

Individual Milestone 6 Report

Executive Summary

I had proposed to produce a daemon feature in the form of a plugin to relay the status information from the daemon to the web-based GUI of the Earth application. The motivation for such feature arose from the fact the existing implementation of the GUI provides very little clue to the user about the operation of the daemon. This lack of information become even more crucial if the daemon and the GUI are running on different physical machines, in which case, erroneous assumptions about the daemon operation could effectively limit the reliability of the Earth application. However, further investigation into the actual implementation revealed that such an elaborate plugin approach would introduce unnecessary redesign and complexity to the codebase, all in return for the trivial benefits of providing daemon status information through the web-based GUI. Instead, a more simplistic approach of attaining the same benefit was pursued and implemented in this Milestone sprint. The approach uses the underlying (Linux) file system to relay the relevant daemon status information to the GUI while still maintaining the design separation between the GUI and the daemon.

Tasks and Activities Assigned

- Design and Implement Display of Daemon Status Information on Web-Based GUI.
 - Description: This task involved capturing the daemon status information and displaying it on the web-based GUI in real-time.
 - Idea/Solution: N/A
 - Affected Files:
 - * app/controllers/servers_controller.rb
 - * app/helpers/servers_helper.rb
 - * app/models/earth/server.rb
 - * app/views/servers/configure.haml
 - * script/earthd
 - Git commits: ([ssurfer/earth.git](#)) 563eb803cfbb8ffb687c86b4c473b718a03dffbd
 - Estimated time taken (planned): 70 hours
 - Estimated time taken (actual): 80 hours

Resource Contributions

At the beginning of this Milestone, some time was spent identifying which particular feature or aspect of the application to be transformed into a plugin-enabled feature. Aside from deciding on the granularity levels of the plugins, some considerations about the relevancy and appropriateness of the plugins to the primary purpose of the Earth application also had to be made. This part of the task initially appeared to be straight-forward and subsequent efforts were made to quickly transform the implementation of some existing features into the relevant plugin forms. Unfortunately, the retrofitting of the daemon to incorporate plugins using extension points had not been fully completed at this point. As a result, the plugin conversion task had to be put on hold until the relevant plugin framework codes were committed.

In the meantime, the development focus was altered slightly to accommodate these delays and one such change involves the relaying of data between the daemon and the web-based GUI. Presently, the only existing mechanism that is able to facilitate any communication between the

daemon and the GUI is the underlying PostgreSQL database. The daemon updates the database based on the changes in the monitored file system and the Rails application retrieves this information from the database for further processing and subsequent display on the browser. This works flawlessly for the file monitoring operations of the daemon, with the GUI passively updating the results when a web request from the application user is received. Unfortunately, this passivity of information relay may not be adequate or suit certain types of operation such as the real-time monitoring of the daemon status. A more direct mean of communication between the daemon and the GUI is more suited for such requirement.

To this end, a more simplistic solution was implemented which involved the periodic updates of a newly created 'daemon status' log file by the daemon. The corresponding GUI front end then simply retrieves the daemon status information from the said log file and displays it on the browser. Alternately, this approach could also be satisfied using the database instead of the log file. However, the approach employing the log file was adopted during this Milestone to quickly complete the proof-of-concept stage in the investigation of this particular design approach and eliminate the inherent complexities that normally arise with the alteration of the database schema. With the successful implementation and demonstration of the proof-of-concept stage results, the feature can then be implemented using the database instead of the log file. This change is necessary to ensure that this particular application feature can operate in a range of situation such as when the daemon and the web-based GUI are running on two different machines. This later expansion to a database-based implementation of the daemon status information update on the GUI is planned for the next development sprint.

Rooms for Improvement

The allocated time of 70 hours which had been initially earmarked for this particular task was in hindsight a bit optimistic. During the planning phase for the current development sprint, it was wrongly assumed that there was some simple way that the daemon could directly relay its status information to the GUI, which would not necessarily involve an intermediate facility such as the log file or the database. The subsequent investigation which revealed that no such simple provision existed also indicated that the incorporation of such feature through direct code manipulation or the creation of an appropriate plugin would take more time than budgeted. This experience clearly highlighted the costly and adverse effect of design decisions based on wrong assumptions on the schedule of software development projects.

— — — —