A RELATIONAL APPROACH FOR MANAGING LARGE FLIGHT TEST PARAMETER LISTS

Sérgio D. Penna, Engineer
EMBRAER Flight Test Engineering Division, Brazil

Antônio Magno L. Espeschit, Engineer
EMBRAER Flight Test Engineering Division, Brazil

ABSTRACT

The number of aircraft parameters used in flight-testing has constantly increased over the years and there is no sign that situation will change in the near future. On the contrary, in modern, software-driven, digital avionic systems, all sorts of parameters circulate through digital buses and can be transferred to on-board data acquisition systems more easily than those converted from traditional analog transducers, facilitating the request for more and more parameters to be acquired, processed, visualized, stored and retrieved at any given time.

The constant unbalance between what parameter quantity engineers believe to be "sufficient" for developing and troubleshooting systems in a new aircraft, which tends to grow with aircraft complexity, and the associated cost of instrumenting a test prototype accordingly, which tends to grow beyond budget limits, pushes for new creative ways of handling both tendencies without compromising the ease of performing an engineering analysis directly from flight test data.

This paper presents an alternative for handling large collections of flight test parameters through a relational approach, particularly in two important scenarios: the very basic creation and administration of the traditional "Flight Test Parameter List" and the transmission of selected data over a telemetry link for visualization in a Ground Station.

KEYWORDS

Flight Test Data Processing, Flight Test Parameter, Telemetry, Relational Database.

INTRODUCTION

In the last 30 years, the number of parameters used for flight-testing probably increased by 2 or 3 orders of magnitude, in particular due to the implementation of more complex digital systems in commercial and military aircraft design.

Digital parameters are easier and cheaper to acquire and process when compared to traditional analog parameters. Several hundred parameters flow in a single-channel digital data bus and can be monitored by connecting only a couple of wires. As aircraft systems complexity increases, more and more digital buses are added, quickly multiplying the total number of parameters available for engineers to analyze, triggering a desire for more and more information after each test flight and putting enormous pressure on instrumentation costs, telemetry bandwidth, not to mention the consequences in real-time processing and test data storage systems.

Statistics performed on parameter usage during a traditional flight-test campaign showed that:

- A small number of parameters is repeatedly used by almost everyone
- A large number of parameters is used most of the time by the same people
- A significant number of parameters is used once or never used

These findings could be seen as symptoms of poor resource management, where parameter instrumentation and parameter usage are totally unbalanced. In fact, part of the cost of maintaining a test aircraft could be reduced by instrumenting parameters only when they are needed. Unfortunately, this is not a trivial task:

- Adding or changing parameter instrumentation in a test aircraft often requires several hours of intensive work, during which the aircraft is not flying
- The precise time a particular parameter or group of parameters is needed changes dynamically when test campaigns are moved in time or migrate from one test prototype to another
- Engineers tend to react negatively - always for justifiable reasons - when asked to “trim” their parameter list
- Parameter instrumentation history must be tracked, even for parameters used just once in a single test flight.

Engineers tend to be more disciplined with respect to parameter usage when they are performing post-flight analysis. This activity requires a lot of their own time and often-interactive computational resources; therefore, engineers tend to optimize how much data is really needed. Better selection of test time-slices combined with a smarter selection of parameters that add significance to the analysis process in general lead to faster and more accurate results. Observing engineers at work, it is often to see parameters being grouped together following some association logic: produced or processed by the same aircraft system (engine, avionics, hydraulics, flight controls, etc), required for a particular analysis (flight qualities, take-off and landing performance, flutter, etc), or simply categorized as troubleshooting aid.

Making groups out of countable items is a common initiative when quantities involved exceed some subjective limit. The next sections will use this technique applied to the management of a large collection of test parameters within a flight-testing environment.

COMMON PROBLEMS BUILDING AND MANAGING PARAMETER LISTS

A "Parameter List" is necessary to guide the installation of on-board data acquisition systems, a task assigned to flight-test instrumentation personnel. Parameters are created by engineers as certification requirements get organized in flight-test campaigns (some of them) and as aircraft systems get designed (most of them). Certification regulations are documented and there is in general little questioning about what test parameters are needed to verify them, while for aircraft systems this is not the case.

In early stages of any aircraft development, information about aircraft systems is incomplete. If a "Parameter List" is built very early, it will contain low quality information about parameters belonging to these systems and will need to be revised constantly, adding more work to the "Parameter List" management task and little quality to the list itself. In this respect, it's not uncommon to have two different parameters created by two different engineering areas representing the same physical quantity and requiring the same type of instrumentation.

As the aircraft project matures, parameter information acquires quality and, after a certain point, it becomes sufficient for a good on-board data acquisition system design, one that matches the engineers need for test data in aircraft certification and development.

Manager's desire for keeping on-board data acquisition systems within budget and engineers desire for flight-test data are opposite forces: there is always a reason for instrumenting more and more parameters! Since resources are finite, some limit line must be drawn in certain point in time and the number of available parameters often becomes "frozen". Under this scenario, entire groups of parameters are occasionally replaced in on-board data acquisition systems as the flight-test campaign progresses, so other systems can be debugged without increasing some agreed total number of available parameters, at the cost of modifying the existing instrumentation.

In summary, following concerns are evident:

- Quality of parameter data must be assured as early as possible
- The balance between availability and usage must be constantly monitored

Failure to address these issues properly can assume catastrophic proportions when parameter quantities involved reach three or four digit numbers. One can expect to spend little effort managing parameter usage in a 500-parameter installation, but practices used in this scenario may not hold in a 10,000 or 20,000 parameter installation.

For large parameter quantities in the range of multiple thousands, new ways to deal with routine parameter acquisition and usage must be created.

AN APPROACH TO PARAMETER LIST MANAGEMENT

As parameters are created and added to a "Parameter List", parameter ownership can be tracked by recording a reference to the parameter user, always a member of an engineering group. Frequently, parameters created by one user are consumed by many others, adding a new dimension to parameter-user relationship. For instance, parameters listed in Table 1 are so basic that they are expected to be consumed by almost every engineer doing analysis on flight-test data. As aircraft development progresses, parameters are eventually no longer required (a certification regulation is already demonstrated or an aircraft system is operational) and could be removed from the current instrumentation installation, saving time in pre-flight activities, processing time and storage space in ground station systems. If "owners" and "consumers" are properly tracked, any particular parameter could be safely removed when it has no "consumers", provided that an agreement from the "owner" is obtained.

1	Airspeed
2	Barometric Altitude
3	Static Air Temperature
4	Total Air Temperature
5	Pitch Angle
6	Roll Angle
7	Yaw Angle
8	Pitch Rate
9	Roll Rate
10	Yaw Rate
11	Mach Number
12	Air-Ground Switch
13	Rudder Position
15	Aileron Position
16	Elevator Position
17	X Acceleration
18	Y Acceleration
19	Z Acceleration
20	Vertical Speed

Table 1 – Very Basic Flight-Test Parameters

If parameters could be divided in classes indicating a different usage or processing priority, managing alternatives in a limited resource scenario could be facilitated, because it would be previously known what groups of parameters could be manipulated. In general, it is possible to segregate parameters in at least three usage classes: those declared "Safety-Of-Flight", those required for certification and those dedicated to system troubleshooting.

Figure 1 depicts a data model that may help addressing the issues above.

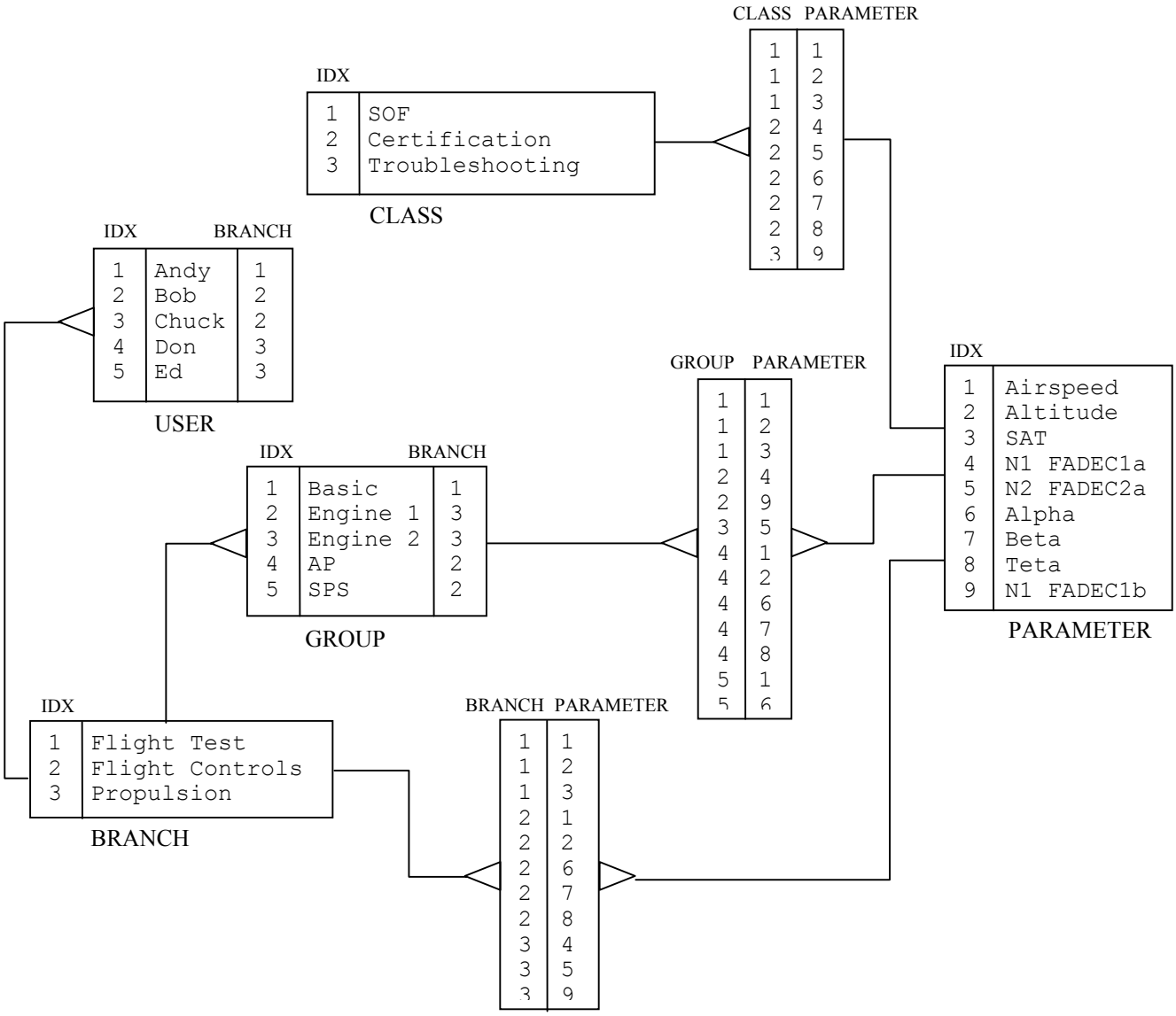

Figure 1 – Proposed Data Model for Parameter List Management

It has five tables, USER, BRANCH, PARAMETER, GROUP and CLASS, and three auxiliary relationship tables ("r-table"), BRANCH x PARAMETER, GROUP x PARAMETER and CLASS x PARAMETER.

Entries in BRANCH table are created before entries in USER table, so each new user is added belonging to an existing branch. Each new parameter creates an entry in PARAMETER table belonging to a particular branch and creates a new entry in BRANCH x PARAMETER r-table. Parameter classes are defined creating new entries in CLASS table. Parameters are associated to existing classes by creating new entries in CLASS x PARAMETER table. Groups of parameters are created by adding new entries in GROUP table. Parameters are added to groups by adding new entries in GROUP x PARAMETER r-table.

From Figure 1, users "Bob" and "Chuck" belong to branch "Flight Controls", branch "Flight Test" shares interest in parameter "Airspeed" with branches "Flight Controls" and "Propulsion", parameter group "Basic" belonging to branch "Flight Test" contains parameters "Airspeed", "Altitude" and "SAT", parameters "N1 FADEC1a" and "N1 FADEC2a" are classified as "Certification", while "N1 FADEC1b" is classified as "Troubleshooting".

The concept of parameter "owner" and "consumer" can be easily accommodated when the "owner" is considered simply the first "consumer", that is, the first branch to add an entry in BRANCH x PARAMETER r-table is the "owner" and any other branches adding new entries for the same parameter are "consumers".

This data model allows for multiple parameter associations, that is, the same set of parameters can be part of different parameter groups, as long as the association makes sense to their creators. From Figure 1, parameter "Airspeed" belongs to groups "Basic", "AP" and "SPS".

Associating parameters to classes can give a hint to parameter usage and help defining priorities for telemetry data transmission, recording and processing.

Using this model, one can formulate a few important queries:
1. Which groups do parameters belong to?
2. Which parameters do not belong to any group?
3. Which branches do parameters belong to?
4. Which classes do parameters belong to?

Query 1 is a measure of general interest for a particular parameter. If the answer to query 2 is a non-empty parameter set, these are serious candidates for removal from on-board instrumentation. Query 3 helps identifying which engineering areas need to be involved for negotiating instrumentation configuration changes and query 4 what parameters could be used for balancing resource limitations.

USING PARAMETER GROUPS IN DATA ACQUISITION

In Ethernet-based on-board data acquisition systems [1] [2] [3], parameter samples are collected in data structures called "datagrams", packets of data exchanged by network nodes. In such systems, datagrams are built in real-time, time-tagged and sent to a fast switched network for processing in one or more digital computers dedicated to data visualization in a monitor screen, data recording in a mass-storage device, PCM formatting for telemetry purposes, etc.

Since datagrams in this context are also an organized collection of test parameters, does the data model presented in the previous section help their generation and processing?

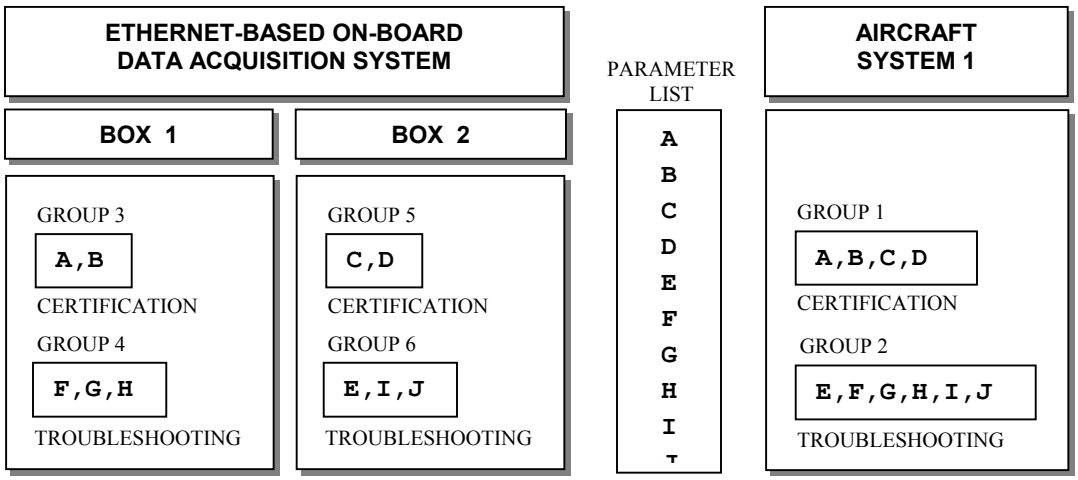

Figure 2 – Parameter Groups in Data Acquisition

Figure 2 suggests a basic parameter set "A-J" that, under the perspective of "Aircraft System 1", associates parameters "A-D" of class "Certification" in "Group 1", and parameters "E-J" of class "Troubleshooting" in "Group 2". The same basic parameter set, under the perspective of "On-Board Data Acquisition System" associates parameters "A-B" in "Group 3" and "F-H" in "Group 4" for "Box 1", "C-D" in "Group 5" and "E;I-J" in "Group 6" for "Box 2". Note that in this scenario parameter classes remain consistent across parameter groups.

"Datagrams" could be created in direct association with "Parameter Groups" under the perspective of an Ethernet-based on-board data acquisition system, preserving certain granularity resulting from the fact that parameters are acquired in small, box-shaped, pieces of hardware.

Groups containing parameters of a certain class could form datagrams that could be routed to different processing nodes in the network. Figure 3 suggests that all datagrams transmitted by the on-board data acquisition system are routed to a test data-recording node, but only datagrams formed after groups containing parameters of class "SOF" and "Certification" are routed to a telemetry formatting node. In the same scenario, if there is enough control in network routing algorithms,

some datagrams formed after groups containing parameters of class “Troubleshooting” could be occasionally re-routed to the telemetry-formatting node in an emergency case.

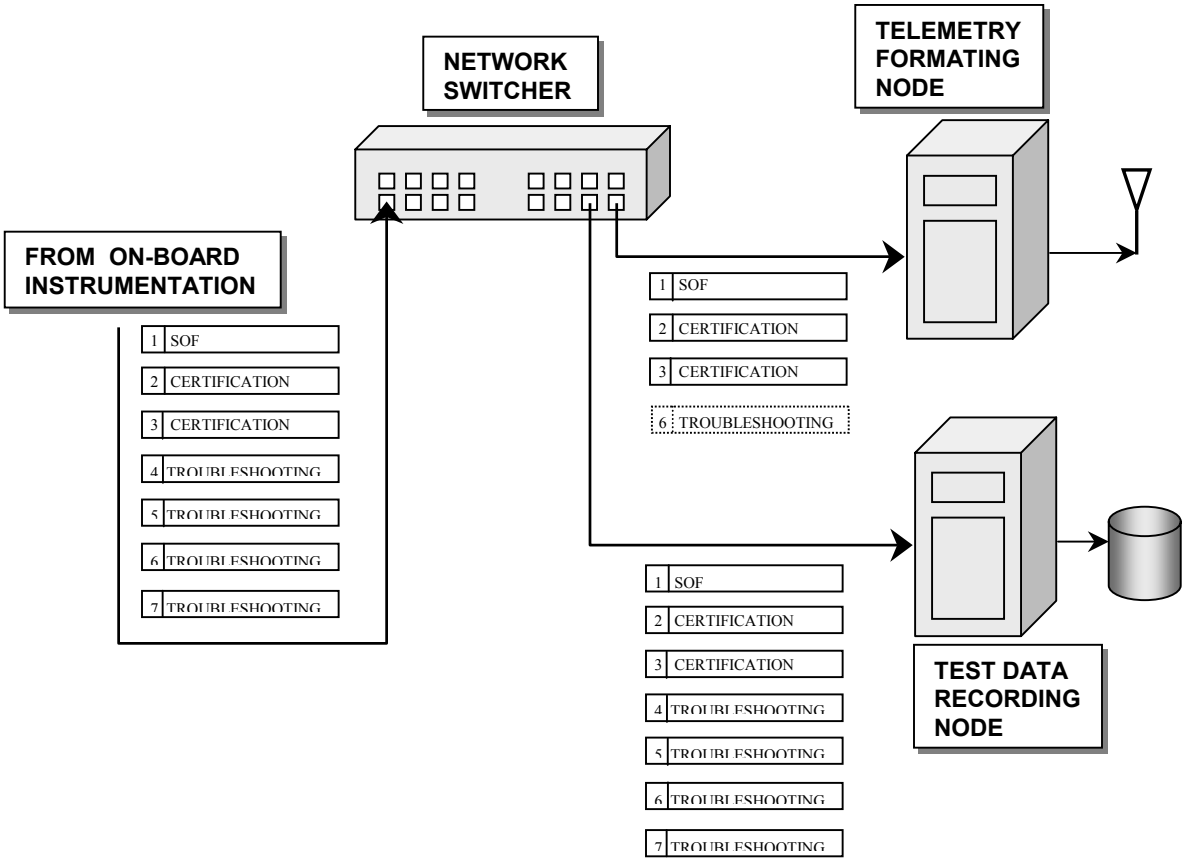

Figure 3 – Datagram Processing by Parameter Class

CONCLUSION

The previous sections tried to demonstrate how relational associations could facilitate a few aspects of parameter information handling in a flight-testing environment.

Benefits of this construction in managing a very large parameter installation were addressed and the direct association of parameter groups with datagram construction in Ethernet-based on-board data acquisition systems and datagram processing was established.

Although a complete discussion on the subject exceeds the context of the present paper, one extension to the presented data model is worth mentioning.

"Parameter Group" is a single level of hierarchy above "Parameter". In a multiple thousand-parameter installation, this may be insufficient for indexing the whole collection of test parameters. As in other indexed file structures, more than one index level may exist, so "Groups of Parameter Groups" could also be formed. In such scenario, parameters could be addressed by text structures similar to file paths, for instance: "Engine1\FADEC1\N1" and "Engine1\FADEC2\N1", or "Engine2\FADEC3\N1" and "Engine2\FADEC4\N1". This naming convention has more information content than the usual short mnemonic text, but may be limiting in some contexts, in particular when designing man-machine interfaces (they naturally become more complex).

Dealing with ever larger parameter installations is a constant challenge to flight-test people, and it will remain so until some new technology comes in and modifies its fundamentals.

REFERENCES

[1] D. Corry – *Building Bridges: Linking CAIS to Ethernet and other protocols*, ETTC Proceedings, 2003

[2] N. Revaux, F. Abadie – *A380 Flight-test architecture: Switched IENA*, ETTC Proceedings, 2003

[3] J. Dai, E. Grozalis, T. DeSelms – *An Ethernet Based Airborne Data Acquisition System,* ITC Proceedings, 2003