AMFESYS: MODELLING AND DIAGNOSIS FUNCTIONS FOR OPERATIONS SUPPORT

J Wheadon

ESA/ESOC European Space Operations Centre
Darmstadt, Germany

ABSTRACT

Packetised telemetry, combined with low station coverage for close-earth satellites, may introduce new problems in presenting to the operator a clear picture of what the spacecraft is doing. A recent ESOC study has gone some way to show, by means of a practical demonstration, how the use of subsystem models combined with artificial intelligence techniques, within a real-time spacecraft control system (SCS), can help to overcome these problems. A spin-off from using these techniques can be an improvement in the reliability of the TM limit-checking function, as well as the telecommand verification function, of the (SCS). The paper describes the problem, how it was addressed in the study, including an overview of the "AMF Expert System" prototype, and proposes further work which needs to be done to prove the concept. The Automatic Mirror Furnace is part of the payload of the European Retrievable Carrier (EURECA) spacecraft, which was launched in July 1992.

1. INTRODUCTION

The recently-introduced technique, of transmitting spacecraft telemetry (TM) to the ground in the form of packets, introduces new problems of processing the data to provide the operations engineer with an accurate view of evolving spacecraft status.

The older TDM telemetry delivered regularly-sampled data in a fixed "TM format" structure, which itself gave a good approximation to successively sampled "snapshots" of spacecraft state; for most practical purposes, the operations engineer had only to display data on a format-by-format basis to understand what the spacecraft was doing.

Packet TM, in contrast, delivers data which are irregularly sampled, and data may even arrive in different packets out of chronological sequence (with respect to the timing of the events which they report). The principle is to deliver data from individual subsystems, only as fast as is needed to track how their status is changing. This means that the spacecraft designer can report a wider range of information for a given downlink capacity, compared to what TDM telemetry allows, but it also makes the processing on the ground more complex.

Most current ground-based Spacecraft Control Systems (SCS), which handle packetized TM, maintain a kind of "pseudo format" (PF) in computer memory by recording the latest values of TM data received, from whatever packets contained them. Although this means that changes to the PF occur irregularly, it gives an acceptable picture of the evolution of the spacecraft state over time, so long as the updated values in the TM arrive at the SCS in true chronological sequence. But when different packets report data out of sequence, or when "on-off" event report packets become lost in transmission, the PF will be inconsistent with the spacecraft.

Limit checking on TM parameters is usually performed by the SCS on the basis that the operating mode of each on-board subsystem can be derived simply from status information in the current TM format or PF, and that in most cases, a fixed set of limits can validly be associated with each such mode. This paradigm works well enough given a steady state of the spacecraft, although it often produces spurious limit alarms during sequences of state changes when switching from one operating mode to another. Spurious alarms will also occur if the PF adopts inconsistent states as described above.

Of course, spacecraft designers should try to make life easy for the operations engineer by ensuring that rapidly-changing data are reported often, that events such as mode switching are reported promptly, and that all information needed for spacecraft control is transmitted in true chronological sequence. ESA has defined standards which aim to promote such good (considerate ?) design, but historically, there has often been a tendency to make compromises in the spacecraft design which push problems of operation onto the ground segment. It should however be recognised, that increasing the complexity of the operations engineer's task in understanding what the spacecraft is doing, also increases the risk of not meeting mission goals; the operations engineers and technicians have to work reliably, day in and day out, over a period of years.

Mission operations also become more problematic for the ground segment when dealing with a spacecraft which does not continuously supply telemetry in near real-time, as for example a near-earth orbiter like EURECA (which itself uses packet TM). Typically these spacecraft have a ground station coverage (visibility from the station) of only about 10% of an orbit period of about 90 minutes. TM data are recorded on-board, and dumped at high speed over the ground station, in parallel with a "real-time" stream to give the operations engineer a "quick look" at the spacecraft status.

Such low-coverage science missions may have a high degree of on-board autonomy, performing complex measurement sequences. They are controlled during each orbit by an on-board Master Schedule of commands, previously uplinked from the ground. The operations engineer must largely rely on the autonomous functions, and can make only a few checks on the state of the spacecraft as it passes over the ground station, taking account account of the last known state of the spacecraft and of the Master Schedule commands which should have been executed during the last orbit. The SCS maintains a display of the Master Schedule, and verifies execution of as many of the commands as possible, mainly by checking defined "verification parameters" in the real-time TM. It also performs mode-based limit-checking on the latest values in the TM format or PF, as described above.

Diagnosis of apparent spacecraft malfunctions or anomolies under these circumstances will be difficult to do in real-time during the station pass (about 7 minutes); spurious anomolies may be flagged by the SCS from time to time, and it will be essential not to take precipitate action when these occur. The operations engineer thus will have no choice but to rely heavily on the on-board safety mechanisms to act correctly when real failures occur.

2. APPLICABILITY OF MODELLING

The actions of the spacecraft operations engineer are in effect dependent on the his view of the state of the spacecraft at any moment, and on the need to act (eg command changes of mode), but respecting defined preconditions.

If the picture given by the PF ("pseudo format") contains inconsistencies, the operator will have to relate what he sees there to a mental image of what the spacecraft should be doing, and try to reconcile the two. He will also find himself having to predict the evolution of spacecraft status in the case of a low-coverage mission with high on-board autonomy, whether the telemetry is actually packetised or not, at times when the spacecraft is out of contact with the ground station.

In effect, the operator will have in mind a model of spacecraft, so why should we not help him by

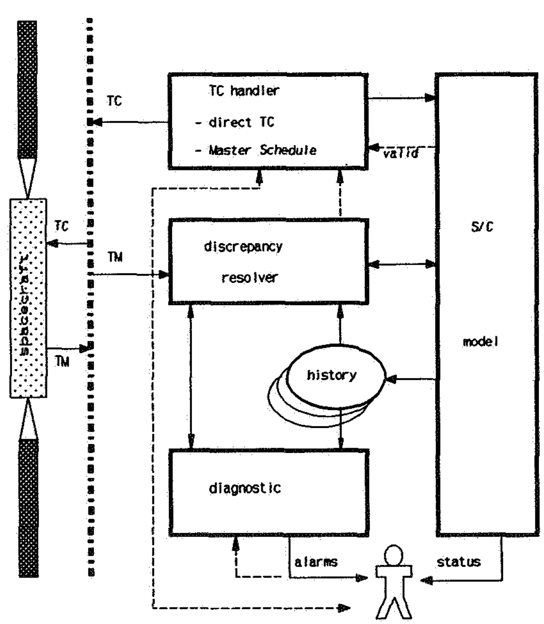

Fig 1: Proposed Concept

building such a time-driven model into the SCS ? The model can then become the reference for limit checking, as well as for command verification. For a low-coverage mission, the model can show the operator what the state of the spacecraft should be, at times when it is out of contact with the ground station. The model can include functions to accept and correctly emulate the spacecraft's handling of telecommands, both direct, and via a Master Schedule if the mission uses one. Fig 1 shows the proposed concept.

Classical "limit checking" in the SCS would then be replaced by a process of "discrepancy resolution", where discrepancies between the model state and the arriving TM data would be resolved by a knowledge-based reasoning process. Only if the TM data and the model were found really to be inconsistent, would out-of-limit alarms be raised to the operator, together with a possible diagnosis of the cause. Reconciliation of the model with the TM could be achieved by adjusting the model state according to the diagnosis, either automatically (in cases where the diagnosis is known to be reliable), or under operator control. In the rare case where a TM parameter is judged to have become unreliable, it would be permanently flagged as such in the discrepancy resolver.

To perform telecommand verification, the model would correctly predict the related TM changes, and any which did not occur would be picked up by the discrepancy resolver. The SCS should continue to keep a log of the execution of all telecommands.

The spacecraft history required for near-real-time evaluation and control would be represented by regularly recording the state of the model (with simple data compression if deemed necessary). Intermediate states at any moment of time could be obtained by extrapolating (modelling) forwards in time, but taking account of adjustments which were made from time to time by the discrepancy resolver.

All raw TM data useful for spacecraft control should also be filed, such that it can be used in long-term spacecraft performance evaluation, and for investigating and correcting possible deficiencies in the model. All payload data would continue to be processed and/or and delivered to end-users as is done at present.

3. AMFESYS STUDY

A modestly-funded ESOC study contract completed in March 1992 has done some initial work to demonstrate the feasibility of the above approach to spacecraft control. A time-driven model of the Automatic Mirror Furnace (AMF) payload subsystem of the EURECA spacecraft has been developed, and integrated with a TC schedule handler, a TM tracker, a discrepancy detector and a diagnosis module. Fig 2 shows the architecture of this AMFESYS ("AMF Expert System") prototype, which is implemented on a Sun SPARC 2 workstation using the C-based object-oriented-programming tool ProKappa (Intellicorp).

The AMF is one of the most complex of the EURECA payloads, although much of the complexity is effectively hidden from the spacecraft controller because of the high degree of autonomy afforded by its embedded microcontroller. The scientific purpose of the AMF is to make experiments in the growth of crystal structures under micro-gravity conditions. The AMF oven comprises an ellipsoid mirror with a hole in the shell at each end of the major axis. One of 7 available halogen lamps is inserted through one hole, such that it radiates from that focus of the

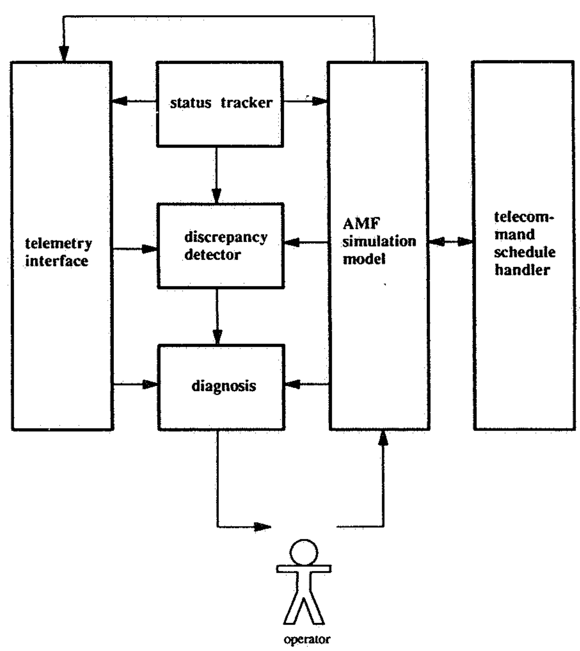

Fig 2: AMFESYS Architecture

ellipsoid onto the other focus. One of 24 cylindrical sample holders (ampoules), containing the crystalline material to be melted and re-crystallised, is inserted through the other hole, heated up to melt the crystals, and then withdrawn and simultaneously rotated about its axis at programmable rates through a cooled neck around the hole in the oven shell.

The 7 lamps are stored on a "lamp disk", whose axis is parallel to those of the lamps, and which transports them around and over the oven insertion hole. The lamps are driven in and out of the hole by a motor mechanism. The 24 samples, some of which are actually calibration probes, are stored on a similar disk at the other end of the oven, and are moved in and out of the oven by a motor mechanism which can also rotate them on their own axis.

Essentially, by monitoring temperatures and other information from along the sample, its holder, the cooling system and the oven shell, it is possible to gain an insight into the growth process (which is not simulated by AMFESYS !).

The AMFESYS model can simulate all operating cycles of the AMF as a function of time, and of the appropriate input telecommands which configure and program those cycles. The model simulates the status of all the mechanisms, and emulates the processing and measurement cycle control being done by the AMF microprocessor. The input of the required telecommands for any desired demonstration test case is effected by feeding to the AMFESYS model a stream of time-tagged TC history records which it interprets as a schedule of commands for execution at the stated times. This thus emulates the reception, by the real AMF, of commands from the EURECA On Board Data Handling System (OBDH) which processes the on-board TC Master Schedule. On the other "side", the AMFESYS TM tracker inputs a (history) stream of corresponding AMF housekeeping TM packets, using the packet time stamps to synchronise the handling of these TM packets with the currently-simulated time of the model. The model time for any test run is therefore set up to run through the time period covered by the corresponding TM and TC history streams.

The discrepancy detector module tries to deal with discrepancies between the TM parameter values, and the state of the model, either by ignoring them if they are small enough (eg minor differences in measurements from analog sensors, or transient status inconsistencies), or by modifying the current status of the model if a reasonable and "nominal" explanation can be found to explain the differences. For example, the heating-up of the sample may sometimes take less time than the model predicts, thus resulting in an apparently premature start of the sample withdrawal, but this could be explained by the discrepancy detector, and it could advance the processing cycle in the model correspondingly.

If no "nominal" explanation for the discrepancies can be found, the diagnosis module is called on to make a more thorough analysis, which will usually result in the conclusion that a malfunction of some kind has occurred. The diagnoser presents its diagnosis to the operator, and proposes a corresponding "forced" change to the status or performance of the model, to correspond to the malfunction. The operator may accept the diagnosis and authorise the change to the model, or simply reject it, in which case AMFESYS will most likely throw it up again within a short time, or he may change the model himself because he decides that his diagnosis is the true one.

One could claim (although many would disagree) that the ability of the diagnoser to force changes to the model under the supervision of the operator, and in particular those changes which affect subsequent performance of the model (such as marking certain components as "failed"), constitutes a primitive form of adaptive "learning" about degradations of the real AMF, under the guidance of the operator. This is a little analogous to a pupil who says to his teacher (AMF engineer) "I think that motor1 has failed, don't you ?" to which the teacher will either agree, or supply his own better judgement.

Either AMFESYS will "learn" about degradations in the AMF, by recording them in its model under the operator's supervision, or the operator can impose the new knowledge directly. An "OK" status slot is defined for the general class *AMF devices* which is the parent of all AMF hardware object subclasses (eg switch, motor, spindle, gear, storage disk...). It is thus particularly easy to "fail" any object via the ProKappa Object Browser. However, it is only sensible to do this where there an object method which can use this information, or a corresponding diagnostic rule to detect the failure.

AMFESYS can recognise and adapt its model to six non-trivial anomolies:

1: on-defective sample end-position switch
2: defective sample position sensor pot
3: spare lamp defective
4: sample code unreadable
5: wrongly-coded sample
6: lamp code unreadable

In the real AMF, all of these failures result in some sort of attempt by the AMF microcontroller to circumvent the problem, and an "exception packet" will eventually be emitted in the TM. AMFESYS can however already detect and correctly diagnose these anomolies from the housekeeping TM alone; in fact it does not use report or exception packets yet, although there are plans under consideration to make it do so.

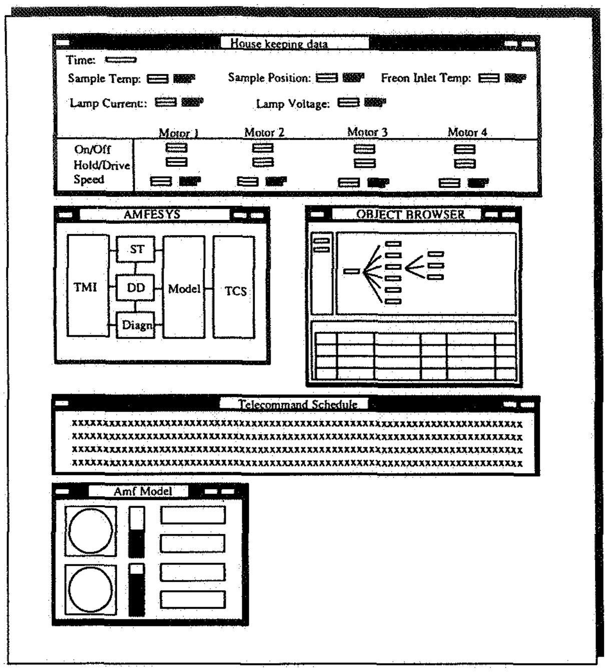

Fig 3: User Interface

4. CONCLUSIONS

The study was completed in March 1992, and the results demonstrate the feasibility of using a model as the basic reference for TM parameter limit checking, combined with automatic reasoning about the discrepancies, and adjustment of the model state under operator control. The available funding has restricted the study to demonstrate the technique for only one spacecraft subsystem, but nevertheless the AMF is a complex device, and entirely suitable as a "test case" for this type of experimentation.

As part of the task demonstrating the viability of the proposed approach to spacecraft control, an analysis was made of what monitoring facilities the user interface of such a system should give to the operator. The AMFESYS was provided with a user interface which not only shows him the evolving state of the model, but also a seperate display showing the comparison between the incoming TM values, and the corresponding values derived from the model state. It is thought that the operator would normally use just the model status display; the "TM comparator display" would be useful when an anomoly occurs, especially during the early lifetime of the model when the users might be rather sceptical of the system's ability to handle all operating conditions. The techniques proposed here will need extensive validation in an environment which represents as closely as possible the real operational world.

The work of the study described above now needs to be extended further, to investigate the problems which may occur in extending the scope of the concept, eg to a range of spacecraft subsystems, in handling historical information, in coping with loss of TM packets, and in handling the non-chronological arrival of TM information. AMFESYS also seems to be a potentially good starting-point to develop the concept further by doing some experiments in model-based hypothetical reasoning, because of its architecture, and also since ProKappa allows the dynamic definition and instantiation of new rules and object classes.

Work needs to be done to set guidelines for the approach to the design and implementation of the models, as well as on the question of how to accommodate such models within the hardware and software infrastructure of future SCS. ESOC's experience over the last 20 years, in developing and using full spacecraft system simulators for ground system and procedures checkout and for operator training, provides a solid foundation for a methodology for the development of models integral with future SCS. However, it is important to keep seperate the development of intra-SCS models from the development of system simulators, which provide an essential independent reference against which to test the behaviour of the ground system. There are questions of both a technical and strategic nature to be considered.

The use of models in SCS is a technique which can be introduced gradually, first concentrating on spacecraft subsystems or functions whose behaviour is simple to model. For every additional subsystem whose TM housekeeping data can be monitored using the model paradigm, there will be a corresponding improvement in the operations engineer's understanding of what the spacecraft is doing. This will allow him to concentrate his attention on events that really matter, and so reduce the risk of human error in operations.

5. REFERENCES

1. "Model-based reasoning for operations support for the Automatic Mirror Furnace in orbit". R Dobiasch, G Schweizer, J Wheadon. Proceedings of the Workshop on AI and Knowledge-based Systems for Space, ESTEC, May 1991.

2. "A model-based approach to fault diagnosis and health maintenance in a spacecraft power subsystem". S Conway, AI Dept University of Edinburgh 1987.

6. ACKNOWLEDGEMENTS

AMFESYS was created as part of an ESA-funded study done by ITV, Karlsruhe (prime) and Siemens Austria, Vienna. Thanks are due to Prof G Schweizer, and G Theiss of Karlsruhe University, who analysed the problem domain, principally the functioning of the AMF. Thanks are also due to R Dobiasch of Siemens who performed the complete design and implementation of the software.