ROADNet

IN13A-1078

Applying a toolkit for dissemination and analysis of near real-time data through the World Wide Web: integration of the Antelope Real Time System, ROADNet, and PHP

Robert L. Newman, Kent G. Lindquist[†], Frank L. Vernon, Todd Hansen, Steve Foley, Jennifer Eakins, John Orcutt

Scripps Institution of Oceanography, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0225

[†] Lindquist Consulting, 59 College Rd. Suite #7, Fairbanks, AK 99701

1. Abstract

The ROADNet project has enabled the acquisition and storage of diverse data streams through seamless integration of the Antelope Real Time System (ARTS) with (for example) ecological, seismological and geodetic instrumentation. The robust system architecture allows researchers to simply network data loggers with relational databases; however, the ability to disseminate these data to policy makers, scientists and the general public has (until recently) been provided on an 'as needed' basis. The recent development of a Datascope interface to the popular open source scripting language PHP has provided an avenue for presenting near real time data (such as integers, images and movies) from within the ARTS framework easily on the World Wide Web. The interface also indirectly provided the means to transform data types into various formats using the extensive function libraries that accompany a PHP installation (such as image creation and manipulation, data encryption for sensitive information, and XML creation for structured document interchange through the World Wide Web). Using a combination of Datascope and PHP library functions, an extensible tool-kit is being developed to allow data managers to easily present their products on the World Wide Web. The tool-kit has been modeled after the pre-existing ARTS architecture to simplify the installation, development and ease-of-use for both the seasoned researcher and the casual user. The methodology and results of building the applications that comprise the tool-kit are the focus of this presentation, including procedural vs. object oriented design, incorporation of the tool-kit into the existing contributed software libraries, and case-studies of researchers who are employing the tools to present their data.

2a. Summary of the ROADNet project

The ROADNet project enhances researchers, policy makers and the general public's capacity to monitor and respond to changes in our environment by developing both the wireless networks and the integrated, seamless, and transparent information management system that delivers seismic, oceanographic, hydrological, ecological, and physical data to a variety of end users in real-time.

2b. Antelope Real Time System (ARTS)

The ARTS software package is a state-of-the-art real-time sensor network monitoring and recording environment. Although primarily developed for seismic data acquisition, the software suite has proven robust and extensible enough to accommodate the acquisition and storage of a wide variety of data types. Within ARTS, data is buffered and transported through a mechanism known as an Object Ring Buffer (ORB).

2c. PHP Hypertext Processor (PHP)

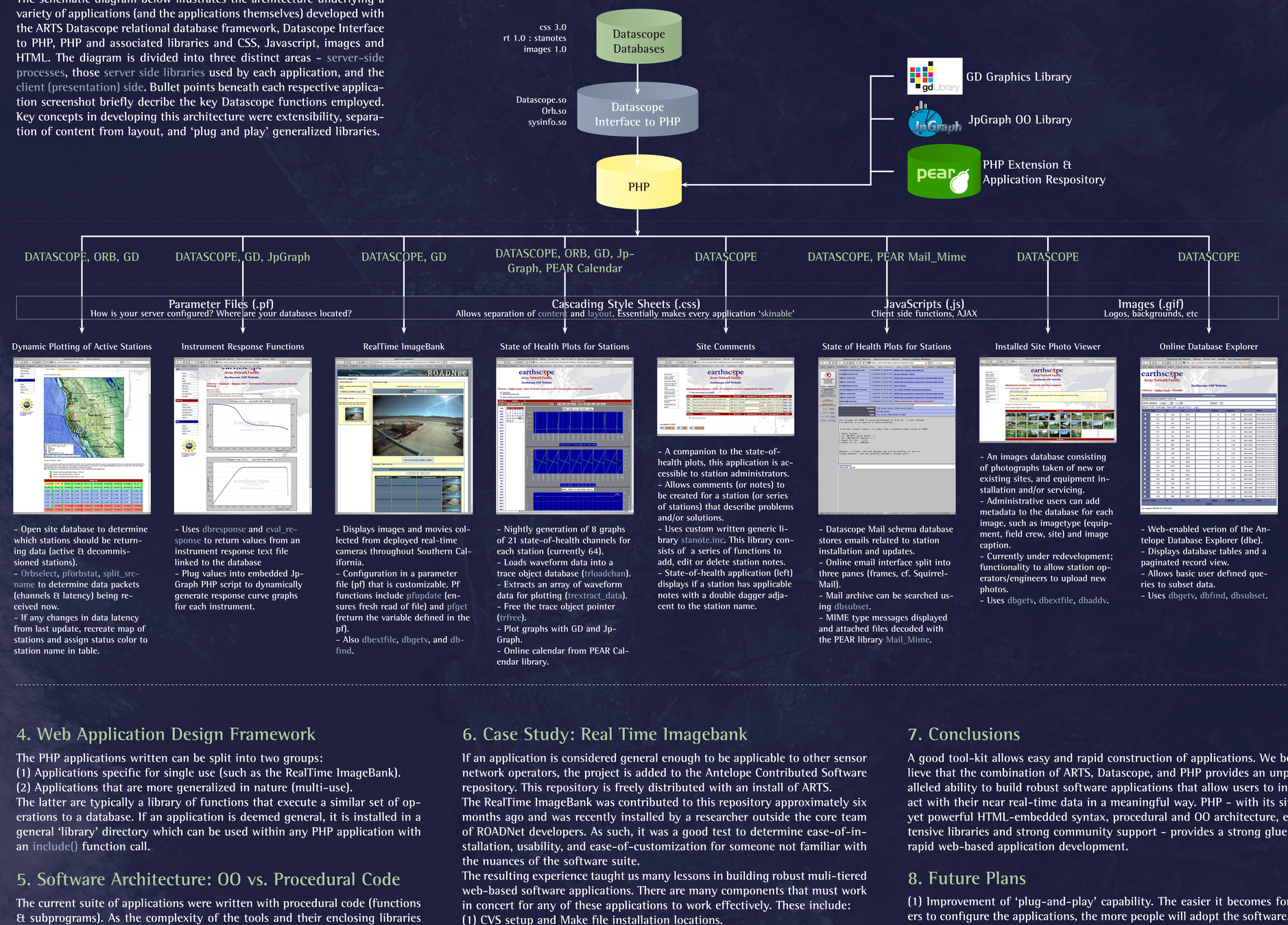
PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. It has been widely adopted as a interface between databases and webpages, with a expansive suite of library tools that encompass database interaction, image creation and manipulation, dynamic graphing, web services, XML parsing, PDF creation and Mathematical functions.

2d. Datascope Interface to PHP

The ARTS software suite provides the acquisition and storage of the various data-types collected (one end of the delivery paradigm). PHP, CSS and HTML provide the manipulation and display of the data (the other end of the delivery paradigm). The 'glue' between the two ends is the Datascope Interface to PHP, which is discussed in a companion poster IN13A-1081 "Improvements to Web Toolkits for Antelope-based Real-time Monitoring Systems", Lindquist et al.

3. Architecture & Example Applications

The schematic diagram below illustrates the architecture underlying a variety of applications (and the applications themselves) developed with the ARTS Datascope relational database framework, Datascope Interface to PHP, PHP and associated libraries and CSS, Javascript, images and HTML. The diagram is divided into three distinct areas - server-side processes, those server side libraries used by each application, and the client (presentation) side. Bullet points beneath each respective application screenshot briefly decribe the key Datascope functions employed. Key concepts in developing this architecture were extensibility, separation of content from layout, and 'plug and play' generalized libraries.



- Open site database to determine which stations should be returning data (active & decommissioned stations).
- Orbselect, pforbstat, split_srcname to determine data packets (channels & latency) being received now.
- If any changes in data latency from last update, recreate map of stations and assign status color to station name in table.

- Uses dbresponse and eval_response to return values from an instrument response text file linked to the database
- Plug values into embedded JpGraph PHP script to dynamically generate response curve graphs for each instrument.

- Displays images and movies collected from deployed real-time cameras throughout Southern California.
- Configuration in a parameter file (pf) that is customizable. Pf functions include pfupdate (ensures fresh read of file) and pfget (return the variable defined in the pf).
- Also dbextfile, dbgetv, and dbfind.

- Nightly generation of 8 graphs of 21 state-of-health channels for each station (currently 64).
- Loads waveform data into a trace object database (trloadchan).
- Extracts an array of waveform data for plotting (trextract_data).
- Free the trace object pointer (trfree).
- Plot graphs with GD and JpGraph.
- Online calendar from PEAR Calendar library.

- A companion to the state-of-health plots, this application is accessible to station administrators.
- Allows comments (or notes) to be created for a station (or series of stations) that describe problems and/or solutions.
- Uses custom written generic library stanote.inc. This library consists of a series of functions to add, edit or delete station notes.
- State-of-health application (left) displays if a station has applicable notes with a double dagger adjacent to the station name.

- Datascope Mail schema database stores emails related to station installation and updates.
- Online email interface split into three panes (frames, cf. Squirrel-Mail).
- Mail archive can be searched using dbsubset.
- MIME type messages displayed and attached files decoded with the PEAR library Mail_Mime.

- An images database consisting of photographs taken of new or existing sites, and equipment installation and/or servicing.
- Administrative users can add metadata to the database for each image, such as imagetype (equipment, field crew, site) and image caption.
- Currently under redevelopment; functionality to allow station operators/engineers to upload new photos.
- Uses dbgetv, dbextfile, dbaddv.

- Web-enabled verion of the Antelope Database Explorer (dbe).
- Displays database tables and a paginated record view.
- Allows basic user defined queries to subset data.
- Uses dbgetv, dbfind, dbsubset.

4. Web Application Design Framework

The PHP applications written can be split into two groups:
(1) Applications specific for single use (such as the RealTime ImageBank).
(2) Applications that are more generalized in nature (multi-use).
The latter are typically a library of functions that execute a similar set of operations to a database. If an application is deemed general, it is installed in a general 'library' directory which can be used within any PHP application with an include() function call.

5. Software Architecture: OO vs. Procedural Code

The current suite of applications were written with procedural code (functions & subprograms). As the complexity of the tools and their enclosing libraries expands, it is becoming increasingly apparent that an Object Oriented (classes & data objects) approach to the code base may provide a more efficient and extensible solution due to data encapsulation methods and code reusability. The advent of PHP (v.5), with its emphasis on a more object oriented coding structure, may provide a catalyst for making the switch. Additionally as the Datascope interface to PHP expands to encompass more ORB functionality, an object oriented solution may be preferred as many of the pre-existing Perl and Tcl functions can be most simply replicated with an adoption of objects. We would be interested to discuss the pro's and con's of these two distinct approaches to coding with the audience.

6. Case Study: Real Time Imagebank

If an application is considered general enough to be applicable to other sensor network operators, the project is added to the Antelope Contributed Software repository. This repository is freely distributed with an install of ARTS.
The RealTime ImageBank was contributed to this repository approximately six months ago and was recently installed by a researcher outside the core team of ROADNet developers. As such, it was a good test to determine ease-of-installation, usability, and ease-of-customization for someone not familiar with the nuances of the software suite.
The resulting experience taught us many lessons in building robust muli-tiered web-based software applications. There are many components that must work in concert for any of these applications to work effectively. These include:
(1) CVS setup and Make file installation locations.
(2) Web server version, configuration and install location; Apache 1.3.* and 2.0.* are supported.
(3) PHP version and compile structure; whether PHP was compiled with GD support, if the image libraries required by GD are installed, defined parameters in the PHP configuration file (php.ini) such as the extension directory definition and included file directories.
(4) Assumed knowledge of the researcher and/or their system administrators: We have gone to reasonable lengths to make sure the applications are as 'plug-and-play' as possible, but there is still an assumption of system architecture knowledge and how the component parts fit together.

7. Conclusions

A good tool-kit allows easy and rapid construction of applications. We believe that the combination of ARTS, Datascope, and PHP provides an unparalleled ability to build robust software applications that allow users to interact with their near real-time data in a meaningful way. PHP - with its simple yet powerful HTML-embedded syntax, procedural and OO architecture, extensive libraries and strong community support - provides a strong glue for rapid web-based application development.

8. Future Plans

(1) Improvement of 'plug-and-play' capability. The easier it becomes for users to configure the applications, the more people will adopt the software, and help improve it.
(2) Persistent database connections for iterative database queries.
(3) Providing downloads of data query results, including any associated metadata.
(3) Round Robin (decimated) databases for time series data.
(4) Online administration areas to allow users to modify server settings without reverting to the command line.