IMPLEMENTATION OF A LOW COST COMMERCIAL-OFF-THE-SHELF COMMANDING SYSTEM

Richard J. Grich Jr.
Chris R. Bourassa
Storm Integration, Inc.

ABSTRACT

Traditional satellite and launch control systems have consisted of custom solutions requiring significant development and maintenance costs. These systems have typically been designed to support specific program requirements and are expensive to modify and augment after delivery. Over the past five years, technical advances have resulted in Commercial-Off-The-Shelf (COTS) products which greatly reduce the complete life cycle costs associated with satellite and launch control system procurements. These advances, however, have been restricted to specific functional areas of the satellite and launch control system - most notably, telemetry processing and simulation. Until recently, technological advances in the development of COTS products which support functional areas like commanding and mission planning have lagged behind. This paper describes the development and application of a COTS product which provides a highly advanced commanding capability that is tightly integrated with the processing of telemetry data. This closed loop telemetry and commanding system forms the basis of a satellite or launch control system at a fraction of the cost normally associated with systems of this kind.

KEY WORDS

Commanding, COTS, Real-time

INTRODUCTION

During the past few years, significant advances have taken place in the commercial market to satisfy the need for low cost, yet highly reliable industry standard products to perform telemetry processing for spacecraft and related applications. Many companies currently offer telemetry processing hardware and software integrated into COTS packages. These packages provide a variety of cost effective alternatives to customers with spacecraft telemetry processing requirements. During this same period, however, little effort has been invested to develop a commercially available software package for spacecraft and launch vehicle commanding. Many technical issues have discouraged potential developers from building a "generic" COTS commanding package. These issues center around the ability to offer a robust and reliable system to a sufficiently large customer base while maintaining low procurement and integration costs.

This paper describes the key design and implementation details of a COTS product which addresses these issues - Storm Integration's Intelligent Mission Toolkit (IMT)™. In addition, this paper discusses the advantages of developing the initial version of the IMT to be integrated with one of these COTS telemetry processing systems. Loral Instrumentation's System 500. By developing a system which makes maximum use of existing COTS hardware and software, for the first time spacecraft and launch vehicle manufacturers and operators will have low cost options for satisfying their full-loop command and telemetry processing requirements.

PRODUCT REQUIREMENTS

The first step in the development of this system was to clearly define the design drivers and requirements associated with a complex command and control system. The three most important elements of the system were determined to be vehicle adaptability, system tailorability, the definition of clearly defined interfaces.

The first major consideration relative to the design of a COTS commanding system is that it must be readily usable by a variety of different types of spacecraft programs without the addition of new software components. In order to satisfy this requirement, the commanding system would have to be database driven.

Closely tied to the need for a database driven system is the ability to easily configure or tailor the system to meet the needs of a specific spacecraft program. Where appropriate, the system should be configurable by the operators who may not be computer experts. This means the user interface, the database configuration and the command plan execution functions should be easily modifiable by the users of the system. A commanding COTS package which is consistent with industry standards in the use of relational databases and user interfaces will provide the required tailorability. System users should be able to configure and tailor the system, however, this should not be the only method of ensuring that all spacecraft program objectives are met. A core system which is database driven should be capable of being augmented by program specific software and hardware, if necessary. In short, the system should be extendible.

Finally, a clearly defined interface to other spacecraft functions must be provided. The primary functions to which commanding will need to interface are telemetry processing and mission planning. An interface to external users may also be required. This external interface needs to have an element which permits commands from an external source to be inserted into the commanding database for eventual processing and transmission.

PRODUCT DESIGN/DEVELOPMENT

Important elements of the IMT product design and development effort included: commercial product selection, use of an iterative design approach, adherence to strict software development standards and methodologies, and a belief that product documentation should begin during the design phase.

The first step in the design process was a search for COTS products which could satisfy one or more functional requirement areas. The goal was to maximize the use of commercial products, reducing the overall cost of the product and enhancing the flexibility and functionality of the system. To ensure the best possible combination of commercial products, Storm Integration performed an extensive study of products available in the area of complex command and control. A detailed set of criteria were applied to this evaluation, however, the characteristics considered most important were: adherence to industry standards, product flexibility and a willingness by the vendor to evolve their product to meet the complex requirements of command and control systems. Products selected during this evaluation included Loral Instrumentation's 500 series front end telemetry processor, Gensym's G2 Real-time Expert System, and SYBASE's family of relational database products.

Use of an iterative design process was critical to overall product quality and customer satisfaction. Storm Integration felt it was important to demonstrate the IMT to actual system users and within an aggressive and complex system environment. This was to ensure that the product would satisfy overall system requirements and to elicit detailed product assessments from people who actually use and operate complex systems. Storm presented the first IMT system prototype at the International Telemetering Conference in San Diego, California in October of 1992. In addition, the IMT has been used to successfully command and control United States Air Force satellites using the US. Air Force's Satellite Control Network (AFSCN). Comments received during these and other demonstrations were incorporated into the IMT product design increasing product usefulness and quality.

To ensure a high degree of quality, the IMT software development effort followed a strict set of documented corporate software engineering standards and guidelines.

These corporate standards include detailed processes and methodologies in the areas of software engineering, configuration management, project management, software testing and product quality management.

PRODUCT DESCRIPTION

Storm's Intelligent Mission Toolkit (IMT) is an open, modular Commercial Off The Shelf (COTS) package which provides the ability to support multiple satellite processing applications with a complete closed-loop commanding and telemetry processing capability. Processes are distributed across an Ethernet or FDDI Local Area Network (LAN) with functionality resident on UNIX-based workstations and VME-based industry standard front-end equipment. The user interface is provided via multiple graphical Human Computer Interfaces (HCI) COTS software packages. The system is configured and driven by a relational database, making it a generic, high fidelity commanding capability.

The IMT was designed to meet the complex requirements of a wide variety of satellite and launch control programs, including the US Air Force Satellite Control Network, commercial cellular communications programs and launch modernization initiatives. Storm's extensive experience with commanding systems used by multiple US Air Force and commercial programs was used to form the necessary core functionality of the IMT. This experience as used to improve upon the existing, limited commanding architectures. The IMT has been designed and developed as a "Commanding Shell" with a supporting toolkit which allows maximum flexibility in system configuration through the use of sophisticated, integrated database manipulations. In addition. Storm incorporates an Object-Oriented Design (OOD) methodology so that for unique installation requirements (i.e. unique hardware interface format), the customization is encapsulated to a minimal set of objects, usually only one.

Storm has developed the IMT using other, well established COTS products as sub components. This has been done to minimize overall cost, development time and risk. The initial release of the IMT uses multiple COTS products which Storm believes, after detailed product assessment and tradeoffs, to be best of breed for their role in this application. Storm has selected SYBASE as its relational database manager, Gensym's G2 Real-Time Expert System as its inferencing tool set and Loral Instrumentation's (LI) System 500 product line as the provider of telemetry acquisition, processing and display capabilities. Because of the use of OOD design methodologies, minimal effort would be required to re-host to or incorporate comparable COTS software products.

The IMT is partitioned into two component sets, a workstation set and a front-end set. The functions which are resident within each set are listed below.

Workstation

- User Interface
- Command Database (Command structure, bits and parameters)
- Pass Plan Development/Database
- Pass Plan Execution
- Command Builder

Front-end

- Intelligent Command Queue
- Transmission Manager
- Transmission Driver
- Validation Manager & associated Validation Algorithms
- Telemetry processing (decommutation, calibration, unit conversion, archival, etc.)

USER INTERFACE

The user interface is provided via three sources. The first is the ability to enter data into the database, which is performed via a Storm-developed GUI to the SYBASE SQL. The second user interface gives the user the ability to execute pass plans through the use of the G2 GUI. Finally, telemetry data is presented to the user via LI's DataScope, which is based upon V. I. Corporation's DataViews. The DataScope product allows the user to build telemetry views containing alphanumeric data, graphical (dials, gauges, strip charts, etc.) representations, schematic diagrams or any combination of these presentation methods. The goal of all three user interfaces is to provide the user the option of having a consistent "look & feel" throughout the IMT functions. These interfaces are all configurable by the users of the system.

COMMAND DATABASE

The IMT database interface allows command lists (ASCII files containing the vehicle command bits) and individual pass plans in a specific ASCII format to be entered into the database avoiding manual entry of this information. Storm utilizes the SYBASE relational database product line in its initial release of the IMT. These products (SQL Server, APT Workbench, Data Workbench) combined with a user interface developed by Storm provide the ability for users to define and store their command structure, the individual command bits, command parameters and pass plan information.

PASS PLAN DEVELOPMENT

Executable pass plan development can be performed via the ASCII file interface previously described or via a Motif-like interface which allows the user to define the commands and ground equipment control directives to be processed during a specific contact with the satellite vehicle. The result of this IMT pass plan development process is a set of objects stored in the database which are translated into G2 objects during the execution of the pass plan.

PASS PLAN EXECUTION

Execution of the IMT pass plan provides system control and flow management to system users during all periods of real-time operations. The complexity of the IMT pass plan is configurable by the user. This complexity ranges from manually stepping through a basic, sequential pass plan to the automatic execution of a complex pass plan which contains branching based upon real-time evaluation of externally generated data (telemetry, equipment status, etc.). Operator execution of the pass plan is augmented by the G2 expert system. Software bridges between G2 and the other IMT components are utilized to execute steps within the pass plan. The powerful G2 inferencing capability allowed Storm to develop a standard set of rules which control the flow of the pass plan and perform constraint checking. Additionally, the IMT uses G2 to provide the structure that allows the user to easily customize or define rules which functionally validate command execution.

COMMAND BUILDER

As the pass plan is executed, the Command Builder is invoked to extract the command structure and relevant command bits and parameters from the database. The complete command is then built and transmitted to the Intelligent Command Queue (ICQ) module which resides in the front-end. Commands can be built and loaded into the front-end as early as the user wishes. The storage of these pre-loaded commands is constrained only by the amount of memory available on the Field Programmable Processor (FPP) card in the LI System 500 front-end. The standard FPP configuration provides 2 MB of memory, and can be expanded, if necessary.

INTELLIGENT COMMAND QUEUE

The IMT functions which execute on the LI System 500 front-end are related to the transmission of commands, the acquisition and processing of telemetry data and the verification and validation of commands. Telemetry processing is performed through the use of functions provided by LI. Commands selected for transmission and constructed by the Command Builder are stored in the front-end in the ICQ. This queue and associated software hold the binary representations of the commands and release them according to instructions which are transmitted from the workstation. The ICQ monitors transmission and validation status in order to determine when commands can be released. When a range of commands is requested for release, the command definition of each will indicate whether to release the commands as fast as is possible (i.e. release after the previous one has transmitted, wait until validation of the previous command occurs, etc.). The ICQ is also responsible for retries, if this option has been selected for the command and the command does not validate. Time critical commanding is initiated by the ICQ, ensuring commands are released to the Transmission Manager based upon their specified vehicle arrival times.

TRANSMISSION MANAGER

Commands released by the ICQ are received by the Transmission Manager. The Transmission Manager is responsible for controlling the rate at which commands are released and for initiating the validation process. The Transmission Manager generates the correct preamble, interval and postamble sequences and passes the command to the Transmission Driver. The Transmission Manager controls the time critical commanding through logic which sends the commands to the Transmission Driver at precisely the time required to ensure the vehicle arrival time is met. This time is calculated using the known communication delays.

TRANSMISSION DRIVER

The Transmission Driver performs the final formatting of the command and releases the command to the hardware device which actually transmits the command from the front-end. The LI System 500 uses a Differential Input/Output (DIO) card for this purpose. Prior to releasing the command, the Transmission Driver inserts the correct count(s) into the command as well as any necessary parity bits, pad bits or other vehicle/installation specific data. The Transmission Driver is also responsible for the generation of S tones, if required, during any periods of vehicle contact without command transmission.

VALIDATION MANAGER AND ALGORITHMS

The Validation Manager, upon notification of command transmission, initiates the proper validation algorithm for the proper validation window. The IMT currently can handle loading and utilizing up to 32 different validation algorithms for each vehicle. If a larger number of algorithms is required for a specific program, the number can be easily increased by Storm. The IMT is delivered with a set of standard algorithms and the capability for the user to define customer specific validation routines. The Validation Manager reports the validation status upon successful validation or completion of the validation window.

APPLICATIONS AND BENEFITS

The database driven design of Storm Integration's Intelligent Mission Toolkit makes it possible for a single system to be used throughout the entire life cycle of a satellite or launch system development program. For example, the US. Ballistic Missile Defense Organization (BMDO - formerly the Strategic Defense Initiative Organization - SDIO) Miniature Seeker Technology Insertion (MSTI) program requires concurrent support for satellite test, training/simulation, launch control and on-orbit operations for multiple satellites. As one satellite is launched, another satellite may in test or require training/simulation. The flexibility of the IMT is ideally suited to support all phases of this or similar program life cycle.

IMT can also significantly improve the effectiveness of Ground Support Equipment (GSE). An IMT system can be used to capture the knowledge of satellite engineers and technical advisors as they test a vehicle. This knowledge can then be incorporated into test scripts effectively eliminating operator error during long tests. During on-orbit operations, the existence of this extensive technical knowledge-base reduces the need for satellite and launch vehicle Technical Analyst's (TA) to be involved in nominal (and even some anomalous) situations. In addition, if TA support is required, their involvement in the development of the knowledge-base and familiarity with the system enhances their ability to resolve problems quickly.

CONCLUSION

Storm Integration's Intelligent Mission Toolkit provides a flexible, modular, commercial product capable of meeting the needs of a variety of complex command and control systems. Its modular design allows it to be used effectively in both large and small scale systems. It is capable of operating in a stand-alone environment or becoming part of an existing system. In addition, IMT's ability to retain the knowledge of project experts - in a expert system knowledgebase - minimizes the impact associated with project staffing changes.

The IMT product concept and implementation has proven to be useful to a variety of new and existing complex systems. IMT has been selected by the United States Air Force as the commanding and telemetry element for its next generation distributed mission control complexes. In addition, IMT is currently being adapted to address the complex network management requirements of NASA and to a variety of Intelligent Vehicle Highway System (IVHS) programs.