. The relevant data includes:

- 1) Broadband seismic data from the TERRASCOPE array.
- 2) Continuous (SCIGN) and "campaign style" geodetic data.
- 3) Paleoseismic data collected on the major faults of southern California.
- 4) Near field strong motion accelerograms of recent earthquakes.
- 5) Field structural geology of major active faults.
- 6) Other data including GPS data, leveling data, pore fluid pressure, *in situ* stress, heat flow, downhole seismic data, multi-channel seismic data, laboratory measurements of mechanical properties of various rocks, 3-D geologic structure, gravity data, magneto-telluric data, hydrology data, and ocean tide data (to constrain coastal uplift).

These will be used, for example, to update the fault geometry models used by GEM, and to update fault slip histories used to validate earthquake models.

A new and extremely promising type of geodetic data is *Synthetic Aperture Radar Interferometry* (InSAR), which permits "stress analysis of the Earth." A number of SAR missions are currently acquiring data over southern California, including the C-band (5.8 cm) European ERS 1/2 satellites and the L-band Japanese JERS satellite. These missions have already produced revolutionary images of the complete deformation fields associated with earthquakes in the United States and Japan (e.g., *Masonnet et al.*, 1993). These techniques rely on radar interferograms that represent the deformation field at a pixel resolution of a few tens of meters over areas of tens of thousands of square kilometers, and over time intervals of weeks to years. We are now essentially able to see the complete surface deformation field due to an earthquake, with excellent prospects for similar views of deformation due to interseismic strain-accumulation processes.

There are productive scientists who have built entire careers on each individual type of data mentioned above. No individual scientist at this time has access to complete collections of all these data types, nor the detailed expertise that would be required to use all these data in a computational model. Consequently, the traditional models are based on some subset of the data, with

which we can try to predict other sets. However a more powerful approach can be based around an environment in which people can collaborate to construct and test models that span multiple data sets and which can then be embellished by other investigators.

To maximize the utility of any given data set, the people who collect and care about the data must be the ones who archive it and make it available. Attempts to construct monolithic databases of diverse earth science data using dedicated people and facilities have met with limited success. It appears to work better if the people or community who "own" the data are the ones who archive it. Thus, we need an environment, which will allow easy collaboration, and access to diverse distributed data sets. To return to our example of modeling the seismicity of California, an investigator might start with a map of faults in the region, a constitutive law for determining the slip on those faults derived from laboratory measurements, and some sort of loading criteria perhaps derived from GPS measurements. From these inputs, a model is constructed which reproduces the statistics of rupture as shown by paleoseismicity data and earthquake catalogs. Subsequently, another investigator takes that model and adds pore fluid pressure modeling in order to investigate the effects of fluids on the nucleation of earthquakes and the migration of seismicity. This process is repeated with other investigators collaborating and developing competing and/or complementary models. The models can be benchmarked against other data sets and against each other. This kind of an environment will enable us to make the most rapid possible progress at understanding earthquake processes.

### *5.2: Future Modeling Directions*

The basic method for constructing simulations is to compile and process data to obtain estimates of fault geometry and properties, including the properties of the bulk material surrounding the fault. Forward models can be run for hundreds of thousands of years using the above parameters as inputs in order to establish dynamic patterns of the system. Pattern analysis can then be performed on the calculated deformation and seismicity to establish temporal and spatial correlations in the simulated data. The simulations can be compared to real geophysical data, which sample only a short timespan of the entire system. In parallel, pattern analysis can be performed on actual geophysical data to search for temporal anomalies. By using a statistical mechanical approach we can address whether statistical patterns form in the system and if so, characterize them. We can also address whether earthquake fault systems can be parameterized without a detailed knowledge of rheological details and assess how the dynamics of space-time patterns relate to the dynamics of fault systems.

There are several questions in earthquake physics that can be addressed with crustal deformation data, a relatively new data set. Some of the questions identified at the Plate Boundary Observatory workshop in Snowbird, Utah, 1999 follow. What processes occur on faults that cause strain localization? What is the architecture of fault zones below the seismogenic zone? What is the relation between strain and fault processes? How do faults interact? What is the nucleation process of earthquakes, or how do they initiate and stop? And what are the conditions (temperature, pressure, stress, . . .) within active fault zones? These questions can be answered by combining high temporal resolution data (seismic, strain, and GPS) and spatially detailed data (InSAR) into models of the deformation process. By answering these questions through the construction of detailed models we can then address how space-time patterns relate to the dynamics of fault systems.

Combining the broad range of models from kinematic to quasi-static dynamic will result in a physically realistic model of earthquake systems based on observations and data. One major question of earthquake physics is whether broad scale stress correlations build up that produce large events or clusters of events. If such correlations exist, are they detectable either in seismicity patterns or in geodetic data? We describe below a means of developing models to test for this without looking at the details of rupture processes.

Geologic, geodetic, and seismic data can all be used to establish fault geometry and slip rates. Paleoseismologic data provide details on slip per event, slip rate and timing of past events on surface rupturing faults. Seismic imaging data as well as borehole data provide details of geometry at depth. Structural analysis of retro-deformed cross sections adds information on slip rate and fault evolution. The geodetic data provide information on fault locations, including blind faults, as well as slip rate. Both strain and rotation rates can be determined from the full velocity gradient tensor provided by GPS data. Strain rates indicate the style of faulting expected from various faults (strike-slip, thrust, oblique) and spatial variations in strain identify fault bounded blocks.

The fault geometries and rates must be coupled with parameters of the mechanical properties of the crust to generate realistic quasi-static models in which stress is transferred and diffused between faults and in the crust. Crustal rigidity structure can be determined from seismic velocity models [e.g. Hauksson, 1997]. These data can be combined with surface deformation observations into viscoelastic models to estimate rheological parameters of the crust as well as place added constraints on slip rates [e.g. Hager et al., 1999].

Fairly abrupt changes of velocity have been observed across the Ventura and Los Angeles basins [Donnellan et al., 1993; Argus et al., 1999]. These velocity profiles can be incorporated into viscoelastic models. The surface deformation data are particularly useful when combined with paleoseismologic data because viscoelastic models indicate that the velocity profile across faults should change with time as stress relaxation occurs in the lower crust following earthquakes. When available,

paleoseismologic data make it possible to determine (or at least estimate) the point in the earthquake cycle that the geodetic data have been collected, thus improving estimates of lower crustal viscosity.

Crustal viscosity or relaxation time determines how well the upper crust is coupled to the mantle, which drives the processes that occur at the surface. A weak lower crust tends to generate broader features and effectively results in far-field or side driving stresses, while a strong lower crust results in strong coupling between the crust and the mantle, localizing effects and amplifying the importance of driving stresses from below. These two styles of driving forces result in widely different observed patterns of deformation at the surface. Furthermore, a strong lower crust results in near-fault strain localizations even late in the earthquake cycle, while a weak lower crust rapidly dissipates stress.

The surface deformation profile is further complicated by fault zone properties. Recent geodetic results indicate that after-slip on the rupture plane can occur for up to two years following moderate to large earthquakes [Pollitz, *et al.*, 1998; Donnellan and Lyzenga, 1998]. Results from the Northridge earthquake are consistent with rate and state parameters determined in the lab, suggesting that faults are weaker than the surrounding material throughout the earthquake cycle. Future detailed measurements following large earthquakes will clarify behavior of both faults and the lower crust.

Until recently faults have been treated fairly independently from neighboring faults in models. Computer power is available now to model systems of faults. Rundle *et al.* [1997] model how stress is transferred to other faults following earthquakes. The models do not allow a viscous lower crust, but future calculations that allow for both elastic stress transfer and viscous dissipation will elucidate whether or how stress correlations build up in the crust. Running models of interacting fault systems that are constrained by observations will be the first step in applying pattern recognition and statistical mechanics to the earthquake problem.

### 5.3: Response to an Earthquake in Southern California

An illustrative scenario is shown in the first part of Figure 14 which links multiple datasets, modeling codes, research centers and scientists for real-time earthquake response. A prototype interface supporting this scenario is shown in Figure 15. Other data types and models (eg., seismic and field data) would be included in any operational system, but are omitted here for purposes of exposition. The goal is to form a rapid consensus among researchers concerning the characterization of the deformation field and the location, size, and direction of slip on a fault following an earthquake. This consensus can be used to guide decisions on both civil and scientific responses to the quake.

This initial analysis is followed at the bottom of Figure 14 by detailed simulations of the type described in Section 4. We describe the whole process in detail below.

Following an earthquake in Southern California, the location and magnitude are calculated based on seismic data within minutes by Caltech/USGS, and currently are broadcast to several users via email and pagers. The information on location and magnitude could then be automatically used to define an area wherein instruments might be expected to record a signal (the program *disp*). Data from these stations would be given priority in retrieval and analysis. In this example we will assume that the data in question is GPS data from the Southern California Integrated Geodetic Network (SCIGN) array. Retrieval in this case is done by telephone modem. As soon as the list of possibly affected stations has been generated, the database at the USGS is checked. If any of the stations on the list have not had data downloaded since the quake, computers at the USGS begin dialing the selected stations and retrieving the data.

Data from these stations would then be processed for rapid analysis to determine the measured displacements of the stations (program *GIPSY*). If the measured displacements are large enough, emergency and scientific personnel are notified via email and pager. These displacements are then automatically fed into an inversion routine (program *simplex*) which solves for the best fit single fault displacement. This single fault displacement is in turn fed back into a forward elastic half space model, which yields a preliminary map of displacements over the whole area (program *disloc*).

At this point this map is shared between various scientists and emergency personnel, using systems like Tango Interactive described in Section 6 that allow the collaboration and interaction of multiple people viewing and manipulating the same data set over the Internet. The emergency personnel can use the preliminary map in combination with a Geographical Information System data about utilities, lifelines, etc. to help assign resources to various areas. The scientists will use the preliminary map to help design a strategy for collecting additional measurements. They can also collaborate on refining the single fault model, possibly breaking the single preliminary fault into several segments, introducing more realistic material properties, or including more data, before rerunning the inversion.

This environment permits the rapid determination and dissemination of preliminary information about the earthquake and the collaborative refining of that information following an event. The rapid dissemination of information can greatly aid both the civil and scientific response to the quake. Resources can be more efficiently allocated to the areas where they are needed, and scientific measurements can be focused to provide information critical to refining our understanding of the earthquake system. Once an acceptable model of the earthquake has been determined, various models can be used to estimate the updated



earthquake hazard for adjacent areas. Since there are currently several competing models for this, consensus will undoubtedly involve multiple runs of multiple models and significant discussion among scientific colleagues.

Each of these models as well as the various pieces of the automated processes described above have been developed by different people under different assumptions, and is developed, run, and maintained on computers under the control of the developer. The technologies such as CORBA and Enterprise Javabeans described in Section 2, allow appropriate access and security mechanisms in this complex evolving distributed system.

The 1999 Izmit Turkey earthquake provided a recent example of how a system like this could have been useful. Following that earthquake, many geoscientists got together in a series of conference telephone calls to try to piece together what had happened, and what was an appropriate response. Some participants initially only knew what had been reported in the media. Others knew of specific pieces of data concerning the earthquake or of actions being taken by various groups and individuals. It is safe to say that no one had a complete picture. Much of the conference call was devoted to informing everyone about all the pieces of data and all the various initiatives that people were pursuing or might pursue. Similar calls and emails occurred after the 1992 Landers and 1994 Northridge earthquakes. Having a system such as has been described above wherein participants could share maps, descriptions, programs, data sets, and graphs and wherein they could interactively and collaboratively manipulate the data and programs, both synchronously and asynchronously, would immeasurably aid the rapid and accurate diagnosis of what has happened and what should be done next.

#### 5.4 Fundamental Computational Science Studies in Earthquake Forecasting

Above we described a relatively complicated real-time analysis scenario. Another important problem-solving environment supports that process typically used in computational science. Here the main steps are development of simulation codes, refinement of analysis techniques, initiating runs with multiple parameter values, and comparison of results with multiple data sets. This type of analysis has similarities with the previous two scenarios but now the collaboration for instance would not emphasize real-time issues so much but extend over many months or years. Technically this implies the need for powerful asynchronous tools linking over greater spans of time and space than the tightly coupled scenario of Section 5.3. The tools needed here include the interactive rapid prototyping environment supporting pattern analysis and visualization. This aspect entails somewhat different trade-offs than the core simulations, in that interactivity is perhaps more critical than performance. As discussed in Section 6, this could suggest that this part of the problem-solving environment would be implemented client-side using interpretative languages. We would also need to experiment with many different ways of linking programs together and so we would have to support both program development and execution. The real-time constraints of Section 5.3 would on the other hand emphasize execution of pre determined program modules.

This problem-solving environment would naturally link with that in Section 5.1 as GPS, InSAR and broadband seismic (TERRASCOPE) data, together with archived and newly developed paleoseismic data can be used in conjunction with the simulation capabilities to establish the relevant model parameters. These parameters include the current geometry of faults, slip rates on any given segment, recurrence intervals and historic variations in slip during earthquakes (leading to estimates of frictional parameters), deformation data (leading to estimates of elastic plate thickness and sub-crustal stress), relaxation times, poroelastic stress, relaxation in the crust following earthquakes (leading to estimates of drained and undrained elastic moduli), and variations in seismicity (leading to estimates of the variable properties of friction and fault geometry at depth). One must support the fitting of models to data with a suite of techniques (e.g., *Menke*, 1989), including least squares, evolutionary programming, and simulated annealing (*Michalewicz*, 1996; *Holland*, 1975; *Rawlins*, 1991), among others. In addition, one can expect to develop new methods so as to adapt models to assimilate new data as that becomes available, a concept that has served meteorological and climate studies extremely well. Self-adaptation techniques can be based on the same kinds of back-propagation methods that have been useful in analysis of neural network models (*Hertz et al.*, 1991).

#### 5.5: Seismic Waves and Earthquake Engineering

The most mature computations in the earthquake field are perhaps those used to calculate seismic waves and the corresponding response of buildings to them. In fact Clayton in the Caltech Geophysics department performed one of the very first Caltech Cosmic Cube computations to simulate the motion of earthquake waves in the Los Angeles Basin. This wave motion can in principle be generated from the earthquake "events" calculated in the Virtual California simulations described in Sections 4 and 5.4. These are large-scale finite element problems with complex grids and a recent NSF Grand Challenge project at CMU was very successful in this area (<http://www.cs.cmu.edu/~quake/>). The wave motion can be used as a forcing function for structural dynamics computations of buildings, roads and other civil infrastructure (<http://www.curee.org/>).

## 6: SUMMARY OF GEMCI

We introduced in Section 2.1 a computational environment GEMCI and described its overall framework in terms of distributed objects supported by commodity technologies like CORBA COM Java and XML. In the following sections we have discussed some especially important modules and how they are integrated into overall systems aimed at particular problem areas. Although we have not discussed them in detail, we will use some basic services, in particular security, collaboration and visualization. Security does not need much customization for computational science from the capabilities available from the commodity technologies for electronic commerce. Both Globus and Legion have developed good security approaches for high performance computing and GEMCI will be adopt such solutions. There is much experience in visualization in many projects although it is not clearly understood how to formulate this generally in systems like GEMCI. We can however expect that the substantial efforts by both Department of Energy and NSF will lead to visualization approaches appropriate for GEMCI. Collaboration corresponds to sharing of objects in GEMCI and this could correspond to shared rendering of data or the results of simulations. This technology is less mature than that for security and visualization although the Web has enabled powerful “asynchronous” collaboration where one posts information on web pages and your collaborators access your information at a later time. Synchronous collaboration can be important – especially in scenarios like those of Section 5.2. Here approaches like Tango Interactive from NPAC at Syracuse (<http://www.n-pac.syr.edu/tango>) are early prototypes that are expected to mature into powerful synchronous sharing for web pages and related data sets. An important area that we have not yet discussed in detail is the handling of data and its linkage to simulation. Here one of the leaders is the NSF NPACI partnership (<http://www.npaci.edu/DICE/>) whose data-intensive activity is led by Reagan Moore. This approach makes extensive use of the same object standards advocated in GEMCI and we expect to be able to use the best available hardware and software to link data and simulations.

In summary, we believe that GEMCI has been designed to take advantages of the best available high performance and commodity information and simulation resources. We cannot predict the future of this rapidly changing field but GEMCI has made choices that should be able to track and exploit advances.

The earthquake community is uniquely positioned to take advantage of technology advances on many fronts. The data volume and quality is rapidly increasing as geodetic and seismic networks become denser and broader band. Recent advances in information technology will enable modeling environments to be constructed in which data and parameters are easily incorporated into and passed between models. Computational power is such that complex three-dimensional fault system problems can now be approached, and algorithms developed in other fields (plasma physics, astrophysics) are well-suited to be applied to the earthquake problem.

## REFERENCES

- Allègre, C., J.L. Le Mouel and A. Provost, Scaling rules in rock fracture and possible implications for earthquake prediction, *Nature*, 297, 47-49, 1982.
- Allègre, C.J. and J.L. Le Mouel, Introduction of scaling techniques in brittle fracture of rocks, *Phys. Earth Planet Int.*, **87**, 85-93, 1994.
- Allègre, C.J., J.L. Le Mouel, H.D. Chau and C. Narteau, Scaling organization of fracture tectonics and earthquake mechanism, *Phys. Earth Planet Int.*, **92**, 215-233 1996.
- Anderson, C.R., *SIAM J. Sci. Stat. Comp.*, **13**, 923, 1992.
- Appel, A W. , An efficient Program for Many-Body Simulation, *SIAM J. Sci. Stat. Comp.*, **6**, 85, 1985.
- Appel, A.W., *SIAM Journal on Computing.*, **6**, 85, 1985.
- Archambeau, C.B., General theory of elastodynamic source fields, *Rev. Geophys. Space Phys.*, **16**, 241-288, 1968.
- Archambeau, C.B. and J.B. Minster, Dynamics in prestressed media with moving phase boundaries: A continuum theory of failure in solids, *Geophys. J. R. Astr.*, **52**, 65-96, 1978.
- Argus, D. F., M. B. Hefflin, A. Donnellan, F. Webb, D. Dong, K. Hurst, D. Jefferson, G. Lyzenga, M. M. Watkins, J. F. Zumberge, Shortening and thickening of metropolitan Los Angeles measured and inferred by using geodesy, *Geology*, **27**, 703-706, 1999.
- Backus, G and M. Mulcahy, Moment tensors and other phenomenological descriptions of seismic sources - I: Continuous displacements, *Geophys. J. Roy. Astr. Soc.*, **46**, 341-361, 1976.
- Barnes, J. and Hut, P., *Nature*, **324**, 446, 1986.
- Barnes, J.E. and P Hut, A Hierarchical ( $O(N \log N)$ ) Force-Calculation Algorithm, *Nature*, **324**, 446-449, 1986.
- Beeler, N.M., T.E.Tullis, M.L. Blanpied and J.D. Weeks, Frictional behavior of large displacement experimental faults, *J. Geophys. Res.*, **101**, 8697-8715, 1996.
- Beeler, N. and TE Tullis, Self healing slip pulses in dynamic rupture models due to velocity dependent strength, *Bull. Seism. Soc.*, 1996.
- Ben-Zion, Y. and J. R. Rice, earthquake failure sequences along a cellular fault zone in a three-dimensional elastic solid containing asperity and nonasperity regions, *J. Geophys. Res.*, **98**, 14109-14131, 1993.
- Ben-Zion, Y. and J. R. Rice, Slip patterns and earthquake populations along different classes of faults in elastic solids, *J. Geophys. Res.*, **100**, 12959-12983, 1995.
- Ben-Zion, Y. and J. R. Rice, Dynamic simulations of slip on a smooth fault in an elastic solid, *J. Geophys. Res.*, **102**, 17771-17784, 1997.

- Berryman, J.G. and G.W. Milton, Normalization constraint for variational bounds on fluid permeability, *J. Chem. Phys.*, **83**, 754-760, 1985.
- Blanter, EM, MG Schnirman, JL Le Mouel and C.J. Allègre, Scaling laws in block dynamics and dynamic self-organized criticality, *Phys. Earth. Planet. Int.*, **99**, 295-307, 1997.
- Bosak, J. and T. Bray, XML and the Second-Generation Web, *Scientific American*, **280(5)**, 89-93, 1999.
- Brown, S.R, C.J. Scholz and J.B. Rundle, A simplified spring-block model of earthquakes, *Geophys. Res. Lett.*, **18**, 2, 215-218, 1991.
- Burridge, R and L. Knopoff, Model and theoretical seismicity, *Bull. Seism. Soc. Am.*, **57**, 341-371, 1967
- Burridge, R. and C.A. Varga, The fundamental solution in dynamic poroelasticity, *Geophys. J. Roy. astr. Soc.*, **58**, 61-90, 1979.
- Carlson, J.M and J.S. Langer, Properties of earthquakes generated by fault dynamics, *Phys. Rev. Lett.*, **62**, 2632-2635, 1989.
- Cleary, M.P., Fundamental solutions for a fluid-saturated porous solid, *Int. J. Solids Str.*, **13**, 785-806, 1977.
- Dahmen K, Ertas D, Ben-Zion Y, Gutenberg-Richter and characteristic earthquake behavior in simple mean-field models of heterogeneous faults, *Phys Rev E*. **58**, 1494-1501 1998
- Dieterich, J.H., Time dependent friction in rocks, *J. Geophys. Res.*, **77**, 3690-3697, 1972
- Dieterich, J.H., Time dependent friction and the mechanics of stick slip, *Pure Appl. Geophys.*, **116**, 790-806, 1978.
- Dieterich, J.H., Constitutive properties of rock with simulated gouge, In *Mechanical Behavior of Crustal Rocks. AGU Geophys. Monograph 24*, Washington, DC, American Geophysical Union, 103-120, 1981.
- Donnellan A., BH Hager, RW King, and TA Herring, Geodetic Measurement of Deformation in the Ventura-Basin-Region, Southern California, *J Geophys Res* **98**, 21727-21739, 1993.
- Donnellan A., Lyzenga G.A., GPS observations of fault afterslip and upper crustal deformation following the Northridge earthquake *J. Geophys Res* **103** 21285-21297 1998 .
- Fisher, D. S., K. Dahmen, S. Ramanathan and Y. Ben-Zion, Statistics of earthquakes in Simple Models of Heterogeneous Faults, *Phys. Rev. Lett.*, **78**, 4885-4888, 1997.
- Goil, S., Primitives for problems using heirarchical algorithms on distributed memory machines, NPAC Tech Rept. SCCS-687, <http://www.npac.syr.edu/techreports/index.html>, 1994.
- Goil, S. and S. Ranka, Parallelization of heirarchically structured applications on distributed memory machines, NPAC Tech Rept. SCCE-688, <http://www.npac.syr.edu/techreports/index.html>, 1995.
- Greengard, L. and Rokhlin, V., A Fast Algorithm for Particle Simulations, *J. Comp. Phys*, **73**, 325-348, 1987.
- Gross, S., On the scaling of average slip with rupture dimensions of earthquakes, pp.157-166 in JB Rundle, D. Turcotte and W. Klein, eds., *Reduction and Predictability of Natural Disasters*, vol XXV in the Santa Fe Inst. series on the Sciences of Complexity, Addison Wesley, 1996.
- Heimpel, M., earthquake size-frequency relations from an analytical stochastic rupture model, *J. Geophys. Res.*, **101**, 22435-22448 1996.
- Hauksson, E., and J. Hasse. Three-dimensional V-P and V-P/V-S velocity models of the Los Angeles basin and central Transverse Ranges, California, *J. Geophys Res.* **102**: (B3) 5423-5453, 1997
- Hertz, J. , A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Lecture Notes I, Santa Fe Inst. Addison Wesley, Reading, MA, (1991).
- Hill, R. A self consistent mechanics of composite materials, *J. Mech. Phys. Sol.*, **13**, 213-222, 1965.
- Holland, J.H., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.
- Houstis, E. N., J. R. Rice, S. Weerawarana, A. C. Catlin, P. Papachiou, K. Y. Wang, and M. Gaitatzes, PELLPACK: A Problem Solving Environment for PDE Based Applications on Multicomputer Platforms, *ACM Trans. Math. Software*, **24**, 30-73, 1998.
- Ivins, E.R. and C.G. Sammis, On lateral viscosity contrast in the mantle and the rheology of low frequency geodynamics, *Geophys. J. Int*, **123**, 305-322 1995b.
- Ivins, E.R., *Composite Constitutive Theory for a Solid Earth of Strong Lateral Viscosity Contrast*, Ph.D. Dissertation, Univ. Southern Calif., 1995a.
- Kanamori, H. and D.L. Anderson, Theoretical basis for some empirical relations in seismology, *Bull. Seism. Soc. Am.*, **65**, 1073-1095, 1975.
- Li, V.C., Mechanics of shear rupture applied to earthquake zones, in *Fracture Mechanics of Rock*, ed. B.K. Atkinson, Academic Press, London, 1987, pp. 351-428.
- Lyakhovskiy, V., Y. Ben-Zion and A. Agnon, Distributed Damage, Faulting, and Friction, *J. Geophys. Res.*, **102**, 27635-2764, 1997.
- Maruyama, T., Fluid pressure responses of a water-saturated porous elastic multilayered half space to water pumping and atmospheric loading, *J. Phys. Earth*, **42**, 331-375, 1994.
- Massonnet, D., M Rossi, C. Carmona, F. Adragna, G. Peltzer, K. Feigle, T Rabaute, The Displacement Field of the Landers Earthquake Mapped by Radar Interferometry, *Nature* **364**, 138-142 1993.
- Menke, W., *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, 289p 1989
- Michalewicz, Z., *Genetic Algorithms+Data Structures = Evolution Programs*, Springer-Verlag, New York, 1996.
- Minster, J.-B., Twenty five years of seismic source theory, in *The VELA Program, A Twenty-Five Year Review of Basic Research*, ed. A.U. Kerr, Defense Advanced Research Projects Agency, pp. 67-116, 1985.
- Newman, W.I., D.L. Turcotte and A. Gabrielov, A hierarchical model for precursory seismic activation, pp. 243-260, in J.B. Rundle, D.L. Turcotte and W. Klein, *Reduction and Predictability of Natural Disasters*, vol. XXV in the Santa Fe Institute series on complexity, Addison Wesley, 1996.
- Okada, Y., Surface deformation due to shear and tensile faults in a half-space, *Bull. Seism. Soc. Am.*, **75**, 1135-1154, 1985.
- Okada, Y., Internal deformation due to shear and tensile faults in a half-space, *Bull. Seism. Soc. Am.*, **82**, 1018-1040, 1992.
- Pollitz FF, Burgmann R, Segall P., Joint estimation of afterslip rate and postseismic relaxation following the 1989 Loma Prieta earthquake, *J Geophys. Res.* **103**, 26975-26992 1998.
- Rabinowicz, E., *Friction and Wear of Materials*, John Wiley, New York, 1965.

- Rawlins, G.J., *Foundations of Genetic Algorithms*, Morgan-Kaufman publishers, San Francisco, 1991.
- Rice, J.R. and M. Cleary, Some basic stress diffusion solutions for fluid saturated elastic porous media with compressible constituents, *Rev. Geophys. Space Phys.*, **14**, 227-241, 1976.
- Rice, J.R., Spatio-temporal complexity of slip on a fault, *J. Geophys. Res.*, **98**, 9885-907, 1993.
- Rice, J.R. and Y. Ben-Zion, Slip complexity in earthquake fault models, *Proc. Natl. Acad. Sci. U.S.A.*, **93**, 3811-3818, 1996.
- Rice, J.R. and A.L. Ruina, Stability of steady frictional slipping, *J. Appl. Mech.*, **105**, 343-349, 1983.
- Ruina, A., Slip Instability and State Variable Friction Laws, *J. Geophys. Res.* **88**, 359-370, 1983.
- Rundle, J.B. and D.D. Jackson, Numerical simulation of earthquake sequences, *Bull. Seism. Soc. Am.*, **67**, 1363 - 1378, 1977.
- Rundle, J.B., Viscoelastic - gravitational deformation by a rectangular thrust fault in a layered earth, *J. Geophys. Res.*, **87**, 7787 - 7796, 1982.
- Rundle, J.B., Some solutions for static and pseudo - static deformation in layered, nonisothermal media, *J. Phys. Earth*, **30**, 421 - 440, 1983.
- Rundle, J.B., A physical model for earthquakes: 1, Fluctuations and interactions, *J. Geophys. Res.*, **93**, 6237-6254, 1988.
- Rundle, J.B. and S.R. Brown, Origin of rate dependence in frictional sliding, *J. Stat. Phys.*, **65**, 403-412, 1991.
- Rundle, J.B. and W. Klein, Nonlinear dynamical models for earthquakes and frictional sliding: An overview, *Proc. 33rd Symposium Rock Mech.*, A.A. Balkema, Rotterdam, 1992.
- Rundle, J.B. and D.L. Turcotte, New directions in theoretical studies of tectonic deformation: A survey of recent progress, in *Contributions of Space Geodesy to Geodynamics: Crustal Dynamics*, ed. D.E. Smith and D.L. Turcotte, Geodynamics Ser. Vol. 23, AGU, Washington, DC, 1993.
- Rundle, J.B., W. Klein and S. Gross, Dynamics of a traveling density wave model for earthquakes, *Phys. Rev. Lett.*, **76**, 4285-4288, 1996.
- Rundle, J.B., Salmon, J.K. and M.S. Warren, Parallel supercomputing with commodity components, *Proc. Intl. Conf. Parallel Distributed Processing Tech. Appl. (PDPTA '97)*, ed. M.S. Warren, D.J. Becker, M.P. Goda, J.K. Salmon and T. Sterling, Las Vegas, NV, June 30-July 3, 1997.
- Smalley, R.F., D.L. Turcotte and S.A. Solla, A renormalization group approach to the stick slip behavior of faults, *J. Geophys. Res.*, **90**, 1884-1900, 1985.
- Stein, R.S., The role of stress transfer in earthquake occurrence, *Nature*, **402**, 605-609, 1999.
- Stuart, W.D. and T.E. Tullis, Fault model for preseismic deformation at Parkfield, California, *J. Geophys. Res.*, **100**, 24079-24101, 1995.
- Winckelmans, G.S., J.K. Salmon, M.S. Warren, A. Leonard and B. Jodoin, Application of fast parallel and sequential tree codes to computing three-dimensional flows with vortex element and boundary element methods, in *Vortex Flows and Related Numerical Methods*, Montreal, 1995.

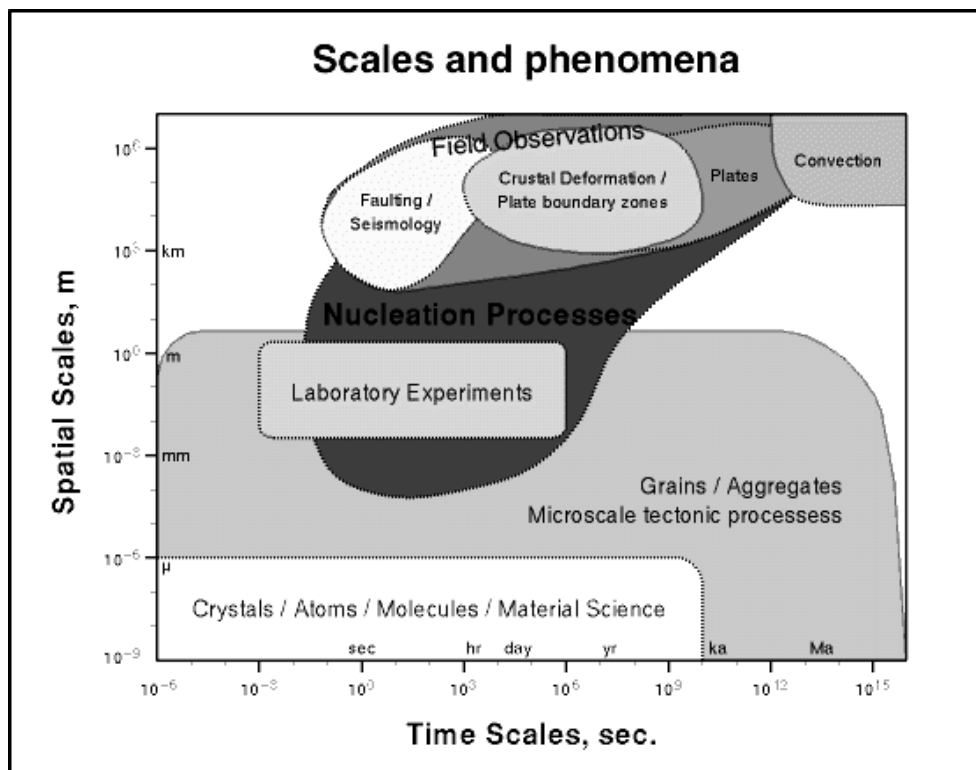


Figure 1. Spatial and temporal scales in earthquake science.

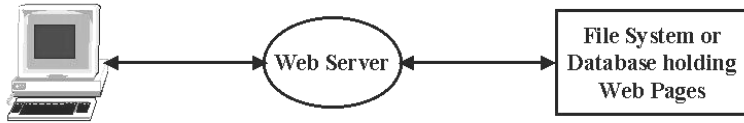


Figure 2. Three-tier computing model illustrated by basic web access.

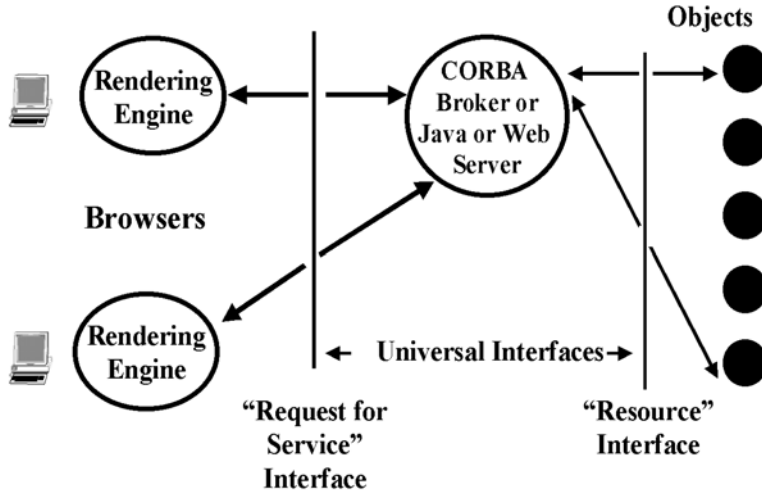


Figure 3. General three-tier architecture.

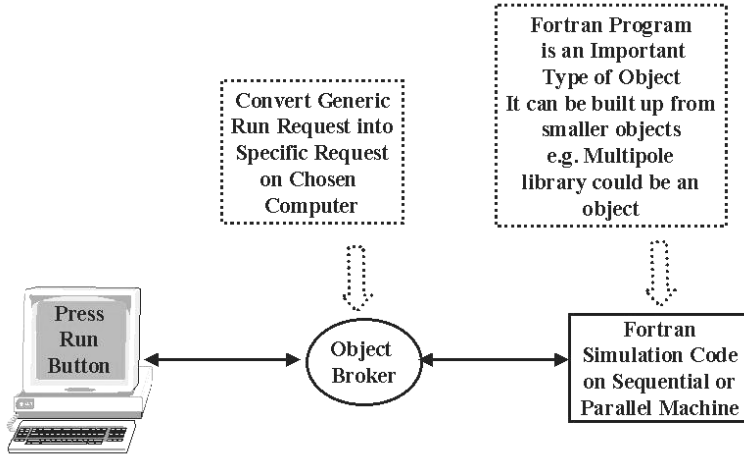


Figure 4. Simple simulation formulated as a “distributed computational object.”

```

<?xml version="1.0"?>
<!DOCTYPE application SYSTEM "ApplDescV2.dtd">
<application id="disloc">
<target id="osprey4.npac.syr.edu">
  <status installed="Yes"/>
  <installed>
    <CmdLine command="/npac/home/webflow/GEM/JAY/dis2loc" />
    <input>
      <inFile Path="/npac/home/webflow/GEM/JAY/" Name="disloc.output"/>
      <source Host="osprey4.npac.syr.edu" Path="/npac/home/Jigsaw/WWW/tmp"
        Name="disloc.out"/ >
    </input>
    <output>
      <outFile Path="/npac/home/webflow/GEM/JAY/" Name="simplex.input" />
      <dest Host="osprey4.npac.syr.edu"
        Path="/npac/home/webflow/GEM/JAY/simplex/" Name="s.in" />
    </output>
    <stdout Host="aga.npac.syr.edu" Path="/npac/home/haupt/webflow/history/"
      Name="job2001.out" >
    <stderr Host="aga.npac.syr.edu" Path="/tmp/" Name="haupt_job2001.err" >
  </installed>
</target>
</application>

```

Figure 5. XML Used to describe the computational problem defined in Figure 14.

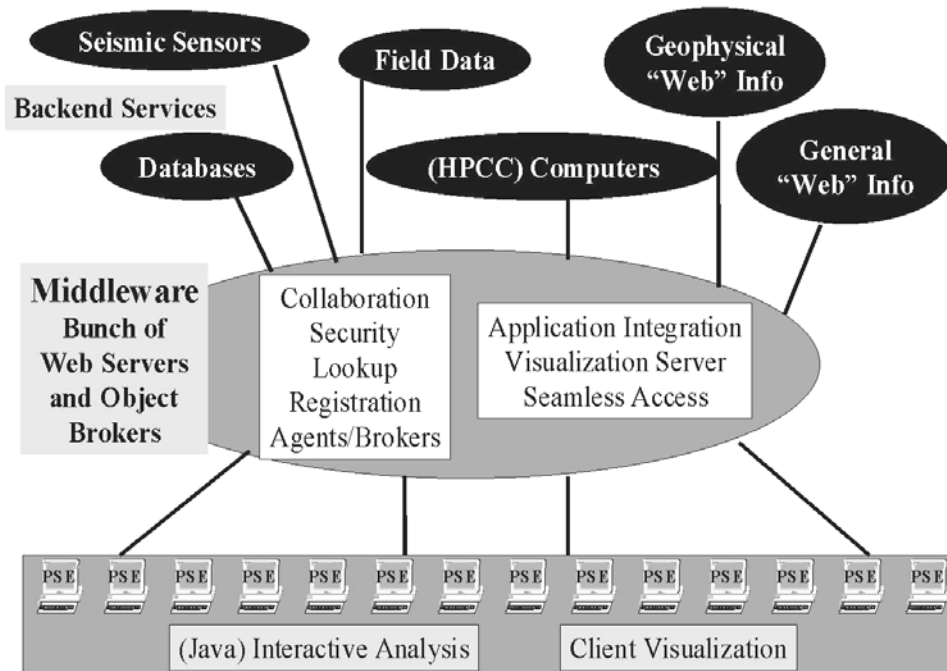
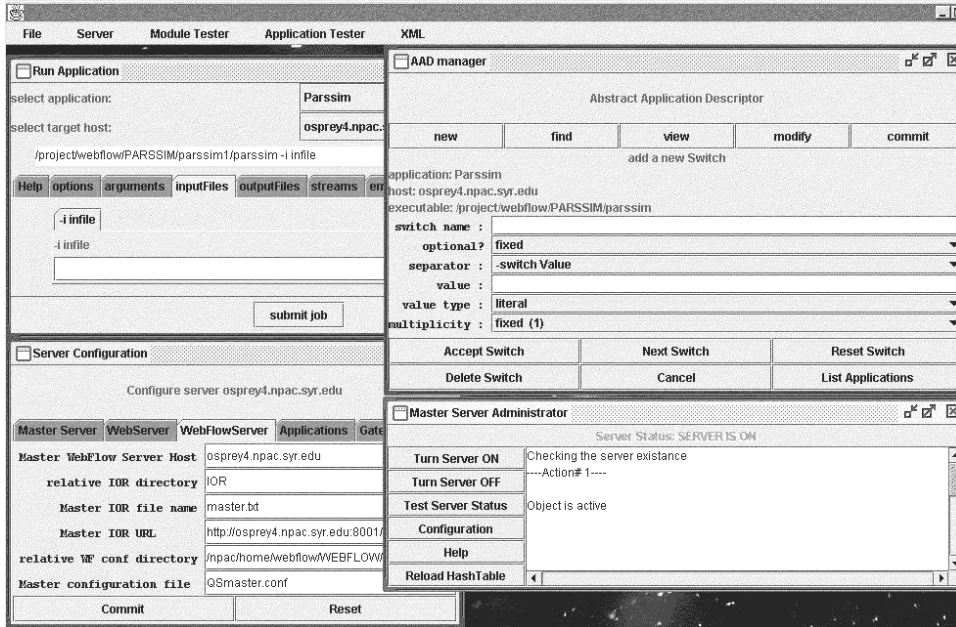


Figure 6. Three-tier architecture of a GEM Problem Solving Environment.



**Figure 7.** This illustrates several components of the WebFlow system from NPAC. (Counter-clockwise from bottom-left): The Server Configuration and Master Server Administrator are WebFlow administrative tools that allow one to configure and monitor the middle tier servers. AAD manager (Abstract Application Descriptor) is a tool to incorporate new applications to the system: the interface shown simplifies creation of an XML definition of the application. This definition can be used for dynamical creation of front-end application interfaces (window "Run Application").

# WebFlow on Globus -- LMS at CEWES

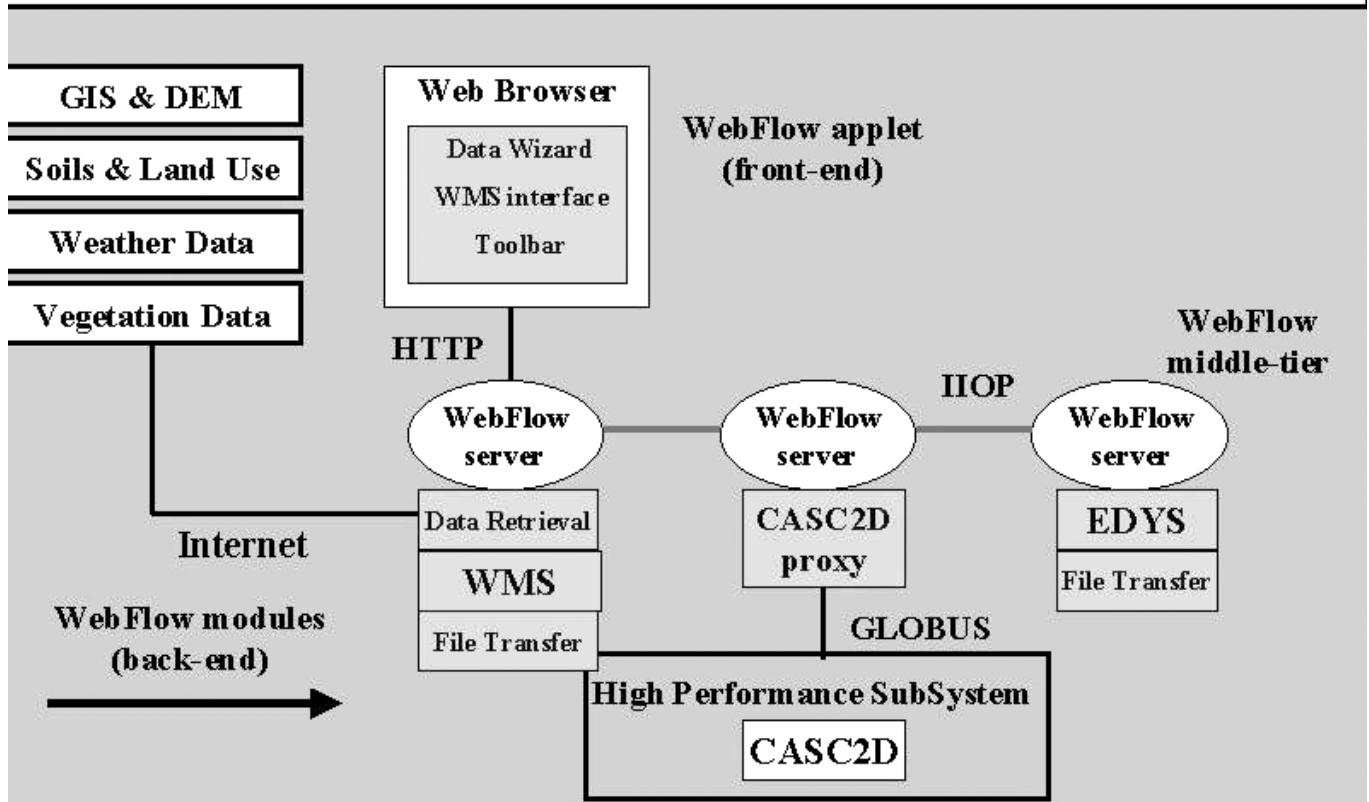


Figure 8. Three-Tier Architecture of DoD "Land Management" application whose front end is shown in Figure



## Example of a custom Web User Interface DoD Land Management System

Navigate and choose  
an existing application  
to solve the problem at hand.  
Import all necessary data.

Department of Defense Land Management System

This project was sponsored by the U.S. Army Corps of Engineers Waterways Experiment Station  
Major Shared Resource Center (MSRC) at Vicksburg, MS  
under the DoD HPC Modernization Program, Programming Environment & Training (PET)

**LMS**

Data Wizard WMS CASC2D

Retrieve data  
Pre/post-processing  
Run simulations

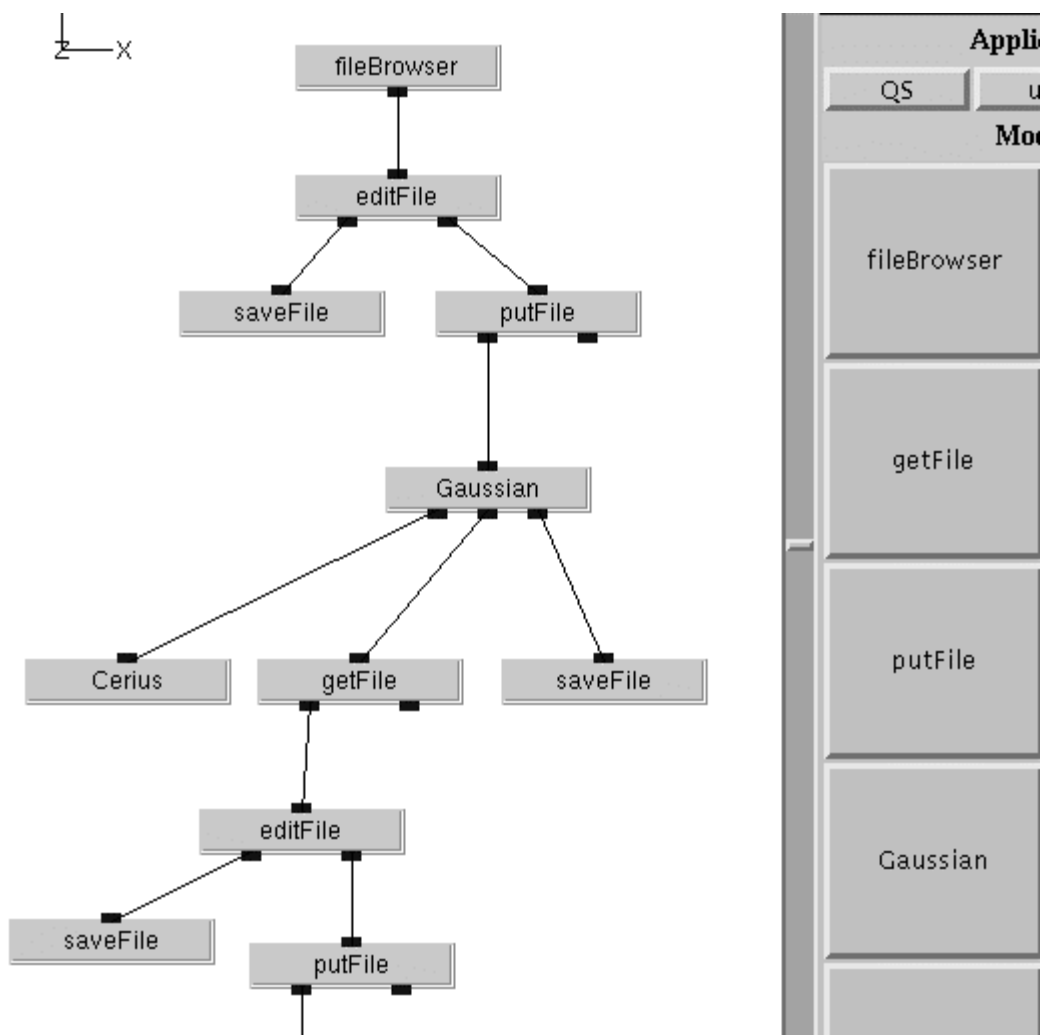
Select host  
Select model  
Set parameters  
Run

select region  
Type in bounding box coordinates  
W:100 N:100 E:100 S:100  
get data

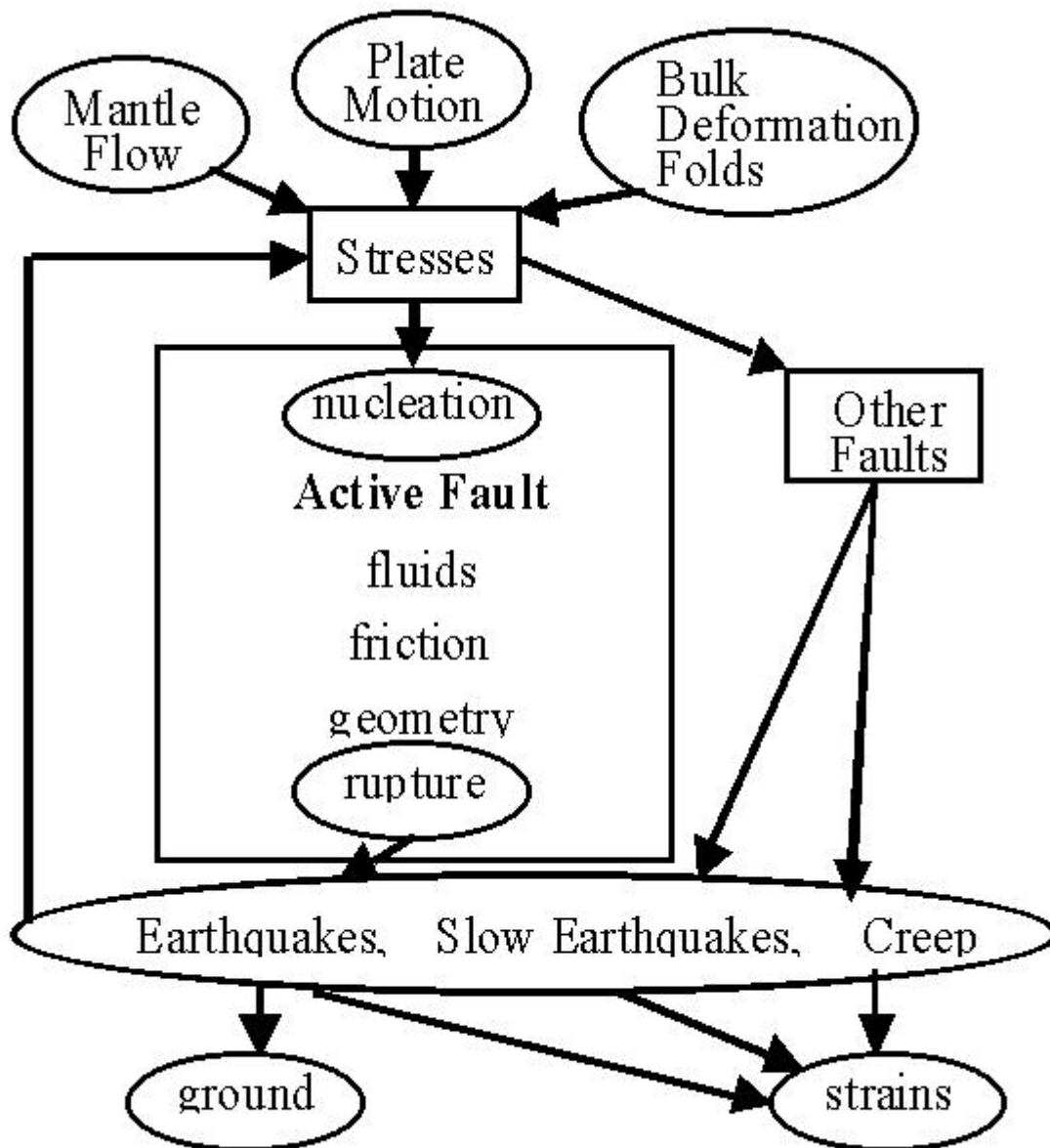
load USGS data  
download DEM data, in tchell...  
file size is 1.40014

CASC2D Interface  
select host: obsidian.wes.army...  
select model:  CASC2D  EDYS  CASC2D and EDYS  
EDMS parameters:  
start date: Jan 1 1997  
end date: Jan 1 1998  
visualize data every month: no  
disturbances: 0  
run selected application

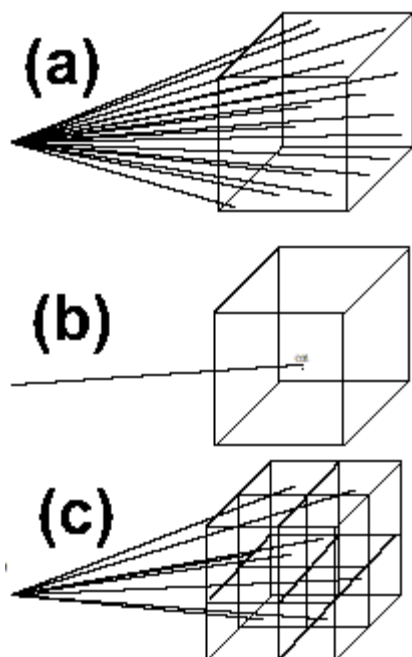
**Figure 9.** Example of a Web Interface for a “Land Management Problem Solving Environment” built by NPAC for the Department of Defense ERDC Laboratory in Vicksburg, MS.



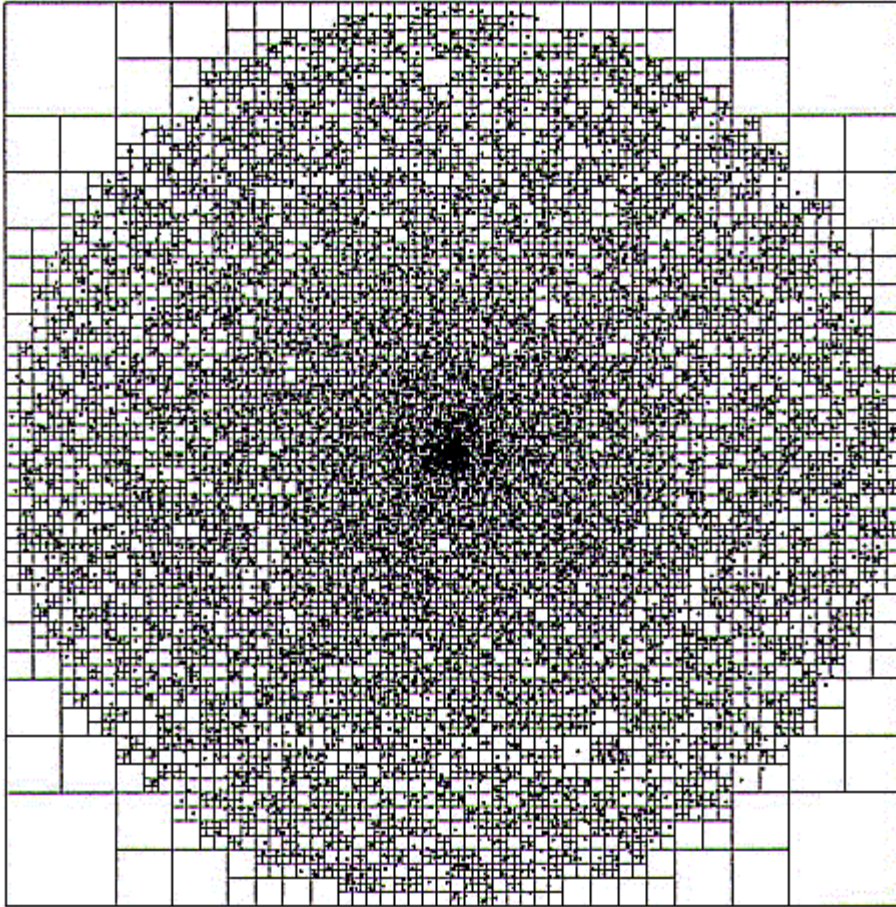
**Figure 10.** Fragment of WebFlow Composition Tool linking modules in a quantum simulation (chemistry) application.



**Figure 11.** Logical structure for model of Section 4.



**Figure 12.** Three ways of calculating long range forces.



**Figure 13.** Adaptive division of disk of particles into cells.

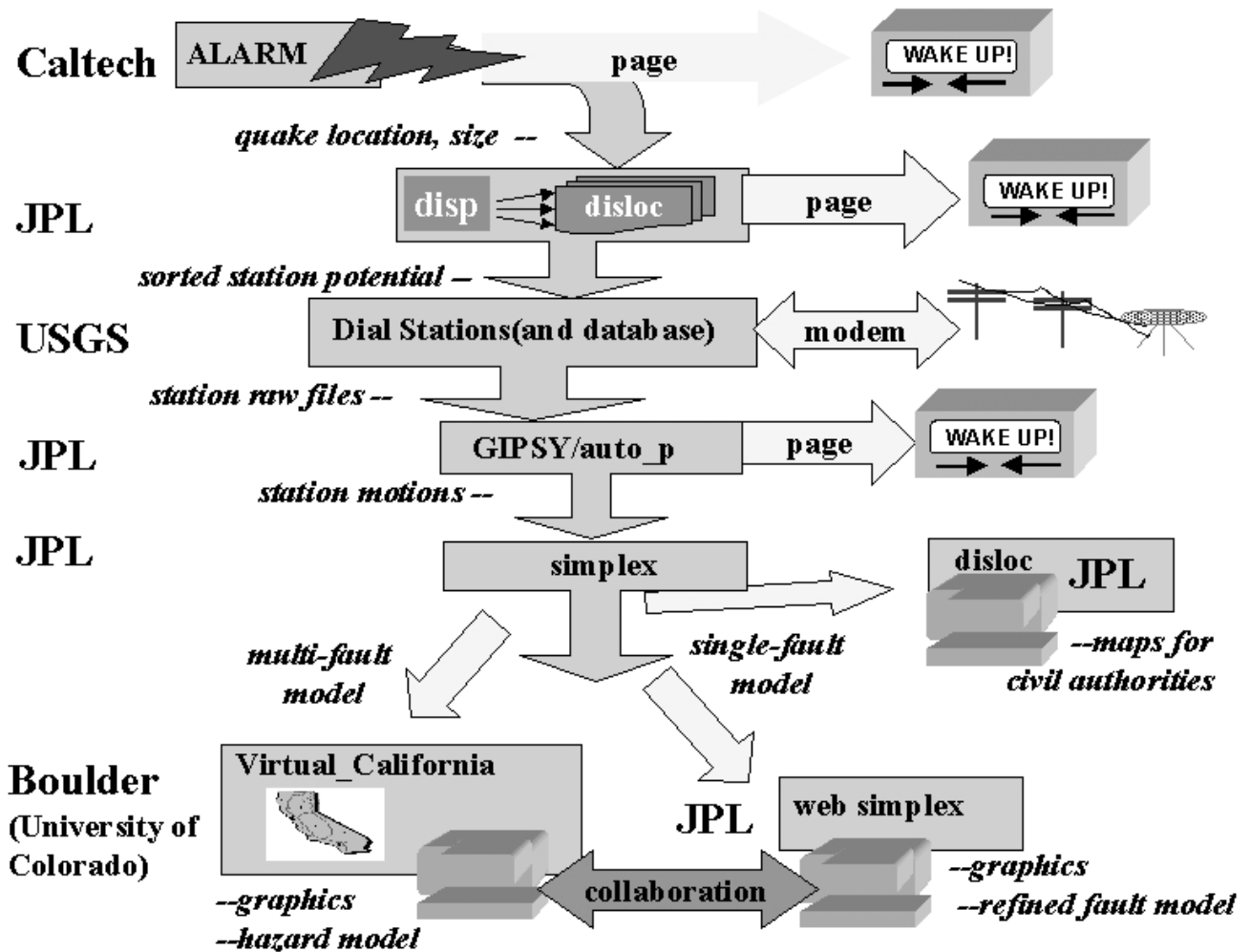
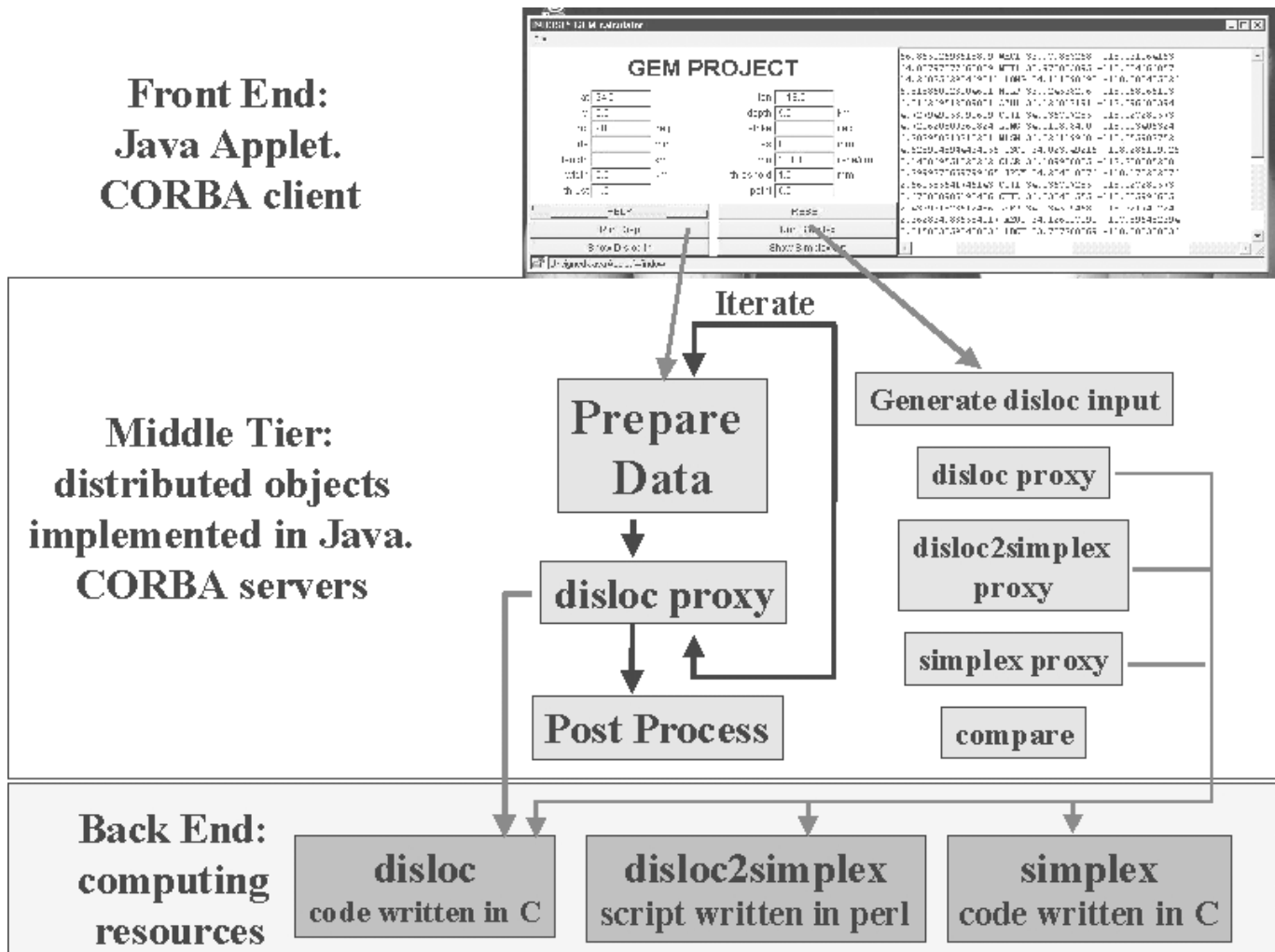


Figure 14. Simulations used in Response to an Earthquake.



**Figure 15.** Computational structure of a fragment of the scenario in Figure 14, which was implemented by NPAC as a prototype GEM Problem Solving Environment.