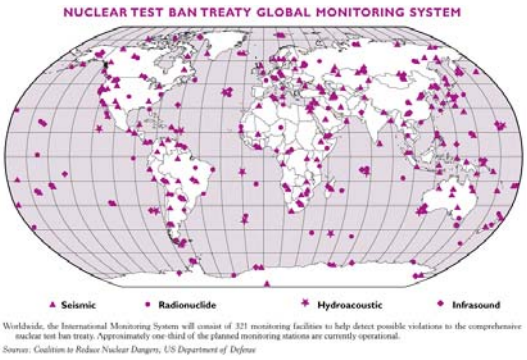
Java Viewer

For Radionuclide Data Messages

I Abstract

The global monitoring of nuclear explosions is an important task which requires the assistance of up to date, easy to use software that can organize and generate visuals from Beta-Gamma data messages. The Java Viewer was not only designed to provide an accurate representation of the data, but also to be aesthetically pleasing and user-friendly. The project was developed in the Eclipse IDE, using the QT / Jambi toolkit and the Subclipse plugin for version control. While the Java Viewer is really a project to which ongoing modifications will be made, the work done thus far has provided both the basic functionality and numerous extra features. Software such as this may eventually be used in a CTBT (Comprehensive Test Ban Treaty) monitoring system to assist the effort to enforce the ban on nuclear testing.

II Introduction



Worldwide, the International Monitoring System will consist of 321 monitoring facilities to help detect possible violations to the comprehensive nuclear test ban treaty. Approximately one-third of the planned monitoring stations are currently operational.
Source: Coalition to Reduce Nuclear Dangers, US Department of Defense

The CTBT (Comprehensive Test Ban Treaty) bans "all nuclear weapon test explosion or any other nuclear explosion," in an attempt to ensure global safety. As part of that effort, a global monitoring system has been (and is continually being) set up to make sure that the ban is enforced. Stations have been set up worldwide that continually take samples using radioxenon analyzers such as ARSA (Automated Radioxenon Sampler / Analyzer) or SAUNA (Swedish Automated Noble gas Analyzer), which provided the data used to test the Java Viewer at PNNL. When one (and inevitably more) or these stations detects activity indicative of a nuclear explosion, the time it takes for each station to pick up that activity, as well as meteorological data, can be used to determine exactly where the explosion in question could have taken place. The Java Viewer reads files from a database that were produced by a machine such as ARSA or SAUNA, displays the information in the files, and uses it to generate helpful graphs and histograms. Its three-dimensional color-coded histograms make it easy to see if a sample contains very much activity or not, which generally makes the monitoring process easier.

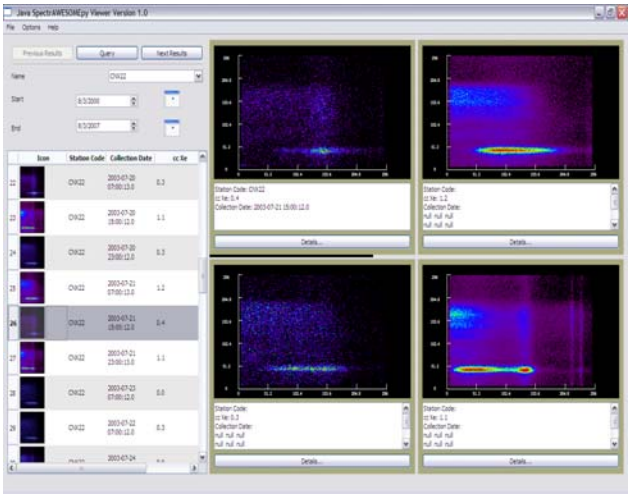
III Software and Toolkits Utilized

The Java Viewer was obviously coded in Java, and used the QT / Jambi toolkit for its GUI elements. The Jambi toolkit's Widgets, like Java, work on basically any platform, and include conveniences such as a pre-made calendar Widget. Popular software that also utilizes the Jambi toolkit includes Google Earth, Motorola A760, and Photoshop Elements. The Subversion Eclipse integration (Subclipse) was used as version control for this project, and the actual coding was done in the Eclipse IDE. The Viewer connected to an Oracle database and queried its tables using SQL.

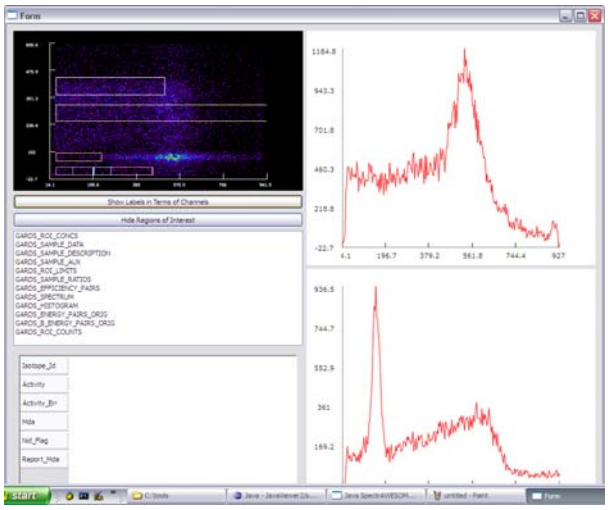
IV Project Challenges

There was already software in existence at PNNL whose functionality is almost identical to that which was expected of the Java Viewer. However, that viewer was coded in C#.NET, and the current client specified that the project be completed in Java. In the older software, calculations were performed using C++ code which was already written and completely accurate. Although it is possible to try and translate the C++ to Java for the new viewer, it would be entirely possible that through the translation, the code would lose some of its accuracy or functionality. So, one of the expectations for the Java Viewer was to include the C++ code in the software that was coded in Java. Unfortunately, this problem was solved only temporarily by entering the calculations into the tables in the database to be read, instead of calculating them within the program.

V Project Screenshots and Description of Features



The Java Viewer's main interface has several elements, but the first that is noticed / used is the form in the upper right hand corner for specifying which records to query from the database. When the default database is loaded at the beginning of the program's execution, or if the user goes to File > Open Database… and successfully connects, the drop down box labeled "Name" will fill with all of the possible station codes in that database. The user can also specify a start and end date to bound their query by date in which the data was collected. For convenience, buttons with calendar icons are located next to the start and end date fields. When clicked, they pop up a calendar that can be used to select dates more quickly and easily. The user can also specify the maximum amount of records in the query's result set by going to Options > Set Number Of Records In Query. Once the query has been executed, the next or previous set of that size can be brought up using the Next Results and Previous Results buttons. With each query, the result set will be displayed in a table immediately underneath the query form. Each record will be shown as a row with a thumbnail of its two-dimensional histogram, its station code, collection date, and the cc of Xenon in the sample. If the user wishes to get more information on any record, the records can either be double clicked or dragged and dropped to transfer them one of the four panels on the right hand side. This allows for side by side comparison of up to four records at once. From there, any one record may be viewed in maximum detail by clicking the detail button below its enlarged view. The detail window holds a larger version of the two-dimensional histogram, one dimensional beta and gamma histograms, and the values for every entry in most tables in the database corresponding to that record. Mouse interaction is available for the one dimensional histograms; the user can click and drag a box to the size that they would like to zoom into, and right click to zoom back out. Buttons are available to toggle back and for the between displaying either energy units or channel numbers on the axis of the histograms, and to toggle the display of ROIs (Regions of Interest) on the two dimensional histogram only.



VI Future Revisions to Project

There are still many features that it would be ideal to have added to the Java Viewer. Among those are increased mouse interaction with the histograms in the detail window, including the ability to zoom in on the two-dimensional histogram as is currently possible with the one-dimensional histograms. Another revision to made as far as mouse interaction is the ability to click-and-drag to resize the regions of interest to the user's liking on the two-dimensional histograms. Of course, there are plenty more ideas that could be later added to improve the project.

VII Conclusion and Acknowledgements

In conclusion, it is always important to be constantly revising old software and developing new software to allow for as much functionality as possible. With more development, the Java Viewer will be able to provide much help to those tracking activity from nuclear explosions at CTBT monitoring stations.

Thanks to my mentor, Brian Schrom, manager, Mary Peterson, the Electronics & Measurement Systems Group, the National Security Internship Program, and Battelle all for the opportunity to work on this project at Pacific Northwest National Laboratory.