Managing Telemetry Information in the New Era of Test and Evaluation

Lee S. Gardner
6510 TW/TSRCP
(805) 277-2628
gardner@nosve.elan.af.mil

Catharine M. Hoaglund
6510 TW/TSRCP
(805) 277-2351
choaglun@tsr.elan.af.mil

Darcy S. Painter
6510 TS/TS
(805) 275-8796

Air Force Flight Test Center
Edwards AFB, California

ABSTRACT

Terms like efficiency, quick response, and interoperability are becoming the bywords of the test and evaluation (T&E) community as the Defense Department tightens its corporate belt [1]. These changes mark the end of an era of manual processes and duplication of effort and the beginning of an era of cooperation, standards, and Total Quality Management (TQM). Managing the huge volume of telemetry information required to support flight test at the Air Force Flight Test Center (AFFTC) has required new paradigms and system development strategies. These new ideas have resulted in the Aircraft Information Management System (AIMS), a system designed to meet the challenges of a new era in T&E.

This paper discusses the AIMS design and function as background for the deeper issue of effective, efficient management of telemetry setup information. The information history model used in AIMS is presented and discussed. In the process of developing standards for the AIMS a methodology was discovered and successfully implemented for resolving information management issues in the framework of system development.

KEY WORDS

Information Management Systems, Interoperability, Total Quality Management, Process Automation.

WHAT WAS THE PROBLEM TO BE SOLVED?

The AIMS was created for two primary reasons. The first goal was to help mission support personnel at the AFFTC handle the growing mountain of telemetry data required by the flight test programs of a new generation of aircraft more effectively and efficiently. The second goal was to shorten the time required to set up various systems to support a flight test. In the beginning, the problems to be solved by AIMS automation seemed to be clear: lengthy response time, human input errors and duplication of effort. However, as we complete the second year of experience with AIMS, we realize that the task of working smarter, faster, and cheaper requires a more complete analysis of the processes involved in the way we do business.

The implementation of AIMS brought immediate help for the simpler issues, and enabled us to focus on the deeper concerns. We found that our old methods of dealing with calibration updates suffered from multiple, nonstandard, short term solutions, often insupportable in the long run. Each flight test project had unique processes and paper trails, creating duplication of effort. Often there was a loss of control of the vast volumes of data moving between the various organizations, and the lines of responsibility were frequently fuzzy or undefined.

WHAT DOES THIS PAPER COVER?

Previous papers have discussed the way in which AIMS has reduced cost and turnaround time through automation [2, 3]. In this paper we explain how the emphasis on standards in the implementation of AIMS has improved the management of the entire telemetry setup process. The payoffs have included improved efficiency, reduced turnaround time, and increased interoperability of people and processes as well as of software and hardware. We will discuss how the lessons learned from two pilot projects supported by AIMS are guiding our on-going development efforts. A glossary of current terminology appears at the end of this paper.

WHAT IS AIMS?

AIMS may be simply described as a program that manages a three dimensional data base of calibration and telemetry information via a standard, readable file format [4]. However, AIMS is more than just a piece of software, it is a way of doing business. The AIMS software design is based on fundamental principles of software engineering. These fundamentals have propagated into related working methods, including automatic interchange of information by predefined standard formats, historical traceability of changes, and standardized procedures to ensure consistency between systems.

The information presently stored in an AIMSfile, (as the data base itself is named) consists of a number of parameter names and their associated attributes [4]. Parameter names generally represent measurement data being telemetered from the aircraft. The attributes are the key values needed to describe each measurand in enough detail to let ground systems successfully translate the raw data stream into meaningful information. The process of breaking the telemetry stream into raw data for each measurand is called decommutation. The follow-on process of converting the raw data into the engineering units initially measured is called calibration. The AIMSfile serves as a central source of decommutation and calibration information for both real-time and postflight systems, and retains a full historical record of all changes.

One of the key features of the AIMS design is the keystream, a standard, readable format for input and output of data and for maintaining a historical record. The keystream is characterized by its flexible nature, in which each parameter name is described by a set of keywords and associated values (Figure 1). The things that need to be described about a parameter will vary with the source of the data and with the kind of systems that are expected to process the data later, so the number and kind of keyword/value pairs per parameter are completely flexible. Any data that can be described can have that description stored in an AIMSfile. The same AIMSfile that holds detailed decommutation and calibration data can also hold global information about the project, such as the tail number of the aircraft and the last date and time information was added to the file itself [4].

A major requirement the AIMS design had to satisfy was to provide both an audit trail for changes and a way to “roll back the clock” to provide accurate recreation of earlier states of the aircraft. Reprocessing of data from previous flight tests requires recalling the correct decommutation and calibration information from a date in the

```
PARAMETER = 'XXXXXXXX'
  TITLE = 'AOA. CORRECTED'
  NWORDS = 2
  RAW_FORMAT = '2S_COMP','2S_COMP'
  WORD =   105,     106
  FRAME =   4,      4
  DEPTH =   4,      4
  SUPER =    0,      0
  LSB =      0,   8
  MSB =     11,  11
  UNITS = 'DEG
  NCOEF =   2
  COEFS = 0.00000000E+00, .12345000E-02
  RMAX =  .25600000E+03
  RMIN = -.25600000E+03
  STATUS = 'ACTIVE'
ENDPAR
PARAMETER = 'YYYYYYYY'
  TITLE = 'AOA INDICATED'
  NWORDS = 2
  RAW_FORMAT = 'UNSIGNED_BINARY','UNSIGNED_BINARY'
  WORD =   10,  11
  FRAME =    8,   8
  DEPTH =   16,  16
  SUPER =    0,   0
  LSB =    0,   8
  MSB =   11,  11
  UNITS = 'DEG
  NCOEF =   2
  COEFS = 0.00000000E+00, .100000000E+01
  RMAX =  .32767000E+05
  RMIN =  .00000000E+00
  STATUS = 'INACTIVE'
ENDPAR
PARAMETER = 'YYYYYYYY'
  TRACK =  2
  TITLE = '16 BIT SWITCH'
  NWORDS = 2
  RAW_FORMAT = 'UNSIGNED_BINARY','UNSIGNED_BINARY'
  WORD =   10,  11
  FRAME =    8,   8
  DEPTH =   16,  16
  SUPER =    0,   0
  LSB =    0,   8
  MSB =   11,  11
  UNITS = 'DEG
  NX =   2
  TABLE_COUNTS = 0.00000000E+00, .100000000E+01
  TABLE_PQ = 0.00000000E+00, .100000000E+01
  RMAX =  .32767000E+05
  RMIN =  .00000000E+00
  STATUS = 'ACTIVE'
ENDPAR
```

FIGURE 1: EXAMPLE KEYWORD STREAM

past. Moreover, this had to be accomplished without overwhelming computer system resources. To do this, the designers worked from an information history management model.

Every piece of data stored is associated with an “effective test number” keyed to a particular state of the aircraft. This test number is also related to the flight number in the test program. As information for a specific test number comes in from instrumentation engineers and other sources, the AIMS program compares the incoming data to the data for the previous test and stores only the changes under the new number. Each change is documented in a report that is automatically generated by AIMS during data updates. This report provides an audit trail for configuration control of the AIMSfile. By maintaining the information history as an initial state of the decommutation and calibration information and all changes indexed by test numbers, AIMS allows the user to extract any previous state of the information desired [4].

Once the function for retaining historical test information was in place, it still took considerable experimentation to evolve a standard process to handle decommutation and calibration changes. The information needed to update the AIMSfile is received by the Ridley Mission Control Center via electronic media - on a tape, diskette or as a file sent over the base network. The format of the data has been negotiated with the provider, and the agreement formalized in an Interface Control Document. The data may arrive all at once, or may come in several installments from several sources. The data are converted to a keystream (if not already in keystream format) and a trial update is performed, generating a change report that details both the old and the new pieces of information affected.

This change report is in the readable keystream format and is immediately made available on-line for verification by the people who provided the data. When verification is received, the true update is completed, and the new information is immediately distributed to the various systems that will need to be set up for the next flight test mission. This verification performed before the AIMSfile is changed guarantees that all the downstream systems get the same correct information.

At the same time the true update is made, other reports are generated and put on-line, including a status and tracking bulletin, the full current keystream, and the change report. This gives all the multiple organizations that must coordinate to make a flight test mission successful a common and current source of information. This information is on-line, accessible to remote locations by network or modem access to the central scientific computer.

Work is currently under way to extend the AIMS to new flight test projects and to new domains of knowledge. The existing process and keystream are flexible enough to work effectively with all currently forecasted requirements. The Software Engineering Section of the 6521st Range Squadron is coordinating the work.

THE STRUGGLE FOR STANDARDS

The handling of aircraft instrumentation information for a modern flight test program is a complex coordination task. The aircraft instrumentation can be a unique, airframe contractor system, an AFFTC provided system, or a hybrid, which uses components from contractor and AFFTC systems. Each Combined Test Force (CTF, the basic testing unit at the AFFTC) has data managers to coordinate the delivery of instrumentation changes and data product requests, and to collect and distribute data products as they are produced. Flight test engineers (FTEs), the end users of this data collection and processing machine, specify what is to be telemetered in real-time, what is to be recorded for post-flight processing, and how measurements are combined into plots, listings, and data files. Instrumentation engineers are the interface between the FTEs and the instrumentation system, programming telemetry streams to meet the FTE’s requests for measurements. Separate systems and staffs support real-time and post-flight data processing activities. Just tracing where data goes and who does what in this process is difficult. The process by which information and data flows through the data production machine can be unique to each project adding to the confusion created by many players and unique test article instrumentation systems.

This unrestricted approach to collecting and processing flight test data has worked when unlimited resources were available to customize the process to each program’s requirements. But, today in the world of finite, even scarce resources, standard processes and procedures are required. One of the first problems tackled by the AIMS team was the establishment of a standard way of doing business by creating standard software and supporting policy concurrently.

The AIMS keyword stream (keystream) illustrates this concept. The keystream evolved by observing the type of information communicated between the players in the test setup process. it began as a structure with general information about a telemetry stream at the highest level, the different measurands at the second level, and the attributes of those measurands at the lowest level. This three-level hierarchy was well suited for Pulse Code Modulation (PCM) telemetry streams, no matter whose instrumentation system produced them. The keystream also provided the capability to add or subtract parameter attributes as required, making it flexible enough to handle the requirements of many projects.

In addition, the keystream provided an easy way to incorporate the information management model into the AIMSfile. As a keystream is processed, each parameter’s attributes are compared to the previous state of those attributes. Changes are candidates for new entries in the AIMSfile once they are validated.

These two features, an appropriate hierarchical structure with sufficient flexibility and a way to retain historical information using minimal storage, made the keystream structure the start of the AFFTC standard for telemetry decommutation and calibration information processing. For the keystream standard to succeed, it had to be supported by a standard process for updating instrumentation system changes. These processes had to incorporate concepts of maintainability and interoperability between projects. Another goal was to avoid the need for gurus and customized, one-of-a-kind processes. The AIMS design philosophy dictated that the standard be based on automated processing concepts that minimized human intervention. The design had to utilize current networking capabilities. Considering these goals, the AIMS team with CTF data managers and test support technical staffs began inventing and using a standard instrumentation system change process.

With a standard structure, the next level of standardization for the AIMS keystream was at the measurand attribute level. Attributes are described in keyword = value statements, where the keywords describe the meaning of the values that follow. The keywords convey the semantics while the values communicate the data to be used. Sets of keywords can be used to distinguish different types of parameters. For example, the keywords NCOEF and COEFS describe a polynomial calibration (of order NCOEF - 1) while NX, TABLE_COUNTS, and TABLE_PQ describe a table lookup calibration. The presence (or absence)of certain keywords indicates differences in calibration processes. In effect, the choice of keywords incorporates the instrumentation engineer's expert knowledge of how the measurand is processed into engineering units.

A current effort is in progress to standardize the keywords (and their definitions) to improve interoperability of software and personnel across projects. Although several emerging inter-service and industry standards have been proposed, none fully satisfied our needs. The proposed Range Commander's Council standard [5] uses cryptic keywords which violate our human readability requirement for a keystream. The current Loral standard [6] comes closest to the AFFTC keystream, differing in the volume of information described (Loral's contains more information we consider static for the life of the project).

Once keywords are standardized, projects should be able to reuse each other's AIMS software. In spite of differences in instrumentation systems and personnel, the project's keystream should be the universal language that allows the use of common software. Of even greater benefit is that a person (instrumentation engineer, FTE, or test support staff) who understands the standard keywords adapts easily to other projects. Further, others involved in the project's data collection and processing can read a keystream and understand how the data was processed too.

The future of this keystream with standard keywords looks bright, provided that the oversight committee gives appropriate weight to the principles behind the standard before extending it to include new keywords. Adaptation of keystreams with standard keywords will maximize reuse of AIMS software and minimize cross project maintenance effort. It will form the foundation for the addition of intelligent error checking of data stored in an AIMSfile. Bridges to other systems, like the evolving Test Instrumentation Management System (TIMS), will be simpler to build due to this standard.

DISTRIBUTED ORGANIZATIONAL RESPONSIBILITIES

Fuzzy and undefined lines of responsibility have been clarified by AIMS development and operations. For example, the manual instrumentation calibration change process blurred the responsibilities between instrumentation engineers and test support system staffers. Although the instrumentation engineers made (or were responsible for making) the calibration change, their responsibility ended with completion of a calibration change sheet. They were provided a copy of whatever system specific file format the staffers produced from the calibration sheet, but these files varied greatly from system to system so that validation of changes by instrumentation experts required inside knowledge of support system specific file formats. Likewise, staffers had little or no knowledge about which calibration changes made sense; they lacked the inside knowledge about instrumentation systems. This lack of ownership of the vital function of change validation caused many reprocessing runs due to improper setup of support systems. No one had responsibility for the whole process from instrumentation system change through updating the test support systems. A whole breed of analyst was created to determine whether an instrumentation change resulting in incorrect test data was due to an error in recording the change by the instrumentation engineers, or due to an error in translating the change into support system setup files by the support staffers.

With a common language, the AIMS keystream, the lines of responsibility were clarified. Since the keystream embodied language common to instrumentation engineers and support system staffers, it became clear that the engineers were responsible for the validity of the changes, i.e. the values changed in specific keywords representing the recalibration of the instrumentation system. The staffer’s laborious task of translating these changes into support system specific files was incorporated into the automation that AIMS brought to the process. With the removal of human error, ownership of the calibration change process was clearly established with the instrumentation engineers. They learned (a process taking a few minutes) how to validate their changes by reading the changes AIMS reported during a trial update and accepted that AIMS would faithfully propagate their changes among the various support systems.

With the ownership of the calibration change process firmly established, support staffers became more flexible. Instead of becoming experts in one support system specific file format, they could become generalists, supporting all systems through AIMS. With the translating task automated by AIMS, they were free to concentrate on other support tasks. This automated translation increased their productivity two ways. Automation removed errors from the translation process thereby reducing data reruns. Further, it eliminated the time-consuming task of translating changes into system specific formats.

The development and operation of the AIMS had a positive impact on interorganizational communications. By clarifying lines of responsibility, organizations involved with the calibration change process became less oriented to fixing blame for mistakes and more open and communicative. With clear lines of responsibility came clearer communication because organizations had less to hide from each other. Also, since process ownership was well defined, process goals emerged from the previous confusion. These goals often required interorganizational cooperation where one process ended and another began. Communication became critical at these process interfaces and organizations quickly developed skills to improve their interactions.

This clarification of roles and responsibility is paving the way toward a more open, interoperable, distributed environment. Distributed computing is the paradigm of the future. Current AIMS operations are distributed in the sense that decommutation and calibration changes are entered into one system, perhaps a mile or several miles away, and stored and distributed to test support systems at another location. The interface between these two systems is currently a file transfer via nine track magnetic tape, but when local network links are complete, network file transfers will be possible. The use of protocols like network file system (NFS), where two computer systems share the same disk storage space, will not be far behind. Plans for future generations of AIMS include exploring the potential of distributed, object oriented data base technology.

Clarifying responsibilities and establishing ownership of complex processes was difficult. In many cases process definitions and boundaries had never been explored. Things were done a specific way because that was how they had always been done. In some instances it was even hard to tell if all players in a particularly poorly defined process had been consulted about their role or about changing their role. Several operational working groups were established, usually one per project converted to the use of AIMS. These groups worked out how the new way of doing business would affect the various players in the current process. Surprisingly, after only two projects, a standard, automated process using AIMS is emerging. It appears that the chaos inherent in the previous way business was conducted was a consequence of that process and its convoluted, project specific nature. The underlying process of updating support systems with instrumentation system changes appears to be generic across projects. This only became visible when the development and operation of AIMS forced the people involved to take a new look at what they do, how they do it, and why.

CONCLUSIONS

The emphasis on standards in the implementation and operation of AIMS has improved the management of the entire instrumentation change process. By developing a standard language, the AIMS keystream, that both instrumentation engineers and test support system staffers could understand and use, clear lines of process ownership emerged and tedious, error prone, manual processes were replaced with automatic, validated, efficient ones. Beginning a process automation project by developing a standard, common, domain specific language is an unconventional approach, however, the potential payoffs for improved management of the process under study are enormous. The AIMS team expected to speed up calibration change updates to test support systems, which they achieved, but far beyond those initial expected benefits were clarification of process ownership, increased interorganizational communication, and a renewed attitude of cooperation.

The AIMS has achieved its original goal of interoperable software between AFFTC projects, but more importantly, AIMS working groups have facilitated interoperability of personnel between projects and support systems. By removing a tedious, error prone subprocess, personnel were released from the burden of project and system specific operations and had the opportunity to see the commonality between projects. By exposing the common ground, the AIMS has led to more efficient workers, saving time and money in the process.

In the new era of test and evaluation, where interoperability, efficiency, and effectiveness are requirements, the AIMS project represents a new paradigm for success. By careful application of standards at critical interfaces and by empowering workers to improve processes and interorganizational cooperation and communication, AIMS succeeded in redefining and greatly improving a mission critical process at the AFFTC. The methodologies used by the AIMS team are portable to other projects. The lessons learned by the AIMS team represent one path to achieving goals of the new era.

REFERENCES

[1] Adolph, Charles E., Keynote Address to the International Telemetering Conference ITC/USA/91, Las Vegas, Nevada, November 4-7, 1991.

[2] Gardner, Lee S., "Introduction to the Aircraft Information Management System," in Minutes of the Data Reduction and Computer Group, Range Commander's Council, Fort Huachuca, Arizona, February 25 - March 1, 1991.

[3] Gardner, Lee S., "Collecting, Managing, and Distributing Instrumentation System Calibration Information in a High Volume, Data Production Environment," in Proceedings of the Sixteenth Transducer Workshop, Telemetry Group, Range Commander's Council, San Antonio, Texas, June 18 - 20, 1991.

[4] Flanders, Joey; Gardner, Lee S.; Hoaglund, Catharine M.; and Jones, Charles H., <u>Aircraft Information Management System User's Guide, Version 1</u>, Computer Sciences Branch, 6510 Test Wing, AFFTC, Edwards AFB, California, April 1991.

[5] "Draft of Proposed Standard of Telemetry Attributes," Data Multiplex Committee of the Telemetry Group, and Data Reduction and Computer Group, Range Commander's Council, June 1991.

[6] “Software Interface Design Document: Data Base Interchange,” Draft copy, Loral Data Systems, Inc., EMR Telemetry Systems, Sarasota, Florida, July 1991.

GLOSSARY

AFFTC - Air Force Flight Test Center at Edwards Air Force Base, California.

AIMS - Aircraft Information Management System. AIMS maintains calibration, decommutation and other data retrievable by data format, parameter names and effective test. Input and output are done in a standard file called a keystream.

AIMSfile - A collection of information stored in a binary file in AIMS format.

ASCII - American Standard Code for Information Interchange. Data represented in this format is said to be “readable” and can be manipulated with a text editor.

Calibration - The translation process in which a raw measurement from the aircraft is turned into a meaningful data value.

CTF - Combined Test Force.

Decommutate - The process by which the incoming telemetry data are broken into tag/data pairs. The initial decoding of the signal.

Format - A particular arrangement of measurands in a PCM data stream, commonly referred to as the PCM matrix. A project may use multiple formats, which may differ in either the arrangement of the matrix or in the parameters within the matrix, or both.

FTE - Flight Test Engineer.

Keystream - Readable (ASCII) format output used by AIMS and by programs that interface with AIMS. Contains data organized by parameters, with keyword/data value pairs describing the attributes of each parameter.

Keyword - A name that describes a particular attribute of a parameter. In a keystream, the portion of a keyword/data pair to the left of the equals sign.

Measurand - A numeric or discrete parameter indicating a value being measured by an instrument in an aircraft. An entry in a PCM matrix.

Parameter - In telemetry, one item of measured data being sent in the telemetry stream, a measurand. In AIMS, a collection of information about a particular item, in which the item is usually a parameter in the telemetry sense. In general, a collection of information about a measurand or about a data value derived from measurands at the same logical level.

PCM - Pulse Code Modulation telemetry. This may represent one of the major schemes by which telemetry data are organized, or may refer to an incoming stream of data in that format.

PCM Matrix - A two-dimensional matrix used to visualize the PCM stream or format. The rows constitute PCM subframes; the columns represent word locations within each subframe.