Web Based Prognostics and 24/7 Monitoring

Miryam Strautkalns[1] and Peter Robinson[2]

[1] *SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA*
Miryam.Strautkalns@nasa.gov

[2] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
Peter.I.Robinson@nasa.gov

ABSTRACT

We created a general framework for analysts to store and view data in a way that removes the boundaries created by operating systems, programming languages, and proximity. With the advent of HTML5 and CSS3 with JavaScript the distribution of information is limited to only those who lack a browser. We created a framework based on the methodology: one server, one web based application. Additional benefits are increased opportunities for collaboration. Today the idea of a group in a single room is antiquated. Groups will communicate and collaborate with others from other universities, organizations, as well as other continents across times zones. There are many varieties of data gathering and condition-monitoring software available as well as companies who specialize in customizing software to individual applications. One single group will depend on multiple languages, environments, and computers to oversee recording and collaborating with one another in a single lab. The heterogeneous nature of the system creates challenges for seamless exchange of data and ideas between members. To address these limitations we designed a framework to allow users seamless accessibility to their data. Our framework was deployed using the data feed on the NASA Ames' planetary rover testbed. Our paper demonstrates the process and implementation we followed on the rover.

1. INTRODUCTION

Through observing the vast number of stove-piped data plotting tools used at NASA as well as the current practices for prognostics, diagnostics, and the concerns of their analysts, the framework for Web Based Prognostics and 24/7 Monitoring (WBPM) was created. Most projects use hard-drive based applications for data gathered during testing (e.g. Labview, ICE, C#, Java, and Matlab). Some will have options for delivering data to servers for storage. The dilemma is that the compatibility for most analysts is limited to a single platform. While most software packages provide multiple releases intended for different platforms, the known issues in the program would vary depending on the user's chosen operation system (OS). This inconsistency forces analysts to collaborate on uniform devices and software packages. The limitations change yet again depending on the length of project. Both short-term and long-term projects present their own unique needs and limitations for data acquisition and analysis. These range from reliability of storing data locally to human factors associated with long-term projects. Many analysts will plan a visit with researchers at National Aeronautics and Space Administration (NASA) Ames Research Center and other facilities to have the ability to contribute with research being conducted. Once their time has elapsed there are limited ways for them to have input and access to current data. The need to remove boundaries is of great importance in order to support increased collaboration and innovation at NASA and other institutions.

For the majority of analysts, the expectation of Hyper Text Markup Language 5 (HTML5), Cascading Style Sheets Level 3 (CSS3), JavaScript (JS), and PHP: Hypertext Preprocessor (PHP) are for the creation of static websites. In the past, this may have been the case; PHP once was an acronym for Personal Home Page. The latest updates to web-based computer languages have given power to accomplish tasks that were once only executable with languages traditionally OS based. The case study using a data feed from the NASA Ames' planetary rover testbed encompasses the majority of tasks WBPM is designed to address. The rover gathers data with the aid of one researcher overseeing the controls, it stores the information on a laptop directly on the rover, users can subscribe to the outputs, and it is considered to be conducting short term (ST) tests. These tests happen frequently providing many opportunities to test with the WBPM framework. Adjustments in rover mission profile based upon feedback

Miryam Strautkalns et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

from the data take time. The WBPM will accept data and the modifications would make the data received relevant.

The contribution of the paper is a framework that allows analysts the ability to monitor their data on any smart mobile device, computer, or laptop. Platform independent storage and retrieval of data is possible.

The paper is organized as follows. Section (2) Motivations, explains the reasons that drove this framework to be created. Section (3) The Flow of Data, explains the languages used and how the data is stored. Section (4) WBPM Framework, describes the data being monitored for the rover test bed. Section (5) Case study with NASA planetary rover test bed, overviews the observations and use of the rover test bed simulator to test the framework. Section (6) Framework Abilities, Benefits, and Future, explains the capabilities and benefits of using a web environment for data acquisition. Section (7) Conclusion, summarizes the findings made.

2. MOTIVATIONS

During most research projects, it was observed that certain restrictions were problematic to the overall quality of projects. The solution was more involvement and time on-site by analysts. More information was gathered about the common ST and long term (LT) research practices and it was realized that there was an extraordinarily large amount of physical energy expended to remain present and ensure the quality of one's project, such as remaining in close proximity to the data (e.g. sleeping onsite or at a lodge).

Concerns changed depending on the parameters of the project, though the overall concern was maintaining the results from the time spent with testing. With projects whose data acquisition was maintained on a single computer, the concern was maintaining the system. There were experiences where a day's work was lost due to the failure of the system, resulting in a loss of data collected. Live data storage on a server would have prevented any data loss experienced from relying on the computer. Many other scenarios would end on the same concern and the same solution was available to all projects: data acquisition that did not rely on the computer alone to gather and store project data.

One server, one web based application. The solution was available because of the strength web based languages had gained with the latest updates made. Once the framework as a solution was established the previous ideas of a project's process changed, and the options available had increased. The environments for collaboration had broadened; the ability to store and present live data during experiments was available with more customization. With every test more features that benefit the research process present themselves.

3. CORE TECHNOLOGIES USED

The WBPM framework is built using HTML5, CSS3, JS, PHP, and MySQL with rover test bed simulator. The following subsections explain and summarize the languages, technologies, and content used.

3.1. HTML5

To create a framework that would be malleable and easily customizable, HTML5, JS, CSS3, and PHP were used. HTML5 comes with many updates that continue what HTML4 started with its updates and contributions in changing HTML from a document-based language to a dynamic and interactive user experience. Canvas is a powerful visual tool that is available in HTML5, and with JS, the visualizations can be used to create graphing systems, charts, and visualizations that update live. Prior to HTML5, the canvas was usually defined using a Java plugin or a Flash plugin. Form validation offers a security to the quality of information collected; HTML5 validation is available within the browser.

3.2. CSS3

CSS3 is a way to organize and represent information through the browser environment in a useful and relevant structure for the content and audience. This includes the style sheet parameters required by the Canvas object. CSS3 introduces 2d and 3d transformations, transitions, and animations. These features organize the visualization of information in an effective manner for the user. Studies have shown that different categories of information are more susceptible to understanding if presented accordingly. In addition, CSS3 styles can be dynamically modified through JavaScript operating on the Document Object Model (DOM). CSS3, like its predecessors, tailor the presentation of the data depending on whether it is on a computer, smart phone, or tablet.

3.3. JS

JS is a dynamic object-oriented language used to control and moderate actions and behaviors of webpages and web-based applications. Whether it is a computer, smart phone, or tablet, JS is the language used to aid activities ranging from game play to live information feeds. HTML5's tools in combination with JS create environments ideal to handle the viewing and controlling of live content. Because of its popularity and power, JS is commonly referred to as the language of the web. JS in this work provides a way to facilitate both the visualizations of the content viewer and a portal to initiate the transmission of data. Developer groups are still performing extensive work to increase the performance of JS.

3.4. MySQL

My Structured Query Language (MySQL) is a database management system that with commands can create different layouts for information storage. While MySQL alone does not contain an interface to make these modifications, there are many user interfaces in many different languages that are created for the task. MySQL is the world's most widely used database system and can be used on any platform. A standard interface to MySQL is through PHP. Modern interfaces can also be through languages such as Python.

3.5. PHP

A server-side language, PHP is a lightweight language used for accessing and altering databases and their content. In addition to being used to work with databases, PHP can be embedded and used to create webpages with HTML. PHP executes the chosen connector for the framework data, stores the data in the MySQL database and passes the information forward as a response back to the JS and HTML in an array.

3.6. Modern Browsers

Web browser software development is far greater than all other software development combined. As of June 2012, there are approximately 2.4 billion Internet users and most of those users have access to browsers via smart phones and tablets outside of the traditional browser facilitated by a desktop or laptop computer. Since HTML5's updates, the browser environment now has the power to take on tasks thought to be limited to computer based software packages and stand alone applications. Current browsers are capable of accomplishing tasks ranging from new transitions and movement, 3d Canvas drawing, and reading motion sensor data for accelerometers, gyroscopes, and magnetometers. Development for the web helps alleviate concerns that most analysts face in keeping their software current so their data is readable and usable as hardware and software platforms evolve. This is due to the fact that the cost and person power used to fix existing bugs, create new functions, and extend the software will not be wasted on a software platform that may not be around in the future.

4. WBPM Framework - Current Rover Test Bed Data Acquisition

The NASA planetary rover currently monitors 84 parameters/second, including the statuses of twenty-four batteries, the voltage output and temperature of each, the currents, commanded speed, and actual speed for each of the four wheels, the global positioning system for latitude, longitude, and elevation, and the accelerometer, gyroscope, magnetometer readings for x, y, and z values. The output of data comes in the form of an array including the timestamp and a new updated array of values comes every second for the duration of a test. In order to observe the live tests performed while the rover is driven, the users must install Windows based communication software used to facilitate subscription.

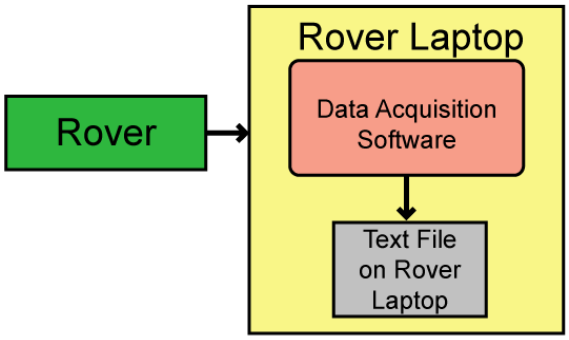

Figure 1. Current Flow chart of Rover Data.

Figure 1 shows the current sequence for Rover Data storage:

- Rover begins transmitting data.
- Local Data Acquisition Software monitors data being transmitted and sorts data into an array
- Array is saved to the rover laptop's hard drive in a text document file.

Once the text file containing rover data for the test session is saved, a researcher will upload the file to be stored on their server. Here people are able to download the file, and, using a corresponding Matlab file, can distinguish between the different values listed.

4.1. WBPM Framework

Depending on the project environment, the initialization of how the framework begins will change to accommodate different levels of security.

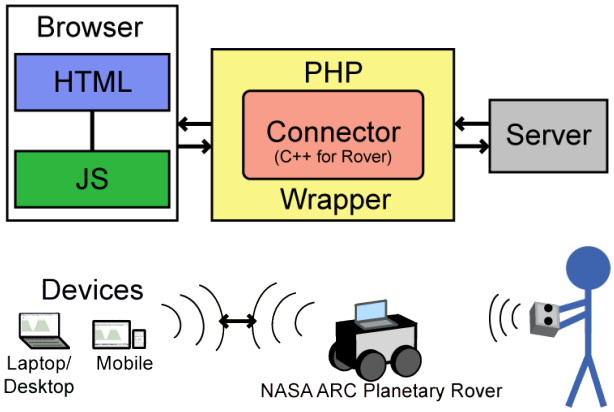

Figure 2. Framework Chart

Figure 2 shows the steps for implementing the WBPM Framework on the rover test bed, with the arrows signifying data being transmitted between languages and devices:

- Browser is initialized; the JS creates an Asynchronous JavaScript and Extensible Markup Language (AJAX) request.
- The AJAX request is created, establishes a successful connection and the PHP is opened.
- The PHP executes the connector program used to communicate data being transmitted from the project.
- The data string is received by the PHP and is split into its individual parameters and separated into an array.
- The array's data is stored by the PHP into the server's MySQL database along with other parameters including originator and access list (for security).
- When specific parameters in a time frame are requested, the information is then sorted into new arrays and sent to the JS as a response to the initial query.
- The AJAX response receives a new value for each area of the rover being monitored. Each value is added to the corresponding array.
- The array values in JS are sent to populate the HTML5 Canvas drawn graphs.
- User will be able to use common thumb gestures to scroll the data.

In real-time mode this process repeats with a set time interval. The rover receives new data every second and the AJAX is set to request data in accordance.

5. Case Study

Initial steps were to incorporate the standard methods users have been following to observe the data. After installation of the communications software there were compatibility problems encountered. The connectors and programming created for users to observe the rover's testing were written in a platform specific manner around many Windows functions – confirming one of the rationale for developing the one server, one web-based application approach. The computer being used in this case is an Apple MacBook Pro. The result was that observing the data with the same research environment, software that allow analysts to observe the rover test bed during a live experiment, was not platform-independent. This situation was a real life motivation for the WBPM framework.

The installation of the platform-dependent communication software was not an option and the solution was to recode the simulation available to be compatible. Understanding the goal is to store and retrieve data the flexibility of framework allows analysts choices on how to present and store it.

New data from the rover is transmitted every second and that dictates the cycle for AJAX requests sent. There is an empty array created within the JS and each AJAX response from the rover is added per second. The AJAX requests initiate the communication between the PHP and the Matlab rover simulation. The Matlab rover simulation provides simulated rover data. The rover data is transmitted through a Java connector embedded in Matlab in a string of values lead by a timestamp, this contains the values for each sensor on the rover for that time. The Java connector and PHP have several tasks to perform.

Java connector:

1. Initiate a connection to the MySQL database.
2. Connect with the Matlab Rover simulation to retrieve the data string.
3. Separate the string into arrays for organization.
4. Store the data into the MySQL table.

PHP Query:

1. Opened by an AJAX request.
2. Initiate a connection to the MySQL database.
3. Send a MySQL query to gather data from the database.
4. Send sorted data back as a response to the AJAX request.

Figure 3. Entire set of rover data: 4 min 3 sec test - Battery 1 Voltage

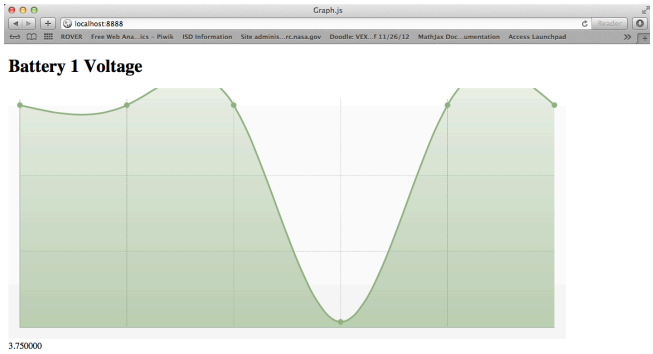

Figure 4. Six latest rover data values - Battery 1 Voltage

The Canvas graph has been programmed to show two views when selected, one for the values over the entire length of the rover test, Figure 3, or the six most recent plots of data, Figure 4.

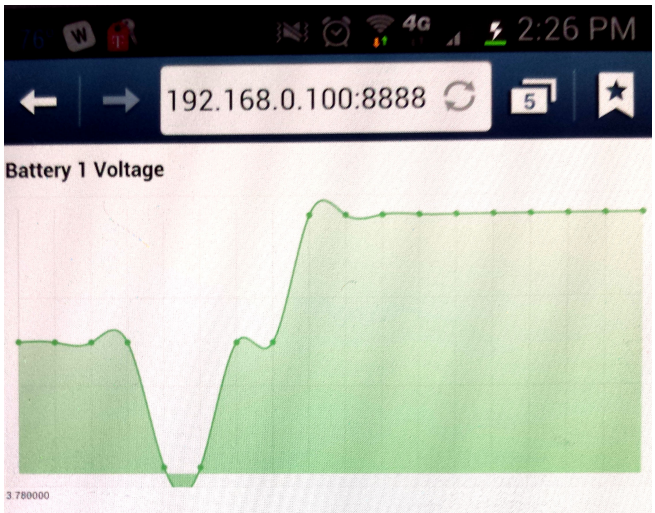

Figure 5. Smartphone test.

The WBPM framework then required mobile device testing to review future possibilities for project monitoring. A connection to be established between the simulation on localhost and a smartphone meant creating an open port for communication over a wireless connection. Figure 5 shows the mobile WBPM framework monitoring the rover's battery 1 voltage, running on a smartphone.

6. WBPM Abilities and Benefits

WBPM is providing a flexible base that can provide secure and reliable monitoring. The resources, browsers, and programming languages used have an extremely vast network providing regular updates and support. All because of the fact that the online world changes so rapidly, when engineering for the web you have to create for the future. Programming for the present in some cases is already too late. Despite the fast paced speed of innovation the support maintained for older web standards is a part of planning web applications. There are common programming practices to ensure that users with older browsers can use applications. This in combination with the browser's built-in cross OS features create an environment where analysts do not need to worry about whether collaborators have the required software installed on their computer, much less the correct OS installed on their system to support the software itself. Developing for the web helps prevent software development in directions which are not upgradable.

6.1. WBPM – Future

Having a WBPM system will only increase in relevance and strength. The capabilities to monitor in real time the performance of a project from a smartphone exists, receiving text messages when errors occur, auto generated emails upon completion of a project cycle – these are all valuable options for analysts. Not only providing a new definition for a project's environment scope, but the opportunity for an analyst to make greater use of their time outside of the scope of in person monitoring. There is no need to physically observe a task during the duration of its test. (Except for parameters for which there are no methods of measuring). The timeliness of data monitoring is not confined to a computer's storage to be viewed at a later date. With secure connections and encryptions it is a safe and cost effective solution to have instantaneous access to the data and to share that data in worldwide collaborations. Not only can you share data, but remotely adjust the settings to the project itself.

Much development of the plotting capabilities needs to be addressed. The plotting of multiple parameters at once is a challenging issue. The ability to zoom in and focus analysis on different portions of the data is also a challenge. In addition methods to perform remote commanding will also require addressing new TBD solutions.

In addition, though the central notion is one server, one web-based application, each data acquisition source will require the development of additional code to push the data to the server.

7. Conclusion

The WBPM framework creates an environment that allows remote viewing of rover data on all OSs and mobile devices. WBPM is the beginning of a cloud based monitoring and fault diagnosis application. Projects rely on computer based software packages; with the framework analysts can incorporate stability for data acquisition. The next steps for effective prognostics will be to have monitoring applications that are server based with a connector receiving data from the project source.

The WBPM resources and languages used create an environment that is ever changing and updating. By using

the browser to monitor live data analysts have new options to how they conduct projects and collaborations and monitor PHM systems. The rover case study shows the benefits of data monitoring and storage when comparing the current method of stove-piped applications to the WBPM framework's method. Future extensions of the framework will only expand the number of systems that can benefit from the framework. The interactivity will increase the number of ways analysts remotely manage their data.

ACKNOWLEDGEMENT

I would like to acknowledge Adam Sweet (NASA Ames Research Center) and Sriram Narasimham (UCSC), they took the time to make their rover test bed data available for my case study. Matthew Daigle (NASA Ames Research Center) for his counsel during the creation of the WBPM framework. Peter Berg (SGT Inc. NASA Ames Research Center) for initiating the course of events that led up to this.

NOMENCLATURE

NASA	National Aeronautical and Space Administration
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
JS	JavaScript
OS	Operating System
LT	Long Term
ST	Short Term

REFERENCES

Flanagan, David. *JavaScript: The Definitive Guide*. O'Reilly Media, Inc., 2013. Print.

Tatroe, Kevin. Programming PHP. O'Reilly Media, Inc., 2013. Print.

E. Balaban, S. Narasimhan, M. Daigle, I. Roychoudhury, A. Sweet, C. Bond, and G. Gorospe, "Development of a Mobile Robot Test Platform and Methods for Validation of Prognostics-Enabled Decision Making Algorithms," *International Journal of Prognostics and Health Management*, vol. 4, no. 1, May 2013