Software Engineering / Development Plan

Numerical Simulations for Active Tectonic Processes:Ê Increasing Interoperability and Performance

JPL Task Plan No. 83-6791
Milestone A:

A - Administration	***Software engineering plan completed. Review board selected.***	***03/29/02***

Introduction

Purpose

We plan to develop a solid earth system science framework for creating an understanding of active tectonic and earthquake processes. It will include:

- A database system for handling both real and simulated data.
- Fully three-dimensional finite element code with adaptive mesh generator capable of running on workstations and supercomputers for carrying out earthquake simulations.
- Inversion algorithms and assimilation codes for constraining the models and simulations with data.
- A collaborative portal (Object Grid Framework) allowing for seamless communication between codes, reference models, and data.Ê The Object Grid Framework is a multi-tier middleware that acts as an object broker, arranging access to data, and enabling the use of programs and models to facilitate rapid exploration of ideas and datasets.
- Visualization codes for interpretation of data and models.
- Pattern recognizers capable of running on workstations and supercomputers for analyzing data and simulations.

Background

This project will create an overall system. Many of the individual components already exist in some form (see code descriptions) andÊ will be parallelized to allow us to address larger problems and wrapped as objects to be brought into an integrated problem solving environment (PSE). The PSE will be built using these components and the Gateway and Community Grids Collaboratory also described in the code descriptions.

Organization and Responsibilities

Project Personnel

The development team is composed of a group of collaborators distributed across the US at various institutions. Each group has a well defined part of the project in which they have recognized cognizance and authority which derives from the fact that they are contributing a significant module and expertise that they have already spent considerable effort developing.

Dr. Andrea Donnellan is the PI of this project and will oversee all aspects of the work.

ÊMany of the team members have worked extensively and effectively together in the past. We plan to conduct regular reviews during the course of this work. We will also hold meetings at JPL in which all of the team members can interact and plan work. Project-wide decisions will be made by the co-investigators lead by the PI, Andrea Donnellan.

Team members will be responsible for developing and completing different components of the system as outlined in the following table in which we itemize the components of the system, the responsible members and their background and skills relating to that component of the framework.

Dr. Donnellan will be assisted by the Software Engineer in the person of Dr. Jay Parker, and the Software Architect Dr Geoffrey Fox.

Component	Team Members	Task, Background, and Skills
Database system	Dennis McLeod	Database systems
	Lisa Grant	Geology/paleoseismology
	Andrea Donnellan	Tectonics, earthquakes
Finite element code	Greg Lyzenga	Parallel computation/FEM
	Jay Parker	Parallel computation/FEM
	John Lou	Parallel computation/FEM/AMR
Inversion and assimilation codes	Greg Lyzenga	Inversion
	John Rundle	Viscoelastic models
	Terry Tullis	Fault nucleation, fast multipoles
	John Lou	Fast multipoles, data assimilation
Object Grid Framework	Geoffrey Fox	Science interoperability
	Dennis McLeod	Science interoperability
Visualization codes	Geoffrey Fox	Visualization and integration in portals
		Geophysical applications
	Andrea Donnellan	
Pattern recognizers	Robert Granat	Hidden Markov Modeling
	John Rundle	Pattern dynamics

Interfacing Groups

Component	Inst.	Person	Phone	Email
Overall Management	JPL	Andrea Donnellan	See above	See above
Configuration and overall Software engineering	JPL	Jay Parker	818-354-6790	Jay.W.Parker@jpl.nasa.gov
Software Architecture	Indiana	Geoffrey Fox	315-254-6387	gcf@indiana.edu
Database system	USC	Dennis McLeod		mcleod@pollux.usc.edu
	UCI	Lisa Grant	949-824-5491	

	JPL	Andrea Donnellan	818-354-4737	lgrant@uci.edu Andrea.Donnellan@jpl.nasa.gov
Finite element code	JPL	Greg Lyzenga	818-354-6920	Gregory.A.Lyzenga@jpl.nasa.gov
	JPL	Jay Parker	818-354-6790	Jay.W.Parker@jpl.nasa.gov
	JPL	John Lou	818-354-1146	John.Z.Lou@jpl.nasa.gov
Inversion and assimilation codes	JPL	Greg Lyzenga	See above	See above
	U Colo	John Rundle		rundle@hopfield.colorado.edu
	Brown	Terry Tullis	401-863-3829	Terry_Tullis@brown.edu
	JPL	John Lou	See above	See above
Object Grid Framework	Indiana	Geoffrey Fox	315-254-6387	gcf@indiana.edu
	USC	Dennis McLeod		See above
Visualization codes	Indiana	Geoffrey Fox		See above
	JPL	Andrea Donnellan	See above	See above
Pattern recognizers	JPL	Robert Granat	818-393-5353	Robert.A.Granat@jpl.nasa.gov
	U Colo	John Rundle	See above	See above

Many of the codes will be run on computers at NASA facilities at Goddard and Ames, and others will run on computers located at JPL and various other institutions. They will be joined by the Object Grid Framework

We will develop a written plan for development of this middleware. This plan will be placed under change control after it is ready.

The Object Grid Middleware is built around Java Servers linked by SOAP and RMI connections. The needed software will run on IBM, Sun UNIX and Linux servers.

The system is intrinsically integrated as all components have XML interfaces and as they are web-based, they will support a web documentation interface and distributed hosting and updating. The performance of production systems may require co-location of services on a given machine but initial development can be totally distributed. Research is ongoing in the community as to appropriate registration and discovery services and the simplest approach for our relatively small effort is WSIL.

Statement of the Problem

We are building a new Problem Solving Environment for use by the seismological, crustal deformation,Ê and tectonics communities.

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

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

- Pattern recognizers capable of running on workstations and supercomputers for analyzing data and simulations.

The details of our approach will be refined and solidified as we develop the interoperability framework plan for milestone H (*Come to agreement on design policy for interoperabilityÊ and community delivery - Review board approves requirements and a preliminary design for functionality)*. That plan will outline our approach and philosophy to achieve milestone J (*Full implementation using improved codes. Review board approvesÊ integration into completed framework andÊ updated documentation for:*·).

The Southern California Earthquake Center (SCEC) has recently received funding for an NSF ITR proposal.Ê We will work with SCEC from the outset to assure that efforts are mutually productive.Ê Both projects strive to improve our understanding of earthquakes, however, the methods for achieving that goal are different, but complementary.

Both organizations have needs for fault and other databases and both will focus on using federated databases.Ê SCEC is interested in putting known, agreed upon faults into their database.Ê Our simulations require agreed upon faults and possible faults in a database in order to do testing and validation of faults.Ê SCEC will be focusing on seismic wave propagation and hazard analysis, whereas our project will be focusing on simulations, fault networks, and crustal deformation data.Ê Not only will our models be different, but our data types will also differ to some extent.Ê As discussed in the proposal, our efforts will be coordinated with other efforts as much as feasible.Ê Lisa Grant is the representative from this group to the SCEC/RELM/USGS efforts.Ê Due to the different needs we are unable to simply adopt their fault database.

In order to strengthen both projects and earthquake research overall we will work together from the outset to share technologies where possible, develop common standards, and develop interfaces between all of the models.Ê We have begun to do so already.Ê SCEC and GEM held a joint meeting in Maui in early August 2001. ÊWe met again at the SCEC annual meeting in September, had a subsequent meeting at JPL with relevant developers in early October, and a two day workshop at the USGS in Pasadena in January 2002.

We expect to share interface definitions.Ê In particular, XML definitions for faults, seismicity, and deformation will be common. We are working collaboratively with the SCEC community to jointly develop acceptable standards. At this point, we have no plans to share software components.

Technical Approach

It is our intention that everything that can reasonably be put on the web be accessible there. Since this project involves several groups of people at several locations around the country, it is necessary that such common information be easily accessible to all. The SW development plan, milestones, progress, and bug reports will all be on the web.

Development Environment

Due to the nature of the project, it will be implemented on several different types of hardware platforms, using a variety of languages. Some codes and problems will require implementation on large parallel supercomputers, while others may reside on desktop machines. Similarly, some large datasets my have to reside on dedicated servers while others are small and simple enough to be able to reside almost anywhere. Many of the software modules already exist. The maturity of the existing legacy modules varies. They are written primarily in C, C++, FORTRAN77, or FORTRAN90. We plan to write Java wrappers for such existing codes such that they can be invoked, and data can be read into and out of them, typically inÊ XML for metadata and NetCDF for larger binary data. The central Object Grid FrameworkÊ (Gateway) will be written in Java and EJB.

Detailed plans for how this will be accomplished will be addressed for Milestone H (*Come to agreement on design policy for interoperabilityÊ and community delivery - Review board approves requirements and a preliminary design for functionality*).

Activities, tools and products (including documentation)

We plan to make use of lightweight methodologies for developing the requirements and the codes for this project. We expect to have many smallÊ incremental changes to the system we are building. This will allow us to review the requirements,Ê to adapt to changes in the requirements, and find errors in integration on the upstream side of the project, yielding higher quality in the long run. We will have monthly goals which will be set at an incremental planning meeting involving all the developers and the SW engineer. If a goal is too large to fit into one month, it will be subdivided.

Activities to be performed

Requirements

The first activity to be performed in the execution of this project is to develop a set of requirements for each part of the collaboratory.Ê To do this, we will :

- Identify end users (scientists) who have the credentials to define the software that will be developed, and interview them to derive preliminary requirements regarding interfaces, functionality, security, algorithms, and error reporting. The result of this step will be a set of scenarios that will illustrate how scientists would like to be able to operate. The users will additionally place priorities on each of the scenarios.
- Use these draft scenarios to develop preliminary user interfaces to be shown to users for revising requirements and refining the interfaces.
- Develop a prototype to demonstrate the functional area of the system. This prototype may or may not be subsequently extended to become part of the final system. The goal here is to explore and illuminate the required functionality as quickly and cheaply as possible to help further refine the requirements. Hence the prototype may not even be in the same language as the final implementation. It may rather be in a quick scripting language form, with word-processor-generated mock output.

The final set of requirements can then be written and will then be placed under change control. Subsequent changes to the requirements will need consensus of the development and user teams. This procedure for developing requirements will be repeated after each code delivery and as part the annual report generation process.

Preliminary/Detailed Design

Based on the requirements,Ê the design of each sub-system will be fleshed out with a short description of the philosophy, a specification of the language(s) to be used,Ê and the approach to be taken. The detail in the description of the approach will be commensurate with the complexity of the task at hand and will include the requirements by reference. It will describe how that component will be built, and exactly what it should do/not do. The format will be flexible but will likely include diagrams, and certainly will include interface specs**.**

The design description will be subject to the approval of the investigators and the Software Engineer. This description will be maintained along with the requirements,Ê and will become part of the project documentation.

Testing

There will be three phases to the testing necessary to ensure a quality system. The first phase will be done by the developers, testing their code as they develop it. This phase will be informal and will not be documented.

In the second phase, the Software Engineer (Parker) will develop a suite of tests to ensure that the software meets the requirements and contains theÊ necessary functionality. This suite, and the results of the tests will be maintained along with the requirements and will become part of the project documentation.

For the third and final phase, we have assembled a team of independent scientists who have agreed to act as testers for the project. The Software Engineer will work with members of this team to design a test plan that will do real science problems to push the envelope and complete the third phase.

Both the second and third phases will in many cases involve comparison with other software for purposes of validation.

Defects uncovered during the second and third phases of testing will be tracked with a tool such as Bugzilla or Seapine Software Test Track Pro 3.0 which provides a simple web interface to allow testers to report defects.

Maintenance

The heart of the maintenance plan will be to ensure that the software is easy to understand and therefore easy to modify. A high degree of modularity will also enable maintenance on one area without affecting others.

We intend to monitor the maintainability of the code using Halsteadâs Effort Equation and McCabeâs Cyclomatic Complexity index.Ê Codes exhibiting large values of these metrics will receive additional scrutiny.

Tools to be used

Microsoft Project or Fast Track will be used to develop and track scheduling.

Bugzilla, or Seapine Software Test Track Pro 3.0 or a similar tool will be used to track defects uncovered in the software.

We will maintain version control using the Concurrent Versions System (CVS).

We will identify tools to collect maintainability metrics on the software.

Products generated

The products generated will include not only the problem solving environment, but also the documentation, consisting of the requirements, design documents, test cases, and user-guides.

Build strategy

See the accompanyingÊ Milestone chart.

Management Approach

Milestones and Schedules

The milestones and schedule is listed in an accompanying Excel spreadsheet.

The optional milestones listed in the Excel spreadsheetÊ use in-house HPCC funds which are not part of this proposal.

Metrics

The primary metric that will be used to capture project status will be the number of mini-milestones accomplished. At the beginning of each month, the CO-Is will meet by telephone to decide on a set of specific mini-milestones to be accomplished in the next month. These mini-milestones will be constructed such that they will each result in some useful piece of the whole project. The pieces will be delivered at the end of the month, and the cycle will repeat. The mini-milestones will be chosen from a collection such that the major milestones will be met by their assigned dates.

Other metrics include lines of code converted/generated, relative speed increases, problem size increases, and work-hours spent.Ê We also will use the Halsteadâs Effort Equation and McCabe Cyclomatic Complexity index to gauge the maintainability of the codes developed. If tools for deriving these last two measures from the source code are not available, we may generate a tool in perl, or we may forego using these metrics.

Risk Management

We plan to use a staged delivery plan to identify risks and problems as early as possible in the development effort.

Technical Reviews will be conducted of all products developed by this project. Individual modules will be reviewed by the team developing them. Other products, with wider impact such as this Software Engineering Plan, will be reviewed by a wider group of people.

At the start of each activity, an apriori estimate of the time required to complete the task will be estimated. This estimate will be compared with the actual time spent. The purpose of this comparison will be to train our intuition in estimating the level of effort required, to allow us to more accurately predict needed resources as the project matures. The time spent will be broken down into categories. Our initial list of categories is:

- Management
- Administration
- Process development
- Requirements development
- User-interface prototyping
- Architecture
- Detailed design
- Implementation
- Component acquisition
- Integration
- System testing
- Software release
- Metrics

Dr. John Lou will serve as Risk Officer and will actively seek out and identify potential risks. Part of each staff meeting will be devoted to identifying and discussingÊ risks. We expect to maintain a list of risks in our defect tracking system.

The current list of risks is:

Personnel changes	Maintain necessary documentation to minimize loss of an individual. Guiding individuals are stable and unlikely to leave.
Creeping requirements	Requirements will be developed carefully and then kept under change control. Revisions will need the agreement of the change committee.
Unachievable schedule	Schedule is reestimated several times over the course of the project. Upstream reviews are used to detect and correct problems when it is least expensive to do so.
Released software has low quality	Prototypes developed and presented to the user community before coding begins

	Test planning assures all functionality will be covered by system testing. Independent reviewers will be involved in testing.

Deliveries

Deliveries will be accomplished as specified in the Task Plan. Documentation, required source code developed under this project, and reports will be placed on the web athttp://quakesim.jpl.nasa.gov/index.html or another web site which will be linked to our CT web site, and an email notification will be sent to the interested parties. We anticipate that the portals developed in this project will themselves be a primary deliverable. We will notify the interested parties as these portals become functional.

Product Assurance

Configuration Management

The Software Engineer (Parker) will be responsible for overall configuration management for the project. Due to the distributed nature of the project, the individuals leading the efforts at each location will implement local configuration management procedures compatible with the procedures at JPL. Each location will implement procedures that will ensureÊ there are at least monthly complete backups, and that the media used for backup is rotated to an off-site location to guard against complete loss in the event of catastrophic loss of the buildings. These plans and recovery plans in the event of a disaster will be developed and reviewed with the PI.

We plan to use the Concurrent Version System (CVS) or equivalent for version control.Ê Items to be maintained under CVS will include requirements,Ê development plans, detailed design documents, source codes, test suites, and user documentation.

Quality AssuranceÊ

John Lou will oversee QA efforts andÊ will be responsible for auditing requirements, architecture, design, code, and test plans to ensure that the project meets its own standards. He will lead technical reviews of all phases of the project and will report directly to the PI.

We will conduct reviews of products as they are developed starting early in the project. The purpose of these reviews will be to identify defects as early as possible.

System tests by independent testers will be used to find defects downstream. A requirements traceability matrix will be employed to make sure that the testing is complete.

Glossary of Acronyms

GEM General Earthquake Model.

HPCC	High Performance Computing Challenge
ITR	Information Technology Research
NASA	National Aeronautics and Space Administration
NSF	National Science Foundation
PSE	Problem Solving Environment
RMI	Remote Method Invocation, or Remote Multi-channel Immersion
SCEC	Southern California Earthquake Center
SOAP	Simple Object Access Protocol
RELM	Regional Earthquake Likelihood Model
USGS	United States Geological Survey
WSIL	Web Service Inspection Language
XML	Extensible Markup Language