ENHANCING SEISMIC CALIBRATION RESEARCH THROUGH SOFTWARE AUTOMATION AND
SCIENTIFIC INFORMATION MANAGEMENT

Annie B. Elliott, Douglas A. Dodge, Michael D. Ganzberger, Teresa F. Hauk, Eric M. Matzel, and
Stanley D. Ruppert

Lawrence Livermore National Laboratory

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Development
Office of Defense Nuclear Nonproliferation

Contract No. W-7405-ENG-48

ABSTRACT

The National Nuclear Security Administration (NNSA) Ground-Based Nuclear Explosion Monitoring Research and Engineering (GNEM R&E) Program has automated significant portions of the processes of both seismic data collection and processing, and of determining seismic calibrations and performing scientific data integration by developing state-of-the-art tools. We present an overview of our software automation and scientific data management efforts and discuss frameworks to address the problematic issues of very large datasets and varied formats utilized during seismic calibration research. The software and scientific automation initiatives directly support the rapid collection of raw and contextual seismic data used in research, provide efficient graphics-intensive and user-friendly research tools to measure and analyze data, and provide a framework for research dataset integration. The automation also improves the researcher’s ability to assemble quality-controlled research products for delivery into the NNSA Knowledge Base (KB). The software and scientific automation tasks provide the robust foundation upon which the synergistic and efficient development of GNEM R&E Program seismic calibration research may be built.

The task of constructing many seismic calibration products is labor intensive and complex, hence expensive. However, certain aspects of calibration product construction are susceptible to automation and future economies. We are applying software and scientific automation to problems within two distinct phases or “tiers” of the seismic calibration process. The first tier involves initial collection of waveform and parameter (bulletin) data that comprise the “raw materials” from which signal travel-time and amplitude correction surfaces are derived, and is highly suited for software automation. The second tier in seismic research content development activities, which we focus on in this paper, includes development of correction surfaces and other calibrations. This second tier is less susceptible to complete automation, more complex and in need of sophisticated interfaces, as these activities require the judgment of scientists skilled in the interpretation of often highly unpredictable event observations. Even partial automation of this second tier, through development of tools to extract observations and make many thousands of scientific measurements, has significantly increased the efficiency of the scientists who construct and validate integrated calibration surfaces. This achieved gain in efficiency and quality control is likely to continue and even accelerate through continued application of information science and scientific automation.

Data volume and calibration research requirements have increased by several orders of magnitude over the past decade. Whereas it was possible for individual researchers to download individual waveforms and make time-consuming measurements event by event in the past, with the terabytes of data available today, a software automation framework must exist to efficiently populate and deliver quality data to the researcher. This framework must also simultaneously provide the researcher with robust measurement and analysis tools that can handle and extract groups of events effectively and isolate the researcher from the now onerous task of database management and metadata collection that is necessary for validation and error analysis. Lack of information management robustness or loss of metadata can lead to incorrect calibration results in addition to increasing the data management burden. To address these issues we have succeeded in automating several aspects of collection, parsing, reconciliation and extraction tasks, individually. We present several software automation tools that have resulted in demonstrated gains in efficiency of producing scientific data products.
OBJECTIVES

The NNSA GNEM R&E Program has made significant progress enhancing the process of deriving seismic calibrations and performing scientific integration with automation tools. We present an overview of our software automation efforts and framework to address the problematic issues of improving the workflow and processing pipeline for seismic calibration products, including the design and use of state-of-the-art interfaces and database-centric collaborative infrastructures. These tools must be robust, intuitive, and reduce errors in the research process. This scientific automation engineering and research will provide the robust hardware, software, and data infrastructure foundation for synergistic GNEM R&E Program calibration efforts. The current task of constructing many seismic calibration products is labor intensive and complex, expensive and error prone. The volume of data and calibration research requirements have increased by several orders of magnitude over the past decade. The increase in quantity of data available for seismic research over the last two years has created new problems in seismic research; data quality issues are hard to track given the vast quantities of data, and this quality information is readily lost if not properly tracked in a manner that supports collaborative research. We have succeeded in automating many of the collection, parsing, reconciliation and extraction tasks individually. Several software automation tools have also been produced and have resulted in demonstrated gains in efficiency of producing derived scientific data products. In order to fully exploit voluminous real-time data sources and support new requirements for time-critical modeling, simulation, and analysis, continued expanded efforts to provide scalable and extensible computational framework will be required.

RESEARCH ACCOMPLISHED

The primary objective of the Scientific Automation Software Framework (SASF) efforts are to facilitate development of information products for the GNEM R&E regionalization program. The SASF provides efficient access to, and organization of, large volumes of raw and derived parameters, while also providing the framework to store, organize and disseminate derived information products for delivery into the NNSA KB.

These next generation information management and scientific automation tools are used together within specific seismic calibration processes to support production of tuning parameters for the United States National Data Center run by the Air Force (Figure 1). The calibration processes themselves appear linear beginning with data acquisition (Figure 1) extending through reconciliation, integration, measurement and simulation through to the construction of calibration and run-time parameter products. However, efficient production of calibration products requires extensive synergy and synthesis not only between large datasets and a vast array of data types (Figure 1), but also between measurements and results derived from the different calibration technologies (e.g., location, identification, and detection) (Figure 1). This synergy and synthesis between complex tools and very large datasets is critically dependent on having a scalable and extensible unifying framework. These requirements of handling large datasets in diverse formats and facilitating interaction and data exchange between tools supporting different calibration technologies has led to an extensive scientific automation software engineering effort to develop an object oriented database-centric framework (Figure 3), using proven research driven workflows and excellent graphics technologies as an unifying foundation.

The current framework supports integration, synthesis, and validation of the various different information types and formats required by each of the seismic calibration technologies (Figure 1). For example, the seismic location technology requires parameter data (site locations, bulletins), time-series data (waveforms), and produces parameter measurements in the form of arrivals, gridded geospatially registered corrections surfaces and uncertainty surfaces. Our automation efforts have been largely focused on research support tools, RBAP (Regional Body-wave Amplitude Processor) and KBALAP (Knowledge Base Automated Location Assessment and Prioritization). Further, increased data availability and research requirements have driven the need for multiple researchers to work together on a broad area, asynchronously. Interim results and a complete set of working parameters must be available to all research teams throughout the entire processing pipeline. Finally, our development staff has continually and efficiently leveraged our proprietary Java code library, achieving 45% code reuse (in lines of code) throughout several thousand Java classes.

968
Figure 1: Summary of the processes of data collection, research and integration within the LLNL calibration process that result in contributions to the NNSA KB. The relationships of the current LLNL calibration tools, scientific automation tools, and database coordination framework to those involved in the assembly of the NNSA KB or within the Air Force Technical Applications Center (AFTAC) operational pipeline are delineated.

Database-Centric Coordination Framework

As part of our effort to improve our efficiency we have realized the need to allow researchers to easily share their results with one another. For example, as the location group produces GT information, that information should become available for other researchers to use. Similarly, phase arrival picks made by any qualified user should also become immediately available for others to use. This concept extends to sharing of information about data quality. It should not be necessary for multiple researchers to have to repeatedly reject the same bad data, or worse, miss rejecting bad data. Rather, once data are rejected because of quality reasons, they should automatically be excluded from processing by all tools. We are implementing this system behavior using database tables, triggers, stored procedures and application logic. Although we are at the beginning of this implementation, we have made significant progress over the last year with several kinds of information sharing using the new database-centric coordination framework. These are discussed below.

Significant software engineering and development efforts have been applied successfully to construct an object-oriented database framework that provides database-centric coordination between scientific tools, users, and data. A core capability this new framework provides is information exchange and management between different specific calibration technologies and their associated automation tools such as seismic location (e.g. KBALAP), seismic identification (e.g. RBAP), and data acquisition and validation (e.g. KBITS). A relational database (Oracle) provides the current framework for organizing parameters key to the calibration process from both Tier 1 (raw parameters such as waveforms, station metadata, bulletins etc) and Tier 2 products (derived measurements such as ground-truth, amplitude measurements, calibration and uncertainty surfaces etc.). Efforts are underway to augment the current relational database structure with structured queries based on semantic graph theory for handling complex queries. Seismic calibration technologies (location, identification, etc.) are connected to parameters stored in the relational database by an extensive object-oriented multi-technology software framework that includes elements of schema design, PLSQL (extension to Oracle), real-time transactional database triggers, constraints, as well as coupled Java
and C++ software libraries to handle the information interchange and validation requirements. This software framework provides the foundation upon which current and future seismic calibration tools may be based.

Sharing of Derived Event Parameters

We have long recognized the inadequacies of the CSS3.0 origin table to serve as a source of information about the “best” parameters for an event. One origin solution may have the best epicenter but poor information on other parameters.

Another may have the correct event type, but be poor in other respects, and so on. We have discussed producing origin table entries with our organization as the author, but that approach has difficulties. Different groups would have responsibility for different fields in the origin. Because their information would not be produced in synchronization, we would either have to always be updating the preferred origin or else producing new preferred origins. Also, there would be difficulties in tracking the metadata associated with each field of the preferred origin. Our solution was to create a set of new tables and associated stored procedures and triggers that collectively maintain the “best” information about events.

In order to calibrate seismic monitoring stations, the LLNL Seismic Research Database (SRDB) must incorporate and organize the following categories of primary and derived measurements, data and metadata:

**Tier 1: Contextual and Raw Data**
- Station Parameters and Instrument Responses
- Global and Regional Earthquake Catalogs
- Selected Calibration Events
- Event Waveform Data
- Geologic/Geophysical Data Sets
- Geophysical Background Model

**Tier 2: Measurements and Research Results**
- Phase Picks
- Travel-time and Velocity Models
- Rayleigh and Love Surface Wave Group Velocity Measurements
- Phase Amplitude Measurements and Magnitude Calibrations
- Detection and Discrimination Parameters

Automating Tier 1

Corrections and parameters distilled from the calibration database provide needed contributions to the NNSA KB for the Middle East, North Africa and Western Eurasia region and will improve capabilities for underground nuclear explosion monitoring. The contributions support critical functions in detection, location, feature extraction, discrimination, and analyst review. Within the major process categories (data acquisition, reconciliation and integration, calibration research, product distillation) are many labor intensive and complex steps. The previous bottleneck in the calibration process was in the reconciliation and integration step. This bottleneck became acute in 1998 and the KBITS suite of automated parsing, reconciliation, and integration tools for both waveforms and bulletins (ORLOADER, DDLOAD, UpdateMrg) were developed. The KBITS suite provided the additional capability required to integrate data from many datasources and external collaborations. Data volumes grew from the 11,400 events with 1 million waveforms in 1998 to the 6 million events with 70 million segmented waveforms and terabytes of continuous data today (e.g. Ruppert et al.; 1999, Ruppert et al. 2005). This rapid increase in stored parameters soon led to two new bottlenecks hindering rapid development and delivery of calibration research.

Automating Tier 2

As the number of data sources required for calibration have increased in number and source location, it has become clear that the manual, labor intensive process of humans transferring thousands of files and unmanageable metadata cannot keep the KBITS software fed with data to integrate, nor could the seismic researcher consistently find, retrieve, validate, or analyze the raw parameters necessary to effectively produce seismic calibrations in an efficient
manner. Significant software engineering and development efforts were applied to address this critical need to produce software aids for the seismic researcher. Thus, our development efforts are focused on the development of two scientific automation tools, RBAP and KBALAP, for seismic location and seismic identification calibration tasks, respectively.

Both of these tools include methods and aids for efficiently extracting groups of events and waveforms from the millions contained in the SRDB, and for making large numbers of measurements with metadata in a batch mode. The concept of event sets (groups of related seismic events or parameters that can be processed together, e.g. either station-centric or event-centric) was introduced, as previous seismic analysis code (SAC) scripts and macros could not scale to the task.

All analysis results go directly into the LLNL production schema where they become available for other users. Because users of KBALAP and RBAP may be able to write to our core tables, these tools implement a rule system that uses database roles to control which users can modify data, and which users can modify other users data or modify bulletin data. For example, some users may be able to rank picks, but not save new origin solutions. Some users may be able to perform an array analysis, but not rank picks, etc. By this means we are able to support use of RBAP and KBALAP by analysts with different skill sets and different research priorities. Users get the convenience of being able to produce the results they want and have them immediately available in the production schema without worrying about the impact of their work on a different research group.

The RBAP Program

The Regional Body-wave Amplitude Processor (RBAP) is a station-centric Tier 2 automation tool; it is an interactive, graphical (Figure 2) and highly specifiable software program that acts as a picker and a magnitude and distance amplitude corrections (MDAC) calculator. RBAP helps to automate the process of making amplitude measurements of regional seismic phases for the purpose of calibrating seismic discriminants at each station. RBAP generates station-centric raw, and MDAC-corrected Pn, Pg, Sn and Lg amplitudes along with their associated calibration parameters (e.g. phase windows, MDAC values, reference events, etc.) in database tables. It strictly follows standardized MDAC processing, and it replaces the original collection of LLNL scripts described by Rodgers (2003). RBAP has a number of advantages over the previous scripts. It is much faster, significantly easier to use, allows for collaboration, scales more easily to a larger number of events and permits efficient project revision and updating through the database.

RBAP integrates the functions of the modules in the previous LLNL scripts into a single, unifying program that is designed to both perform the amplitude measurement task efficiently and to require a minimum effort from the users for managing their data and measurements. For well-located events with pre-existing analyst phase picks, the user reviews for quality control and then generates all the amplitudes with just a few mouse clicks. For events needing more attention, the user has complete control over the process (e.g. window control, ability to mark bad data, define regions, define MDAC parameters and define the events to be used in the overall calibration process). RBAP shortens the time required for the researcher to calibrate each station while simultaneously allowing an increase the number of events that can be efficiently included. RBAP is fully integrated with the LLNL research database. Data is always read directly from the appropriate tables in the research database rather than from a snapshot as was done in the previous system. All RBAP result tables have integrity constraints on the columns with dependencies on data in the LLNL research database. This design makes it very difficult for results produced by RBAP to be stale and also ensures that as the research database expands, RBAP automatically becomes aware of new data that should be processed, as well as data marked as unusable by other applications (such as KBALAP), which should no longer be used for processing. In a like manner, when a segment is marked as bad in RBAP, it is excluded from further processing in KBALAP.

RBAP projects are station-centric; stations can be either single stations or arrays, where arrays focus on a reference element. Each project also specifies one or more regions, which can be simple rings or user-defined polygons; each region may be assigned its own velocity model. Once defined, concepts such as geographic regions are available to other researchers and other projects; interfaces include extensive use of modern mapping technologies and data tables the design of which are driven by research workflows. RBAP makes use of the data type manager concept extensively, and includes separate managers for velocity models, regions and events. Events are shown color-coded on a map for ease of use. RBAP also includes a graphical phase picker that generates windows automatically for the
Pn, Pg, Sn and Lg phases using times predicted by the velocity model. The picker is geared towards using signal-to-noise ratios for regional body wave amplitude measurements, and picks are automatically advanced according to applied velocity models.

Figure 2: A sample of some of the RBAP graphical and data manager windows.

Some key features of RBAP are listed below:

• Based on WG 2 Standardized Algorithm
  - RBAP is built on the standardized MDAC body-wave amplitude measurement algorithms. Its results are completely consistent with the last version of the LLNL scripts (Rodgers, 2003) that were vetted in the February 2003 exercise between LLNL, LANL, and AFTAC.

• Fast and Efficient Calibration
  - RBAP is self-contained and optimized for station-centric body-wave processing. “Good” events can be handled with just few mouse clicks. The researcher has direct control over key calibration parameters within the tool such as phase amplitude windows and migration, marking bad segments, defining distinct geophysical regions, event types to process, etc. We expect RBAP, when fully developed, to provide roughly a factor of 5 increase in calibration speed compared with the original scripts, enabling us to calibrate more stations, with more events per station.
• **Project Management**
  - RBAP is designed so that a calibration project can be put down for a day, month or a year, and easily picked up, by the same researcher or a new one. All processing metadata is saved and events are easily tracked as processed, unprocessed or outside the current project definitions. This allows a researcher to efficiently work through a huge data list without repetition and to easily identify and incorporate new events as they become available in the database.

• **Utilizes Database for Up-to-Date Results**
  - RBAP can draw on the latest calibration parameters being generated by other working groups, such as the most recent phase picks, relocations, magnitudes, instrument response information, or event type ground truth.

• **Batch Processing**
  - RBAP is designed to allow simple batch updating of the amplitude results, whether the change is small (e.g. one-event is relocated) or large (instrument response is changed affecting all events).

• **Engenders Collaboration, Consistency and Efficiency**
  - RBAP’s complete database integration allows multiple researchers to access the finest-grained tuning parameters for all projects; no data is lost in collaboration, and parameters may be reused.

**The KBALAP Program**

The KBALAP program is another Tier 2, event-centric automation effort in the GNEM program. It is a highly interactive, graphical tool (Figure 3) which uses a set of database services and a client application based on data selection profiles that combine to efficiently produce location ground truth data which can be used in the production of travel time correction surfaces, and as part of the preferred event parameters used by other tools in our processing framework.

KBALAP’s database services are responsible for evaluating bulletin and pick information as it enters the system to identify origin solutions that meet pre-defined ground-truth criteria with no further processing, and for identifying events that would likely meet a predefined ground truth level if a new origin solution was produced using available arrivals. The database service is also responsible for identifying events that should have a high priority for picking based on their existing arrival distribution, and the availability of waveform data for stations at critical azimuths and distances.

The interactive portion of KBALAP has the following principal functions:

- production of GT origins through prioritized picking and location,
- specification of GT-levels for epicenter, depth, origin time, event type,
- batch-mode location of externally-produced GT information,
- production of array azimuth-slowness calibration data, and
- easy review and modification of event parameters used by all GNEM researchers.

Users of KBALAP are able to easily search for data relevant to the production of GT and filter the results by processing status, GT level or potential GT level. The user can select any GT or potential GT event and observe the distribution of stations with picks and stations which have available waveforms. The tool can indicate whether a selected event has the potential to become a GT event if appropriate picks were made on available waveforms that currently have no suitable picks. The user can also select any station with available waveforms and open a picker with any current picks displayed, and adjust existing picks, add new picks, mark bulletin picks as unusable, and relocate the event. When a new GT level is calculated, the user can choose to accept that origin solution and GT level, or continue working with other stations. Traces marked as unusable in KBALAP are automatically viewed by RBAP as bad, and thus not used in processing in that program as well.

With a single mouse click, the user can open a selected event for review and further analysis. In this review mode the user can review and rank existing picks, calculate new origin solutions, and, if appropriate, produce calibration origins. At any point in this process, the user can see the current spatial distribution of arrivals and stations with
waveforms. The tool can also guide the user toward analyzing stations that are important to achieving an origin solution with the best possible GT level.

The interactive GT entry mode of KBALAP allows the user to retrieve information about a specific event and add or update that event’s GT parameters. The program can also create a new event with a GT level for cases where epicenter, time, depth and magnitude GT data are available. Similarly, KBALP’s batch mode allows for the specification of flat files containing GT data for events already in the database. KBALAP’s research driven interface design includes dedicated graphical user interfaces (GUIs) for station filtering, event selection and single and multi-station phase-onset pick windows. Further, there are GUIs that allow users to specify, store and apply different types of bandpass filters. Finally, there is a waveform pick editor window, as well as multiple pick “views,” including band filtered views for low signal-to-noise problems and filtering by attributes such as analyst or load date.

Figure 3: A sample of some of the KBALAP graphical and data manager windows.

Some key KBALAP features are listed below:

- **Fast and Efficient Location**
  - KBALAP data selection profiles are self-contained and optimized for the event-centric task of location. KBALAP displays all available picks with available waveforms, and allows picks with waveforms to be modified.

- **Project Management and Collaboration**
  - KBALAP is designed so that a profile can be put down for a day, month or a year, and easily picked up by the same researcher or a new one. All processing metadata is saved, and events are easily tracked as processed, unprocessed or outside the current project definitions. This allows researchers or research teams to efficiently work through a huge data list without repetition and to easily identify and locate new events as they become available in the database. Once a senior researcher has reviewed a profile’s picks, these picks are finalized and the arrivals and associated metadata made available to other researchers and tools.

- **Batch Processing**
  - KBALAP is designed to allow simple batch loading of externally produced GT information.
Further Enhancements to Efficiency Through Cluster-Based Computing

We have begun to leverage scalable and reconfigurable cluster computing resources to improve the efficiency of our computational infrastructure. Just as the database-centric approach to information management provided important gains in efficiency, we have realized the need to move to a different computational paradigm to provide the computational power necessary during calibration production and research. We have begun developing a set of flexible and extensible tools with platform independence that are parallelizable. These research tools will provide an efficient data processing environment for all stages of the calibration workflow, from data acquisition through making measurements to calibration surface preparation. We are also scheduled to implement Oracle 10g’s clustering capabilities to further push the performance envelope for our production database. This scalable and extensible approach will result in more coupled and dynamic work flow in contrast to the linear work flow of the past, and allow more interaction between data, model creation, and validation processes.

Initial development and modification of existing codes and algorithms of the cluster based computing environment has yielded significant efficiency improvements in RBAP and other measurement tools. Modification of RBAP to incorporate threads to isolate computationally intensive operations has provided a more interactive and responsive environment for the researcher, as well as laid the ground work for moving the threads to cluster-based computing resources. Other areas under investigation that leverage cluster resources are waveform correlation and subspace detector operations, as well as large-scale event relocations to support the evaluation of ground truth and model calibrations.

CONCLUSIONS AND RECOMMENDATIONS

We present an overview of our software automation efforts and framework to address the problematic issues of consistent handling of the increasing volume of data, collaborative research efforts and researcher efficiency, and overall reduction of potential errors in the research process. By combining research driven interfaces and workflows with graphics technologies and a database-centric information management system coupled with scalable and extensible cluster based computing, we have begun to leverage a high performance computational framework to provide increased calibration capability. These new software and scientific automation initiatives will directly support our current mission including rapid collection of raw and contextual seismic data used in research, provide efficient interfaces for researchers to measure and analyze data, and provide a framework for research dataset integration. The initiatives will improve time-critical data assimilation and coupled modeling and simulation capabilities necessary to efficiently complete seismic calibration tasks. This scientific automation engineering and research will provide the robust hardware, software, and data infrastructure foundation for synergistic GNEM R&E Program calibration efforts.

ACKNOWLEDGEMENTS

We acknowledge the assistance of the LLNL computer support unit in implementing and managing our computational infrastructure. We thank Jennifer Aquilino, John Hernandez and Laura Long for their assistance in configuration and installation of our Linux cluster and workstations.

REFERENCES


PGL SERVER: DEVELOPMENT OF A STAND-ALONE SERVER-BASED EARTH-MODEL LIBRARY FOR SEISMIC MONITORING

Lee A. Jensen, James R. Hipp, Rose Q. Villanueva, Daniel S. Myers, Breanna Ammons, and Jack H. Gauthier

Sandia National Laboratories

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Development
Office of Defense Nuclear Nonproliferation

Contract No. DE-AC-04-94-AL85000

ABSTRACT

Sandia National Laboratories is developing a server version of the Parametric Grid Library (PGL), a shared-object library that allows storing and accessing Earth models for the National Nuclear Security Administration (NNSA) Knowledge Base. Creation of a server version of the PGL—PGL Server—allows several improvements when compared with the existing version of the PGL, which bundles an Earth model directly with an application.

1. Computational efficiency. From a practical viewpoint, currently when an application calls or starts up (i.e., instantiates) the PGL, there is a significant startup delay while the PGL reads in the Earth-model data. This delay can occur frequently if the application is not always in use. The PGL Server has the advantage of always having the Earth-model data available, thus there is no delay. However, the PGL Server does accrue delays from message passing.

2. Platform independence. A server-based Earth model allows applications to talk to the server across a network; thus the Earth model and the applications can run in a heterogeneous computing environment.

3. Architectural flexibility. A server-based Earth model can support many architectures, including client-server architectures, service-oriented architectures, and distributed-objects architectures. A server-based Earth model can also support parallelism, either with a single server for multiple applications or multiple servers for one or more applications.

Several issues are still being addressed with the PGL Server. These issues include assessing network communication performance between applications and the PGL Server, managing application state in order to manage simultaneous requests from multiple applications, and evaluating parallel distributed architectures that allow the PGL Servers to scale for large problems and large numbers of applications. Architectural issues are also being addressed, including the most appropriate service level for an Earth model. The PGL Server used at its lowest level (atomic components) can require many small messages to be passed. But defining a server at a higher interface level (e.g., the level where event-locations are calculated) might require only a few large messages (a server-based event locator would have the Earth model instantiated with it). Because of the message-passing overhead, a higher-level server could provide performance improvements in many circumstances.
OBJECTIVE

We are developing a server version of the PGL called the PGL Server. The PGL is a shared-object library used in seismic analysis and seismic monitoring to store, access, and manipulate geophysical data representing the Earth. With the PGL Server, we want to improve a delay problem when the PGL is instantiated. In addition, we want to use the PGL Server to investigate advanced computing architectures that might be used in future seismic-monitoring systems.

RESEARCH ACCOMPLISHED

Background

The PGL is a shared-object library used to store, access, and manipulate geophysical data representing the Earth—spatial dimensions (lat, lon, and depth), density, travel time of seismic waves, the difference between observed and actual travel times, etc (Hipp et al., 2005). These “Earth models”—the geophysical datasets—are used in seismic monitoring to identify, locate, and measure the yield of nuclear explosions. There are many Earth models and they are often updated.

The PGL can be used in research to build Earth models and in operations to monitor seismicity in near real time. It works with many Earth models, and it incorporates features that allow it to be readily modified. It is used by many applications, including tools to create and modify Earth models (e.g., KBCIT) and seismic-event locators (e.g., LocOO). It is an object-oriented program, written in C++, designed to be flexible, modifiable, and computationally efficient. The PGL contains approximately 250,000 lines of code, of which approximately 150,000 lines are source code. It has almost 200 classes, with several thousand public methods.

Currently, an application must instantiate the PGL directly. This situation requires that the PGL be on the same machine as the application, and that the application be in the same language as the PGL (although applications written in Java can also access the PGL through special Java Native Interface calls). This situation also typically necessitates a lengthy startup delay while the PGL reads the Earth model dataset and constructs the appropriate objects. This “startup problem” can occur frequently if the application uses the PGL in a repetitive start/stop execution cycle.

The PGL Server was initially conceived as a means of circumventing the startup problem. A PGL Server runs all of the time; thus, the Earth model is always loaded and the objects always instantiated. An application—or several applications—can access the PGL immediately, whether the application is in constant use or not. Figure 1 shows the fundamentally different way that the PGL and the PGL Server can be used.

![Figure 1. Comparison of current the PGL implementation with a simple the PGL Server implementation.](image)

Although the initial impetus for a server version of the PGL was the startup problem, the PGL Server also offers the opportunity of investigating more advanced architectures that could be used in a seismic-monitoring system. Server versions of software products are seeing widespread use in industry, for example enterprise databases, which are often used in seismic monitoring, are typically server-based products. Architectures built on server-based products
(e.g., service-oriented architectures [SOA] and web services [SOA over the Internet]), are of increasing interest (Binstock, 2005; Reuters, 2006). These advanced architectures are being developed because they offer advantages in cost, reliability, upgradeability, and performance. As a component within an advanced architecture, the PGL Server could offer similar benefits to seismic monitoring.

1. Cost: the PGL Server can run on a variety of platforms, allowing use of commercial-off-the-shelf (COTS) hardware. Also, applications can be located on the same or different machines. The ability to operate in a heterogeneous computing environment not only allows reduction in capital costs, but protects against obsolescence.

2. Availability, reliability, and maintainability. Multiple PGL Servers could be implemented redundantly so that if one fails, the job can be completed by another. Such a configuration could also allow a machine running the PGL Server to be disconnected from the network without failure of the system, thus simplifying maintenance.

3. Upgradeability: Legacy code is difficult to upgrade for two reasons. First, it usually requires single-supplier hardware. Second, it usually requires a transition that cannot be easily performed within the existing system. The PGL Server is platform independent. And multiple PGL Servers can be readily deployed, including upgraded versions, so that transitions can be done easily.

4. Design flexibility: As a stand-alone component with access through an application programming interface (API), the PGL Server can fit into many system designs, including client-server, distributed-objects, and SOA. This flexibility is important because many applications have need of an Earth model, and these applications might have different design requirements; therefore, the PGL Server would not interfere with design decisions. In this manner the PGL Server also allows ease of upgrading.

5. Data and parameter control: The separate, stand-alone nature of the PGL Server allows different applications to have the same or different datasets or parameters.

6. Performance. Not only could the PGL Server fix the startup problem, but increased speedup and higher throughput could be achieved by implementing multiple distributed PGL Servers or by having a PGL Server running on a faster machine. Different configurations are possible, as discussed later.

Several issues need to be addressed before an efficient and effective PGL Server can be realized. These issues include understanding (1) the design of a PGL Server, (2) the delays incurred from message passing, (3) how a PGL Server should keep track of multiple client applications, (4) what is the most appropriate service level for an Earth model, and (5) how to best implement parallelism. These issues are interrelated. We have conducted and continue to conduct experiments to examine these issues. This paper presents our work and conclusions.

**Basic Design of the PGL Server**

The PGL Server is a stand-alone, continuously running program. It instantiates one or more Earth models from File Databases (FDB) as specified by clients accessing the models. In addition to the objects instantiated from the FDB, the PGL Server also contains a set of stub classes and methods that match one-to-one with PGL classes and methods. These stubs contain the processes necessary to create, send, receive, and decipher messages. In the case of the version of the PGL Server that we are currently using, the messages are in eXtensible Markup Language (XML) and the message passing is handled by XML-Remote Procedure Call (XML-RPC). The PGL Server is written in C++ and uses an XML-RPC library for C++.

XML-RPC was chosen as the message-passing protocol for this work after we performed a survey of existing distributed-system software. The XML-RPC protocol makes use of two well-established technologies: XML and HTTP. XML-RPC has several desirable properties. In particular, the protocol is lightweight and is specifically designed to emphasize compatibility between different platforms and programming languages. This is important, since we have many Java applications that need to communicate with the C++ PGL library.
When a request is generated, XML-RPC parses the relevant information—such as the name of the procedure, argument types, and argument values—and generates an XML document with all of the necessary information. This document is transmitted to the server using the HTTP protocol. After the server receives the message, it parses the XML document to reconstruct all the information it needs to complete the request. In order to handle complex objects, the PGL Server creates an object map to retain the information on any instantiated object. In this map, the PGL Server saves the object and a unique identifier as its key. This key, in most situations, is provided by the client. Usually the client is the one creating objects (e.g., it can use Java’s toString() method to create a unique key), which it then sends to the server for the object map. This work is all done in the stub classes, so use of XML-RPC is completely transparent to the user, and the user interface to PGL is almost, if not exactly, the same.

A client using the PGL Server can run on the same computer or on a different computer. It can be written in any computer language that has support for the same message passing protocol as the PGL Server. Among other languages, XML-RPC has libraries for C/C++, Java, Lisp, and a number of scripting languages including Perl, Python, PHP, and Tcl (XML-RPC, 2003; Apache, 2006). The client must have a set of stub classes and methods that represent the PGL classes and methods that it uses. These stubs are similar to the PGL Server stubs; they contain the processes to create, send, receive, and decipher messages.

Currently, a version of the PGL Server exists that runs on Sun servers with the Solaris operating system, and clients have been written that run on Sun servers, Linux servers, and personal computers.

Figure 2 contains an illustration of how a PGL Server interacts with the PGL and with client stubs.

---

**Figure 2. Basic design of the PGL Server.**

Because the PGL has over 5,000 methods, programming all of the necessary stubs is a daunting task, compounded by the prospect of making stubs for different interfaces (e.g., XML-RPC, sockets, and Java Native Interface [JNI]). The PGL also has several Java-based clients, which interface to PGL in a manner similar to that shown in Figure 2, but using Java Native Interface. For the PGL Server project, we also developed a method for automatically generating the PGL interface stubs. Automatic stub generation consists of three parts: Doxygen (an open-source documentation tool) (van Heesch, 2006), a specialized parser, and the generating programs themselves.

First, the Doxygen program scans each of the PGL source files and produces an XML document summarizing the information in each class: methods, members, arguments, etc. Next, this XML document is processed by a specialized parser that examines each XML document and produces a tree data structure containing the same information as the raw XML, but in a more convenient form. Finally, the tree is processed using a standard pre-order traversal. As it encounters each new class, the code-generating program creates a new source file, and applies its rules to create XML-RPC interfaces to each of the PGL functions.
As mentioned previously, the XML-RPC protocol has no innate method of dealing with complex objects. The PGL library is object-based, so this is an obstacle to fully implementing the new system. Our solution to this problem is to create all objects in pairs, with one object created on the server-side, and one on the client-side. The server-side object is the “real” copy that maintains its state and actually executes any RPC requests. The client-side object is only a stub, and is used only to supply an interface from the client to the server. The two objects are linked by a common handle, created during the construction process. When an RPC request involves an object, the client will collect the appropriate stub object and extract the handle linking it to a server-side object. The handle is sent with the other data as part of an XML-RPC request. When the server unpacks the handle, it uses a hash-table data structure to look up the corresponding “real” object. This server object performs any requested work, saves its state, and is then re-hashed until it is needed again. This approach does require more sophisticated methods to manage object life-cycles, so that objects are correctly discarded when their useful lives are over, but this functionality is already part of the PGL, so we do not have to manage object life-cycles explicitly.

Delays Incurred by Message Passing

One of the potential problems with the PGL Server (and indeed all server-based software and SOAs) is the overhead associated with message passing. We are interested in understanding the best method for implementing message passing (a lower-level technique, such as sockets, or a higher-level system, such as XML-RPC). Delays incurred by message passing can also be mitigated by choosing the appropriate level of service of an Earth model. This issue is discussed below.

Important variables in this problem are the format of the message, size of the message, the compression of the message, and the platforms that the client and PGL Server are running on. Preliminary work showed that binary (unformatted) messages without any compression are transferred the most efficiently. Binary messages are shorter and generally require less processing than formatted (ASCII or unicode) messages, although the communicating processes must know how to decipher the message. Our investigation confirmed fewer large messages are more efficient than numerous small messages. This preliminary work also showed that compression only helped with formatted messages, and then only with a low level of compression. Additional preliminary work also showed that Fast Infosets (Sandoz and Pericas-Geertsen, 2005), a binary XML implementation, show promise in speeding message passing under some circumstances.

For the work presented here, we programmed a client in Java that communicates with the PGL Server. The client is based on an original test code for the PGL. The client calls the same methods using the same PGL interface as the original, but it can be running on the same or a different machine. The Java client sends method parameters to the server via XML-RPC, and the C++ PGL Server sends responses back to the client in the same way. The major complication occurs when the original test code passes pointers to large datasets. Pointers have no meaning on a different machine, so what the pointers are pointing to - generally large data vectors - have to be passed back and forth.

To address this complication, the client calls the PGL stub classes (Figure 2). The purpose of these stubs is to execute the PGL method calls via XML-RPC, performing whatever manipulations are necessary to make communication between it and the server as simple as possible. An object map on the server side retains the information on any instantiated object. In this map, the server saves the object and a unique identifier as its key. This key is provided by the client, using Java’s toString() method, which it then sends to the server for the object map. This work is all done in the stub classes, so use of XML-RPC is completely transparent to the user, and the user interface to the PGL is almost if not exactly the same.

The first step of the performance testing is to determine the baseline performance. Of particular importance is the total amount of time spent doing XML-RPC-related processing. The baseline times are used below to adjust the the PGL Server times to determine the additional fraction of time spent on XML-RPC-related processing. To determine the baseline times, the original C++ test code, directly connected to the PGL, was timed performing operations on various sized data vectors. For the test, the PGL was actually exchanging six vectors of doubles of the indicated size between the client and server: two containing interpolatory coordinate information and four containing interpolation results vectors for some arbitrary attributes. The results of the baseline PGL performance testing are given in Figure 3.
The same tests were then conducted with the PGL Server and the Java client. The data messages were passed as uncompressed binary payloads, as the preliminary testing indicated that this type of message would give the best performance. Performance testing was conducted with the PGL Server running on a Sun server. The Java client communicating with the PGL Server was tested running on the same Sun server, a Linux server and a Windows XP workstation. The network connection to the Sun server was 100 Mbps.

The results of the testing are presented in Figure 4. The plot shows the fraction of XML-RPC-related processing time required for various message sizes. The metric shown on the Y-axis was calculated as the time it took to execute with the PGL Server minus the baseline time (Figure 3), normalized by the baseline time: \[
\frac{(\text{XML-RPC time}) - (\text{baseline time})}{(\text{baseline time})}.
\]

The result is the additional fraction of time spent doing XML-RPC related processing, such as parsing and serializing XML documents and sending these over the network.

As shown in Figure 4, in all cases XML-RPC related processes reduce performance of the PGL. This finding is especially true when dealing with short data-vector lengths, with the total run times taking more than twice as much time as the baseline. Note however, that the total run times for these short data-vector lengths are typically small fractions of a second. For larger amounts of data, the performance suffers by only 20 to 40 percent.
These results indicate that there is a penalty associated with using the PGL Server. However, the additional time needed for passing short data vectors is small in absolute terms, and the additional time needed for passing long data vectors is small in relative terms. And these delays are much less than the typical 30-second or more startup time for the PGL with a large Earth-model FDB.

How the PGL Server Should Keep Track of Multiple Client Applications

Server software should be able to handle multiple clients. Thus, the software should be able to save the state of each process for the clients that are being served. However, the PGL was not designed with this specific capability as it was originally intended to be bound to a single client for its entire execution lifetime. In the future, however, we plan to modify the PGL Server to support a message queue that tags each client’s requests with a unique identifier. The PGL Server will then use the identifier with an object map to determine various “state” settings associated with a particular client. With this approach the PGL Server will be able to change its state settings, process a request, and store the results for each client that it is serving. The client will be able to request these tasks be performed in separate asynchronous functional calls, if desired, without having to worry about other clients modifying the server’s state and thus producing a result that is different than the original request submitted by the client.

The Most Appropriate Service Level for an Earth Model

An outstanding architectural issue is whether the most appropriate service level for an Earth model is the PGL Server. Consider that the PGL Server, used at its lowest level (atomic components), can require many small messages to be passed. But defining a server at a higher interface level (e.g., the level where event-locations are calculated), might require only a few large messages. In this case, a server-based event locator would have the PGL and the Earth model instantiated with it. Because of the message-passing overhead, a higher-level server could provide performance improvements in many circumstances. This architectural issue is illustrated in Figure 5.

Figure 5. Comparison of architectures at the PGL/Earth model level and the event-locator level.

To address this issue, we have programmed a server version of the event-locator LocOO (Ballard, 2003). LocOO takes as input data for one event and (based on a given FDB) returns the location. LocOO calculates locations by repeated successive queries to the PGL, continually refining its estimate of the event’s hypocenter and epicenter and the error volumes associated with these locations. Many requests for small collections of data are made. The version of LocOO that is typically used instantiates its own copy of the PGL to access Earth-model data, so these numerous requests to the PGL carry little overhead. LocOO can also use the PGL Server, but when it does the numerous message-passings that occur cause substantial overhead.
Contrast this situation with LocOO Server. An instance of the PGL is created within LocOO Server when LocOO Server is started. LocOO Server takes the same input data as LocOO. LocOO Server then calculates the location by repeated successive queries to the PGL, but because the PGL is bound directly to LocOO Server on the same machine, the communication overhead is minimal.

We are currently defining tests to quantify the differences between using the PGL Server and LocOO Server in typical location problems. We are also interested in looking at implementing parallelism in the PGL Server and LocOO Server (see the next section), because LocOO is often used to locate or relocate hundreds of events.

Location, however, is only one specific problem where Earth-model data are needed. Earth models are also used in phase identification, magnitude, discrimination of explosions from earthquakes, etc. It is possible that a next-generation data center for seismic event monitoring could have Earth models available at several different service levels within the processing system.

How to Best Implement Parallelism

Parallelism, the execution of two or more similar processes concurrently, is one technique for improving processing speed and achieving higher processing throughput in a system. It is possible that next-generation data centers for seismic event monitoring will have to deal with many difficult processing issues (e.g., 3D Earth models). At this time, we have only dealt with parallelism primarily at a theoretical level. However, the PGL Server and other server-based components of a seismic event monitoring system lend themselves immediately to parallelism. Here we discuss some ideas concerning parallelism, including running the PGL Server in parallel processes.

There are two ways of implementing parallelism in an Earth-model tool such as the PGL Server: “internal” parallelism, where subprocesses within a single instance of the PGL Server can execute in parallel, and “external” parallelism, where multiple instances of the PGL Server work on the same job in parallel. Examples of these types of parallelism are outlined in Figure 6. Note that in the figure the job being processed is implied within the PGL Server. In both types of parallelism, a single job is split into parts, the parts are worked at the same time, and then the results from each part are collected (the parts of the job cannot be dependent on one another).

Internal parallelism is the type of parallelism that is usually envisioned. Several software systems have been developed to implement this type of parallelism, including Message Passing Interface (MPI) and Parallel Virtual Machine (PVM). Although not strictly necessary, this type of parallelism tends to work best on multi-processor computers or a network of homogeneous computers with exotic network hardware. We believe there are processes that the PGL performs that are amenable to distributed parallel processing (e.g., loop oriented numerical processes such as matrix solvers). In addition to implementing MPI parallelism within the PGL for multiprocessor machines, we are investigating how to implement the PGL Server parallelism using a network of heterogeneous computers and standard gigabit Ethernet hardware.

More germane to this paper is how to implement parallelism using multiple instances of the PGL Server. Two basic questions arise: (1) Would there be an improvement in speedup or throughput; and, (2) how would multiple PGL Servers be controlled? This second question includes issues such as how would the job be divided among multiple servers, how would load balancing be performed, how many servers would be running, and how would errors be handled (e.g., fault tolerance). Answering the second question involves theoretical work and creating prototypes of controllers. In our initial approach, each the PGL Server instance has an agent that communicates with agents at each client to manage these issues. This work is in the formative stages, and we will only address the first question here.

Using the Condor batch-job-farming software (Condor, 2006), we investigated the extent of the speedup afforded by using multiple instances of the PGL on different processors. The problem was to visualize a travel-time model by using kriging to interpolate on a grid of given resolution (e.g., two degree spacing over some extent). By varying the grid resolution and the number of processors, it was possible to investigate performance improvements afforded by using Condor to facilitate parallel processing with the PGL. In this particular test with the interpolation of grid points was split between 9 processors, we see parallel performance bettering single processor performance at a grid resolution of 0.5 degrees (approximately 2,000 total grid points in this test). The performance improvement approaches linear speedup when the grid resolution is 0.025 degrees (approximately 750,000 total grid points in this test).
These results are shown in Figure 7. Thus, to answer the first question, we have shown that multiple instances of the PGL (or PGL Server) can improve speedup under certain circumstances within a seismic monitoring system.

Figure 6. Potential ways of implementing parallelism with the PGL Server.
CONCLUSIONS AND RECOMMENDATIONS

We currently have a version of the PGL Server that uses XML-RPC for communications. The PGL Server can be used to solve the PGL startup problem. In the process of developing this version of PGL Server, we are also developing an automatic method for generating the PGL Server stub libraries (e.g., using XML-RPC, sockets, JNI). We also developed a server version—LocOO Server—for the event-location program LocOO.

With these programs, we investigated issues associated with advanced system architectures, in particular SOAs. We conducted performance testing that indicated the best way to conduct message-passing and the best form of the messages. Using LocOO Server, we have begun to investigate the structure of different architectures, including whether it is better to have a service level at the Earth model or at the locator. This preliminary work suggests that different processes might require different service levels. In anticipation of more computation-intensive operations being required for seismic monitoring in the future, we began to study ways of parallelizing Earth models. A preliminary finding from this work is that many PGL calculations scale linearly with the number of PGL instances (e.g., PGL Servers) running.

Much of this work is in the initial stages. We would like to implement a method for saving state within the PGL Server. We would like to better understand how to manage multiple PGL Servers and how to best implement parallelism with the PGL Server. Additionally, we would like to test the concepts that we are developing on real-world problems.

In performing this work, we understand that architecture development is not accomplished by introspection, and it is not just a paper study. Development of an advanced architecture requires construction of prototypes and testing the behaviors of these prototypes in realistic situations. Ultimately, we believe that the knowledge that we have gained from this work can be applied to the design of an advanced-architecture for a seismic monitoring system.

ACKNOWLEDGEMENTS

We would like to thank David Gallegos, Mark Harris, and Chris Young; without their support and guidance this work would have never been accomplished.

REFERENCES


INTEGRATED SEISMIC SENSOR/DIGITIZER EVALUATION

Richard P. Kromer and Darren M. Hart

Sandia National Laboratories

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Development
Office of Defense Nuclear Nonproliferation

Contract No. DE-AC04-94AL85000

ABSTRACT

Sandia National Laboratories have tested and evaluated an integrated seismic sensor/digitizer from Science Horizons Inc. (SHI), Melbourne, FL. A Geotech GS21 single component short-period vertical borehole seismometer was integrated with a SHI AIM24-S1/GS21 modified borehole digitizer. The integrated seismometer/digitizer concept may provide a lowered system noise due to the elimination of long analog cables (up to 100 m), digitizer cable harness, and potentially quieter pre-amplifier design.

The modified SHI AIM24-S1/GS21 digitizer was compared to a standard configuration AIM24-S1/GS21 digitizer with a 30-m analog cable for self-noise measurement comparison. A sensor impedance simulator was used as a digitizer load.
OBJECTIVE

Introduction
The Air Force Technical Applications Center (AFTAC) is tasked with monitoring compliance of existing and future nuclear test treaties. To perform this mission, AFTAC uses several different monitoring techniques to sense and monitor nuclear explosions, each designed to monitor a specific domain (e.g. space, atmosphere, underground, oceans, etc.) Together these monitoring systems, equipment and methods form the United States Atomic Energy Detection System (USAEDS). Some USAEDS seismic stations may be included in the International Monitoring System (IMS). Some of these monitoring systems are deployed in extremely quiet locations that challenge the performance of the digitizing waveform recorder (DWR).

Some Sensor Sub-Systems (SSS) built for AFTAC applications use passive sensors such as the Geotech GS21, 23900, GS21a and GS13. These sensors are typically installed in a borehole at up to 100 meters depth. A DWR is typically installed at the top of the borehole in a Wellhead Terminal Unit (WTU). A long analog cable connects the seismometer output to the DWR input through an Interface Box (IB). The sensor impedance is terminated using a programmable resistance in the DWR. The DWR provides an internal preamplifier to set the seismic signal level appropriate to the application.

A set of tests has been developed to (1) determine if the long analog cable contributes noise to the separated sub-system and (2) determine if an integrated sensor/DWR can lower sub-system noise.

Evaluations Performed
Evaluations include determination of the GS21 seismometer impedance model, construction of a seismometer impedance simulator and tests to determine sensor sub-system performance.

Tests included:
DWR Seismic Sensor Application Tests
  - Seismic System Static Performance Tests
  - DWR Seismic System Noise (DWR-SSN)
Sensor Sub-Systems Seismic Sensor Application Tests
  - Sensor Sub-Systems Seismic System Static Performance Tests
  - Sensor Sub-Systems Seismic System Noise (SS-SSN)

RESEARCH ACCOMPLISHED

Determination of Seismometer Impedance Model
The complex impedance of a seismometer can be modeled using equation 1.

\[
\text{Complex impedance}(\omega) = \frac{R_1 + \left(\frac{R_{DG}}{M}\right)^{\frac{1}{2}} \omega}{\omega^2 + 2 * D_c * \omega_0 + \omega_0^2}, \tag{1}
\]

Where

\[
R_1 = \frac{R_{DC} * R_{ED}}{R_{DC} + R_{ED}} \quad \text{and} \quad R_2 = \left[\frac{R_{ED}}{R_{DC} + R_{ED}}\right]^2,
\]

\(R_{ED}\) is the external damping resistor (ohms), \(R_{DC}\) is the data coil resistor (ohms), \(M\) is the seismometer mass (kg), \(D_c\) is the seismometer damping coefficient, \(G_c\) is the seismometer generator constant (V/m/s), \(\omega_0\) is the natural frequency (radians/second) of the seismometer. These parameter values for the seismometers used in this study are shown in Table 1.

Table 1. List of seismometer parameters used to calculate complex impedance.

<table>
<thead>
<tr>
<th>Seismometer Type</th>
<th>Generator Constant (V/m/s)</th>
<th>External Damping Resistor (ohms)</th>
<th>Damping Coefficient</th>
<th>Data Coil Resistance (ohms)</th>
<th>Natural Frequency (Hz)</th>
<th>Seismometer Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS21</td>
<td>458.0</td>
<td>4255</td>
<td>0.707</td>
<td>467</td>
<td>1.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>
The complex impedance plot for the GS21 seismometer is shown in Figure 1. The plot shows at low and high frequencies the seismometer impedance is approximately equal to $R_1$ (~420 ohms) and the seismometer impedance rises to the value of the external damping resistance at the natural frequency of the seismometer.

Figure 1. GS21 Complex Impedance.

Table 2. Impedance terminators used to model GS21 seismometer’s complex impedance. X’s indicate impedance values used to model specific seismometers complex impedance. For this report the best and worst case impedances were used (XX).

<table>
<thead>
<tr>
<th>Impedance Termination (ohms)</th>
<th>402</th>
<th>1050</th>
<th>1500</th>
<th>2100</th>
<th>2500</th>
<th>3010</th>
<th>3480</th>
<th>4420</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS-21</td>
<td>XX</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>XX</td>
</tr>
</tbody>
</table>
Construction of Seismometer Impedance Simulator
A Seismometer Impedance Simulator was constructed using parts of an old seismometer. Changeable resistor terminations are installed on the inside. This simulator, shown in Figure 2, was used in place of the seismometer.

![Image of Seismometer Impedance Simulator]

Figure 2. Sensor Impedance Simulator with interchangeable loads.

Evaluation of AFTAC Typical Sensor Sub-System and Alternative Integrated Sensor Sub-System
An AFTAC Typical Sensor Sub-System consists of a Geotech GS21 sensor installed at the bottom of a borehole and connected to a DWR installed in a WTU. Analog signals from the seismometer are connected by borehole cable to the WTU Interface Box and distributed to the DWR installed at the top of the borehole. This is shown in Figure 3, left.

The Alternative Integrated Sensor Sub-System consists of a Geotech GS21 sensor directly integrated to a DWR installed at the bottom of a borehole. Digital signals from the integrated DWR/seismometer are connected by borehole cable to the WTU breakout box. This is shown in Figure 3, right.
Figure 3. A cartoon drawing of the two deployment configurations tested in this study. The boxes in red represent the digitizer, light blue boxes represent the seismometer and the orange objects represent the WTU. The drawing on the left is the typical deployment configuration. The drawing on the right shows the alternative configuration where the digitizer is directly connected to the seismometer.
Evaluation of AFTAC Typical Sensor Sub-System using Science Horizons AIM24S1 DWR separate from Geotech GS21 Short-period Vertical Borehole Seismometer (Figure 4).

The following tests were conducted on the AFTAC Typical SSS DWR.

DWR Seismic Sensor Application Tests
   Seismic System Static Performance Tests
   DWR Seismic System Noise (DWR-SSN)

The following tests were conducted on the AFTAC Typical SSS.

Sensor Sub-Systems Seismic Sensor Application Tests
   Sensor Sub-Systems Seismic System Static Performance Tests
   Sensor Sub-Systems Seismic System Noise (SS-SSN)
DWR Seismic System Noise (DWR-SSN) Test
Purpose: The purpose of the DWR seismic system noise test was to determine ability of the DWR to resolve the expected seismic background using a specific seismometer. The DWR self-noise should be below the expected seismic background.

Configuration: The DWR sensor input connector was terminated with the equivalent output impedance of the application sensor. For the GS21 comparison purposes, the range of values was chosen to approximate the minimum (402 ohms) and maximum (4.4 K ohms) impedance.

Evaluation: For GS21 sensor application, the system noise of the DWR was converted to ground motion using the GS21 seismometer response mathematical model. The results of this computation were overlaid with the USGS New Low Earth Noise Model (NLNM) to demonstrate the ability of the DWR to resolve the local seismic background.

Sensor Sub-System Seismic System Noise (SS-SSN) Test
Purpose: The purpose of the SSS seismic system noise test was to determine ability of the SSS to resolve the expected seismic background using a specific seismometer. The SSS self-noise should be below the expected seismic background.

Configuration: The SSS Sensor Simulator at the sensor end of the SSS sensor cable was terminated with the equivalent output impedance of the application sensor. For the GS21 comparison purposes, the range of values was chosen to approximate the minimum (402 ohms) and maximum (4.4 K ohms) impedance.

Evaluation: For GS21 sensor application, the system noise of the SSS was converted to ground motion using the GS21 seismometer response mathematical model. The results of this computation were overlaid with the USGS New Low Earth Noise Model (NLNM) to demonstrate the ability of the DWR/SSS to resolve the local seismic background. Comparisons were made between the two configurations.

Results: DWR-SSN and SS-SSN results are shown in Figure 5. There were no appreciable differences between the two test configurations.

Figure 5. DWR Seismic System Noise (DWR-SSN) Test and Sensor Sub-Systems Seismic System Noise (SS-SSN) Test Results.
Evaluation of Alternative Integrated Sensor Sub-System using Science Horizons AIM24S1 DWR integrated with Geotech GS21 Short-period Vertical Borehole Seismometer (Figure 6).

The following tests were conducted on the Alternative Integrated SSS.

Sensor Sub-Systems Seismic Sensor Application Tests
  Sensor Sub-Systems Seismic System Static Performance Tests
  Sensor Sub-Systems Seismic System Noise (SS-SSN)

Sensor Sub-System Seismic System Noise (SS-SSN) Test
Purpose: The purpose of the SSS seismic system noise test was to determine ability of the SSS to resolve the expected seismic background using a specific seismometer. The SSS self-noise should be below the expected seismic background.

Configuration: The Alternative Integrated SSS DWR/Sensor Simulator was terminated with the equivalent output impedance of the application sensor. For the GS21 comparison purposes, the range of values was chosen to approximate the minimum (402 ohms) and maximum (4.4 K ohms) impedance.

Evaluation: For GS21 sensor application, the system noise of the SSS was converted to ground motion using the GS21 seismometer response mathematical model. The results of these computations were overlaid with the USGS New Low Earth Noise Model (NLNM) to demonstrate the ability of the DWR/SSS to resolve the local seismic background.

Results: SS-SSN results are shown in Figure 7.
Figure 7. Alternate Integrated Sensor Sub-Systems Seismic System Noise (SS-SSN) Test Results.

Comparison of AFTAC Typical Sensor Sub-System and Alternative Integrated Sensor Sub-System Seismic System noise

Sensor Sub-System Seismic System Noise (SS-SSN)

Purpose: The purpose of the SSS seismic system noise test was to determine ability of the SSS to resolve the expected seismic background using a specific seismometer. The SSS self-noise should be below the expected seismic background.

Configuration: The Alternative Integrated SSS DWR/Sensor Simulator was terminated with the equivalent output impedance of the application sensor. For the GS21 comparison purposes, the range of values was chosen to approximate the minimum (402 ohms) and maximum (4.4 K ohms) impedance.

Evaluation: For GS21 sensor application, the system noise of the SSS was converted to ground motion using the GS21 seismometer response mathematical model. The results of these computations were overlaid with the USGS New Low Earth Noise Model (NLNM) to demonstrate the ability of the DWR/SSS to resolve the local seismic background.

Results: SS-SSN results are shown in Figure 8. There were no appreciable differences between the two test configurations.
CONCLUSIONS

Evaluation of AFTAC Typical Sensor Sub-System
The DWR seismic system noise was not degraded or increased by installation into an AFTAC Typical SSS installation for either sensor impedance. SSS seismic system noise was equivalent to within 0.2 dB.

Evaluation of Alternative Integrated Sensor Sub-System
The Alternative Integrated SSS seismic system noise was measured without difficulty.

Comparison of AFTAC Typical Sensor Sub-System and Alternative Integrated Sensor Sub-System
The Alternative Integrated SSS seismic system noise was not improved over the AFTAC Typical SSS installation for either sensor impedance.

The pickup of unwanted electronics noise can be a problem when separating a passive seismometer like the Geotech GS21 and the application DWR with up to 100 meters of cable. The AFTAC Typical installation technique for this configuration including power, cable interconnection and radio communications does not appear to contribute electronics noise to the SSS.

The DWR/Seismometer tested is the least affected impedance match. Other passive sensors can have impedances to greater than 100K ohms. This series of tests could be applied to these sensors to confirm rejection of unwanted electronics noise.

RECOMMENDATIONS

The present generation of AFTAC DWR components have inherent rejection of common mode electronics noise of up to the analog power supply voltages (+/- 12 volts). The next generation of low power DWR component technology utilizes electronics components shared with the cell phone and similar industry. The typical power supply voltages are +/- 3 volts or even lower. These components will not have the electronic noise rejection of the present technologies.

The use of the alternative integrated technique should continue to be evaluated as next-generation electronics components become more common in DWR design. This design might require more integrated functions into the DWR/SSS design such as internal GPS, integrated power systems and direct digital communication to the downhole integrated sub-system.
INTEGRATED SEISMIC EVENT DETECTION AND LOCATION BY ADVANCED ARRAY PROCESSING

T. Kvaerna\textsuperscript{1}, S. J. Gibbons\textsuperscript{1}, F. Ringdal\textsuperscript{1}, and D. B. Harris\textsuperscript{2}

NORSAR\textsuperscript{1} and Lawrence Livermore National Laboratory\textsuperscript{2}

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Development
Office of Defense Nuclear Nonproliferation

Contract Nos. DE-FC52-03NA99517\textsuperscript{1} and W-7405-ENG-48\textsuperscript{2}

ABSTRACT
As the trend in nuclear explosion monitoring continues in the direction of detecting and locating seismic events of ever smaller magnitudes, the number of events which are detected and which require processing increases enormously. With more candidate phase determinations and event hypotheses, it is of paramount importance that the accuracy of parameter estimates and automatic event locations is good enough to prevent analyst time being used on examining signals which can, with a high level of confidence, be attributed to known industrial seismic sources. It is similarly important to try to maintain a low false alarm rate. Many seismic arrays of the International Monitoring System (IMS) are within regional distances of many sources of repeating seismic events. In recently funded studies, large numbers of events from known sources of seismicity have been identified providing an excellent basis of ground-truth (GT) events by which we have been able to benchmark phase determinations and subsequent event location estimates. Some of these events are mining events for which the mine operators have provided explosion times and locations, and other events have been attributed to source locations on the basis of waveform correlation studies.

The aim of the current project has been to construct a prototype system for the automatic monitoring of seismic events from sites of interest using regional seismic arrays. Current automatic location procedures frequently show rather large mislocations. In many cases, this is due to multiple events whereby several sequences of similar regional phases may reach a given seismic array within a short time period and be subsequently associated incorrectly. The prototype system is designed to mitigate errors of this kind by considering a system of site-templates whereby, under the hypothesis of an event at a given time from a calibrated source location, we consider only the observation of wavefield parameters in a number of very carefully defined time-windows relative to the hypothetical origin time. Another major source of error in the current automatic location estimates is a result of the use of azimuth and slowness estimates measured using variable frequency bands which show a demonstrably greater spread than the corresponding estimates measured in fixed frequency bands. The prototype system is designed to mitigate these errors by using a calibrated phase-template for each of the time-windows specified by the site-template. Azimuth and slowness are measured in one or more fixed frequency bands which are demonstrated to give the most stable estimates for a given phase from a given site, and appropriate corrections are applied prior to the location procedure. The template-based system gives automatic location estimates which are demonstrably better for the sites considered than the corresponding generalized beamforming (GBF) solutions. Anticipated and observed azimuth values can vary dramatically, emphasizing the need for calibration.

An additional problem with site- or phase-templates relying upon conventional frequency wavenumber (F-K) analysis and beamforming is the loss of coherence across an array aperture. Conventional algorithms assume that the wavefields incident upon an array satisfy a plane-wave model. When they do not, as happens when refraction and scattering are significant, especially in higher frequency bands, empirical matched-field processing may improve performance. In the matched-field processing approach, narrowband plane-wave steering vectors are replaced with empirical steering vectors derived from many observations of the array signal of interest. Last year we demonstrated empirical steering vectors for the ARCES array for the Pn phase from mines of the Khibiny Massif. These calibrations provided up to a factor of 3 improvement in energy detection at frequencies above 10 Hz. This year we report progress on extending the calibration from a single phase to the entire seismogram. The innovation is a non-stationary calibration or set of steering vectors that changes continuously throughout the duration of the seismogram as different phases come and go.
OBJECTIVE

This two year collaboration between the Norwegian Seismic Array (NORSAR) and Lawrence Livermore National Laboratory (LLNL) has explored improvements to the automatic detection and location of seismic events using regional arrays. At the heart of the study has been the calibration of processing parameters for the detection and location of events from a specific region using observations of previous ground-truth events at the sites of interest. The goal is to attribute, with a high degree of confidence, automatically located events to active mines or areas with known recurring seismicity. The study has examined sites in Fennoscandia and Kazakhstan using the seismic arrays in these regions.

The signals at a given array station, resulting from a set of events from a site with recurring seismicity, are likely to display common characteristics which may be exploited in order to identify subsequent events from the same region. A template describing the measurements which can be anticipated at a given station at a given time can be used to judge whether or not a detected signal is the likely result of an event from the site of interest. Such templates must be calibrated by investigating the variability of measurements made from events confirmed to have taken place at the sites; such calibrations have been the main focus of this investigation. We have, in addition, explored the potential of applying advanced new “matched field” array processing methods in order to compensate for array processing loss due to refraction and scattering, thus enhancing array gain at high frequencies.

RESEARCH ACCOMPLISHED

Automatic Event Detection and Location Procedures

The aims of nuclear explosion monitoring are being widened continually to encompass the detection, location, and identification of seismic events of ever smaller magnitude. This leads to a dramatic increase in the number of seismic events which require processing which in turn leads to increased demands upon automatic processing procedures. Such procedures need to be sensitive (to ensure that no events are missed), with a low false-alarm rate, and accurate. In particular, the large number of routine industrial seismic events should be associated (fully-automatically and with a high level of confidence) with the correct source location. Recent improvements in GT information for repeating seismic sources have improved dramatically our ability to assess the quality of fully automatic event location estimates. Figure 1 shows the location of numerous sites of repeating seismicity in northern Fennoscandia and north-west Russia for which we have excellent GT information. The four regions indicated in Russia are all sites of several different mines and information regarding the location and origin times of explosions has been collected from all these sites under the NNSA-funded contract “Ground-truth Collection for Mining Explosions in Northern Fennoscandia and Russia” (Harris et al., 2003). Explosions at the Finnish ammunition destruction site were identified largely from the GBF (Kvaerna and Ringdal, 1989) automatic event bulletin due to the characteristic explosion times; their identification was subsequently confirmed using array-based waveform correlation (Gibbons and Ringdal, 2006) which also provided excellent constraints upon the origin time of each event. For the remaining sites (all large-scale mining operations in the north of Sweden), information regarding a few selected events was provided by the mining companies, and large numbers of additional events were subsequently identified using waveform correlation methods. Figure 1 (a,b) allows a direct comparison of known event locations and fully-automatic GBF location estimates. Whilst the location accuracy is generally sufficiently good for an analyst to take an event and relocate it interactively, it is clear that associating signals with a specific source region is not possible without some form of post-processing algorithm. The reasons for the large spread in the GBF solutions are well-understood and motivate the development of a template-based system as depicted in Figure 2. Observations of many events from the same source region allow for a very good evaluation of which phases are well-observed at which stations and at which times following a seismic event; this is the concept of a site template. Within a carefully defined time-window for each such phase, we can define a set of diagnostic tests together with a range of accepted values by which we can determine whether or not the observed phase is consistent with corresponding observations from previous events at the same site. This is the concept of a phase template (see Figure 3). Gibbons et al. (2005) describe in detail the algorithm and results for such a system for the identification and location of events from the Kovdor mine. In this case, observations from only a single array station are used. This is a case of increasing relevance in the observation of increasingly small events using a relatively sparse international seismic network.
Figure 1. (a) Sites of repeating seismicity in the vicinity of the ARCES primary IMS array. (b) Fully automatic location estimates (Kværna and Ringdal, 1989) for events known to have taken place at the sites in (a). (c) Analyst reviewed location estimates for selected events from (b). (d) Fully automatic slowness vector estimates for initial P-arrivals for events from the mining sites on the Kola Peninsula in North West Russia (Zapoljarni, Olenegorsk, Khibiny, Kovdor). (e) Fixed frequency band slowness estimates (2.0 - 4.0 Hz) for the arrivals displayed in (d). (f) The same phase arrivals processed in the fixed frequency band 4.0 - 8.0 Hz.
Figure 2. Schematic representation of a template-based event identification and location algorithm. The site template specifies which seismic phases should be observed at which times at which stations, given a hypothesis of a seismic event at a site of interest at a given time. For each such phase, a phase-template (symbolized by the black boxes) defines a set of test parameters, evaluation criteria and, optionally, calibration information which determine whether or not the observed phase is consistent with observations of that phase from previous events at that site (see Figure 3).

Figure 3. The principal components of a site template in regional array event identification and location algorithms: autoregressive arrival time re-estimation (left) and measurement of backazimuth and apparent velocity using broadband F-K analysis in fixed frequency bands. As displayed in Figure 1 (d,e,f), the use of carefully chosen fixed frequency bands can lead to a great improvement in the stability of slowness estimates and allows for far tighter hypothesis acceptance criteria.

Under the current project (Kværna et al. 2004, 2005), the principles applied in Gibbons et al. (2005) have been extended to encompass a far wider range of sites and station configurations. In all cases, the application of template-based algorithms has led to an improvement in fully-automatic event location estimates. In almost all cases in which the procedure failed, the reason for failure was the rejection of event hypotheses due to the inability to measure a phase arrival time with a sufficient level of confidence (signal-to-noise ratio [SNR] or Akaike Information...
Criterion measurements did not satisfy required thresholds) or the failure of F-K analysis to return a slowness estimate within the required time-window. In almost all cases, this was the result of “interfering signals” which were almost invariably multiple events from the same site. Such multiple event sequences are unfortunately very characteristic of the type of event we are attempting to classify in the current investigation. In the GBF algorithm, this typically leads to a spurious association of P- and S- phases from distinct events and a corresponding location estimate at the wrong site (see Figure 1b). This eventuality is precluded from the template-based system since we only consider observations within the time-windows defined by the site-template (Figure 2); phases observed outside of these time intervals have no influence on the evaluation of the event hypothesis. Gibbons et al. (2005) found that approximately 40% of the confirmed Kovdor events could not be located automatically due to such interfering phases. To deal with this eventuality, a new category “very likely Kovdor events” was defined; these are signals which show many characteristics of events from the site of interest, albeit not sufficiently many that the event can be located automatically. This category is useful but a setback in our quest to mitigate the need for analyst interaction. It is useful to note that in the cases where a fully automatic location was not possible, the analyst location was also subject to a far higher degree of error. This failure rate (40% of events not locatable) turned out to be quite typical for all the sites covered.

Considerations in the Use of Small-Aperture Arrays for Slowness Measurements and Consequences for Event Location Estimates

Whilst the majority of the location estimates in Figure 1b which are “qualitatively incorrect” are the result of spurious phase association, the general large spread in location estimates is primarily the result of applying slowness and azimuth estimates which are measured in variable frequency bands. Figure 1d shows slowness estimates for the initial P-arrivals from the Russian events in the dataset, measured in a frequency band optimized on an event-by-event basis, as they appear in the NORSAR detection lists. Figure 1 e) and f) show the slowness estimates for exactly same arrivals when measured in the fixed frequency bands 2.0-4.0 Hz and 4.0-8.0 Hz respectively. Two results are immediately evident:

1. The fixed frequency band estimates are far more stable than the variable band estimates.
2. The frequency band offering the most stable estimate is not the same for all sites.

It is, for example, clear how the spread in slowness estimates for the Zapoljarni mines (blue symbols) in Figure 1d translates directly into the spread in event location estimates observed in Figure 1b. It is likely that if only slowness estimates obtained in the 2.0 - 4.0 Hz band were applied, then the corresponding spread in location estimates would decrease dramatically. If the slowness estimates obtained between 4.0 and 8.0 Hz were used instead, the spread in location estimates would probably remain large. The situation is reversed for the Khibiny mines with the higher frequencies resulting in more stable estimates.

The set of Finnish explosions provides a nice example case since, due to the simpler source-time functions, none of the events were subject to interfering phases and the spread in event location estimates can be attributed almost entirely to variable slowness estimates and error in phase onset-time measurements. Figure 4 shows the fully-automatic P-phase slowness estimates (left) and the same estimates made in the fixed frequency bands as indicated (right). The fixed-band estimates not only display a significantly lower spread for each band applied, but also display almost no overlap between the different bands. It is important to note that at this epicentral distance, a triplication occurs whereby the Pg crustal phase and the higher frequency Pn phase arrive essentially at the same time; this may explain why the velocity in the 4.0-8.0 Hz band is higher than the velocity at lower frequencies. At even higher frequencies, waveform incoherence over the array is probably significant. However, what is not explained by this triplication is how the azimuth varies with the frequency band. The smallest spread is observed for the 2-4 Hz band and restricting location estimates to these values results in location estimates for the fully-automatic template-based method which are superior to the analyst interactive location estimates (Figure 5). The reason for this is evident from Figure 4; the analyst picks a frequency band on an event-to-event basis and locates the event using velocity and azimuth measured in the chosen band. The analyst does not (currently) have the calibration information available and is not aware that the 2-4 Hz band gives by far the most stable azimuth estimates for events from this site. It must also be stressed that this result is by no means intuitive since the SNR is significantly lower in this frequency band than it is at higher frequencies. The temptation will be to use a frequency band which produces an optimal combination of SNR and high F-K power (beam gain) - this is judged subjectively by the analyst for each
event encountered and results consequently in a directional estimate which is a direct function of the frequency band applied.

Figure 4. (Left) slowness estimates in the routine automatic processing for the defining P-arrival at the ARCES seismic array for each of 108 ammunition destruction explosions at a military site in the north of Finland, at a distance of approximately 175 km. (Right) slowness estimates for in the indicated fixed frequency bands for the same arrivals, albeit measured in an identical time window for each event.

Figure 5. Locations of Finnish explosions using different location methods. The exact explosion site coordinates are not known but are assumed to be approximately 68°N, 26°E (see arrow), the green stars represent the GBF fully-automatic location estimates, the red diamonds indicate the NORSAR analyst reviewed location estimates, the orange squares indicate the single array (ARCES) template-based location estimates, and the blue circles indicate the 3-component station network solutions using the Finnish National Seismic Network (courtesy of the seismic bulletin of the University of Helsinki). Note in particular how the template-based fully automatic solutions have a far lower standard error (approximately 5 km) than either the NORSAR reviewed analyst solutions (approximately 15 km) or the Finnish network solutions.
Matched Field Processing

Last year, we tested the concept of applying empirical matched field processing to a single phase (Pn) at high frequencies (above 10 Hz) for a small aperture array (ARCES). For a single phase it is reasonable to make the assumption that the spatial statistics of the signal are stationary. Using a narrowband (essentially monochromatic) assumption, we made the approximation that the complex analytic form of the array signal in a narrow band centered on frequency $f_0$ has the form:

$$
\mathbf{r}(t) = \begin{bmatrix}
\mathbf{r}(\mathbf{x}_1, t) \\
\mathbf{r}(\mathbf{x}_2, t) \\
\vdots \\
\mathbf{r}(\mathbf{x}_N, t)
\end{bmatrix} = \mathcal{E}(f_0) \mathbf{s}(t, f_0) e^{-i2\pi f_0 t}
$$

where the signals $\mathbf{r}(\mathbf{x}_i, t)$ are the signals recorded by individual sensors in the array located at positions $\mathbf{x}_i$. The baseband signal $\mathbf{s}(t, f_0)$ is a slowly-varying complex envelope. This narrowband approximation has the effect of separating the spatial and temporal variations of the signal into multiplicative factors. The so-called steering vector $\mathcal{E}(f_0)$ encodes all of the spatial variation of the signals across the array, and the baseband signal encodes the remaining temporal structure. This approximation becomes more accurate the narrower the processing band and becomes exact for monochromatic signals.

In the plane-wave approximation, the steering vector embodies the complex phase factors induced by propagation delays across the array. In reality, effects of refraction (e.g. focusing), diffraction, and scattering cause the steering vector to depart from this simple model. As a consequence, we advocate measuring the steering vector for particular source regions as a calibration and applying the measured steering vector in FK and beamforming operations instead of its theoretical plane-wave counterpart. Measuring the steering vector is conceptually straightforward using a large number ($M$) of waveforms from events in the target region of interest. We recommend estimating the spatial covariance matrix over the ensemble of events as:

$$
\mathbf{R}(f_0) = \mathbf{r}(t)(\mathbf{r}(t))^H \sim \frac{1}{M} \sum_{j=1}^{M} \int \mathbf{r}_j(t)(\mathbf{r}_j(t))^H dt = \alpha \mathcal{E}(f_0)(\mathcal{E}(f_0))^H
$$

where $\alpha = \frac{1}{M} \sum_{j=1}^{M} \left| \mathbf{s}_j(t, f_0) \right|^2 dt$.

If the assumption that a single phase is present is correct, the covariance matrix as shown in equation (2) is rank one and the steering vector may be obtained as the principal eigenvector of the matrix. This line of reasoning can be extended to calibrate the entire wavetrain for events from a particular region. However, the signal can no longer be considered to be stationary, even approximately. The signal model must be extended to permit changing spatial covariance structure along the wavetrain. We propose the following narrowband model:
This is a collection of $d$ signals representing all of the phases present, which may be a great number if multipath arrivals are considered. For simplicity, the frequency dependence of the model has been suppressed, but it is understood that the $d$ individual steering vectors and baseband signals are functions of $f_0$. The support of the individual baseband phase signals $s^i(t)$ may be expected to be transient. With phases coming and going, the spatial structure of the signals will vary as first one steering vector, then another comes to dominate array signal. At higher frequencies, as scattering becomes more significant, it is likely that multiple steering vectors will be required simultaneously to capture the range of spatial structure of signals originating from a specific source region.

$$r(t) = E(f_0)S(t, f_0)e^{-i2\pi f_0 t} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_d \end{bmatrix} e^{-i2\pi f_0 t}$$

It is convenient to compute and represent samples of the covariance function of equation (4) in discrete form as a matrix

$$R(t_1, t_2) = \frac{r(t_1)(r(t_2))^H}{M} \sim \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_d \end{bmatrix} \frac{1}{M} \sum_{j=1}^{M} \begin{bmatrix} s^1_j(t_1) \\ s^2_j(t_1) \\ \vdots \\ s^d_j(t_1) \\ s^1_j(t_2) \\ s^2_j(t_2) \\ \vdots \\ s^d_j(t_2) \end{bmatrix}^H$$

An example covariance matrix for the signals from the Kirovsk mine observed by four of the channels (ARA0, ARD1, ARD4, ARD7) of ARCES is shown in Figure 6. This covariance matrix was computed for a narrow band around 2.5 Hz using 236 events. The moduli of the individual (complex) elements of the matrix are represented in grayscale in the image with the darker elements having larger moduli. The individual phases ($P_n$, $P_g$, $S_n$ and $L_g$) are apparent in the matrix. It is interesting to note that the separate phases are correlated. Some decorrelation might be expected given the relative moveout of the different phases as the events are distributed over some source region. In
this case, the source region is small (several kilometers in extent), which may account for the relatively high cross-correlations among the phases.

Figure 6. Example covariance matrix for 236 events from the Kirovsk mine on the Khibiny massif as recorded by a subset of the ARCES array. Note the correlations between the different phase types: Pn, Pg, Sn, and Lg.

It is our intention to extract systematically from covariance matrices of the type shown time-varying steering vectors for particular source regions and for a large number of narrow bands. We anticipate using these steering vectors in beamforming operations that use the entire time-history of the signal in an temporally incoherent, but spatially coherent detection operation.

CONCLUSIONS AND RECOMMENDATIONS

We have constructed a prototype algorithm for the fully-automatic identification and location of seismic events from a given source region. This has been applied to a wide range of sources of repeating seismicity in Fennoscandia and North West Russia, and in Kazakhstan. The use of carefully calibrated processing parameters has led to greatly increased stability in location estimates which in many situations are comparable to solutions obtained interactively by an analyst. We have also documented a situation whereby the fully automatic template-based location estimates were significantly better than those obtained by an analyst. This is due to the fact that the solutions obtained are heavily influenced by the backazimuth estimates obtained from a regional array station and these are estimates vary with the applied frequency band. For each phase analyzed, the analyst makes a decision of which frequency band to use and does not have available information regarding which frequency band is optimal for that particular site (we have also demonstrated that the optimal frequency band varies significantly from site to site). The template-based automatic system also applies a calibrated correction term to the backazimuth measurement prior to the location
procedure which takes into account the azimuth bias associated with the specific frequency band applied. An analyst would require detailed calibration information for the site involved in order to know how great a correction to apply for the processing parameters chosen. This scenario motivates a completely different approach whereby the steering vectors applied during F-K analysis are modified to take account of the demonstrable apparent curvature of the wavefield with frequency, in order to provide a slowness estimate with a zero bias for all frequency bands. It is not yet known to which degree such a calibration is possible. The greatest unknown is the extent to which the time-delay corrections are a continuous function of the theoretical slowness vector.

Towards the aim of source-region-specific detection procedures, in situations where complicated and highly heterogeneous source-time functions preclude the effective use of waveform correlation detectors, we anticipate that time-varying steering vectors calculated from multiple narrow-band observations of large numbers of events from that region will facilitate an effective beamforming operation which utilizes the full wavetrain.

ACKNOWLEDGEMENTS

We are grateful to Vladimir Asming and colleagues at the Kola Regional Seismological Center in Apatity, Russia, for collecting Ground Truth information from the Russian mine-operating companies as to the sites and times of routine industrial explosions. We would also like to thank staff at the Kazakhstan National Data Center for providing data from the Kazakhstan array network together with automatic and reviewed event bulletins.

REFERENCES


SEL0: A FAST PROTOTYPE BULLETIN PRODUCTION PIPELINE AT THE CTBTO
Ronan J. Le Bras\textsuperscript{1}, Tim Hampton\textsuperscript{1}, John Coyne\textsuperscript{1}, and Alexander Boresch\textsuperscript{2}

Provisional Technical Secretariat of the Preparatory Commission for the Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO)\textsuperscript{1} and Angewandte Wissenschaft Software und Technologie GmbH (AWST)\textsuperscript{2}

Sponsored by Comprehensive Nuclear-Test-Ban Treaty Organization

Contract No. 2005-0376\textsuperscript{1,2}

ABSTRACT

The Preparatory Commission for the Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO) is exploring and assessing which data and products, beyond the raw waveform data from the International Monitoring System (IMS), might be useful and can be provided by the CTBTO’s Provisional Technical Secretariat (PTS) as input to tsunami warning centers. One of the critical features of such input is the timeliness of the delivery. The earliest standard automatic network processing at the International Data Centre (IDC), standard event list 1, or SEL1, forms events 1 hour and 40 minutes after they occur. This is too late for consideration by a tsunami warning system. Starting in February 2005, the PTS has conducted a technical test in its IDC development environment where the standard IDC software was tuned and improved to minimize the time difference between occurrence of the event and writing into a new standard event list in the database. This new standard event list is called SEL0. Time differences on the order of 20 minutes are routinely achieved between event occurrence and the writing of an event into the database, which should make it a usable input to an organization tasked with generating a tsunami warning. We present the basic characteristics of the configuration of this new processing pipeline, some statistics on the results achieved with this prototype, and future work to be conducted under this project. The basic conclusions are that, under stable conditions, no large events of interest are missed by the SEL0 pipeline and that the false alarm rate has shown some improvement during the last year of development of the pipeline.
OBJECTIVES

The objectives of this project are to develop a prototype processing pipeline using the IMS network that can produce a timely bulletin useful as input for agencies charged with warning the general public about impending disasters such as tsunamis.

Specifically, the elements of interest in a bulletin for such an agency would be as follows:

- Timeliness of the bulletin. This is the most critical element of the prototype system, as the warning has to be issued as early as possible to allow sufficient time for civil authorities to proceed with such actions as evacuation. We set this objective at an indicative time of 20 minutes after the occurrence of the event, given that the IMS network is not sufficiently dense to provide a faster bulletin uniformly on the surface of the globe.

- Minimal missed event rate. This is another critical element of the prototype system, as missing a large event would be unacceptable. There are several stages in an operational system that would be prone to failure leading to missed events. This includes failures of the acquisition system, hardware failures, database problems, or failures of the software system to detect a large event. At this stage of the prototype, since we have operated using minimal resources and in a development environment (where interruptions to processing are not uncommon), we have concentrated on the ability of the software to detect the large events when the acquisition and the hardware environment were stable.

- Location accuracy of the bulletin. This is an important element in determining whether the event is offshore or close to the shore and thus has the potential to generate a tsunami.

- Minimal false alarm rate. This is important in order to minimize disruption in the overall warning mechanism (it is important to remember that civil actions, such as evacuations ordered as a result of a tsunami warning, are themselves not without risk and often result in injuries or event deaths.)

- Accurate sizing and focal mechanism for events in the bulletin. The IDC automatic bulletins produce a measure of the size of the event based on the mb magnitude. It is well known that this magnitude scale saturates for events larger than about 6.5 (e.g., Abe, K, 1995). It is therefore important to develop other methods of assessing the size of large events in a timely manner. We have investigated the possibility of performing moment tensor inversion on long period P waves. This is currently at the development stage. The focal mechanism would also provide additional information for the tsunamigenic potential of the event.

RESEARCH ACCOMPLISHED

Configuration of the SEL0 Pipeline

The software used to produce the SEL0 bulletin is identical to the standard IDC software. The difference is that processing includes an additional set of auxiliary seismic stations received continuously in addition to the IMS primary stations. The configuration of the software is the main difference with the standard IDC processing. The main adjustments to data processing are that station processing intervals are 2 minutes instead of 10 minutes long in standard IDC processing, and network processing is run every 5 minutes instead of every 20 minutes in standard processing. The timing of network processing is also moved from 1 hour and 40 minutes to 20 minutes after the time of the event. An additional feature of SEL0, compared with the standard IDC processing, is that it keeps track of all instances of events written to the database as soon as they are written to the origin database table. To achieve this, in addition to the standard ORIGIN table, the ORIGIN_EWS table contains all versions of events and may contain the same event several times.

SEL0 processing is currently done in a non-operational context, on the IDC development environment, using a mixed Solaris-Linux environment, where most of the heavy-duty processing is done under Linux (detection
processing, phase identification, and network processing.) This non-operational mode implies that processing stops when forwarding of data is interrupted. To ensure timeliness of results, the SEL0 pipeline is configured to ignore data arriving more than 30 minutes after real time. This is in contrast to the normal IDC Operations where “catch up” after an extended outage significantly delays automatic bulletin production. Maintaining a constant data flow to the SEL0 pipeline would be a high priority if it were to become operational. In the current context, events are missed when the data are late or interrupted, which is not acceptable for an operational system.

**Reporting on a Monitoring Web Page**

Figure 1 shows the web interface through which detailed information about events generated by the SEL0 pipeline can be accessed. The page is continually updated by the processing to display the most recent events built by the processing. The final column of the figure shows the delay in minutes between the origin time of the event and the time at which it was written to the database.

![Figure 1](example.png)

**Table 1:** Highlights of the PDE events during the two time periods.

<table>
<thead>
<tr>
<th>Event</th>
<th>Origin Time (h:mm)</th>
<th>Magnitude</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Depth (km)</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2006-07-04 02:55:32</td>
<td>7.0</td>
<td>14.8</td>
<td>114.6</td>
<td>14.8</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>2</td>
<td>2006-07-04 09:26:28</td>
<td>7.2</td>
<td>17.1</td>
<td>116.8</td>
<td>14.8</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>3</td>
<td>2006-07-08 10:33:39</td>
<td>7.3</td>
<td>17.3</td>
<td>116.9</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>4</td>
<td>2006-07-09 14:32:41</td>
<td>7.4</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>5</td>
<td>2006-07-09 18:02:34</td>
<td>7.5</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>6</td>
<td>2006-07-09 22:13:25</td>
<td>7.6</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>7</td>
<td>2006-07-10 02:56:27</td>
<td>7.7</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>8</td>
<td>2006-07-10 07:57:33</td>
<td>7.8</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>9</td>
<td>2006-07-10 12:58:39</td>
<td>7.9</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
<tr>
<td>10</td>
<td>2006-07-10 17:59:45</td>
<td>8.0</td>
<td>17.3</td>
<td>116.8</td>
<td>14.2</td>
<td>NORTHEAST CHINA</td>
</tr>
</tbody>
</table>

**Events with Mₚ > 6 for Test Periods between June 7 and June 21, 2005, and between June 14 and June 30, 2006**

The events of interest for a warning system are large events. Our testing has concentrated on events with Mₚ larger than 6.0 as reported in the Preliminary Determination of Epicenters (PDE) bulletin of the U.S. Geological Survey (USGS). During the 15 days between June 7 and 21, 2005, processing was relatively stable, with the exception of 13.5 hours between 02:30 and 16:00 on June 13, 2005, caused by a hardware failure. Similarly, during the period between June 14 and 30, 2006, processing was stable with the exception of a slowdown on June 27 between 18:00 and 19:00. Table 1 shows the 15 events published in the PDE (USGS) catalog with Mₚ magnitude larger than 6.0 during these time intervals. Note that one of them, event 3, grayed in the table, falls in the SEL0 processing gap previously mentioned, whereas event 15 could not be obtained because data acquisition was delayed by more than 30 minutes. Event 6, offshore California, prompted a tsunami warning (issued, but subsequently cancelled) and resulted in maximum wave heights (peak-to-trough) of between 26.0 cm at Crescent City down to 2.0 cm at Bamfield, Vancouver Island, Canada.

Table 2 shows the SEL0 events corresponding to the PDE events during the two time periods. The last column in the table shows the delay time between the origin time of the event and the time at which the event was written to the database.

![Table 2](example_table.png)
database. Note that all the events were formed in the SEL0 bulletin at the exception of events 3 and 15, when the SEL0 processing was disrupted.

Table 1. PDE events with $M_w > 6.0$ from June 7 to 21, 2005, and from June 14 to 30, 2006 (see Figure 2). The grayed events fall in a processing gap due to hardware failure or late processing.

<table>
<thead>
<tr>
<th>Event number</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Depth  (km)</th>
<th>USGS $M_w$</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.21</td>
<td>96.74</td>
<td>46</td>
<td>6.1</td>
<td>08-Jun-2005</td>
<td>06:28:13.90</td>
</tr>
<tr>
<td>3</td>
<td>2.09</td>
<td>126.61</td>
<td>10</td>
<td>6.0</td>
<td>13-Jun-2005</td>
<td>07:02:33.11</td>
</tr>
<tr>
<td>4</td>
<td>-19.93</td>
<td>-69.03</td>
<td>117</td>
<td>7.8</td>
<td>13-Jun-2005</td>
<td>22:44:33.86</td>
</tr>
<tr>
<td>5</td>
<td>51.23</td>
<td>179.39</td>
<td>51</td>
<td>6.8</td>
<td>14-Jun-2005</td>
<td>17:10:16.35</td>
</tr>
<tr>
<td>6</td>
<td>41.28</td>
<td>-125.98</td>
<td>10</td>
<td>7.2</td>
<td>15-Jun-2005</td>
<td>02:50:53.01</td>
</tr>
<tr>
<td>8</td>
<td>-44.97</td>
<td>-80.57</td>
<td>10</td>
<td>6.5</td>
<td>15-Jun-2005</td>
<td>19:52:24.31</td>
</tr>
<tr>
<td>9</td>
<td>40.76</td>
<td>-126.60</td>
<td>10</td>
<td>6.7</td>
<td>17-Jun-2005</td>
<td>06:21:41.92</td>
</tr>
<tr>
<td>10</td>
<td>51.78</td>
<td>177.07</td>
<td>14</td>
<td>6.4</td>
<td>14-Jun-2006</td>
<td>04:18:42.44</td>
</tr>
<tr>
<td>11</td>
<td>45.41</td>
<td>149.35</td>
<td>95</td>
<td>6.0</td>
<td>22-Jun-2006</td>
<td>10:53:11.63</td>
</tr>
<tr>
<td>12</td>
<td>-0.44</td>
<td>123.30</td>
<td>53</td>
<td>6.3</td>
<td>24-Jun-2006</td>
<td>21:15:04.94</td>
</tr>
<tr>
<td>13</td>
<td>52.24</td>
<td>176.23</td>
<td>37</td>
<td>6.2</td>
<td>27-Jun-2006</td>
<td>02:39:36.02</td>
</tr>
<tr>
<td>14</td>
<td>-19.87</td>
<td>-178.31</td>
<td>571</td>
<td>6.2</td>
<td>27-Jun-2006</td>
<td>02:59:15.72</td>
</tr>
<tr>
<td>15</td>
<td>6.53</td>
<td>92.72</td>
<td>32</td>
<td>6.2</td>
<td>27-Jun-2006</td>
<td>18:07:23.05</td>
</tr>
<tr>
<td>Matching PDE Event</td>
<td>Latitude</td>
<td>Longitude</td>
<td>Depth (km)</td>
<td>IDC mb</td>
<td>Date</td>
<td>Time</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>1</td>
<td>2.21</td>
<td>96.79</td>
<td>67</td>
<td>5.07</td>
<td>08-Jun-2005</td>
<td>06:28:16</td>
</tr>
<tr>
<td>5</td>
<td>51.31</td>
<td>179.48</td>
<td>99</td>
<td>5.51</td>
<td>14-Jun-2005</td>
<td>17:10:22</td>
</tr>
<tr>
<td>6</td>
<td>41.67</td>
<td>-125.51</td>
<td>0</td>
<td>5.03</td>
<td>15-Jun-2005</td>
<td>02:50:54</td>
</tr>
<tr>
<td>7</td>
<td>-4.60</td>
<td>152.91</td>
<td>96</td>
<td>5.48</td>
<td>15-Jun-2005</td>
<td>10:14:02</td>
</tr>
<tr>
<td>8</td>
<td>-44.89</td>
<td>-80.75</td>
<td>0</td>
<td>4.89</td>
<td>15-Jun-2005</td>
<td>19:52:23</td>
</tr>
<tr>
<td>9</td>
<td>40.91</td>
<td>-126.27</td>
<td>19</td>
<td>5.46</td>
<td>17-Jun-2005</td>
<td>06:21:44</td>
</tr>
<tr>
<td>10</td>
<td>51.76</td>
<td>177.21</td>
<td>0</td>
<td>5.05</td>
<td>14-Jun-2006</td>
<td>04:18:40</td>
</tr>
<tr>
<td>11</td>
<td>45.45</td>
<td>149.21</td>
<td>73</td>
<td>5.61</td>
<td>22-Jun-2006</td>
<td>10:53:09</td>
</tr>
<tr>
<td>12</td>
<td>-0.28</td>
<td>123.31</td>
<td>0</td>
<td>5.37</td>
<td>24-Jun-2006</td>
<td>21:14:56</td>
</tr>
<tr>
<td>13</td>
<td>52.06</td>
<td>176.19</td>
<td>112</td>
<td>4.74</td>
<td>27-Jun-2006</td>
<td>02:39:43</td>
</tr>
<tr>
<td>14</td>
<td>-19.77</td>
<td>-178.25</td>
<td>556</td>
<td>5.04</td>
<td>27-Jun-2006</td>
<td>02:59:15</td>
</tr>
</tbody>
</table>
Figure 2. This map shows the 15 PDE events (in red) with Mw >6 during the time period between June 7 and 21, 2005, and between June 14 and 30, 2006. The matching SEL0 events are shown in green. Owing to a hardware failure, the SEL0 processing could not process the time period including the Molucca Sea event. Similarly, due to late processing, one Indonesian event on June 27, 2006, was not obtained by SEL0. All other 13 PDE events were matched by corresponding SEL0 events. Note that some of the events are close together and that the symbols are superimposed on the map.

False Alarms for Test Periods between June 7 and 21, 2005, and between June 14 and 30, 2006

The rate of false alarms observed for SEL0 magnitudes mb 5 and above (IDC mb magnitude) was 40% for the time period in June 2005 (see Figure 3). This was significantly less than the false alarm rate observed for standard IDC operations (including all magnitudes) but not sufficient to reliably issue an alert based purely on this magnitude estimation. Most of the false alarms (14 out of 16) have only 1 or 2 defining station mb magnitudes (9 have 1 defining station mb, and 5 have 2 defining station mb).

In the meantime, we have implemented a quality control module in network processing to eliminate false alarms based on the difference between an estimated local station magnitude Ml and an estimated mb. If an event is defined by fewer than three stations and the difference between an estimated mb and Ml is larger than one magnitude unit, the event is considered to be a false event due to incompatibility in amplitudes between the stations. In addition to this module, we are screening detections based on SNR in the frequency band 0.8 to 4.5 Hz and identifying more of the low signal-to-noise ratio (SNR) detections as noise (Jia, 2006).

The false alarm rate at the level of SEL0 magnitude mb 5 and above has decreased significantly between June 2005 and June 2006, with just 2 false alarms out of a total of 21 events above magnitude 5 (see Figure 4). This is slightly under 10%. We attribute this improvement in part to the introduction of the quality control module eliminating small events with large discrepancies between Ml and mb and in part to an improvement in identifying low SNR detections as noise.

It is still desirable to improve on this magnitude estimation, for instance, by rapidly estimating an Mw magnitude for SEL0 events, given that we know that the IDC mb magnitude scale saturates rapidly for events with Mw > 6. Furthermore, a comparison between IDC and International Seismological Centre (ISC) magnitudes has shown that the IDC estimate of mb is on average smaller than the ISC estimate. We believe that the SEL0 bulletin at this stage could be a useful input to a decision-making process prior to issuing an alert. Its main advantage is its timeliness and
the fact that we have not observed at this stage any missed event (for confirmed events with $M_w > 6$). It would be very advisable to include a quick expert review of the SEL0 results before an alert is released to discard potential false alarms.

Figure 3. This map shows all 40 SEL0 events (in green) with $mb > 5.0$ during the time period June 7 to 21, 2005, and the matching PDE events (in red). Note the false alarm rate of about 40%. The SEL0 bulletin was using data from 41 stations, including some IMS auxiliary stations whose data was requested continuously.
Figure 4. This map shows all 21 SEL0 events (in green) with mb > 5.0 during the time period June 14 to 30, 2006, and the matching PDE events (in red). Note that there are few false alarms during that time period. The SEL0 was using 30 stations during that time period.
Figure 5. This histogram shows the distribution of events during the time period June 14 to 30, 2006, according to the time delay at which they were written in the SEL0 origin database table. Note that the majority of events are in the database 20 minutes after they occur.

Figure 5 shows the statistics of the time delays between the origin time and the time at which the events are written to the SEL0 origin database table, as well the equivalent statistics for an intermediate table (SEL0-EWS) that contains all events ever built by the system, as soon as they are built. The average delay is 16.4 minutes for the SEL0-EWS events and 18.9 minutes for the SEL0 events.

CONCLUSION(S) AND RECOMMENDATIONS

We have shown that it is possible to make adjustments to the CTBTO IDC software in order to produce a bulletin that can be useful as input to tsunami or other natural disaster warning centers. During relatively stable processing windows in June 2005 and June 2006 in a developmental context, we have shown that the timeliness objective can be obtained with a global bulletin produced within 20 minutes of real time. The SEL0 bulletin did not miss any event with $M_w$ of 6 or more reported by the PDE bulletin during these two time windows with the exception of two events occurring during hardware failure and acquisition delay, respectively. We have seen an improvement in terms of false alarm rate for $mb > 5$ between June 2005 and June 2006. We are working on further reducing this false alarm rate and adding automatic fast computation of an $M_w$ magnitude based on long-period P waves, which will give a better estimate of size for large events.

ACKNOWLEDGEMENTS

We thank Lassina Zerbo, IDC Director, for allowing us to publish this research and for his constant support during the project.
DISCLAIMER

The views expressed in this paper are those of the authors and do not necessarily reflect the views of the CTBTO Preparatory Commission.

REFERENCES

ABSTRACT

This project represents a three-year research effort aimed at improving seismic and infrasonic monitoring tools at regional distances, with emphasis on the European Arctic region, which includes the former Novaya Zemlya test site. The project has two main components: a) to improve seismic processing in this region using the regional seismic arrays installed in northern Europe and b) to investigate the potential of using combined seismic/infrasonic processing to characterize events in this region. In the latter case, we plan on using the northern European seismic array network in combination with infrasonic stations either installed or scheduled for installation in the near future.

During this reporting period, we have implemented basic infrasonic processing software for the Apatity infrasonic array and for the ARCES seismic array. In the case of ARCES, there are currently no infrasonic sensors available (the plans are to install an infrasound array in 2006/2007), but the seismic sensors have proved useful as an initial substitute for detecting and processing infrasonic signals from explosions at local and regional distances. We have developed an algorithm for associating detected infrasonic phases (either by ARCES or Apatity) with regional seismic events detected and located by the on-line Generalized Beamforming (GBF) process which is currently in experimental operation at NORSAR. We searched the GBF bulletin for approximately one full year of data for seismic events at local or near regional epicentral distances to ARCES or the Apatity infrasound array. We found that 944 infrasound signals could be associated with 651 different seismic events from the GBF bulletin. The large majority of these events were confirmed mining explosions, mainly on the Kola Peninsula.

We present results from an analysis of seismic and infrasonic signals from a set of 108 surface explosions in northern Finland, carried out for the purpose of destroying old ammunition. We have used waveform cross-correlation on ARCES seismic recordings to determine very accurate origin times for the explosions. The extremely high correlation coefficients observed for this data set indicate that these explosions are all very closely spaced, probably within an area of some hundreds of meters in diameter. We have used this database to study the stability of slowness estimates for both seismic and infrasonic phases, using ARCES and Apatity array recordings. By analyzing various subconfigurations of the ARCES array, we find that the scatter (standard deviation) in the azimuth estimates for the explosions is about inversely proportional to array aperture. When carrying out a similar analysis of infrasonic data, we find that, in contrast to the case for the seismic P-waves, the azimuth scatter using our f-k estimation process does not decrease when the array aperture increases. Furthermore, the average azimuth remains essentially unbiased both with varying array aperture and with varying filter bands. This is also in contrast to the situation for seismic P-waves, where we have found strong frequency dependent and configuration dependent azimuth anomalies.

The recent upgrade of the Spitsbergen seismic array, which has included installation of five new three-component seismometers, has resulted in a significant improvement of S-phase detection. We demonstrate this improvement by presenting analysis of recent small seismic events on Novaya Zemlya, where three events (of \( \text{mb}=2.2, 2.3 \) and 2.7) were detected by the GBF process during March 2006.
OBJECTIVE

The objective of the project is to carry out research to improve the current capabilities for monitoring small seismic events in the European Arctic, which includes the former Russian test site at Novaya Zemlya. The project has two main components: a) to improve seismic processing in this region using the regional seismic arrays installed in northern Europe and b) to investigate the potential of using combined seismic/infrasonic processing to characterize events in this region. In the latter case, we plan on using the northern European seismic array network in combination with infrasonic arrays either installed or scheduled for installation in the near future.

RESEARCH ACCOMPLISHED

Infrasound Data Processing using Apatity and ARCES Array Data

The Apatity infrasound array is a three-element array co-located with the nine-element Apatity short-period regional seismic array, which was installed in 1992 on the Kola Peninsula, Russia by the Kola Regional Seismological Centre (KRSC). For further details see Baryshnikov (2004).

The 25 element ARCES array is a short-period regional seismic array, located in northern Norway. ARCES has no infrasound sensors, but because of special near surface installation conditions, many of its seismic sensors are also sensitive to infrasound signals. The seismic sensors have therefore proved useful as an initial substitute for detecting and processing infrasonic signals from explosions at local and regional distances (see e.g., Ringdal & Schweitzer, 2005). Current plans are to install an infrasound array near the ARCES site in 2006/2007.

In this study, we have developed an initial STA/LTA-based infrasonic processing system for the Apatity infrasound array and for the ARCES seismic array. We have also developed an algorithm for associating detected infrasonic phases (either by ARCES or Apatity) to regional seismic events generated in the on-line Generalized Beamforming process which is currently in experimental operation at NORSAR. Some preliminary results are summarized in the following (for details, see Schweitzer et. al., 2006).

On the average, 23.4 infrasound signals per day were observed with the Apatity infrasound array and 7.6 signals per day with the ARCES array. These numbers of observations result from applying only an initial set of infrasound signal processing rules. We want to determine how many of these infrasonic signals can be associated to sources already known from their seismic signals. To investigate this question in more detail the following test was performed:

The Generalized Beamforming (GBF) algorithm (Ringdal and Kværna, 1989) integrates automatically all observations of local and regional phases from all seismic arrays analyzed at the NORSAR data center in one common bulletin, associates these observations to their common sources, and locates these seismic sources. It can be assumed that this bulletin is quite complete and that it is representative for local and regional seismic events in Fennoscandia and the European Arctic with local magnitudes above 1.5 in on-shore regions and above 2.5 overall. At large distances from the arrays, the threshold could be higher. We searched the GBF bulletin for the first 351 days of the year 2005 (until the 17th of December) for seismic events at local or near regional epicentral distances to ARCES or the Apatity infrasound array. The following association criteria were used to correlate seismic events with presumed, corresponding infrasound signals:

- The epicentral distance of the event must be within 500 km from the array.
- The possible onset time of the infrasonic signals was set to be within the time window spanned by group velocities between 0.2 and 0.7 km/s.
- The difference between the event backazimuth and the backazimuth observed for the infrasonic signal should not be larger than 20 degrees.
Figure 1. GBF bulletin of an event in the Khibiny Massif, Kola Peninsula with associated infrasound signals (marked as Ix) observed at the Apatity infrasound array and at ARCES.

Applying these rules, 944 infrasound signals could be associated to 651 different events of the GBF bulletin. For these 651 events we obtained the following statistics:

- 333 events could be associated only with infrasound signals observed at the Apatity infrasound array.
- 250 events could be associated only with infrasound signals observed at the ARCES seismic array.
- 68 events could be associated with infrasound signals at both arrays, the ARCES seismic array and the Apatity infrasound array.

Figure 1 shows the GBF bulletin entry for an event in the Khibiny Massif, Kola Peninsula, for which infrasound signals were observed at both arrays. The source area is known to have numerous large explosions in open pit mines. The associated infrasound signals show quite small backazimuth residuals, the SNR of the observed infrasound signals at both arrays is of the same order as for the seismic signals, and at both arrays, the infrasound waves are arriving in different onset groups within a time window of 1 to 2 minutes.

Figures 2 and 3 show the results of the associations described above. We note that the seismic events with associated infrasound observations are concentrated around known mining areas. We further note that all of these associations are automatic, and have not been reviewed by an analyst. Nevertheless, we are confident that the vast majority of these associations are in fact real. Further work will include detailed review and statistical analysis of results from this association process.

Case Study of Explosions in Northern Finland

Each year between mid-August and mid-September, a series of explosions in the north of Finland is recorded by the stations of the Finnish national seismograph network and also by the seismic arrays in northern Fennoscandia and NW Russia. Based upon event locations given in the seismic bulletin of the University of Helsinki, the geographical coordinates of the explosion site are assumed to be approximately 68.00ºN and 25.96ºE. The explosions are carried out by the Finnish military in order to destroy outdated ammunition and are easily identified from the automatic seismic bulletins at NORSAR for several reasons. Firstly, they are always detected with a high SNR on the ARCES array, secondly they register very stable azimuth estimates on the detection lists, and thirdly they take place at very characteristic times of day (the origin time indicated by the seismic observations almost invariably falls within a few seconds of a full hour, or half-hour in the middle of the day). A preliminary list of candidate events was obtained by scanning the GBF automatic detection lists for events which appeared to come from the correct region at appropriate times of day.
Figure 2. The map shows the 651 automatically located events (GBF) for which either the ARCES seismic array or the Apatity infrasound array observed infrasound signals. The blue triangles show the GBF event locations and the red stars show the location of known sites with explosions either at the Earth’s surface or in the atmosphere. Note that the automatic GBF locations usually scatter over a larger area around these source regions. Also note that the GBF locations employ a fixed grid, and that many of the grid points shown on the map have a large number of corresponding events.

Figure 3. This map is similar to Figure 2, and shows the 68 automatically located events (GBF) for which both the ARCES seismic array and the Apatity infrasound array observed infrasound signals. The blue triangles show the GBF event locations and the red stars show the location of known sites with explosions either at the Earth’s surface or in the atmosphere.
Between 2001 and 2005, a total of 108 events were found which appeared to fit the general attributes of explosions from this site; the GBF location estimates for these events are displayed in Figure 4. These fully automatic estimates display a somewhat surprisingly large geographical spread and, assuming that these events are in fact essentially co-located, the origin times will be correspondingly spurious. Before we proceed in attempting to detect and analyse infrasound signals produced from these explosions, we must first confirm that all of our candidate events are in fact from essentially the same location and then obtain the best possible origin time for each event. To this end, we applied a waveform correlation procedure, which confirmed that the explosions were indeed closely spaced, probably within an area of some hundred meters in diameter (for details, see Ringdal and Gibbons, 2006).

Figure 4. Estimated location of the explosion site in northern Finland (orange diamond) in relation to the seismic arrays ARCES and Apatity together with the GBF fully automatic location estimates for 108 candidate events between August 2001 and September 2005 (green diamonds). The regular pattern of event location estimates is due to the fixed-grid trial epicenter procedure employed by the GBF.

Thus, this data set of more than 100 surface explosions in almost exactly the same place recorded by the ARCES and Apatity arrays provides an excellent opportunity to investigate the stability of slowness estimates, both for the seismic and infrasonic recordings. The paper by Ringdal and Gibbons (2006) presents results on the effects of filter frequency band, array aperture and number of sensors at both the Apatity and ARCES arrays. In this paper we will focus on using various sub-configurations of ARCES to simulate array configurations of various diameters and number of sensors.

Figure 5 shows the ARCES slowness estimates for the event set as a function of various sub-configuration of vertical-component seismometers. These are, in increasing sizes:

- The 4-element A-ring configuration (seismometers A0, A1, A2, A3)
- The 9-element A,B-ring configuration (by adding the seismometers B1-B5)
- The 16-element A,B,C-ring configuration (by adding the seismometers C1-C7)
• The 25-element A,B,C,D-ring configuration (comprising the full ARCES vertical-component array)

As expected, the scatter of the estimates decreases as the array size and number of seismometers increases, and the amount of decrease in the standard deviations is about proportional to the increase in array diameter. We note that the mean azimuth estimates show significant differences among the array configurations, even if we are applying the same bandpass filter (3-5 Hz) throughout.

Figure 5. Seismic slowness estimates of the 108 events in the data base. The figure corresponds to estimates for the seismic P-phase (25-35 seconds after the event origin time), in the filter band 3-5 Hz. The four subconfigurations are as described in the text. For each subconfiguration, the mean and standard deviation of the azimuth estimates are indicated.

We carried out a similar study of slowness estimates for infrasonic waves recorded at the ARCES seismic array. In this case, we used throughout a 60 second window beginning 620 seconds after the event origin time. Figure 6 shows the ARCES slowness estimates for the infrasonic phases (named Ix) as a function of the same sub-configuration of vertical-component seismometers as used in our studies of P-waves described above.

In contrast to the P-wave analysis, we were not able to make reliable slowness estimates for the infrasonic phases of all the events. This is mainly due to low infrasonic SNR for a number of the events in the database. This makes a
comparison between the performances of different filters and subconfigurations more complicated, and we need to consider both the number of successful estimates and the variance reduction when evaluating the results.

Figure 6. Infrasonic slowness estimates of the 108 events in the data base. The figure corresponds to estimates for the infrasonic phase (620-680 seconds after the event origin time), in the filter band 2-7 Hz. The number of events for which reliable estimates could be made is indicated on each plot. The four subconfigurations are as described in the text. For each subconfiguration, the mean and standard deviation of the azimuth estimates are indicated.

When comparing the infrasonic results to those obtained for seismic P-waves, we see some interesting differences. For example, we see no significant variance reduction as the array aperture and number of sensors increases. Although there appears to be a slight reduction in the standard deviations, the largest number of successful estimates were in fact made using the smallest configuration. Therefore we consider that there is essentially no difference in the stability of the slowness estimates for these four configurations. It is of course possible that other estimation techniques could show such improvements, but it may also be that the variance in estimates is dominated by factors such as varying atmospheric conditions over the 5 years covered by this study. Another important observation is that the average azimuth values are essentially independent of the subconfiguration chosen. This also contrasts to our observations from seismic P-waves.
Detection of Small Seismic Events near Novaya Zemlya

The recent upgrade of the Spitsbergen seismic array, which has included installation of five new three-component seismometers as well as an upgrading of the sampling rate from 40 to 80 Hz, has resulted in a significant improvement of S-phase detection. We demonstrate this improvement by presenting analysis of recent small seismic events near Novaya Zemlya, where three events (of mb=2.2, 2.3 and 2.7) were detected by the GBF process during March 2006 (Table 1).

Figure 7 shows spectrograms of the Spitsbergen B1 seismometer (vertical, radial and transverse components) for the Novaya Zemlya event on 5 March 2006. The most noticeable feature is the high SNR of the P-phase for this small (mb=2.65) event. In fact, the SNR on the array beam is above 100, indicating that even an event at this site more than an order of magnitude smaller could have been detected. This should not, however, be extrapolated to a general statement about detection thresholds for the Spitsbergen array, since the SNR to a large extent depends upon path-specific focusing effects. Nevertheless, the amount of high-frequency energy is remarkable, taking into account the large epicentral distance (more than 1000 km). We note that the vertical and radial components have significant P-wave energy even above 20 Hz. The transverse component shows (not unexpectedly) a small P-wave and a much larger S-wave, indicating that the use of transverse components could be useful in detecting S-phases.

This is further illustrated in Figure 8, which shows selected Spitsbergen array beams for the 5 March 2006 Novaya Zemlya event. The top trace is a beam steered to the epicenter with a P-wave velocity, and using a typical detection filter (3-16 Hz). Note that the S-wave on this trace is fairly small, and would give a fairly marginal detection by the automatic process. The middle trace is an “optimum” beam designed to detect the S-wave. It represents the beams of the transverse components of the six three-component seismometers in the array, filtered in the band 2-4 Hz and steered to the epicenter with an S-phase velocity. Note the greatly improved SNR gain on this trace. The bottom trace shows, for comparison, a P-beam of vertical sensors using the same (2-4 Hz) filter. Clearly, the detection of S-phases could be greatly improved by augmenting the beam deployment with several steered beams, rotated so as to provide transverse components, toward the grid points in the beam deployment system.

CONCLUSIONS AND RECOMMENDATIONS

The initial results from associating infrasonic observations to seismic events are promising. We plan in the future to compare in more detail the infrasound observations with analyst reviewed event locations. This will require a review by an analyst of each infrasound signal, in order to confirm their validity and to identify possible misassociations if appropriate. Furthermore, we will implement and test additional infrasonic array detectors, such as the PMCC detector (Cansi, 1995).

The data set of more than 100 surface explosions in northern Finland in almost exactly the same place recorded by the ARCES and Apatity arrays has provided an excellent opportunity to investigate the stability of slowness estimates for seismic and infrasonic recordings as a function of array geometry, number of sensors and filter frequency band. Future work will focus using this database as well as the ground truth data base of mining explosions in the Kola Peninsula to assess the detectability of infrasonic phases under various atmospheric conditions.
The new Spitsbergen array configuration has shown excellent recordings of high-frequency data from Novaya Zemlya events. The new three-component instrumentation provides a great potential for improving S-phase detection at this array, and an enhanced S-phase detector will be implemented in the near future.
Figure 7. Spectrograms for the Spitsbergen B1 seismometer (vertical, radial and transverse components) for the Novaya Zemlya event on 5 March 2006.
Figure 8. Spitsbergen array waveforms for the 5 March 2006 Novaya Zemlya event. Note the greatly improved SNR gain for the Sn phase shown in middle trace, which represents the beams of the transverse components of the six three-component seismometers in the array.

REFERENCES


ADVANCES IN DATA INTEGRATION AND QUALITY CONTROL IN SUPPORT OF THE NNSA KNOWLEDGE BASE

Richard Stead, Michael Begnaud, and Julio Aguilar-Chang
Los Alamos National Laboratory

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Development
Office of Defense Nuclear Nonproliferation

Contract No. W-7405-ENG-36

ABSTRACT

The goal of the NNSA Ground-based Nuclear Explosion Research and Engineering program (GNEM R&E) is to develop, demonstrate, and deliver advanced technologies and systems to operational monitoring agencies to support ground-based detection, location, and identification of nuclear explosions. One such system is a custom-designed data storage and access system known as the NNSA Knowledge Base (KB), primarily based on relational database schemas (sets of table structures). The GNEM R&E research conducted at the national laboratories to populate the KB requires collection and integration of a remarkably large and diverse collection of geophysical data to develop the types of products needed to improve monitoring capability. The size and diversity of these data present substantial technical challenges to achieve complete, correct, consistent, useful, and accessible information. These data are processed by the labs to produce the higher-level engineering products (e.g., travel time correction surfaces) that are needed for operational monitoring, but the basic data must also be included in the KB to fully test and verify the operational products. Without the supporting data and metadata capturing the processing details, the operational engineering products cannot be validated and thus will not be used for operations. Los Alamos National Laboratory (LANL) has developed and contributed to several versions of the Knowledge Base and in the process we have developed and refined a substantial foundation of software, structures, and procedures to assure high-quality integration of diverse data sets. Software advances include generalized database interfaces and generalized quality assurance/quality control (QA/QC) software. Structural advances include a metadata abstraction of supporting structures (themselves metadata) that we refer to as the schema schema. Procedural advances leverage the software and structures to create robust procedures for definition and transfer of data between groups.

The development and application of automated QC software is the primary topic of this paper. Attention to quality, particularly in the supporting data, has been a subject of growing importance and focus recently. Dealing with data quality in an established KB is difficult and time-consuming. A better approach is automated QC of supporting data before they are integrated into the KB. We have taken several steps in this area over the past year, including the development of automated quality-inspection software. The first critical step in automating QC was to make the information about the schema readily available to the QC software, which was done by developing a set of tables describing the schema itself, or a "schema schema." The schema schema captures information about the content of each table (what the columns are), the relationships between the tables, and information about each column (definition, acceptable range of values). With this in hand, we then developed a Perl-based QC tool to check the content of any set of tables against what is in the schema schema. The software performs three basic types of checks: 1) validity of column data within each table, 2) consistency of column data between related tables, and 3) more complicated consistency relationships between related tables. Because the software makes use of the schema schema, it was written in a generic manner and thus can be used for virtually any sort of check without modifying code. A user sets up a parameter file to designate the database tables that will be checked, writes the specific checks for #2 and #3 above, and then initiates the QC check. The output can then be used to direct the KB integrator to problems in the incoming data. The QC tool was used extensively for the production of the latest KB release with excellent results. Thousands of problems were found and fixed prior to integration thereby greatly improving the quality of the KB and dramatically reducing the amount of post-delivery editing.
OBJECTIVES

The GNEM R&E program has made recent progress in developing and automating QC of the NNSA KB and in the supporting metadata architecture. The goal is to improve the quality of data and derived calibration results in the KB, as well as improving the speed and accuracy of the procedures for incorporating new data and derived calibration results. A related goal is to improve human interfaces to increase the efficiency and effectiveness of KB integrators.

RESEARCH ACCOMPLISHED

Background: The Need for Quality Control

Quality control in this paper will refer to the quality assurance and quality control of both geophysical data and research products derived from those data. QC for the GNEM R&E KB is an issue that has increased in priority and significance recently. While QC has never been ignored, the increased maturity of various processes and products developed under GNEM R&E and advancing toward operations has brought necessary scrutiny. This scrutiny is different and far more intense in many ways than that typically experienced in the research community. At the same time, the volume and diversity of data and products is challenging by any standard (Stead, 2006).

A variety of typical methods of QC exist that are adopted in smaller research efforts. One is to include manual review of every field for every element of data and products by the researchers. This method works well, since the researchers are the experts and will immediately recognize errors, but it is only viable for very small and focused collections of data and products. A second method is sampling, where the expert examines a small sample of data and results to verify the correctness of these, and projecting the QC result to the whole. If properly applied, this can be a viable method for the QC of data, even large data sets. But it takes an intimate knowledge of the particular data source to know how best to properly sample the data and have confidence in the results. It is not a good approach for research products, since it implicitly assumes random errors and normal distributions. A third method is to develop a dedicated application to QC a particular type of data or research product. But this exercise then must be repeated for every new type of data or research product, and the dedicated applications require constant maintenance to account for changes in data or products. This is not meant to be an exhaustive list, but merely a representative sampling to indicate the disadvantages of such ad-hoc approaches when the QC problem deals with very large and diverse data and research products, many of which are inter-dependent, where there are frequently new kinds of information or important changes to existing information, and where much of the information is targeted for operational purposes (see Figure 1).
Faced with this situation, LANL has been developing more comprehensive and flexible QC that can be automated. We have also worked closely with Sandia National Laboratories (SNL) on these issues, but this paper will primarily deal with the work done at LANL.

**Infrastructure for QC**

Any generalized, automated QC capability will require a certain amount of information infrastructure to support the process. At a fundamental level, such QC will require information that describes what qualified data and products are. A generalized QC process then merely has to match the data and products to the information and indicate which elements violate the qualification. The automated, generalized QC will only reach a level of achievement that the information infrastructure on which it is built will permit. Therefore, careful consideration must go into the construction and population of the infrastructure.

Some requirements for the infrastructure are that it be

- sufficiently general to describe all conceivable types of data and products;
- simple to understand and populate, directly or through user interfaces;
- simple to maintain, directly or through user interfaces;
- capture as much information about what qualifies data and products as possible; and
- multipurpose, so that any effort in populating and maintaining it can benefit other work.

To this end, LANL initiated the development of a database-based metadata model to support QC and other database development work beginning about two years ago. The basic concept behind this metadata model is that it is a metadata model to describe other metadata, and because of this design, it has come to be referred to as the “schema schema” (i.e., a schema to describe schemas). It is simple in that it has only four objects in the schema: an object to
describe tables, an object to describe columns, an object to describe the association between tables and columns, and an object to provide definitions of fixed values that columns may take. Additional objects can be added that relate to these basic four, to handle other special-purpose tasks (one is the complexjoin table which will be described below).

This concept was deliberately designed to have a close relationship to the Oracle data dictionary, to make the use and understanding of the schema schema simple and straightforward. However, it does go well beyond the Oracle data dictionary in various ways. In particular, because it exists apart from the tables it describes, it exists at all times regardless of which objects are currently defined or how they are defined. It also supports all of the concepts embedded in earlier documentation of the KB schema, including external flat-file formats and NA values.

The structure is quite amenable to web-based interfaces, making the population, maintenance, and visual interface simple to use and understand.

The structure is also multipurpose. By designing it to capture the information in previous database description documents, providing the capability to maintain multiple versions of such information in a single place and with a single interface, and providing simple tools to maintain the information, the schema schema has replaced the flat-file documentation as the medium of choice for direct maintenance of all schema information and documentation. LANL has developed web-based interfaces that allow users to quickly view database documentation in a familiar format, directly out of this schema, eliminating the piles of massive paper documents that had to be used in the past. SNL has also developed additional interfaces for viewing and maintenance. In addition, the schema provides a direct and simple means for software of all types to determine the objects and there structure at run time, as opposed to maintaining software-based descriptions of all of this information. SNL has made extensive use of this feature in the development of their Java-based dbtoolkit. LANL is using this feature in a variety of Perl-based software.

Figures 2–4 show the LANL-developed web interface for viewing the table descriptions and column associations, the column descriptions, and the glossary (the 4 core schema schema tables). See Begnaud, et al., 2005 for additional information on the schema schema and the web interfaces to the KB.

Figure 2. Table description web GUI. This web panel shows the complete description of a table from a particular schema, along with the column associations for the table and SQL statements and format lines that correspond to the table.
Figure 3. Column description web GUI. This web panel shows the complete description of a column from a particular schema, along with the table associations for the column.

Figure 4. Glossary web GUI. This web panel shows the complete description of selected glossary entries. The entries may be selected by schema name, table name, and column name, or by the search feature.
Automated QC

A first-generation QC tool has been developed at LANL. The tool leverages the schema schema described above to facilitate automated inspection of data and derived products that exist in database tables. The tool can operate on any collection of tables, but it can only do the most basic kinds of inspection unless the tool can match each table to a corresponding description in the schema schema. The tool is applicable in a variety of situations. A version has been adopted by the DOE labs, as a group, to handle QC of deliveries between the two science labs and SNL. LANL has found the tool to be of considerable use in dealing with raw or ‘roughed-in’ data sets and research products. Running the tool on such information allows a wide variety of issues to be identified quickly and corrections then planned and made separately. The results also then are used as feedback to improve the conversion of data sets from their raw form into the KB, and also to improve both the structure of research products, and even the processes that generate research products. The tool has significant advantages over ad-hoc QC, in that it adapts automatically to changes in the underlying schemas, no specialized software is required if new data or products become available, and there is no need for tedious manual inspection or for experts to remember or track all of the possible QC errors that may arise.

The automated tool requires very little information to run. The basic run-time parameters are the connection to the database and the list of tables to be examined. Currently, if cross-reference checks are desired, these must also be specified individually, since there is no generalized table in the schema schema to describe these. In practice, a user of the tool will keep a list of cross-reference parameters, and copy them as needed into a run-time parameter file for a particular set of tables to be examined. The tool learns everything else it needs from the database. The tool executes three stages of QC checks: single-table checks, cross-reference checks, and complex joins.

The single-table checks consist of four table-based checks (beyond simple existence of the table), followed by a large number of column-by-column checks (that depend on the type of column), for each table specified. First, the tool determines the standard names of the schema and table. This verifies that the table matches a documented table structure and provides the tool access to a complete description of that table and its columns, glossary entries and complex joins. It counts the number of rows (a primitive check – a lack of rows or too many may indicate a problem), and then validates the table’s primary key, and its unique key (if any). That completes the table-based checks. The column-based checks may be basically divided into general checks and then checks performed only on string columns and only on numeric columns. The general column checks include counting the unique values, NA values and nulls, determining the minimum and maximum of the column, and finding duplicated values. None of these may be errors, but they can indicate errors when subsequently reviewed. The character column checks include checks for bad character strings, strings that do not fit a columns regular expression (if any), and comparisons to the glossary for columns that have defined or reference set values. The numeric column checks include various range checks, checks for NA values other than that defined for a column, and checks for negative values.

The multi-table or cross-reference checks are individually specified and executed. There is no need for the tables in the cross-reference checks to be specified in the list of tables to check. This allows checks against reference tables, without having the tool perform full checks (single table and complex join) of the reference table itself. There are three basic kinds of checks: single column cross-references between tables, two-column cross-references between tables, and indirect cross-references. A single-column cross reference requires that every value in the specified column of the table being checked is also found in the specified column of the table that the first table is to be compared against. There are different versions of this check, since the underlying queries may be faster or slower under different circumstances. The two-column check is the same as the single-column check, except that each unique pair of values from two specified columns in the table being checked must be found in the two column specified for the comparison table. The indirect cross-reference check handles cases where the reference for a column in a table being checked is specified by a second column in the same table. The archetypical example of this is the wftag table, where the reference for the tagid column is specified by the tagnname column; that is, if tagnname equals ‘evid’, then tagid contains an evid and should be compared to columns in other tables that are evids (like event.evid), but if tagnname equals ‘arid’, then the contents of tagid must be compared to arids (like arrival.arid).

The third and final checks are the complex joins. These are specified in an auxiliary table to the schema schema. This table can specify up to three tables to be involved in the join from the list of tables being checked. The QC tool will automatically format and run a check if the kinds of tables the check requires are among the list of tables the user provided. Since the table contains a generalized SQL query, these checks can be as complex as SQL permits. Some simple examples include verifying that a date in a table corresponds to the time specified in the same table,
verifying that the contents of a field in one table correspond to those in another, when there is no direct link between the tables (such as `wfdisc.instype` versus `instrument.instype`), and verifying that a count listed in a field of one table equals the count of those objects in another table (such as `origin.ndef` versus the count of time-defining arrivals in `assoc`).

Figures 5 and 6 provide examples of the QC tool output.

---

**TABLE AMPLITUDE (658054 rows):**

* 'AMPLITUDE' found as 'amplitude' in 'USNDC P2B2'  
  no PK violations  
  [keychk: 6 s]

---

**COLUMN UNITS:**

NA value is '-', and allowed is n  
Unique values = 2 of 658054 total  
NA values ('-') = not allowed  
Min = nm  
Max = nm/s  
Most common repeated values, top 10:

<table>
<thead>
<tr>
<th>VALUE</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>nm</td>
<td>413060</td>
</tr>
<tr>
<td>nm/s</td>
<td>244994</td>
</tr>
</tbody>
</table>

244994 (1 distinct) entries not in glossary  
Unreferenced values, top 10:

<table>
<thead>
<tr>
<th>VALUE</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>nm/s</td>
<td>244994</td>
</tr>
</tbody>
</table>

No regexp  
No bad string values.  
[bvchk: 0 s]

---

**Figure 5. QC tool: single-table check stage: example output.**
... 

487 (479 distinct) entries in stamag (arid, delta) not in assoc (arid, delta) [ref2chk 130: 1 s]
Most common failed references, top 10:

<table>
<thead>
<tr>
<th>ARID</th>
<th>DELTA</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>65206475</td>
<td>1.009</td>
<td>2</td>
</tr>
<tr>
<td>65206479</td>
<td>1.194</td>
<td>2</td>
</tr>
</tbody>
</table>

... 

'origin.nass = the count of rows in assoc for the orid' 
Enforce: soft 
Checking complex join 27 'ORIGIN.nass' 
= (select count(*) from ASSOC s where s.orid=a.orid) 
991 (69 distinct) values violate join, top 10:

<table>
<thead>
<tr>
<th>VALUE</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>101</td>
</tr>
<tr>
<td>8</td>
<td>87</td>
</tr>
<tr>
<td>7</td>
<td>68</td>
</tr>
<tr>
<td>9</td>
<td>66</td>
</tr>
<tr>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td>61</td>
</tr>
<tr>
<td>11</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>36</td>
</tr>
<tr>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>27</td>
</tr>
</tbody>
</table>

[complexjoin_chk 27: 7 s] 

Figure 6. QC tool: cross-references and complex joins: example output.

Limitations

This is a first-generation QC tool and there are a variety of limitations. Two limitations are prominent. The first is that this is a QC inspection tool, not a QC inspection and repair tool. The second is that this QC tool is limited to inspection of data that are in the database. The NNSA KB schema (Carr, 2005) contains objects, principally instrument and wfdisc, which point to and describe external data. Other tools are required to incorporate these data into overall QC review.

Some tools exist to handle these limitations. In particular, we have recently extended the LANL web interface to the KB to include waveform review. Waveforms may be viewed in a browser, using basic selection criteria, and other information, such as amplitude measurement windows may be overlaid on these images to provide a simple visual inspection (see Figure 7).
CONCLUSIONS AND RECOMMENDATIONS

We have developed a metadata schema to describe metadata, which has matured to the point now that it undergirds a variety of software within the GNEM program. It is now a component that is helping to meet a variety of tasks and requirements in a manner that improves efficiency and allows work done in documenting and maintaining metadata information to have immediate benefits elsewhere.

We have developed a first-generation QC inspection tool that relies in the schema schema. QC, which is always important, has become prominent in GNEM recently. Metadata and the proper handling of metadata are the keys to getting a handle on QC, particularly for large and diverse collections of data and research products like the GNEM KB. The QC tool has proven to be very useful in a variety of situations, and has now been deployed throughout the GNEM labs. We expect that the tool will undergo further development, based on experience, to increase both ease of use and automation.

The QC efforts have not yet extended to automated or semi-automated correction of errors. This is an obvious next step. We plan to investigate this subject as well, and we expect to find that a variety of QC problems can be addressed at least in a semi-automated fashion. By semi-automated, we mean that general rules for the repair of problems can be created that can then be adapted to particular problems in software, and applied to those problems...
given simple approval by an expert. This differs from manual repair in that the repair does not need to be formulated from scratch each time.

**ACKNOWLEDGEMENTS**

The authors wish to acknowledge LANL personnel who are making or have made important contributions to the development and maintenance of the LANL KB, and have provided helpful discussions on these and related topics: Diane F. Baker, Hans E. Hartse, W. Scott Phillips, George E. Randall, and Marian D. Romero-Yeske. We also wish to acknowledge helpful discussions with SNL personnel, which improved the work reported here and its more general application: Sandy Ballard, Dorthe Carr, Marcus Chang, Jennifer Lewis, and Chris Young.

**REFERENCES**


