NOAA TSUNAMI FORECASTING SYSTEM: DESIGN AND IMPLEMENTATION USING SERVICE ORIENTED ARCHITECTURE

Donald W. Denbo[1*], K.T. McHugh[1], J.R. Osborne[2], P. Sorvik[1], and A.J. Venturato[1]
[1]Joint Institute for the Study of Ocean and Atmosphere, University of Washington, Seattle, WA
[2]OceanAtlas Software, Vashon, WA

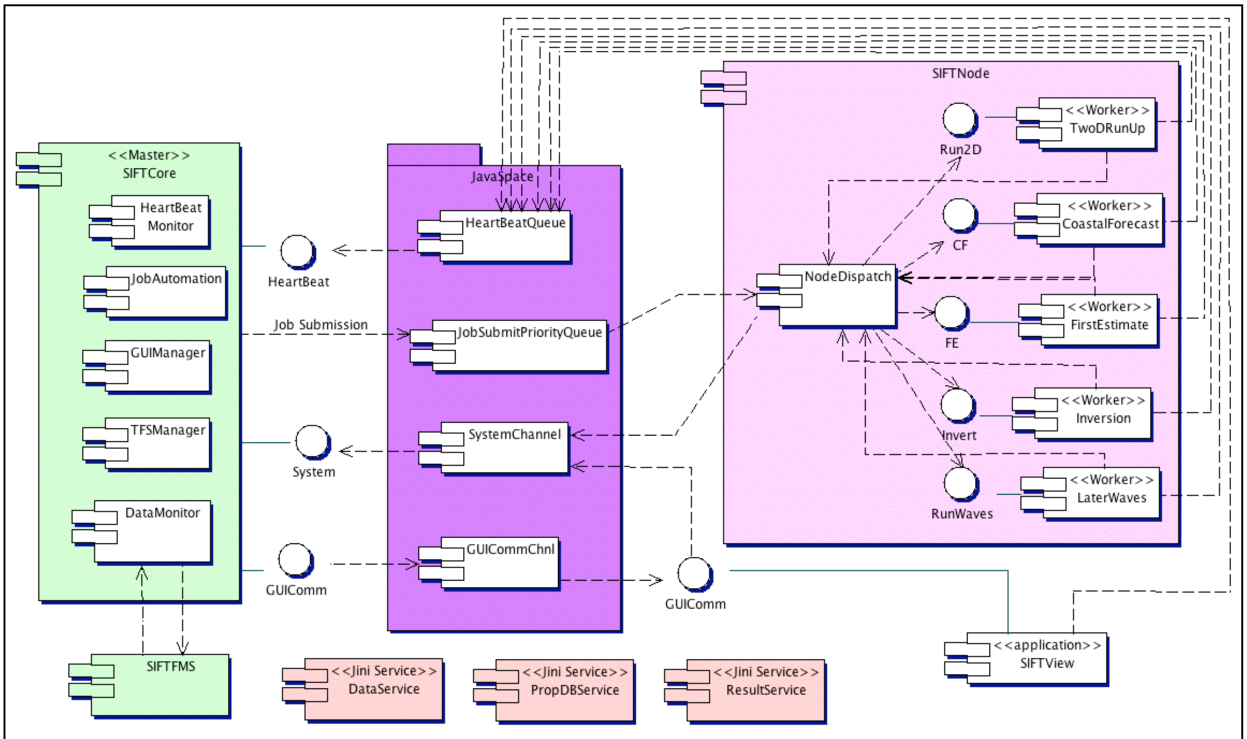

Figure 1. Block diagram of Tsunami Forecasting System architecture.

1. INTRODUCTION

Recent advances in tsunami measurement and numerical modeling technology by the NOAA Center for Tsunami Research in collaboration with National Weather Service Tsunami Warning Centers has lead to the development of a tsunami forecasting system that assimilates real-time event data with numerical model results to provide estimates of tsunami amplitudes and arrival times for potential at-risk communities.

This paper describes the design and implementation of the Tsunami Forecasting System (TFS) to take these advances and produce an operational system. SIFT (Short-term Inundation Forecasting for Tsunamis) is one of the operational component of TFS. The goal of this system is to reduce tsunami warning system false alarms and to issue site-specific forecasts of maximum tsunami flooding and inland penetration.

* *Corresponding author address:* Donald W. Denbo, NOAA/PMEL/OCRD, 7600 Sand Point Way NE, Seattle, WA 98115; e-mail: Donald.W.Denbo@noaa.gov

2. DESIGN

2.1 *Goals*

To provide both the Pacific Tsunami Warning Center (PTWC) and West Coast/Alaska Tsunami Warning Center (WC/ATWC) with an operational tsunami forecast system that meets both centers needs, we have chosen a design that:

- Provides a robust cross-platform architecture.
- Uses proven technologies to increase scalability.
- Uses a modular design for maximum flexibility and reusable components.

The technology chosen to meet these goals include:

- Java 5.0 as the primary programming language and Virtual Machine.
- Java Swing as the user interface framework.
- Jini/JavaSpaces 2.1 to provide the Service Oriented Architecture
- PostgreSQL 8.0 with PostGIS 1.0 for spatially enabled relational databases.

2.2 *Architecture*

The system architecture is illustrated in Figure 1. The main components communicate using objects stored in a JavaSpace. A JavaSpace provides a distributed persistence and object exchange mechanism for Java objects. Additional communications channels exist between the Jini Services and system modules.

SIFTCore. Coordinates SIFT activities via several subsystems. These subsystems include:

- HeartBeat Monitor. Used to monitor the health of the individual system components.
- Job Automation.
  - Controls transactions.
  - Manages duplicate requests and seismic events.
  - Provides required automated behavior. This includes starting the First Estimate, Coastal Forecast, and Composite First Estimate when a new or edited seismic event is received.
- GUI Manager. Coordinates communications between the SIFTView modules (the actual user interface) and the rest of SIFT.
  - Accepts requests and posts results
  - Posts sea-level data
  - Provides system status notifications.
- TFS Manager. Responsible for coordinating between the SIFTCore subsystems and the other modules.
  - Configuration management.
  - Event, Job, and Result processing and message handling.
- Data Monitor. Coordinates with SIFTFMS, the file monitor system, to import seismic events and sea-level data into SIFT.
- Relational Database. Stores system configuration, tidal coefficients, GUI configuration, and performs necessary transaction management.

SIFTFMS. Monitors a local file system to determine if new seismic events or sea-level data are available. When data become available, SIFTFMS reads the new data and using sockets, sends an XML encoded data packet to the SIFTCore Data Monitor.

SIFTView. Provides the means for operators to interact with the SIFT system (Figure 2). One or more instances of SIFTView can be running simultaneously. Basic functionality includes: seismic event handling, display of the offshore forecast and coastal estimates. Future functionality will include sea-level data display, management of inversion from deep-ocean buoy measurements, inundation forecasts, and presentation of the Later Waves Forecast.

SIFTNode. Coordinates the computational needs of SIFT. Any number of SIFTNodes can be run simultaneously and one instance of SIFTNode per CPU will provide optimal system performance. SIFTNode is designed so that only one Job can run at a time. For faster throughput more SIFTNodes can be added, as additional CPUs are made available. The types of Jobs coordinated by SIFTNode include:

- FirstEstimate. Using a single Unit Source nearest to the earthquake epicenter, FirstEstimate prepares fields of maximum wave height and travel time extracted from the database using the PropDBService.
- CoastalForecast. A coastal forecast is computed using all Unit Sources for each warning point that is selected from the relational database. Modeled time series at each warning point is computed by adding each unit source's result multiplied by the slip for that source.
- Inversion. Using time series computed from the PropDBService and actual sea-level data from deep-water buoy measurements, an improved slip distribution for the unit sources is computed.
- LaterWaves. The LaterWaves node computes the estimated time period and amplitude envelope of the full tsunami wave train based on actual and predicted time series at a coastal station. This forecast is computed after the arrival of the initial tsunami wave.

Several Jini Services have been developed to provide needed additional services to the SIFT system.

PropDBService. Accesses data from the multi-terabyte database of open ocean tsunami propagation results. The database consists of a file for each unit source. The PropDBService, is collocated with the actual files to minimize network communication. This service sub-samples the data files and combines multiple unit sources within the service, transmitting the combined data to the client.

DataService. Coordinates the ingestion of sea-level data from the SIFTCore data monitor with the sea-level database and SIFTView modules. In addition, the service detides the data as it is acquired and provides a sea-level data cache for quick response to requests for data from SIFTView.

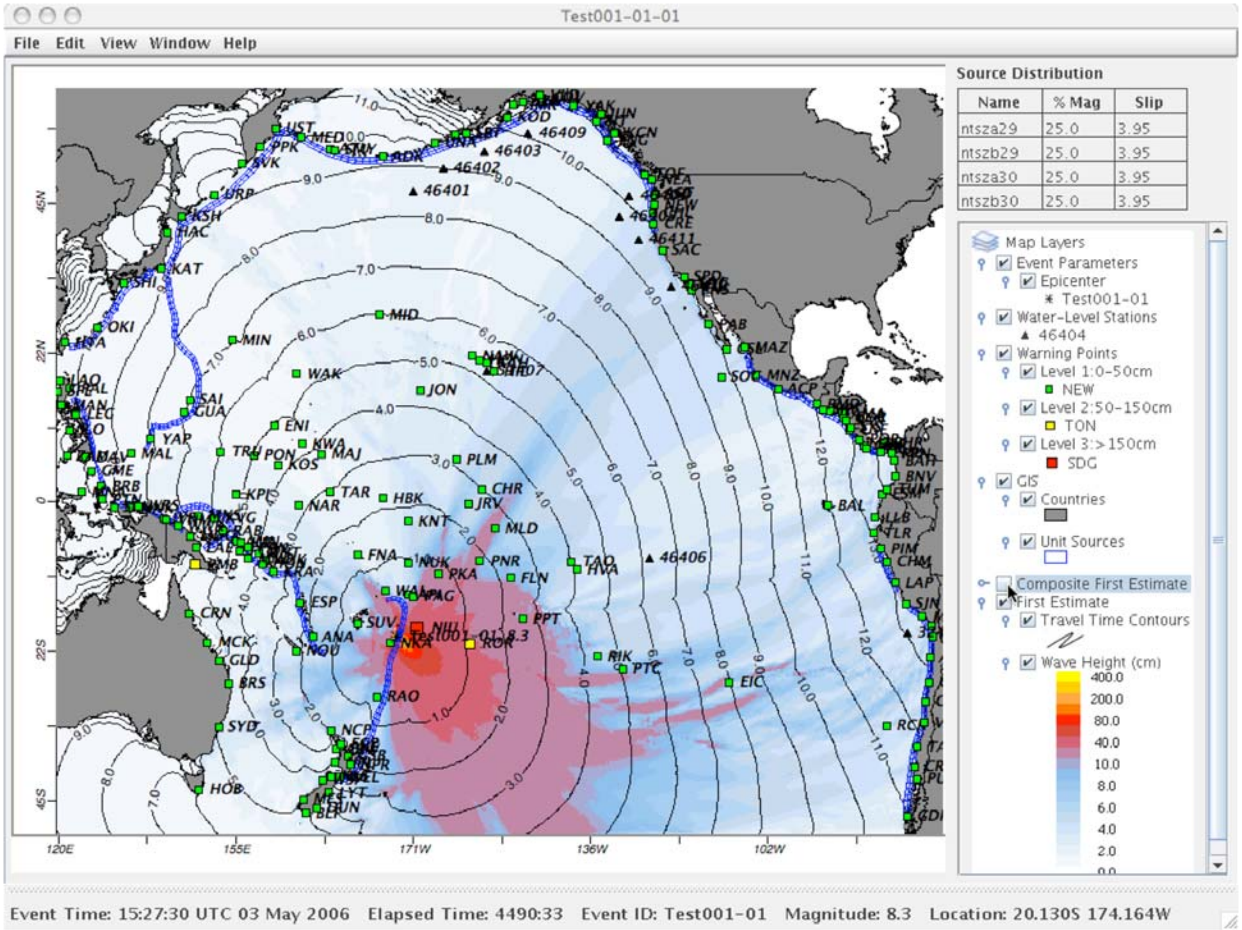

Figure 2. SIFTView, showing a test seismic event at 20.1S 174.2W and magnitude Mw=8.3.

ResultService. The results from the SIFTNode calculations can become unwieldy due to size. The ResultService takes those results and, rather than storing them in the database, creates a netCDF file locally. The data reference is then passed in place of the actual data to SIFTCore. When SIFTView accesses the data reference, the ResultService will read the netCDF data file (instead of querying the database) and send the data to SIFTView. The ResultService caches some results to speed up this process.

3. IMPLEMENTATION

Standardized software development practices were followed in creating SIFT. In developing operational software it is important that the software be robust. Robustness can be more easily be achieved by clearly tracking the software development process. In the development of SIFT we use:

- JBuilder IDE for Java development.
- Follow Java coding standards.
- Unit testing using JUnit.
- Configuration and software source code control with Subversion.
- Bug and issue tracking with Mantis, a php/MySQL/web based bug-tracking system.

4. FUTURE DIRECTIONS

SIFT is presently at version 1.5. In future releases, real-time deep-water buoy measurements will be assimilated with model results to provide a more accurate offshore solution, inundation models will be added to provide estimates of onshore amplitudes, and later waves will be predicted based on the arrival time and amplitude of the initial tsunami wave. More information about the NOAA Center for Tsunami Research and this development program can be obtained at http://nctr.pmel.noaa.gov/.

Acknowledgment. This publication is funded by the Joint Institute for the Study of the Atmosphere and Ocean (JISAO) under NOAA Cooperative Agreement #NA17RJ1232, Contribution #1364. PMEL contribution 3020. The views expressed herein are those of the author(s) and do not necessarily reflect the views of NOAA or any of its sub agencies