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

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

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

There are many factors that have historically impacted on the Deep Space Network (DSN) capability to improve performance in its science-gathering role. Among these factors, the one that is becoming increasingly apparent is the capabilities and vagaries of the various software applications the DSN uses in its systems. Software has been and continues to be one of the prime causes of DSN data outages, and historically is responsible for slightly more than 20% of outages. It is the second highest cause in terms of data effect after hardware. In reality, this level of outage will in fact be higher, as a very significant portion of outages that are originated as of an unknown cause, would, if resources were more readily available, be more likely to be finally determined to be the result of a software problem and reflected as such.

For the 12-month period 1 February 2003 to 31 January 2004, across the DSN, software causes accounted for 1228 hours of outages, whilst problems described with the unknown cause accounted for a further 1024 hours outage. As a percentage of hours scheduled by the projects these are small outages, data delivery is consistently very high, often greater than 98% nevertheless, all science data is of course of great value, and every effort must be made to attain it. It is estimated that 70-80% of these unknown cause outages could be attributed to software. If we look more closely at the data affected and strip out the navigation, monitor, and command data types to arrive at the telemetry outage, we gain a clearer insight into the potential impact on science data. The lost telemetry data that can be attributed to an equipment software problem for that same 12-month period is in the region of 23%.

As the science data requirements and demands inevitably grow, the systems and their software applications must grow in complexity in an ongoing attempt to match the requirements. Consequently, these systems and the software running on them are more often than otherwise, of a "leading edge" nature. External factors can also create an impact, in that "peculiarities" relevant to one spacecraft or one data source, might not evidence themselves with another data source. This is often compounded in that problems may manifest themselves in different ways and in some instances are observed differently at the three different DSN locations in Canberra, Australia, Madrid, Spain, Goldstone, California (USA) and at the Network Operations Control Center (NOCC), JPL (USA). The very nature of software implies that a problem or incorrect outcome may only be present for the blink of an eye, making any effort at describing it a very difficult proposition, with the more usual outcome more closely resembling a description of the effect observed and not the root problem cause.

Gathering some strictly factual or metric information on incidents it is a relatively easy task, however it is a much different proposition to achieve some form of uniformity or consistency in the description of these incidents. The DSN sites are located in America, Australia and Spain, and therefore have incident observers from three core nations, each with their own "version" of English and of course in Spain's case, their English is also a second language. This disparity is compounded by the reality that there are several background nationalities with different linguistic skills within the individual sites, further impacting on the consistency of describing an incident or problem. In addition, naturally there are a variety of levels of experience within and across the DSN sites, and this also has an effect. Even if the problem is one that might be said to be common and perhaps well understood, there is almost no chance that the problem will be categorised and documented in a descriptive manner that could in any way be rated as "consistent". This inconsistency occurs at just one site, let alone across the DSN over any significant period of time.

As a result, it is beyond doubt that there are a significant number of instances of software outages that have insufficient detail to allow any proper investigation or closure, and that accordingly do not enhance the maintenance and engineering's staff's knowledge of their software. Therefore the extensive effort and resource that goes into the problem recording process often achieves no beneficial outcome, and really results in just the collection of outage and problem metrics. No beneficial outcome in reality equates to: -

- No improvement in the understanding of the inherent problem,
- No reduction in its effect, and,
- No correction of the software code to prevent recurrence.

All of which in turn contributes to the loss or degradation of science data.

The result is a continuing challenge for the whole of the DSN, but particularly for the software designers and developers and the site based maintenance engineers. They must continue their role to both improve their product and to ensure that the DSN sites making use of their software, comprehend it and it's functions so that they are able to use the software effectively. They must undertake this "improve the product" role, at least in part based on "observations" made by the operators utilising the software, and as a result of the inconsistent narratives, they are to some extent hampered in their tasks. Despite the "leading edge" nature of their product and all the other difficulties for the sake of the science data they must continue to achieve improvement in their software product.

One solution is to provide them a better level of relevant information by the capturing of the error messages generated by their software, and then provide a tool that allows the team of engineers, developers, and maintenance staff, to research and query the error data collected. To further eliminate any inconsistency, the process where the error messages are captured should be as automated as possible, thereby removing or minimizing any potential observer subjectivity. The basic pre requisite to implement this scenario is that the software error messages themselves must achieve a very high standard of consistency and relevance, and must not be of an ambiguous nature. It is only in the more recent past that the software development teams have begun to achieve this with their error routines and messages. More recent storage capability improvements in the IT industry have also increased the potential merit of this error data acquisition in that the cost of disk space does not prohibit the collection of larger lumps of pertinent data.

For the past couple of years, the DSN Discrepancy Reporting and Management System, (DRMS), has recorded error messages on a trial basis from some sub systems (that have now in fact been replaced by new items). Evaluating this trial period, it was found that the error messages were of limited use due to the difficulty in identifying which message was in fact the most pertinent. This initial attempt at gathering more relevant data was designed to record the first error message generated by the errant system. This was set up as a manual process dependant on the observer reviewing his scrolling log and selecting the first error reported. Being such a subjective manual process proved it of limited usability. It relied on the operator both being sufficiently skilled and having sufficient time available to research his log and correctly identify the first message. This proved a difficult task when antenna time is at a premium and operators often had very minimal time between spacecraft passes. Even if time was found and the message recorded, in the end it may not have been to "prime" or crucial message and all subsequent messages remained buried in the Log file. In fact there were a few instances where valuable investigative time was spent looking in the wrong direction.

An investigation revealed that the observers really never had a chance at extracting the "first" error message. This is mainly caused by the "over classification" of error messages generated where so many are received that are both near identical and/or are not really of a significant importance. In some log files, warning messages account for greater than 60% of the Log lines.

Alongside this first effort error message field, it was recognised that it was possible for the DRMS to provide a little more assistance in gathering the required data together. Recognising that in some instances the pass log file might be most useful the DRMS was modified so that a user could append the URL of the pass Log file into the record. This made it easier for the investigators to at least locate the information, however it still left them with a log file of often thousands of lines to trawl through. This log had to be viewed and examined within a nominal timeframe and relatively promptly, as again storage availability has a major impact with files being purged at a set age. Whilst it was an improvement, it did still rely on the investigator having the time to "manually" peruse through the lines of the log. These drawbacks in part led to the current quest to provide more useful data to those undertaking the role of problem or incident investigation and analysis.

The earlier efforts to improve these capabilities also recognised that in some instances the set up data and pass parameters for a spacecraft pass might also be of relevance, including the various predicts and the sequence of events details. This dataset was the first to be addressed in that each operator's workstation was provided with a parsing script that effectively gathered all the available pre pass material together into a single text file. The operational crew would use their judgement to determine if such material might assist and if so the pass catalogue was generated. The resulting text file was then automatically passed via an email to the particular site's DRMS administrator. They would simply attach it to the DRMS Discrepancy Report (DR) of the incident, where it is available for all DRMS users at the DSN site, but in particular the analysing maintenance staff, to view, examine and perhaps make use of in their investigation.

During the "requirements" phase of this development of the enhanced error message capture capability, it was also determined that there is yet more data that can assist in any investigation. Each operator in building a link follows a checklist that might often include pertinent data. It was decided that this data might also be made available and accordingly the electronic checklist is to be included in the pass material that can be appended to the DR record. As well as seeking structured input from the operator based on automated link building software, this checklist when completed, identifies how the pass has been set up. The final outcome will be that the investigative team will be armed with all the predicted pass parameters, what the operator actually set in place, and the extracted error messages, three key building blocks along with the observers problem description with which to perform their analysis.

The DRMS is a powerful database that maintains records of all data outages experienced across the DSN, as well as real time status data on the DSN equipment that is most useful to the Network Control Centre in allocating antenna time during problem periods. This is the logical vehicle to maintain the newly provided capability. Added to this DRMS outage data on a best efforts basis, (data delivery is the prime goal) are details of any equipment hardware or software resets performed in the building of a link, during a pass or at other times when data is expected. All these records are replicated to the Jet Propulsion Laboratory (JPL), and reside in a centralised master database. Locally, each DSN site also has a private (to that site) Station Log, that used in conjunction with the outage data, allows them to build an electronic and searchable 'corporate history' not only related to equipment, but also to a wide range of other site activities. Engineering teams have a role to play in maintaining visibility of how their equipment is performing across the globe, and providing comment and assistance to the sites as required. To achieve that role they must be as informed as possible regarding their equipment and/or software.

To make these captured error messages more useful, the DRMS is undergoing some schema changes. Firstly this will provide a database field to store and replicate to JPL, the collection of relevant error messages. Whilst this collection alone adds valuable data in that it is a time sequence of reported errors, the addition of other new database fields will permit the investigating engineers and technicians as part of their analysis process, to identify and select their so called "prime" error message. They will select both the unique error number and the descriptive text including any particular variable parameters, and paste them into this second set of new prime error message fields. The message entered is validated as part of the entry process to ensure the error message matches one of a valid list. This ability to track instances of these "prime" errors will be of greater benefit in the longer term.

It has been recognised that there will be some resources needed to maintain the inbuilt lists of error messages. However whilst the number of new systems under development is relatively low and whilst software revisions and also not generally focussed on error message it is considered that the task will not be an onerous one. As time progresses it is intended to build into the DRMS suitable tools that can take the electronic list of error messages and run a comparison with present values. Making use of such utilities should keep the error data as current as possible.

The error message data will be most useful in pointing the developing engineering teams to sections of code that might be requiring more attention in their effort to improve the software performance. To some extent their skill at developing their routines will determine how much they can achieve from the data provided and the query capabilities of the DRMS particularly for the "prime" error message field will assist this process as will the ability to search for particular error message combinations within the "collection" field. 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 also then make use of the DRMS strong text search capability to query the error fields, seeking to identify trends and patterns.

In order to make the best use of the error data it is important to contain the messages to both a relevant time frame and of course to the relevant system at fault. Filtering out both other system's messages and the other messages (not reporting errors or warnings), concentrates the focus where it needs to be. Even though some log files contain more than 60% of lines as warning messages, the ability to cut back to that level is the starting point.

To further reduce the Log file there are two possible methods that can be used independently or in conjunction. The first of these is simple time frame parameters to cut the error log back to messages generated within the time frame requested. It was therefore decided that the link operator would firstly establish a time band within which the error message collation would be performed. Nominally, this will often be determined as the same as the period that the outage occurs however there may be instances where the errors occur prior to the outage, and in such circumstances the operator will need to exercise their judgement.

The second is the ability to specify items or systems that has experienced the problem. This selection of item can also provide a fuller pass long history of item activity, which may also be most useful in some circumstances. The link operator will indicate the system or systems of interest. Generally there will be a simple single system, however at times there are systems that inter-react and errors on one system could have an impact on the other, thus there are cases where two or more systems might be involved in the error message collection. Lastly, a combination of these two methods may allow an even more condensed and focused set of massages to examine.

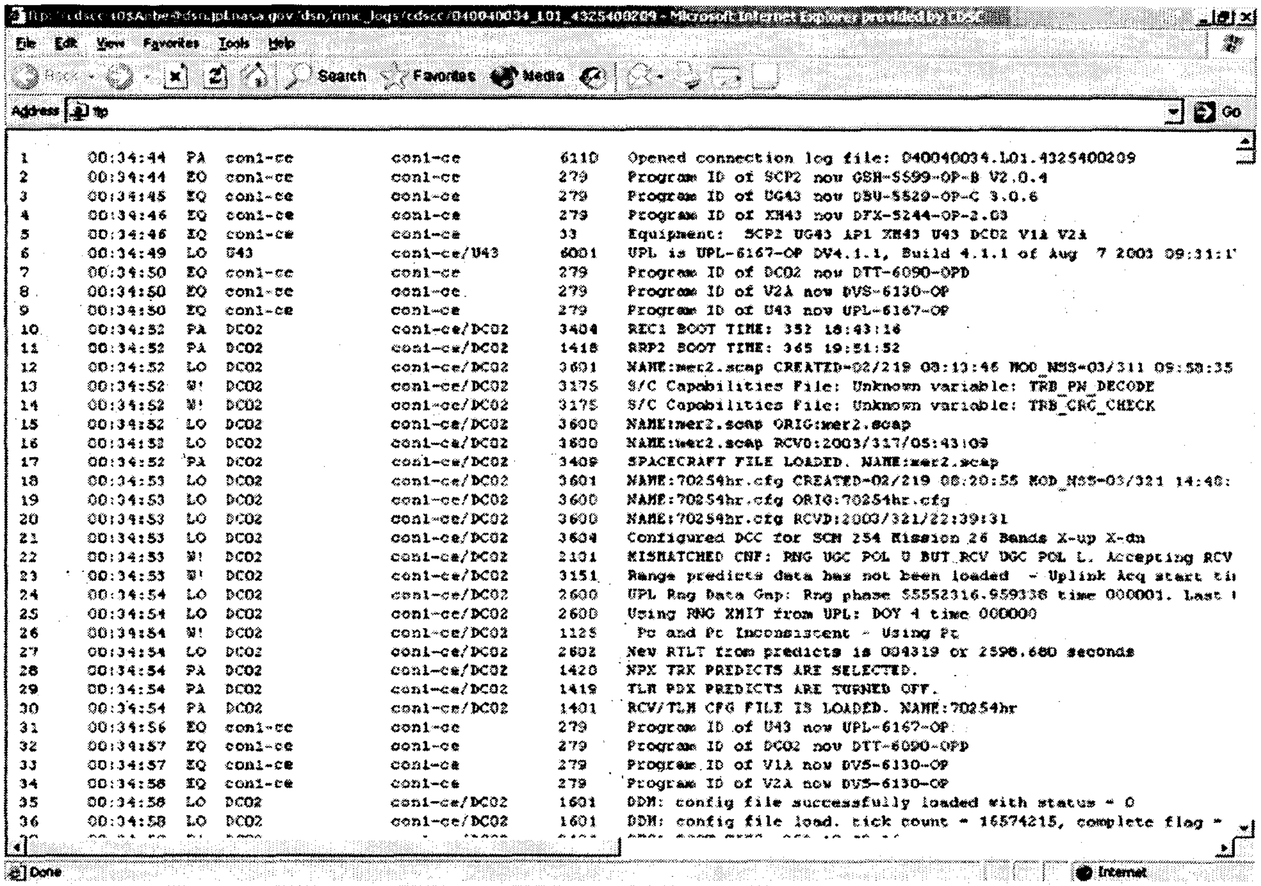
From a sample Log file, (7066 lines), the following statistics demonstrate the usefulness of drilling down to a system and then a time frame: -

- Collating all warning messages for the pass resulted in just over 3900 lines of information,
- Selecting a single commonly used system reduced that number to 206,
- Adding a time period of 10 minutes prior to incident and extending to end of data outage reduced those 236 to 12 lines. This number will vary substantially in some error situations where messages may repeat several times.

The value of this parsing is very evident from the above numbers.

There is still an inherent weakness in the overall process in that the operator might misjudge the time or perhaps not have the experience to realise that a related on interactive system's messages might be of use. Nominally this information is not lost, but is still resident within the overall log file however what is lost are the additional resources required to go and retrieve it. Despite this potential loss of information major improvements have been made.

The Log files themselves are of a reasonably consistent format that enables readers to identify various items within each record line. The descriptive text is the most complex component and at the same time the most variable, including the quite variable message parameters. A small sample of one log file is shown below. It displays the data and the basic format.



1	00:34:44	PA	con1-ce	con1-ce	6110	Opened connection log file: 040040034.L01.4325400209
2	00:34:44	EO	con1-ce	con1-ce	279	Program ID of SCP2 now GSH-5599-OP-B V2.0.4
3	00:34:45	EQ	con1-ce	con1-ce	279	Program ID of UG43 now DSU-5529-OP-C 3.0.6
4	00:34:46	EQ	con1-ce	con1-ce	279	Program ID of XH43 now DFX-5244-OP-2.03
5	00:34:46	EQ	con1-ce	con1-ce	33	Equipment: SCP2 UG43 AP1 XH43 U43 DC02 V1A V2A
6	00:34:49	LO	U43	con1-ce/U43	6001	UPL is UPL-6167-OP DV4.1.1, Build 4.1.1 of Aug 7 2003 09:31:1'
7	00:34:50	EO	con1-ce	con1-ce	279	Program ID of DC02 now DTT-6090-OPD
8	00:34:50	EO	con1-ce	con1-ce	279	Program ID of V2A now DVS-6130-OP
9	00:34:50	EQ	con1-ce	con1-ce	279	Program ID of U43 now UPL-6167-OP
10	00:34:52	PA	DC02	con1-ce/DC02	3404	REC1 BOOT TIME: 352 18:43:16
11	00:34:52	PA	DC02	con1-ce/DC02	1418	RRP2 BOOT TIME: 365 19:51:52
12	00:34:52	LO	DC02	con1-ce/DC02	3601	NAME:mer2.scap CREATED=02/219 08:13:46 MOD_NSS=03/311 09:58:35
13	00:34:52	W!	DC02	con1-ce/DC02	3175	S/C Capabilities File: Unknown variable: TRB_PN_DECODE
14	00:34:52	W!	DC02	con1-ce/DC02	3175	S/C Capabilities File: Unknown variable: TRB_CRC_CHECK
15	00:34:52	LO	DC02	con1-ce/DC02	3600	NAME:mer2.scap ORIG:mer2.scap
16	00:34:52	LO	DC02	con1-ce/DC02	3600	NAME:mer2.scap RCVD:2003/317/05:43:09
17	00:34:52	PA	DC02	con1-ce/DC02	3409	SPACECRAFT FILE LOADED. NAME:mer2.scap
18	00:34:53	LO	DC02	con1-ce/DC02	3601	NAME:70254hr.cfg CREATED=02/219 08:20:55 MOD_NSS=03/321 14:48:
19	00:34:53	LO	DC02	con1-ce/DC02	3600	NAME:70254hr.cfg ORIG:70254hr.cfg
20	00:34:53	LO	DC02	con1-ce/DC02	3600	NAME:70254hr.cfg RCVD:2003/321/22:39:31
21	00:34:53	LO	DC02	con1-ce/DC02	3604	Configured DCC for SCN 254 Mission 26 Bands X-up X-dn
22	00:34:53	W!	DC02	con1-ce/DC02	2101	MISMATCHED CNF: RNG UGC POL U BUT RCV DGC POL L. Accepting RCV
23	00:34:53	W!	DC02	con1-ce/DC02	3151	Range predicts data has not been loaded - Uplink Acq start ti
24	00:34:54	LO	DC02	con1-ce/DC02	2600	UPL Rng Data Gap: Rng phase 55552316.959338 time 000001. Last
25	00:34:54	LO	DC02	con1-ce/DC02	2600	Using RNG XMIT from UPL: DOY 4 time 000000
26	00:34:54	W!	DC02	con1-ce/DC02	1125	Pc and Pc Inconsistent - Using Pc
27	00:34:54	LO	DC02	con1-ce/DC02	2602	New RTLT from predicts is 004319 or 2596.680 seconds
28	00:34:54	PA	DC02	con1-ce/DC02	1420	NPX TRK PREDICTS ARE SELECTED.
29	00:34:54	PA	DC02	con1-ce/DC02	1419	TLM PDX PREDICTS ARE TURNED OFF.
30	00:34:54	PA	DC02	con1-ce/DC02	1401	RCV/TLM CFG FILE IS LOADED. NAME:70254hr
31	00:34:56	EO	con1-ce	con1-ce	279	Program ID of U43 now UPL-6167-OP
32	00:34:57	EQ	con1-ce	con1-ce	279	Program ID of DC02 now DTT-6090-OPD
33	00:34:57	EQ	con1-ce	con1-ce	279	Program ID of V1A now DVS-6130-OP
34	00:34:58	EQ	con1-ce	con1-ce	279	Program ID of V2A now DVS-6130-OP
35	00:34:58	LO	DC02	con1-ce/DC02	1601	DDM: config file successfully loaded with status = 0
36	00:34:58	LO	DC02	con1-ce/DC02	1601	DDM: config file load. tick count = 16574215, complete flag =

Looking more closely at the Log file content, the following expands on some of the content and fields that are used in the parsing process.

10	[redacted]	PA	[redacted]	con1-ce/DC02	3404	REC1 BOOT TIME: 352 18:43:16
11	00:34:52	PA	DC02	con1-ce/DC02	1418	RRP2 BOOT TIME: 365 19:51:52
12	00:34:52	LO	DC02	con1-ce/DC02	3601	NAME:mer2.scap CREATED=02/219 08:13:46 MOD_NSS=03/311 09:58:35
13	00:34:52	W!	DC02	con1-ce/DC02	3175	S/C Capabilities File: Unknown variable: TRB_PN_DECODE
14	00:34:52	W!	DC02	con1-ce/DC02	3175	S/C Capabilities File: Unknown variable: TRB_CRC_CHECK
15	00:34:52	LO	DC02	con1-ce/DC02	3600	NAME:mer2.scap ORIG:mer2.scap

The format of the Log file is as follows:

10 — There is a line number leading off followed by

[redacted] — a time stamp which is followed by

PA — a 2 character key identifying the type of log entry. There are a range of these types including PA – progress advisory, LO - Log Only, however the types that are most relevant to this task are those known as alarms, C! critical, W! warning and E! emergency alarms. There are also some cases where DA (Deviation Advisories) may also be pertinent.

[redacted] — Following on the next set of characters identify the equipment item or system involved in upper case, (or on some occasions the connection involved in lower case).

The next segment is a combination of the connection engine and system item and is not of relevance to the error process.

3404 The next segment is the unique advice number. In the cases of alarms these are in effect error numbers. This is followed by the closely related explanatory text that will often include parameters REC1 BOOT TIME: 352 18:43:18. In this example receiver 1 boot time is provided, the time being a variable.

The parsing process is a relatively simple exercise that makes use of the relevant segments of the text file to extract the relevant error messages within the specified parameters (if provided the start and end time and/or the systems / sub systems that are suspected of causing the problem). The extracted data excludes other 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 entered into the DRMS, which records all outages for tracks, by the incident observer. Once in the DRMS, this data can be readily accessed, interrogated, and searched using the query capability provided. In their examination of a problem, the engineering and maintenance team are encouraged to enhance the value of the data collected by allocating a "prime" or major error message from the set collected. Provided they can confidently identify which is the true cause, they achieve this by copying the selected "prime" error message into the prime error fields in the DRMS database as part of their problem analysis.

The following is an example of the final outcome of the parsed error messages. Initially for the pass concerned there were 206 lines of log written within the time parameters. It is very prominent that the resultant outcome is a much more concise and precise 12 lines.

378 00:46:41 W! DC02 con1-ce/DC02 3112 IF SWITCH: CONNECTION WAS NOT FOUND
382 00:46:53 W! DC02 con1-ce/DC02 3112 IF SWITCH: CONNECTION WAS NOT FOUND
385 00:46:56 W! DC02 con1-ce/DC02 3112 IF SWITCH: CONNECTION WAS NOT FOUND
387 00:47:00 W! DC02 con1-ce/DC02 3123 LNA FOR CHANNEL 2 CHANGED FROM NONE TO X1
388 00:47:00 W! DC02 con1-ce/DC02 3155 STATION DELAY NOT FOUND IN RNG CAL FILE (138 RECORDS CHECKED)
390 00:47:00 W! DC02 con1-ce/DC02 3124 POLARIZATION FOR CHANNEL 2 CHANGED FROM UNKNOWN TO RCP
391 00:47:00 W! DC02 con1-ce/DC02 3125 POLARIZATION FOR CHANNEL 2 (RCP) DOES NOT MATCH S/C TABLE (LCP)
392 00:47:00 W! DC02 con1-ce/DC02 3126 INPUT SIGNAL PATH FOR CHANNEL 2 CHANGED FROM UNKNOWN TO LOW_NOISE
393 00:47:00 W! DC02 con1-ce/DC02 3127 INPUT SIGNAL SOURCE FOR CHANNEL 2 CHANGED FROM UNKNOWN TO HORN
394 00:47:00 W! DC02 con1-ce/DC02 3128 INPUT PRE-COUPLING FOR CHANNEL 2 CHANGED FROM UNKNOWN TO NONE
395 00:47:00 W! DC02 con1-ce/DC02 3129 INPUT POST-COUPLING FOR CHANNEL 2 CHANGED FROM UNKNOWN TO NONE
399 00:47:01 W! DC02 con1-ce/DC02 2101 MISMATCHED CNF: RNG UGC POL U BUT RCV UGC POL R. Accepting RCV UGC.

There is a growing potential for the use of "artificial intelligence" algorithms on the main error message datasets. Presently, all this error data is held in individual files, however as more of these error messages are captured in the DRMS it become much 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.

The use of AI is more of a future goal, however those tasked with the development of the DRMS must keep it in mind and wherever possible ensure that such options can be exercised as the data available increases.

The overall aim is to improve the level of knowledge of our software and its performance. This equates to better data retrieval from our exploration spacecraft, and therefore greater scientific achievement.