INTELMOD
– An Intelligent System for Capturing Operations Knowledge and Providing Advanced Operations Support

A. Donati
Special Projects Division, ESA Directorate of Technical and Operational Support, ESOC, Darmstadt, Germany

E. Romani
Dataspazio, Rome, Italy

This article provides an overview of the implementation of a knowledge-based intelligent system to support flight operations personnel in their mission execution tasks, by means of a decision support system for fault management. After a brief snapshot of expert-system techniques, the inherent complexity of flight-operations knowledge is reviewed, before addressing the concept, architectural definition and implementation of the INTELligent MODeller (INTELMOD). The customisation of INTELMOD for the Cluster-II mission, and potential future applications, are also described.

Introduction

Knowledge-based systems

Artificial Intelligence is a branch of computer science research that, sometimes in the past, has generated expectations in potential future users well beyond the possibilities and results subsequently achieved. The consequent mistrust is somehow influencing today's approach towards the implementation of intelligent applications.

In the meantime, the applied research has made progress and commercial applications of intelligent systems are now available and being used quietly and successfully in several application fields, including manufacturing, communications, and operations management, and in business areas like marketing, logistic and finance.

The evolution in this domain has stepped through the rule-based behavioural system to the knowledge-based system, making use of object-oriented modelling, up to the inclusion of fuzzy logic and neural-network techniques, within the inference engine. A major area for further improvement is still the knowledge representation and the knowledge transfer between human 'experts' and the 'assistant' systems. Examples range from the classical procedural representation, where the knowledge expressed can be represented in the form of algorithms (e.g. operations manuals) to the declarative representation, such as statements stored in the form of symbolic structures accessible by general procedures capable of treating the knowledge thus expressed.

Modelling techniques to represent systems dynamics and decision support systems were researched in early 1990s. Substantial progress has also been achieved in the refinement of expert systems' control strategies and in the definition of methodologies for human-machine systems research. The availability of commercial tools to provide a user-friendly development and implementation environment has facilitated the introduction of intelligent systems in the domains noted above. The enhancement of human capabilities and the automation of specific activities were (and are) the major immediate objectives for such applications, with intelligent systems expected to contribute positively to risk mitigation and cost reduction.

Expert systems in mission-control domains

The demand for ever greater satellite performances has resulted in a continuous increase in the complexity of spacecraft platforms and payloads in recent years. Progressive use of on-board software, implementation of automatic or autonomous functions, increased complexity of on-board data-handling system and fault management are just some examples. The payloads themselves make increased demands on mission control systems, in terms of higher payload duty cycles, more payload modes, and shorter mission planning cycles.

At the same time, the pressure for savings has been reflected in the role of the operators, who are sometimes overloaded with tasks and responsibilities previously assigned to more than one expert. There is therefore an implicit necessity to provide additional support, in particular during critical mission phases, to the flight operations staff in order to make their work safer, more efficient and more competitive.

Expert systems could provide an answer in specific and well-defined domains: monitoring, failure management, trend analysis, planning and resource management are just a few examples among many. However, effective benefits from expert systems can only be experienced after thorough iterative refinement and validation of the stored knowledge.

In the area of supervision and fault detection and diagnosis, for instance, knowledge-based systems can provide a performance enhancement for the traditional monitoring activity. The current limit-value-based supervision method is simple and reliable: it provides an alarm only after a sudden fault or a gradual trend. However, the alarm is generated at individual parameter level, but does not represent a synthesis of the overall situation; moreover an in-depth fault diagnosis is usually not possible. Advanced methods for supervision and fault diagnosis would provide an answer to early detection of small developing faults, to diagnosis at unit or subsystem or system level, and to supervision of processes in transient states.

Lessons learnt from past science and applications space missions show that the majority of on-board failures occur in the attitude- and orbit-control systems and in the data-handling systems, which are known for being highly complex and software-driven. In addition, experience shows that the occurrence of simultaneous multiple failures within the same area is not as rare as is often claimed. Such results merely serve to confirm the urgency to identify additional supervisory and diagnostic tools for the operations staff.

ESA's European Space Operations Centre (ESOC), as a centre of excellence in the research and delivery of flight-operations services to the spacecraft user community, continuously strives for improvement in its own processes, methods and tools in order to maintain its outstanding record of successful mission operations and to make them available to other flight operations centres within Member States. In this context, INTELMOD represents a pioneering activity in understanding and exploiting, at prototype level, the potential benefits offered by presently available knowledge-based system technology when applied to flight-operations processes.

Previous studies in the mission-control field

Past research at ESOC associated with artificial intelligence has covered such topics as automated procedure selection and execution, timeline planning, spacecraft dynamic modelling versus real-time telemetry and diagnosis, architecture concepts for operations automation, and the applicability of advanced technology (ATOS-4).

The INTELMOD activity was preceded by an initial prototyping exercise using the same development platform, with the aim of exploiting the basic capabilities of an object-oriented rule-based system and support system-level monitoring, fault management and resource-consumption evaluation. The software was, at that time, hard coded and took the Automated Transfer Vehicle (ATV) as the reference mission. A simple ATV telemetry and failure-injection simulator was included within the application. The results, both in terms of demonstrated capabilities and potential applications to adjacent areas such as support to flight operations training and procedures definition, were satisfactory and paved the way for the INTELMOD concept.

Operations knowledge

How a spacecraft is operated

The recipe for flying a mission requires, as ingredients, a mixture of trained human resources, pre-programmed computers and pre-validated procedures. The nature of the tasks to be executed varies with the phase of the mission, but they can be grouped into two families: executional tasks and supervisory tasks. The first are usually automated, on board, and used for the implementation of the mission. They require progressive uplinking of pre-programmed automated command sequences. Critical executional tasks, such as GO/NO-GO decisions, are usually performed manually by the flight controllers.

The supervisory tasks include the recognition that all executional tasks are progressing as expected and the verification that all measurements are within the agreed fields of tolerance. The required supervision of the execution of complex, parallel tasks and the monitoring of hundreds or even thousands of parameters gives the flavour of the complexity and the responsibility assigned to the flight controllers, equivalent to those of a crew in an aircraft cockpit.

The mission follows a Flight Operations Plan. It consists of temporally sequenced, pre-validated procedures giving the spacecraft flight controller instructions on the monitoring of telemetry parameters, and on controlling the evolution of the mission by means of pre-configured and validated telecommands. The real trajectory is monitored against the planned one and manoeuvres are prepared and executed according to the plan.

Vigilant supervision implies the capability to quickly recognise anomalies and to react, with the prime objective of continuing the mission, via the implementation of specific contingency recovery procedures (e.g. activation of a redundancy unit). The diagnostic activities are then carried out by the experts, to understand the cause of the failure and to identify the necessary corrective and preventive measures to be implemented.

All of the spacecraft platform and payload subsystems, although loosely coupled, when properly engineered, have anyway a certain degree of interactivity and mutual dependency. This makes matters more complicated if a specific subsystem unit stops working correctly. A deep knowledge of the on-board architecture, functions and interdependencies is mandatory for the flight controller sitting at the console, and makes ‘flying a space mission’ an activity that can only be learned by experience.

The knowledge repository

The fundamental knowledge required to fly a mission is initially based on the information provided by the manufacturer in the Spacecraft User Manual and on the mission-analysis results. It is then complemented with the Mission Database, providing the definition and validity of all telemetry and telecommand parameters, and the Flight Operations Plan, written by the ‘pilots’ of the spacecraft, consisting of detailed timelines and associated operational procedures for nominal and contingency cases. Most important of all is the knowledge that the ‘spacecraft pilots’ have acquired during the mission-preparation phase, in specialised training sessions, while writing and validating operational procedures and participating in simulation sessions.

INTELMOD

The primary objective of the INTELligent MODeller is to demonstrate in practice whether new technologies could be beneficially applied as new tools for supporting human judgement of complex process anomalies and of related decision making. As such, it has to provide the flight-operations experts with an experimental test bed to probe the capabilities and limits of intelligent systems as a sophisticated advisory tool in support of complex and time-constrained operations tasks, using data from real missions.

The ‘toolkit’ was initially developed by knowledge engineers, providing all the necessary features to save and exploit available knowledge, and is then directly ‘programmed’, or customised, and further exploited by the flight-operations expert staff. It is connected to the existing Mission Control System (MCS) and, for cost-efficiency reasons, is based on a commercially available object-oriented software development and utilisation environment for intelligent applications, with already existing high-level functional blocks (e.g. for diagnostics) and interface blocks.

Conceived as an evolutionary toolkit, INTELMOD will provide a sufficiently user-friendly man/machine interface for the user who is knowledgeable about the spacecraft functions and mission plan, but not necessarily expert in low-level programming. The system is open, modular and expandable. Its library allows progressive growth and re-usability of modelled objects. The toolkit is now available for testing in its prototype version, after which the software knowledge engineers will implement/correct further features based on user feedback.

The operational concept

The flight operations staff can use the INTELMOD toolkit during both the mission-preparation and mission-execution phases. In the first phase, the spacecraft operations domain experts implement and organise the knowledge of the spacecraft system, mission phases, and diagnostic rules into the INTELMOD knowledge database. The INTELMOD toolkit provides a user-friendly modelling environment. During mission operations, the stored ‘know-how’ is driven by telemetry and telecommand data and provides advice to operations staff when, and possibly just before, an anomaly occurs. As is usual for such pioneering systems, it will initially be used to support very specific spacecraft units. It will be interfaced to the existing Mission Control System with the objective of optimising and expanding the current MCS-supported functions.

The users

To support the different stages of model development and implementation, three different INTELMOD user profiles have been identified:

– Spacecraft Component Developer (SCD), responsible for the creation of spacecraft components (modules, subsystems and units), which are then inserted into a Component Library to be used later during a model-definition phase. Component

Developers are expected to have a high level of knowledge concerning typical spacecraft 'building blocks'.
– Spacecraft Model Developer (SMD), creates a mission-specific model representation by selecting and configuring items created by the SCD. Models are progressively assembled and configured to provide a physical, functional and mission-related representation of the spacecraft in question.
– Spacecraft Operator (SO), interacts with the models created by the SMD during mission operations / training scenarios.

The toolkit has been conceived to be used in two modes, for two distinct phases of the spacecraft operations lifecycle (Fig. 1):

– *INTELMOD customisation during mission preparation*
The users will access the toolkit as SCD, to create, modify or augment the models of the 'terminal' elements of the hierarchical representation of generic spacecraft, to be then stored in a library. In the very same phase, the user will access also as SMD to model selected subsystems, down to the end item, the related mission modes, the diagnostic and failure propagation rules, the contingency procedures and the trend analysis rules belonging to a specific spacecraft and mission.

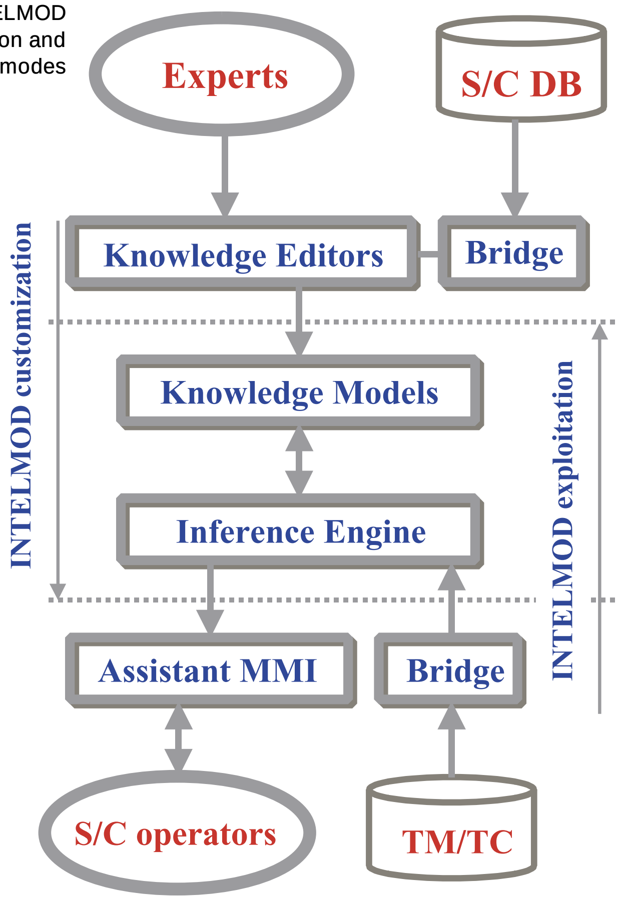


Figure 1. INTELMOD customisation and exploitation modes

– *INTELMOD exploitation during the flight execution phase*
INTELMOD will be connected to the existing MCS, to receive telemetry (and a copy of telecommands). This time the toolkit will provide the INTELMOD SO, as part of the flight-control team, with an operational advisory service throughout the mission-execution phases.

The spacecraft operator can be supported by INTELMOD for:
– enhanced visual monitoring, at system level (synthesised monitoring) and alarm alerts
– diagnostic support, including failure detection and anticipation, failure isolation, diagnosis and recovery, and failure propagation analysis
– resource evaluation and assessment.

The toolkit has been developed taking into account the following requirements:
– support multi-mission environment
– user-friendliness of the interface for the operations and spacecraft experts during both modelling and flight-operations phases
– minimal software customisation effort when applying the toolkit to a specific mission limited to interface adaptation
– hardware-platform-independent application
– open interface, easily adapted to the existing Mission Control System environment.

The toolkit architecture

Knowledge representation techniques are still in an evolutionary phase and the INTELMOD developers have therefore designed the toolkit using structured knowledge domains specifically adapted to contain, respectively, spacecraft knowledge, mission knowledge and functional knowledge, by means of objects, procedures and rules. A set of editors for each knowledge-representation domain has been specifically developed: they represent the meta-knowledge domain of INTELMOD. A logical decomposition of the INTELMOD knowledge architecture is represented Figure 2, which is a high-level object-oriented diagram showing the logical editors' behaviour and user relationships.

The editors implemented are suitable for supporting the acquisition of knowledge represented in the spacecraft model, in the mission model and in the behavioural model, as shown in Figure 3 and described below.

The spacecraft model
This model provides a hierarchical representation of the spacecraft, its subsystems and individual components. For example, a spacecraft may be partially represented in terms of power, thermal and AOCMS subsystems. The power subsystem in turn may be composed of a

power distribution unit, batteries, etc. This knowledge is entered using a breakdown editor to interactively gather and structure knowledge related to the physical organisation of the spacecraft. The breakdown editor configures itself according to the user currently interacting with the system (SCD, SMD or SO).

The mission model
The Mission Model provides a hierarchical representation of the mission in terms of the various activities performed within various phases and modes, together with the expected configuration (e.g. status and resource consumption profile) for all the physical components defined in the physical spacecraft model. As with the physical model, a breakdown editor allows the SMD to gather and structure this knowledge.

The functional model
The functional model uses a graphical rule-based language to define knowledge related to the functions to be performed during the course of a mission. This knowledge falls into the following areas:

– Spacecraft behavioural knowledge: includes knowledge describing the behaviour of the spacecraft systems with respect to the interaction between the various components and subsystems. This knowledge enables the model to perform basic diagnostic functions including failure isolation and recovery.
– Mission behavioural knowledge: describes the spacecraft behaviour exhibited during the execution of different mission phases and the activities performed during those phases. This model also uses the Flight Operations Plan (FOP) to enable INTELMOD to perform resource evaluation.
– Spacecraft/mission relationship knowledge: captures the heuristics used by the operations, spacecraft and payload engineers to identify and rectify problems that occur over the lifetime of the spacecraft. Once defined, the spacecraft/mission relationship knowledge enables INTELMOD to perform trend analysis, failure detection, diagnosis and prevention
– Spacecraft/mission propagation effect knowledge: cause and effect knowledge, available from the Flight Operations Plan and mission specialists, which relates sections of the spacecraft and mission models. A causal network allows the flight controllers to perform an analysis of process and hardware failures and predict the consequences of failure if no corrective action is taken. It also allows the controller to assess the impact on the mission in terms of unavailable hardware and lost functionality.

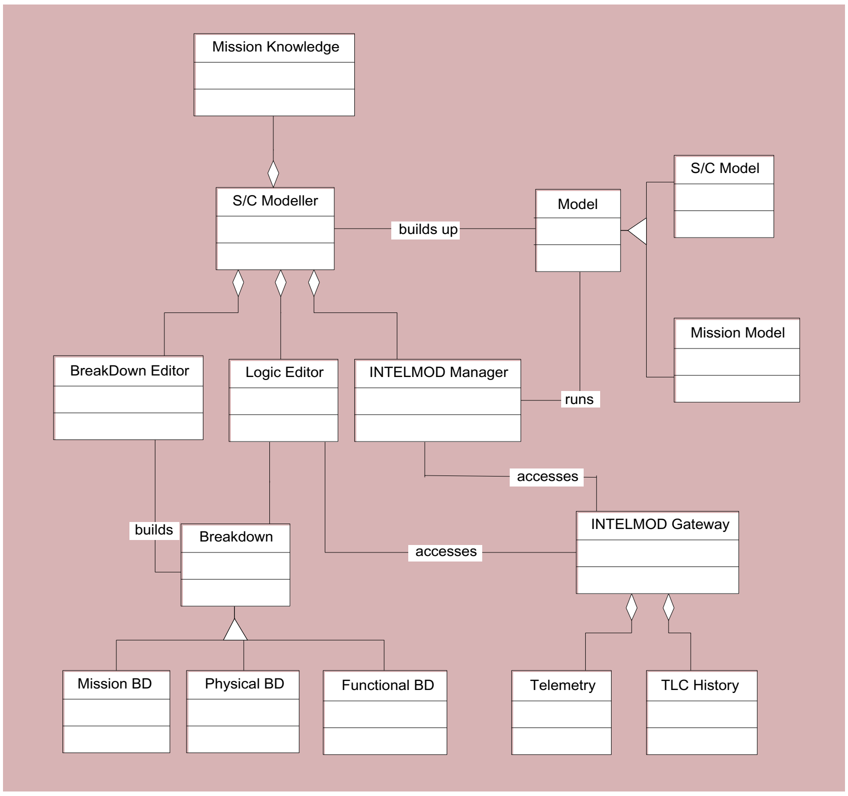
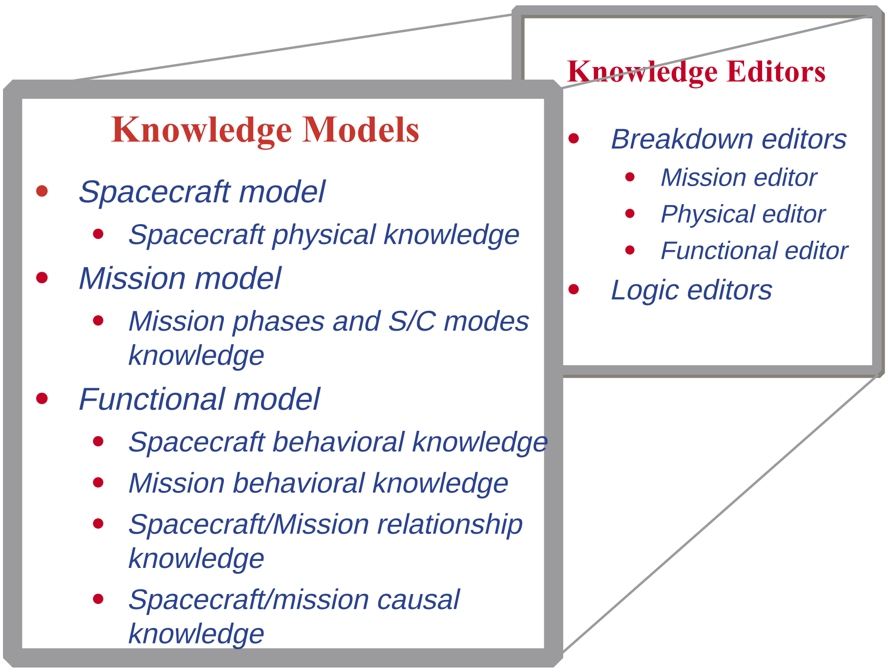


Figure 2. The INTELMOD knowledge architecture

The INTELMOD knowledge models and associated supported functions are summarised in Figure 4.

The Cluster-II test case
The modelling
The INTELMOD toolkit was initially tested, from a functional point of view, by simulating support for the AOCMS and power subsystems of Cluster-II. Taking the power subsystem as an example, this has been decomposed using INTELMOD's breakdown editors into the

Figure 3. INTELMOD knowledge editors and knowledge models



following units:
* power control
* power distribution
* internal power dumpers
* external power dumpers
* battery regulation
* batteries, and
* pyro-electronics.

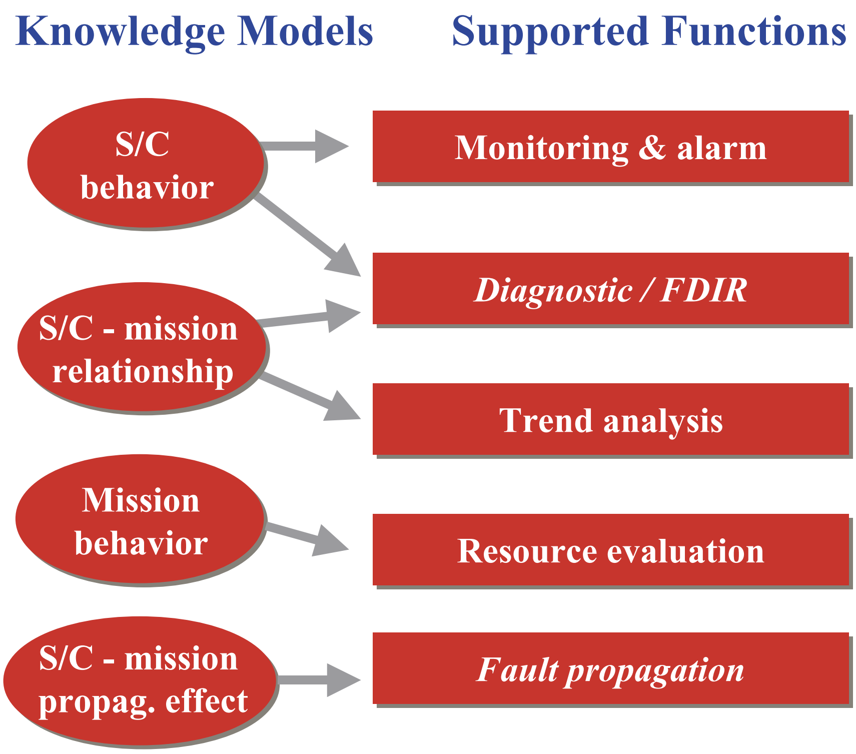


Figure 4. INTELMOD knowledge models and supported functions

Once these breakdown components have been identified, attributes can be added, again using a dedicated editor; e.g. a battery would typically be described using properties such as voltage, temperature, charge current and discharge current (Fig. 5).

Having linked the INTELMOD breakdown components with their external counterparts (i.e. telemetry parameters or groups of telemetry) by interactively querying the spacecraft database, it is then possible to construct the monitoring and diagnostic 'rules'. Monitoring rules allow mapping of the status of breakdown leaves (as derived form the incoming telemetry) with synthesis information: whenever the status changes, the colour of the related component displayed in the breakdown mimics changes according to the following table:

1 - green: OK
2 - red: fault
3 - yellow: off-line
4 - blue: stand-by
5 - orange: redundancy lost.

This information travels upwards in the hierarchy. Once colours have been computed for all the components belonging to a specific level (e.g. units), the same colours are logically combined to derive the colour representing the 'assembling' component at the upper level (e.g. subsystem). Colour propagation and telemetry association logic is entered within

INTELMOD Implementation

The INTELMOD toolkit has been developed using a RAD-style (Rapid Application Development) approach, based upon the Dynamic System Development Method (DSDM), which is a non-proprietary method developed in the UK and currently used worldwide. This approach was partially adopted in this project to help ensure that the system could be developed in a much shorter time scale, and that the final system would more closely match ESA's real needs.

DSDM employs an iterative approach to development with heavy emphasis on end-user involvement and a project-management philosophy focusing on products rather than on the activities needed to achieve them. Time-boxes were used to control the development process, allocating a fixed amount of time to complete a given area of functionality.

INTELMOD has also made extensive use of commercial off-the-shelf (COTS) software products, including: G2, GDA, G2-Weblink, ODBC Bridge and Space UniT. The G2 (Gensym Corporation) software platform provides an object-oriented environment for building and deploying mission-critical, intelligent applications. It is typically used to represent knowledge captured from operations experts performing complex tasks in real-time situations. GDA (G2 Diagnostic Assistant) is a layered application product for G2, which provides an integrated visual development and execution environment for modelling application logic/diagnostics. Its intuitive graphical user interface allows faster development of the complex system models required for INTELMOD. G2-Weblink allows the distribution of intelligent decision-support information to intra/internet users throughout the organisation. Gensym also provides bridges for ODBC-compliant databases, in our test case a Microsoft Access copy of the Cluster-II database. Space UNiT (Universal Intelligent Toolkit, from Science Systems Space Ltd.) has been developed in a partnership programme for ESA to provide a component-based suite of graphical products for procedure execution, schedule execution, monitoring and event handling. It enables INTELMOD to automatically prompt (and execute) contingency procedures following the detection of anomalies by the functional model.

These COTS products were used in order to provide rapid delivery of high-level functionality required. In addition, the industrial-partnership approach with Dataspazio and SSSL allowed the project to remain within the allocated budget.

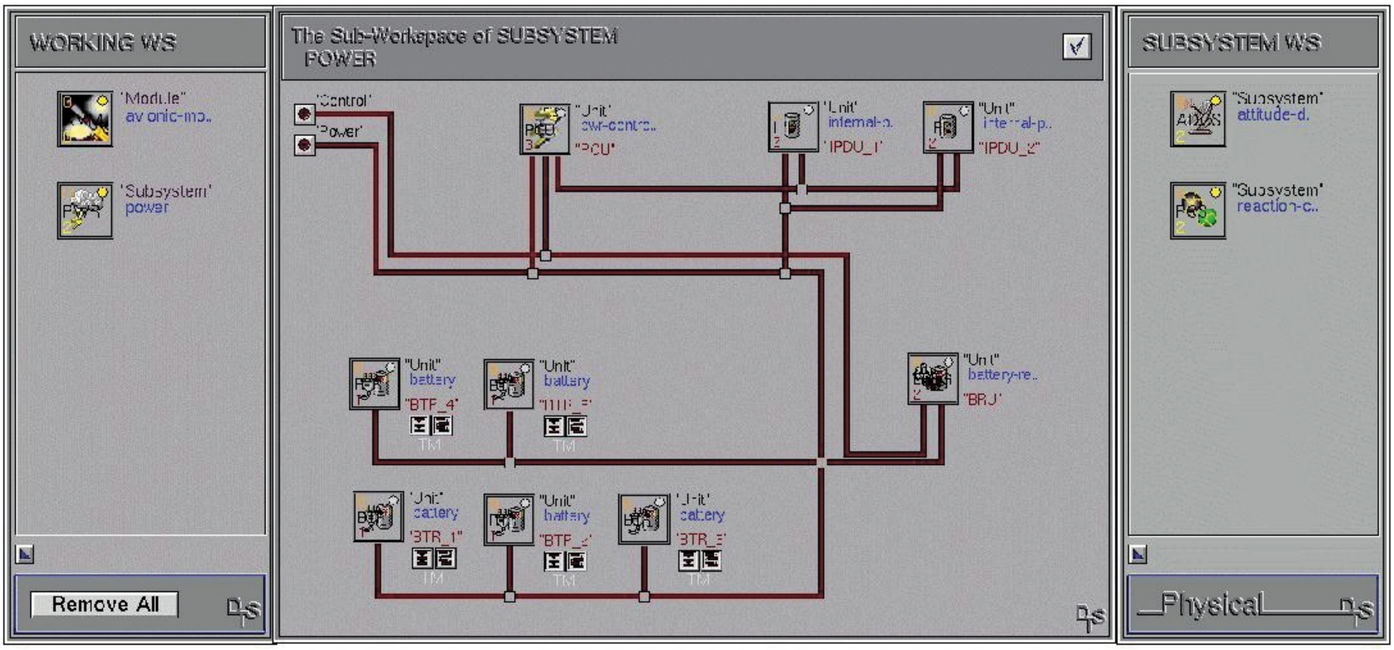


Figure 5. Screenshot of breakdown components

INTELMOD via dedicated parsers, which allow this logic to be defined in a 'natural language' way.

Figure 6 shows, as an example, a simple INTELMOD GDA-based diagnostic (spacecraft behavioural) model. The blocks on the left of the diagram are 'entry points', usually corresponding to a telemetry value that can be automatically created from the breakdown components. Signals are fed through various GDA logic blocks in an attempt to diagnose the cause of operational problems - in this case an internal power subsystem failure arising from a battery over-discharge. If all the logic paths entering the 'AND' block on the left of the diagram are true, then a diagnosis can be made. A message will be sent to one of INTELMOD's message areas, alerting operators to the cause of the problem. It should also be noted that the outputs/conclusions of one GDA diagram could pass information to other diagrams and other INTELMOD model types. Customised GDA blocks are available to link diagnostic models with fault propagation models.

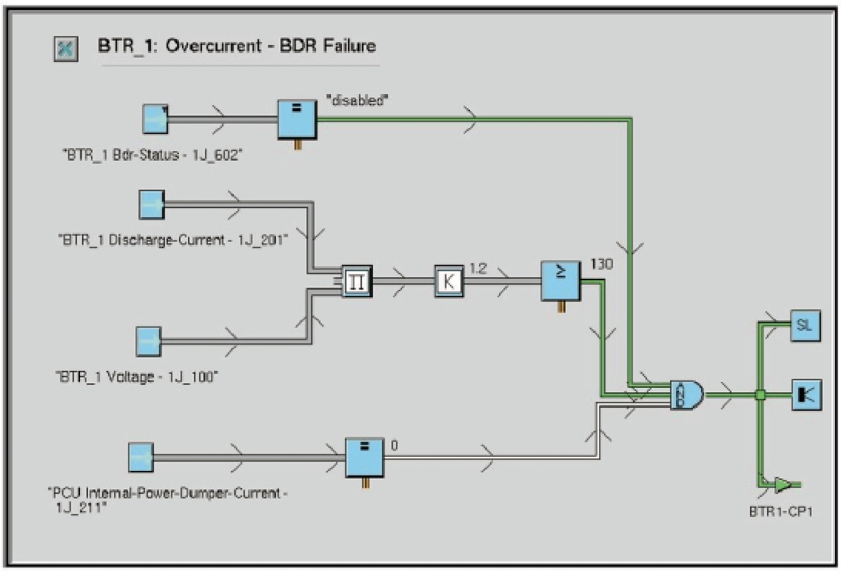
In this way, operators are not only alerted to system failures and their potential causes, they can also be supported in assessing the likely knock-on effects, when these are expected to occur, and the impact on mission operations. In the example shown, the GDA model also incorporates a link to a UNiT procedure (shown by the block labelled 'SL'). Consequently, when the diagnosis is made, a contingency procedure can be automatically invoked to provide failure recovery.

All of INTELMOD's model types share a common mode of use. Model Developers use pull-down menus and palettes to select the building blocks that are required. These are placed on a workspace, configured with any necessary information, then connected together. Such models are then immediately ready for use, allowing the developer to concentrate on the expression of expert domain knowledge rather than writing conventional programs.

The external interfaces

In order to demonstrate that INTELMOD could operate in a realistic manner and could provide the expected support to flight controllers during mission execution, the system had to be provided with a high degree of connectivity. This was achieved using three separate interfaces (Fig. 7): a telemetry/telecommand bridge, a spacecraft database bridge, and an Intranet bridge (for message broadcasting).

Figure 6. GDA-based diagnostic model



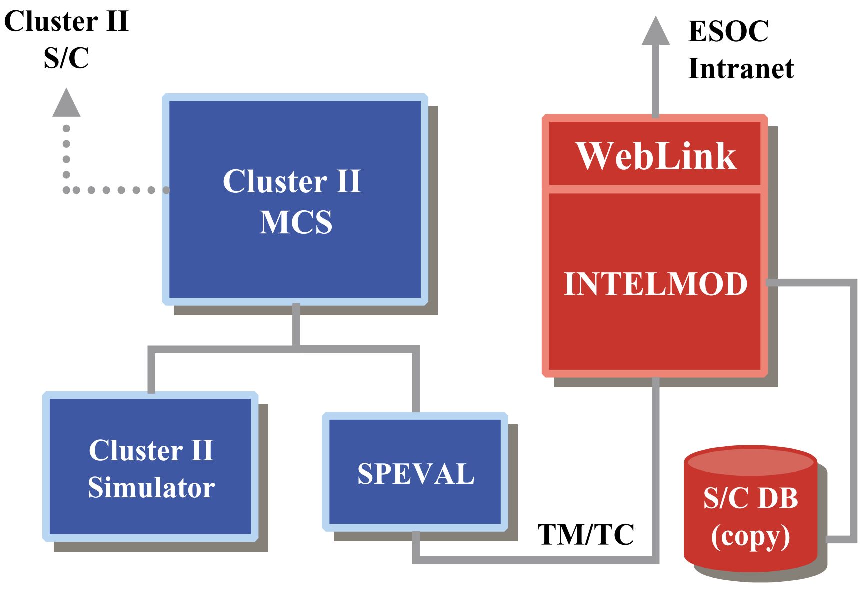


Figure 7. INTELMOD interfaces for Cluster-II implementation

The telemetry and telecommand bridge provides INTELMOD with the information required to perform its analysis. In a 'live' implementation, this information would be supplied directly from the Mission Control System (MCS) software. However, in the time frame of the current study, data files were provided using a client server application (SPEVAL) developed during a previous ESOC study. SPEVAL maintains its own archive of data from the Cluster-II MCS, can be read by INTELMOD using a dedicated bridge.

The purpose of the database bridge is to connect the INTELMOD modelling environment to a satellite database containing information regarding the various parameters, command definitions and scaling/limit data specific to the particular spacecraft under study.

To provide a means for the distribution of alarms, warning messages and general information, G2 Weblink has been incorporated into INTELMOD. Whilst it would have been possible to provide a dedicated bridge to send, for example e-mail messages within ESOC, the use of Weblink enables information to be accessed from the widest possible range of platforms by means of web browsers.

Simulations and evaluation

All of the functions implemented within INTELMOD have been successfully tested through simulation sessions using Cluster-II telemetry history files. The next step will be the operational validation in a real environment with the spacecraft operator using it in open loop during a mission to supervise a target critical area of a spacecraft. This will allow INTELMOD's performance to be judged in the real operational environment.

Conclusion

INTELMOD is a generic satellite-modelling toolkit that has been developed to offer faster, incremental development of spacecraft models. Thanks to its user-friendly man/machine interface, the system requires no formal programming expertise.

The use of AI techniques within INTELMOD has provided a significant enhancement in supporting such operational tasks as system monitoring, anomaly detection and anticipation, failure detection, isolation, diagnosis and recovery, failure propagation analysis and resource management. The toolkit could also be exploited in the training of new operations staff during simulated test-case sessions, making use of the previously captured expertise. Moreover, it could potentially assist engineers during trade-off analysis, specifically to investigate alternative design solutions, alternative operational strategies and contingency recovery procedures.

To be justified, the investment required in the modelling process has to have an economic return. From initial estimates, the toolkit should provide substantial cost savings in the flight-operations budget, especially if applied to long-duration missions (e.g. interplanetary missions) or recurrent/ repetitive missions (e.g. satellite constellations, meteorological satellites, etc.). In the first case a common repository of deep knowledge of spacecraft behaviours is guaranteed, facilitating the turn-over in flight control teams over the years. In the second case, the multi-mission operation supervisory tasks are eased, mitigating risk and/or facilitating the assignment of operations responsibilities. In both cases the 'modelling' investment is either redistributed across several years of operations or across several parallel missions.

The results achieved with INTELMOD to date indicate that it can pave the way for an innovative operations concept in which AI-based tools, integrated into an existing mission control system, will provide more effective and efficient support to the flight controllers during both safety-critical and routine operations. esa

The authors would welcome feedback from others working on similar projects. They can be contacted by e-mail at:
adonati@esoc.esa.de and
enrico.romani@dataspazio.it.