Capturing DSN Software Error Messages to Improve Software, Operations and Data Delivery

Mark Chatillon [1]; James Lin [2]; Tonja Cooper [3], Robert W. Sible, Jr. [3]

[1]Raytheon Australia, Canberra Deep Space Communication Complex, Canberra Australia;
[2]Chase Computing International, Corp., Alhambra, California USA;
[3]Jet Propulsion Laboratory (JPL)/ California Institute of Technology (Caltech), Pasadena, California, USA

The accurate capture of software error messages assists software development engineers in their ongoing role to continually improve their product. The Deep Space Network (DSN) software is generally "leading edge", complex and affected by outside factors such as spacecraft anomalies and spacecraft idiosyncrasies; thus, the software is often difficult to troubleshoot. In addition to the complexity of the software, the problem observers are geographically separated at four sites around the globe, and from three different cultures with different principal languages resulting in a very low potential for achieving consistent problem descriptions.

One solution is to automate the capture of the error messages generated by the software, and then provide a tool that allows the team of engineers, developers, and maintenance staff to research and query the data collected. The first component, the capture of the messages can be achieved by parsing the Log file produced during each spacecraft pass, and extracting the relevant error messages within provided parameters (start and end time and the systems / sub systems that are suspected of causing the problem). The extracted data excludes other fields (lines) of the Log removing "noise" entries. By being able to capture Log error lines for more than one system we provide visibility of scenarios where one system has an impact on another.

The text file can then be copied into the Discrepancy Reporting and Management System database (DRMS), which records all outages for tracks. This data can be readily accessed, interrogated, and searched. In their examination of a problem, the engineering and maintenance team can enhance the value of the data collected by allocating a "prime" or major error message from the set collected. They achieve this by copying the selected "prime" error message into the prime error field in the DRMS database as part of their problem analysis.

As data is gathered and analysed there will be an increased level of more accurate analysis and an improving trend in the allocation of problems to the specific cause. The engineering and maintenance team can then make use of the DRMS strong text search capability to query the error fields, seeking to identify trends and patterns. In addition they will be able to isolate the cause of software problems down to the lines of code involved.

There is also potential for the use of "artificial intelligence" algorithms on the main error message datasets. Presently, data is held in individual files, however, once captured in the DRMS it is more focused and more useful. With the best of intent the engineering team will not be able to discern all the trends or patterns that the available data can offer them. Using "AI" techniques should enhance engineering knowledge of the software components and provide new "views" of their software's performance.

Better knowledge of our software and its performance equates to better data retrieval from our exploration spacecraft, and therefore greater scientific achievement.