

Database Administration for Spacecraft Operations – The Integral Experience

J. Houser & M Pecchioli

Mission Operations Department, ESA Directorate of Technical and Operational Support, ESOC, Darmstadt, Germany

Introduction

Long before the integration of a spacecraft begins, the structure of the spacecraft database is defined and documented. Because of the large amount of data involved, a proper and clear definition of the database structure is crucial to minimise the effort subsequently required to populate and maintain it.

ESA's flight control teams have always used databases for spacecraft monitoring and control. The mission control systems rely on a database to interface with the spacecraft, in order to know how to encode the telecommands sent and interpret the telemetry received. This approach allows the control system software and the data describing the controlled domain to be decoupled. Given the number of changes typically implemented in a database prior to and even after launch, no other approach would be feasible.

The population, validation and configuration control of the spacecraft database therefore play an important role in mission pre-launch activities. Experience shows that a reliable database is possibly the most fundamental prerequisite for the successful execution of spacecraft operations.

The typical spacecraft database consists of several dozen tables, each containing a portion of the data. The tables are interrelated in that one table record can refer to items further described in other tables. Unfortunately, the database structure is influenced not only by the characteristics of the controlled domain (i.e. the spacecraft), but also by the design features of the control system itself. Given that different control systems are typically used by the satellite manufacturer's Assembly, Integration and Test (AIT) team and the flight control team at ESOC, several different database structures are normally defined. To support the exchange of data across the different formats, interfaces are formally agreed and data-conversion applications (import/export) developed.

The database is initially populated, used and validated by the teams responsible for the design and integration of the different satellite subsystems. The flight control team at ESOC would normally see it for the first time two to three years before the launch. At this stage, the database is typically in an incomplete and unstable state, and progressively matures throughout the integration and test phase. During the same period, the flight control team at ESOC will need to augment and sometimes modify it. These parallel activities introduce the need to ensure coherence between these different versions of the database and to coordinate their maintenance.

The size and complexity of spacecraft databases differs significantly depending upon the particular mission, and has been significantly affected by the evolution in spacecraft design over the years. For example, the adoption of modern telemetry/telecommand standards (e.g. the Packet Utilisation Standard) allows great data-handling flexibility on board and therefore implies the need to introduce more sophisticated techniques on the ground to describe variable packets, for example. The amount of data required to describe a spacecraft has also increased dramatically in recent years. For the early spacecraft, only about 500 telemetry parameters had to be defined in the database, whereas those for today's scientific satellites typically contain more than 15 000 parameters. This growth has imposed severe requirements on the functional and performance capabilities of database editors and mission-control systems.

The database maintained by the satellite manufacturer and delivered to ESOC is normally referred to as the 'Satellite Database, or SDB'. The database maintained at ESOC by the flight control team is normally referred to as

the off-line 'Operational Database, or ODB'. For performance reasons, the ODB is converted into an internal format when it is imported into the Mission Control System (MCS), which is normally referred to as the 'Run-Time Database'.

History

The 80-column card era

In the late 1970s and early 1980s spacecraft databases were defined and managed on 80-column punched cards. These cards were a matrix of holes, 80 columns wide by 12 columns deep. Each column was used to record one character of information as a series of holes. The first two rows were used for the header information, and the remaining 10 to represent the characters.

There were many punched-card machines in ESOC during this period. They were as loud and as large as today's photocopying machines. The problem with punched cards was flexibility – there just wasn't any! With punched cards it was almost impossible for the operations personnel to have an overview of the data contained within them.

As computer technology progressed, punched cards were replaced by electronic storage media, but the organisation of the data itself was not modified. The data were accessed using remote terminals connected to a central mainframe computer and edited using very basic text editors. Inputs about the database content were still received from the satellite manufacturer in paper form.

The 80-column card era has been the driving factor in the design of the database structure for many control systems. The database structure of the SCOS-1 control system, still in use today, can be mapped to 80 columns of information per record.

Relational databases

Relational databases (e.g. ORACLE) were introduced in the late 1980s with the development of the SCOS-1 infrastructure and are still being widely used in ESA. The use of relational database technology has opened the door to a progressive increase of the automation in the creation and maintenance of the spacecraft databases. ORACLE, in particular, supports a powerful query language (SQL), which allows interrogation of the database and the manipulation of all data meeting specified criteria (bulk data editing). It also supports a development environment, which has been used extensively to produce the editor forms. The visibility of the database content as well as the sophistication of the

syntactical and on-line consistency checks could be progressively improved. Most of the off-line databases that are managed currently at ESOC are still in ORACLE format.

Object-oriented databases

Object-oriented databases have only recently been introduced at ESOC both for the off-line and for the run-time database. The Rosetta off-line database system and the run-time database of SCOS-2000 (latest ESOC infrastructure for spacecraft control systems) are both based on object-oriented technology.

Database content

The content of the satellite database can normally be split into three main areas: telemetry, telecommand and flight dynamics. This last area is not addressed in this article, which focusses on the telemetry and telecommand databases of the Integral Mission Control System.

Telemetry

All of us are involved in one way or another in monitoring data in our daily lives, for example the water temperature indicators in our cars. What about attaching your thermometer to the inside housing of a spacecraft's service module and launching it into space? As opposed to seeing the data directly, as you can in your car, you encode the temperature value in the form of binary bits and place them in a data unit (e.g. a 'packet'), which is eventually down-linked from the spacecraft. In order to present you with an engineering view of the on-board status, your control system first needs to identify which data unit is being received and then to know how to extract and decode the telemetry data delivered by it. In most cases, the location of telemetry data in the transmission data units is fixed. This is the simplest and traditionally most commonly used approach, but not necessarily the most efficient one. To optimise the usage of the down-link bandwidth, it maybe required to transmit different data depending on some conditions related to the on-board status. The Packet Telemetry Standard allows this flexibility in that different packets can be generated depending on the actual data to be down-linked. A further level of sophistication may be required under some circumstances, whereby the content of some asynchronously generated packets depends on the prevailing on-board states.

Once the bits corresponding to a telemetry parameter sample have been extracted, the control system needs to know how to decode them (depending on the parameter type, e.g. integer, real, string, time) and how to calibrate them to derive a proper engineering view. One

aspect that is of fundamental importance in the monitoring of spacecraft telemetry data is an assessment of the validity of the data received. Is the temperature shown in your car a valid indication if the car is switched off? Validity conditions have to be specified in order to ensure that the data being used are reliable. This may sound trivial, but in fact it is not as the validity conditions are normally based on other telemetry parameters that also require validation. A chain of conditions is established, which is typically organised in a hierarchical manner. Once the validity of the received data is established, the system is supposed to notify the operator if the on-board status is not nominal. This is achieved by applying to each received parameter a set of checks ensuring that its value is within the allowed ranges (limit checks) and also that no unexpected change in on-board status has taken place (status consistency checks). A different set of checks has to be applied, for example, during different mission phases. This implies that applicability conditions have to be specified for each check, each condition also being dependent on telemetry data.

In many cases, the telemetry data received also have to be consolidated to ease the visibility of the on-board status. This is achieved by associating with a 'pseudo' telemetry parameter (derived parameter) a routine that is evaluated based on some trigger criterion, for example the reception of a contributing telemetry packet.

The telemetry databases contain all of the data required to identify, extract, decode, calibrate, validate, check, derive and display the telemetry data.

Telecommands

Telecommands are encoded messages transmitted to the spacecraft to trigger one or more actions on-board. They can perform many different functions, the most obvious one being to switch a unit on or off, as you would with your normal television remote-control unit. Of course, commanding a modern spacecraft implies a much higher level of complexity and also imposes severe safety requirements.

A telecommand is typically composed of several elements, which may be given a fixed value, or a default value which can be changed at each instantiation level (parameters). Mirroring the processing of telemetry parameters, a value for a command parameter can be specified/viewed in engineering form and thus has to be properly (de-)calibrated and eventually encoded in a format that depends on the parameter type. Parameter values can also be subject to

syntactical as well as range checks. Unlike previous standards, the Packet Telecommand Standard, and the Packet Utilisation Standard in particular, allow great flexibility in the encoding of a command packet, including a variable number of parameters, or parameters that may be given values with variable length.

The execution of spacecraft operations typically requires the transmission of many telecommands to achieve a specified goal. To ease this process, telecommands are normally collected in named lists (command sequences), which can also be associated with parameters (mapped to the parameters of the constituting elements). Command sequences are also allowed to contain other sequences (nested sequences), thereby enabling the hierarchical definition of the commanding activities.

The telecommand databases contain all of the data required to (de-)calibrate and check command parameter values, to encode, validate and verify commands and to store the definitions of command sequences.

The Integral experience

The Integral Mission Control System has been developed based on the SCOS-2000 infrastructure. As noted above, SCOS-2000 allows full freedom in the design of the off-line database system for a client mission. In the case of Integral, however, the structure of the off-line database was directly derived from the organisation of data expected by the SCOS-2000 database importer. This approach offered the advantage of minimising the conversions taking place when importing the database into the run-time system, and thus afforded the users access to all of the flexibility offered by SCOS-2000. Also, the organisation of the SCOS import data is fully 'normalised', in that many-to-many relationships between data are avoided by means of mapping tables. The adoption of a normalised database structure simplifies significantly the relationships between data items and thus the development of the database editors and consistency checker.

The size of the Integral database is significant in that it currently contains about 150 000 records, occupying more than 12 Mbyte of memory. As for many other missions currently managed at ESOC, this has imposed the need to electronically import as many data as possible from the database delivered by the manufacturer, thereby avoiding the risk – and effort – implied by manually inputting all data.

The data flow involved in the process of creating and maintaining the Integral off-line database is shown in Figure 1. This process is not a one-off

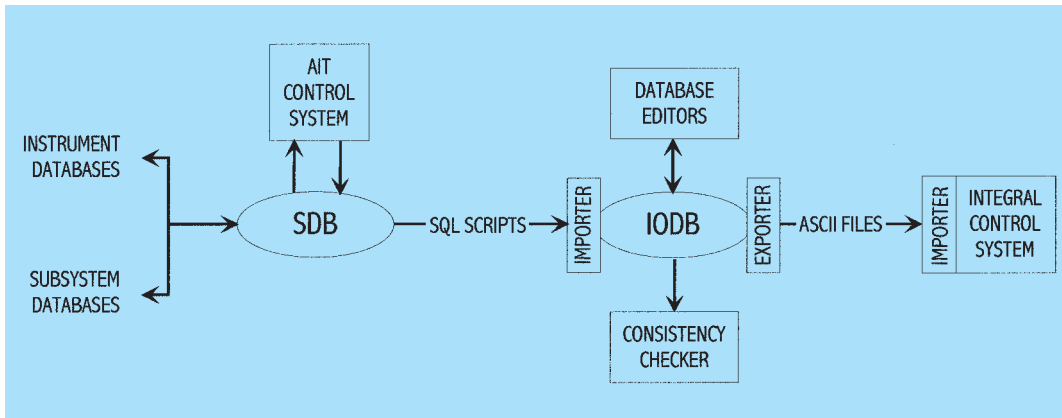


Figure 1. Data flow within the Integral database maintenance process

exercise, because in the years prior to launch population of the database basically takes place in parallel on the manufacturer's side and at ESOC. This is explained and described further in the following sections.

The Satellite Database (SDB)

The SDB is populated and maintained by the spacecraft manufacturer based on inputs from those responsible for each subsystem. It contains all of the basic telemetry/telecommand data required to interface with the spacecraft, as well as some basic Boolean expressions, which are used as telemetry validity criteria, check applicability criteria, and command pre-transmission validation checks. Each SDB version is delivered to ESOC in the form of SQL scripts, from which a Microsoft Access database with the same structure is created at ESOC. This database is not manipulated further at ESOC. It is only used as a reference and for importing data into the Operational Database (ODB, see below). A browser developed in MS-Access is available to all users to visualise the SDB content.

In order to ensure its integrity and consistency prior to importation into the ODB, the SDB is interrogated by the database administrator at ESOC. Any problems with that particular version of the SDB are thereby isolated and addressed during the importation process.

The Operational Database (ODB) editors

As noted above, the telemetry/telecommand data contained in the SDB are automatically imported into the ODB by means of a set of MS-Access queries (basically SQL queries), which convert the SDB data and import them into the appropriate fields of the ODB. Additional information not delivered by the spacecraft manufacturer as part of the SDB is required in the ODB. On the manufacturer's side, only those items essential to safely interface with the spacecraft are introduced into the database. The flight control teams at ESOC also have to take into account how the data are entered (commanding side) and how

they are presented to the end operators (monitoring side). This is crucial both during the execution of critical mission phases (when operations have to be executed under time pressure), as well as during the routine operations phase (when only a single spacecraft operator is responsible for the execution of all operations without any on-line engineering support).

Typical examples of data stored in the ODB but not imported from the SDB are the definitions of monitoring displays and command sequences, which the flight control team directly defines to be fully in line with the Flight Operations Plan. Another area where the ODB requires significant extension after SDB importation is the definition of derived parameters. Typically, hundreds of additional derived parameters are defined by the flight control team to refine the validity, monitoring check applicability and PTV* criteria, and also to present the on-board status to the spacecraft operators in the most convenient manner. Finally, other extensions of the ODB are required to introduce the data relating to all of the functions of the Integral Mission Control System that are not supported by the control system used by the AIT** team (e.g. status consistency checks, load/dump comparison verification checks, report based verification, variable packets definition).

Thus, although a large portion of the ODB data is directly imported from the SDB, database editors are nonetheless required to support the flight control team at ESOC in the definition of the additional data. Severe requirements apply in that the Man/Machine Interface has to be unambiguous, efficient, user-friendly and protective against the introduction of invalid data.

The Integral ODB (IODB) editors have been developed based on Microsoft Access, allowing easy integration of the several elements involved in a database editor, such as the Man/Machine Interface forms and the code written in Visual Basic. It also supports easy

*PTV= Pre-Transmission Validation

**AIT= Assembly, Integration and Test

access to an intuitive query definition environment (based on SQL), which significantly eases the data manipulation and visualisation. The IODB editors were designed with the continuous involvement of the end users, to ensure that the product would meet the actual user needs, including those sometimes not easily specified in the form of requirements. One of the main achievements lies in fact that the editor forms provide a very good overview of the data being edited as well as of any referenced data, as shown in Figures 2 and 3.

The IODB editors are designed for multiple users accessing a centralised set of tables from their own PC in their office environment. The database tables are based on a structure that

mirrors the data organisation and the relationships between data items as expected by the MCS database importer. This allows simplification of the database export function, which is a pure conversion into ASCII format.

Figure 4 provides an overview of the different elements related to the IODB editors and associated tools (importer and consistency checker). The IODB editors support a dedicated on-line function, which enables the user to understand exactly how a data field is going to be used in the Mission Control System and all of the constraints applying to it.

Last but not least, the content of a database needs to be documented. Reports are produced in order to support reviews and also

Figure 2. The calibration curve editor

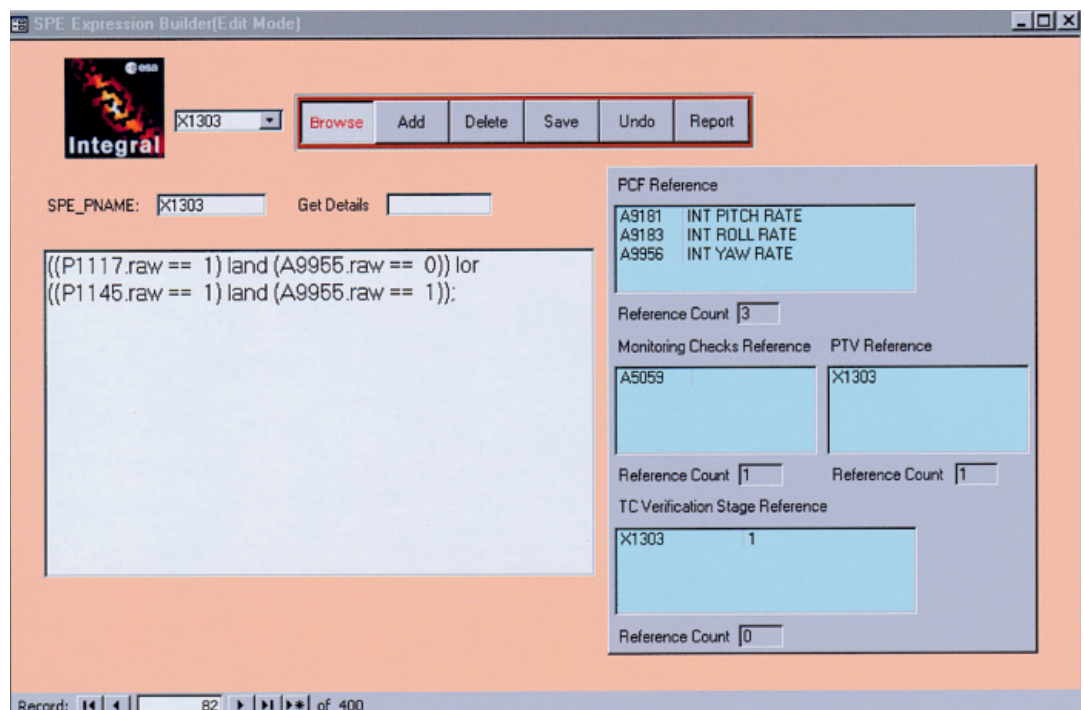
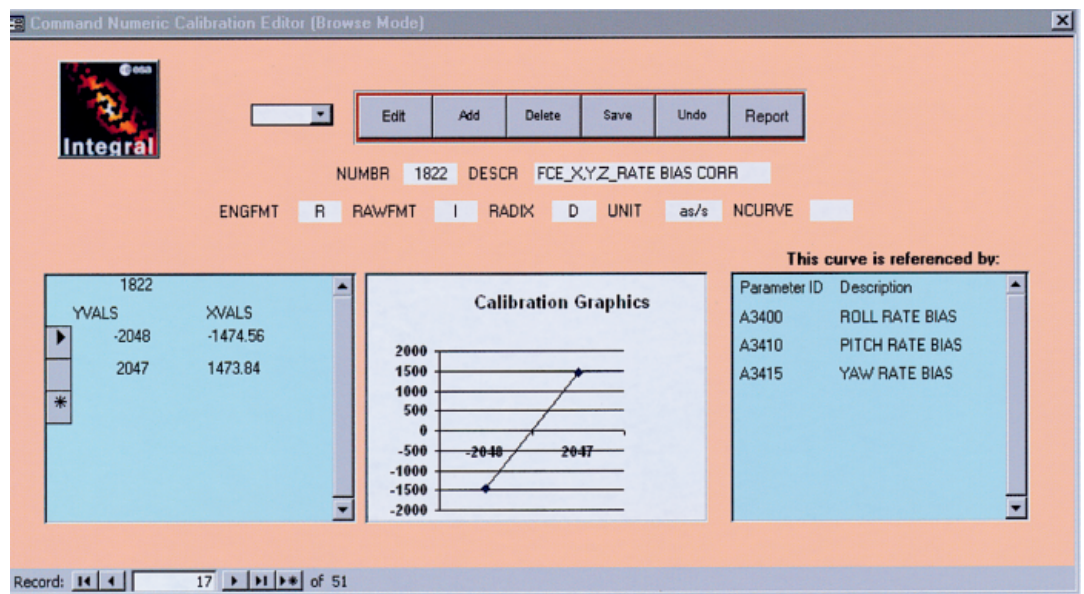


Figure 3. The derived parameters editor

to be used as a reference in the control rooms. The IODB editors support the generation of reports containing not only the attributes of each data item, but also cross-reference lists (i.e. which item is referenced by which other item).

The database consistency checker

The IODB editors are designed so as to avoid as much as possible the inputting of incorrect data. This is achieved, for example, by means of drop-down selection lists for all multiple-choice fields. Relational integrity is also automatically preserved using the relationships between tables by means of cascade delete and cascade update mechanisms. However, the execution of on-line checks is not adequate to ensure in all cases that no database inconsistencies are introduced. The database table organisation is complex, with many interactions between data. Executing all applicable checks following any user input would introduce unnecessary complexity and possibly in some cases even lead to a chicken and egg loop. Furthermore, a large portion of the data is imported from external sources, thereby imposing the need to check the overall consistency of the data after each data import operation.

For Integral, developing an off-line consistency checker tool has solved these problems. This application accesses the same database tables that are manipulated via the editors. It runs a predefined set of sophisticated queries about the current database content and returns a report that can be filtered based on tables, fields and type of checks. This approach has proved to be very powerful in providing users with a 'snapshot' of the current database status in terms of data consistency.

Database validation and configuration control

By the time a spacecraft is launched, its Operational Database must be completely error free and fully tested. There is no system in the world that can ensure this, and only an intensive test campaign can provide the required confidence. Several methods are used at ESOC to accomplish this task.

Spacecraft telemetry playback

The first satellite data typically made available to ESOC for testing are in the form of recorded telemetry data units. These data are useful not only to start testing the telemetry processing in the control system, but also to start exercising the telemetry database.

Simulations

Prior to launch, intensive testing activities using a spacecraft simulator take place. These are

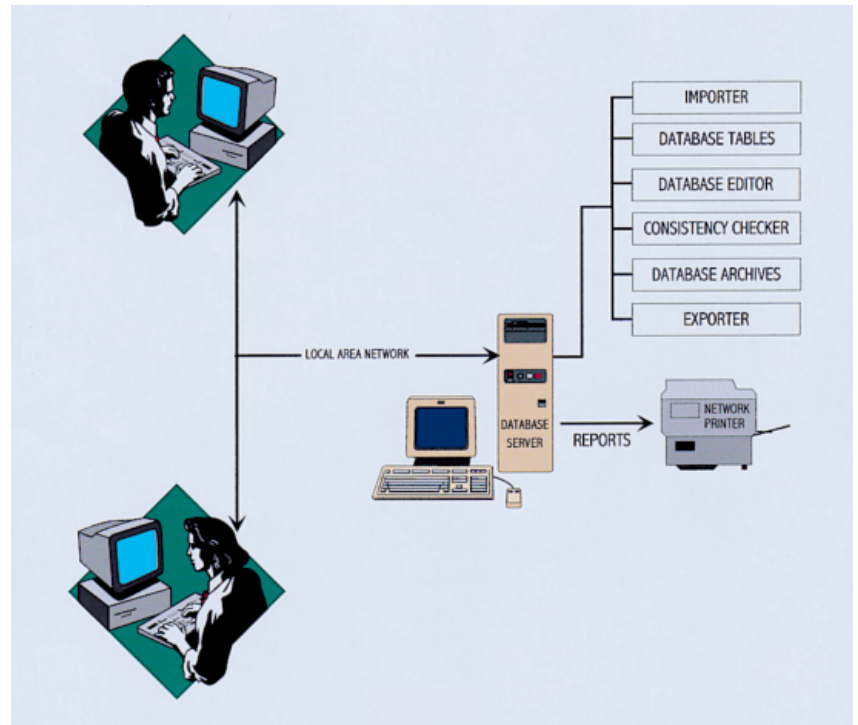


Figure 4. Organisation of the IODB applications

geared to the validation of the Mission Control System, the Flight Operations Plan as well as the Operational Database. The advantage of using the simulator for the validation of the database is that the various test scenarios (including failure scenarios) can be relatively easily recreated. However, the validation achieved using the simulator cannot be considered conclusive. In fact, the simulator's behaviour is also driven in many cases by configuration tables that are populated using the same input as for the Operational Database.

System Validation Tests

The System Validation Tests (SVTs) are conducted by interfacing with the real spacecraft whilst it is still on the ground. The actual ground-segment equipment is used and realistic operational scenarios are exercised. One of the SVT objectives is to confirm the correctness of the telemetry and telecommand database. In fact, SVTs ensure an end-to-end validation of all items in the Operational Database, which are actually used during the tests.

Historical data reprocessing

The execution of the above tests will lead to the identification of any problems with the Operational Database and their subsequent correction, but final confirmation of the database's correctness is still required. On the telemetry side, this can be achieved by simply reprocessing the data that have caused the problem. In SCOS-2000 this is in many cases simply achieved by opening the monitoring displays in retrieval mode, which implies

Table 1

Need/Problem

Solution for Integral

Manage large amount of data

Automatically import TM/TC spacecraft characteristics from the manufacturer's Satellite Database (SDB). Allow the users to access the powerful and intuitive query language supported by MS-Access

Maximise the work efficiency in the definition of the Operational Database (ODB) specific data

Support high-level 'replicate' functions at the level of data items (e.g. commands). Minimise the set of checks performed on-line during editing activities

The first version of the SDB is imported when it is not yet in a final state

Support a 'delta' SDB import option preserving all data added or modified at ODB level

Ensure overall consistency of the imported and edited data request

Support an off-line consistency checker performing all required checks upon user. Establish relationships across database tables with cascade delete and update mechanisms

Allow parallel maintenance of the Operational Database from the users' office environment

Support the remote access of a centralised set of tables directly from the users' PCs

Provide the users with adequate visibility of the database content

Design editor forms in collaboration with the users. Implement navigation techniques across related data items

Keep change and version control of the operational database

Automatically log all user edit actions. Automatically archive all exported databases

complete reprocessing of the raw telemetry packets based on the currently active database.

Database configuration control

After the execution of SVTs, it is crucial to ensure that new errors are not introduced into previously validated data. A rigid database change- and version-control procedure is typically adopted, whereby each change has to be traceable and, even more importantly, it must always be possible to roll back to an older database version if new problems are indeed identified. For this purpose, the Integral ODB editors support the automatic logging of all user edit operations, and all database versions exported to the Mission Control System are automatically archived under a dedicated directory.

Conclusions

This article has described the main problems encountered in administering the Operational Database for spacecraft operations and the solutions adopted for the Integral mission. The problems and the solutions adopted in each case for Integral are summarised in the accompanying table (Table 1).

