

# IMACCS: A PROGRESS REPORT ON NASA/GSFC'S COTS-BASED GROUND DATA SYSTEMS, AND THEIR EXTENSION INTO NEW DOMAINS

E. J. Scheidker, R. D. Pendley, R. M. Rashkin, R. D. Werking (CSC)  
B. G. Cruse (Altair Aerospace)  
M. A. Bracken (NASA GSFC)

## ABSTRACT

The Integrated Monitoring, Analysis, and Control COTS System (IMACCS), a system providing real time satellite command and telemetry support, orbit and attitude determination, events prediction, and data trending, was implemented in 90 days at NASA Goddard Space Flight Center (GSFC) in 1995. This paper describes upgrades made to the original commercial, off-the-shelf (COTS)-based prototype. These upgrades include automation capability and spacecraft Integration and Testing (I&T) capability. A further extension to the prototype is the establishment of a direct RF interface to a spacecraft. As with the original prototype, all of these enhancements required lower staffing levels and reduced schedules compared to custom system development approaches. The team's approach to system development, including taking advantage of COTS and legacy software, is also described

## KEYWORDS

Satellite ground data system, COTS products, automation

## INTRODUCTION

Traditionally, ground data systems for NASA missions were built and integrated entirely by civil servants and contractors. Institutions at NASA centers performed system development and missions operations. Motivated by the desire to aggressively advance space science goals despite shrinking budgets, NASA's approach to all aspects of missions has changed, including ground data systems. Today, missions are conceived and flown in response to Announcements of Opportunity that make the Principal Investigator (PI) responsible for the allocation of funds. The PI can choose to get support wherever he perceives the best value and as a result, NASA centers must compete with each other and non-NASA institutions to provide satellite ground data systems.

At NASA's Goddard Space Flight Center (GSFC), the Mission Operations and Data Systems Directorate (MO&DSD) is charged with building and operating ground systems. Faced with the competitive challenge, MO&DSD sought to reengineer its business and initiated the RENAISSANCE project to lead the way. At its inception in 1993, RENAISSANCE had a modest goal: build an operational ground system in less than 1 year for less than \$5 million. Initial studies by the RENAISSANCE team led to an architecture based on reusable building blocks, garnered from GSFC's legacy systems where possible and built to be reusable (Stottlemeyer et al., 1993). This approach was called the RENAISSANCE first generation architecture. Shortly thereafter, NASA Director Goldin's exhortation to "faster, better, cheaper" was taken to imply far more substantial changes. The RENAISSANCE team responded with a second architecture that allowed for extensive use of COTS hardware and software (Stottlemeyer et al., 1996).

Indeed, in recent years, commercial off-the-shelf (COTS) hardware and software for satellite applications has evolved considerably. COTS tools now surpass the functionality of many custom-built systems and system components. The Eagle testbed, an outgrowth of the CIGSS (CSC Integrated Ground Support System) COTS and legacy system integration project of Computer Sciences Corporation (CSC) provides the experience base for CSC's COTS integration work (Werking and Kulp, 1993; Pendley et al., June 1994). Several other testbed projects, including the United States Air Force's (USAF) Center for Research Support (CERES) (Montfort, 1995), the International Maritime Satellite (INMARSAT) consortium, and the USAF Phillips Laboratory (Crowley, 1995) have produced successful prototypes using COTS components. The Extreme Ultraviolet Explorer (EUVE) Science Operations Center (SOC) at the University of California at Berkeley (Malina, 1994) has adapted a COTS-based system to automate science instrument operations, resulting in significant cost reductions.

### IMACCS 90 DAY PROJECT—A REVIEW

In 1995 CSC, building on its COTS integration experience, proposed that NASA Goddard's RENAISSANCE team build a COTS-based prototype to demonstrate that significant cost reductions were possible. The Integrated Monitoring, Analysis, and Control COTS System (IMACCS), had the following goals: integrate a set of COTS tools, connect them to live tracking and telemetry data, and reproduce the functions of an operational ground system (Bracken et al., 1995). The target mission for IMACCS was the Solar, Anomalous, and Magnetospheric Particle Explorer (SAMPEX) mission, one of the spacecraft in GSFC's Small Explorer (SMEX) series. SAMPEX is a low earth orbiting satellite in its fourth year of operational support. IMACCS was designed to replicate the current real time command and telemetry flight and off-line support for SAMPEX. A time limit of 90 days was imposed, and indeed, proved to be sufficient.

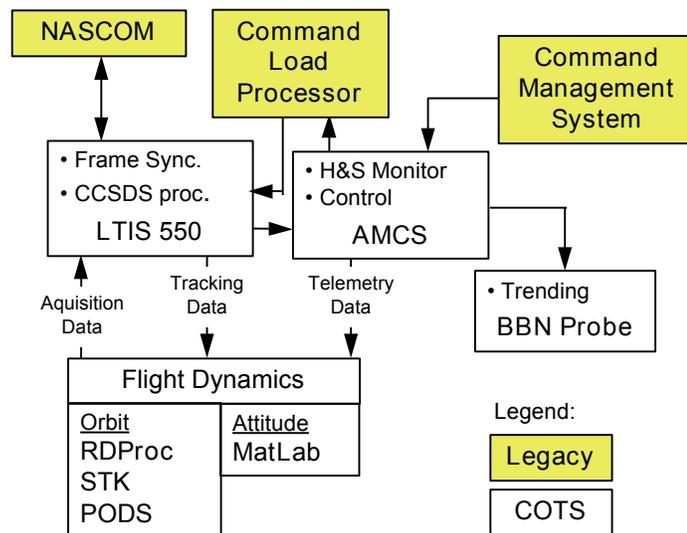


Figure 1. The IMACCS prototype contains six COTS software tools and three legacy tools.

A simplified block diagram of IMACCS is shown below in Figure 1. The COTS hardware and software have capabilities that exceed SAMPEX operations requirements. One tool, the Altair Mission Control System (AMCS), used on IMACCS for command and telemetry, shows substantial promise for automating data monitoring and commanding. CSC, through its Eagle testbed had prior experience with the AMCS and was familiar with its capacity to perform automated operational support. The AMCS provides automation through finite state modeling and state transitions (Wheal, 1993). State modeling and state transitions proved to be easy to implement, and a set of initial state models was built. Other features and capabilities of the IMACCS prototype are detailed in Bracken et al. (1995).

A key characteristic of the initial IMACCS project was the speed with which it was implemented, being fully functional 90 days after project start. This rapid turnaround on the original implementation has been repeated for all of the extensions described in this paper. For both new system development and major enhancement, being able to complete major system lifecycle phases on this timescale of 3 months or less is essential for future ground systems that must be delivered on reduced budgets and schedules.

IMACCS prototypes do not cover the entire traditional waterfall lifecycle. The major phases of this lifecycle for ground system development are shown in Table 1. The phases shown in this table are typical of standard methodologies used by NASA, DoD, and other major institutions. Development of the original IMACCS prototype and its extensions corresponds to the design phase through the testing phase. Because these prototypes have been built in parallel with existing systems (or systems under development), they have started from existing, and therefore stable, requirements. Similarly, because the prototypes can be evaluated against operational systems in most cases, the testing needed to establish their full requirements compliance is less than that required for a new system.

LIFECYCLE PHASE	PRINCIPAL ACTIVITIES
Requirements Development	Determine what the system is required to do and analyze requirements for completeness, feasibility, testability, etc.
Requirements Analysis	Determine system functionality and allocate functionality to high level components.
Design	Allocate functions to low-level components in detail and specify interfaces in detail.
Implementation	Construct components and integrate into the system.
Testing	Verify that the system meets its requirements.
Operational Deployment	Install system for operations and integrate with existing systems.
Operations and Maintenance	Support system as it used operationally.

Table 1. Major phases of traditional waterfall development lifecycle.

Nevertheless, the IMACCS lifecycle of 90 days (or less) for this subset of the traditional lifecycle is still very favorable compared to the 12 to 24 month periods that have been typical of the more traditional approach.

Another element of the IMACCS approach to system development and integration is to make maximum use of COTS products, freeware such as PERL and EXPECT, and other existing software tools. These tools are used intact, with no modifications, and communicate through simple interfaces such as files to the maximum extent possible. The maximum advantage accrues in the design and implementation phases, because there is little detailed design at the subroutine level and similarly minimal code to develop. We expect that fully testing such a system will take about the same amount of effort as for a traditional system, as the full set of requirements must be verified. We also think that reduced effort for the two requirements phases is likely, because the use of existing tools makes prototyping rapid enough to affect requirements development and analysis decisions.

The tools used in IMACCS do not require extensive training to use, and enable a user to directly implement a system function. The benefit of rapid development has been realized by tools that enable experts in the spacecraft and operations domains to adapt the tools to their needs without the intervention of experts from the software domain. User-intuitive interfaces enable spacecraft and operations engineers, unfamiliar with the tool, to rapidly customize the software. With appropriate COTS tools, individuals can easily develop mastery of several packages, thus facilitating their integration.

Another factor in the rapid implementation of the IMACCS prototypes is the use of a small, highly-empowered team of NASA civil servants and CSC engineers that has working relationships with the COTS product vendors. Team members worked in close proximity and a high degree of cooperation. Rapid progress, enhanced by visible results from graphical interfaces of the available tools, accelerated the development pace.

## IMACCS EXTENSIONS

Analysis of IMACCS operational functions (Pendley et al., November 1994) showed that although IMACCS satisfied telemetry and tracking data processing, commanding, mission planning, archiving and trending, and orbit and attitude determination functions, a number of other mission operations functions were not addressed. Furthermore, automation in the first prototype was restricted to real time data monitoring. The next step for IMACCS was to prove that a COTS-based architecture could expand both functionality and automation. Construction of the initial set of state models showed that the AMCS had substantial capability not only to automate operator functions, but also to implement to highly autonomous systems. Methods to automate off-line functions, such as orbit determination, events computation, and acquisition data generation, have also been discovered. The IMACCS team also investigated COTS alternatives to the NASCOM serial telemetry and command interface between the antenna and the system. A COTS RF link, connected directly to a test antenna at GSFC, was implemented and integrated with the IMACCS system. Finally, IMACCS was extended to perform spacecraft integration and test (I&T) functions.

**Automation** Our basic approach to automation was to take advantage of the capabilities available in the products or ensembles of products that constituted IMACCS. Working closely with the SAMPEX flight operations team, the IMACCS team developed five categories of operations activities:

- Data monitoring
- Routine pass activities
- Known contingencies
- Emergencies
- Product generation

The IMACCS team automated data monitoring, routine pass activities, known contingencies, and emergencies with the state modeling capability of the AMCS. These four activities are driven by real time telemetry, and their automation is detailed in Klein et al. (1996). For non-real time product generation, we needed a way to script the execution of interactive, Xwindows-based programs, like Satellite Tool Kit (STK) from Analytical Graphics. Our approach was to utilize a record-and-replay test tool, Xrunner from Mercury Systems (Lin et al., 1996).

**Radio Frequency Interface** The original IMACCS received tracking and telemetry data from, and sent commands to, SAMPEX through ground antennas located at Wallops Island, Va.; Goldstone, Ca.; Madrid, Spain; and Canberra, Australia. These stations communicate through the NASA Communications Network (NASCOM) serial data

interfaces for both downlink and uplink. IMACCS used the Loral Test and Information Systems LTIS550 front end to receive and transmit data via NASCOM.

Driven by the interest of some flight projects to control all their resources, and by the success of IMACCS, MO&DSD sponsored integration of COTS RF equipment with the IMACCS prototype (Butler, 1996). This system is shown in Figure 2. It utilizes a 4.3

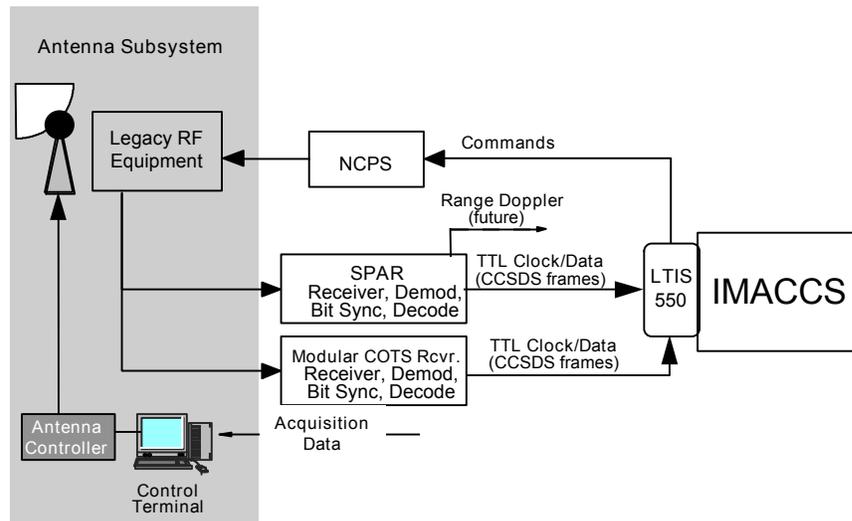


Figure 2. The COTS-based RF interface to IMACCS contained two separate receivers.

meter dish at the Greenbelt test facility and a receiver being developed by Stanford Telecommunications, Inc. under the sponsorship of Goddard Space Flight Center. The multi-functional, Software-Programmable Advanced Receiver (SPAR) (Zillig et al., 1995), couples advanced charge-coupled device (CCD) technology (developed by MIT/Lincoln Laboratory with NASA sponsorship) and digital signal processor (DSP) algorithms. SPAR will provide extremely flexible communications support for multiple modulation format, including PM, PSK, and FM; PN spread and non spread signals; integrated tone ranging of user spacecraft with 1-meter accuracy; and data rates up to 10 Mbps. For the RF interface test with IMACCS, only the receiver portion was used. The complete IF-to-baseband data receiver comprises three standard 220 millimeter, 6U VME cards: the IF module, CCD module, and DSP module. Communications and control connectivity between each of the modules and a local PC controller is achieved using a 5 MBPS industrial ARCNET local area network standard.

The IF module accepts RF input from 370 to 500 MHz (selected for application both at NASA's White Sands and GN ground stations) and at an input power level between -75 and -15 dBm. A key feature of the SPAR is an architecture that optimally leverages the use of new, programmable CCD technology against a powerful, high-speed multiprocessor arrangement of digital signal processors. Organized as a programmable discrete-time analog transversal filter, the CCD technology employed alternately serves as the SPAR's

PN-code matched filter for spread spectrum applications and as an IF-symbol-matched filter for non-spread applications. The resultant signal was passed to the LTIS front end, bypassing NASCOM altogether. The IMACCS/RF system took SAMPEX passes and tracked the spacecraft while monitoring states and telemetry. Other COTS products are available and could have been used in this prototype, as has been demonstrated by JPL. The integration demonstrated the feasibility of a complete, end-to-end COTS-based system and generated excitement and interest among demonstration audiences.

**Integration and Test (I&T)** Spacecraft I&T system functionality substantially overlaps operational ground system functionality, making it likely that a single system can be tailored to perform both roles. GSFC's RENAISSANCE team compared I&T requirements with those of operational systems, and found this overlap in areas such as data packing and unpacking, EU conversion, limit checking, and command and telemetry database ingestion. They also found that I&T systems differ by requiring frequent database updates and bit level data displays and command construction. I&T systems also derive little benefit from automation of monitoring or commanding.

We identify three systems in the lifecycle: the Spacecraft Component Test System (SCTS), the I&T System, and the Operational Ground Data System (GDS). The SCTS is a collection of tools that evolve as satellite components are developed. The I&T system is used to integrate and test the components into the complete spacecraft. The GDS is used to fly the satellite. As each system hands off to the next in the lifecycle, the information developed in the previous phase must be passed along. Traditionally these hand-offs have required that the three lifecycle systems need to read some database representation of the device parameters or have them input manually, and restructure the information for local use.

The IMACCS team approached this expanded requirement set with the same COTS-based architecture used in earlier prototypes. We obtained a spare component from the X-ray Timing Explorer (XTE) mission and reproduced the functionality of its ground support equipment using LabVIEW, a graphical instrument driver package made by National Instruments. Using the same LabVIEW interface to support the integration with the rest of the hardware, the new prototype populates the telemetry and command database, scripts test scenarios, and attaches to a CORBA based network to get data from a variety of data interfaces (the LTIS550, IP sockets, and direct 1553 connection). The operational system reverses the database operation and uses its information to decommutate and convert the data. The obvious advantage to a consistent architecture is that information can be passed along from one phase to another without manual intervention or reformatting. Moreover, users throughout the lifecycle are using similar, if not identical, interfaces to interact with the same spacecraft object.

This architecture evolves smoothly from SCTS through I&T to GDS and bypasses the inefficiencies and risks of data restructuring. IMACCS/I&T differs slightly from the original IMACCS. It is based on PC platforms under Windows NT, because SCTS tools should be on platforms used by spacecraft engineers. The IMACCS team is in the process of implementing a CORBA interface to make the network seem transparent to all users regardless of platform. This extension of IMACCS demonstrates that the benefits of a common architecture now extend from component testing to end of life.

## CONCLUSION

In the past, satellite missions required costly, custom-built systems because each new mission advanced the state of spaceflight art. Near the end of the fourth decade of spaceflight many more satellites are flown, and the domain of knowledge needed to operate these vehicles is better bounded, allowing development of general purpose tools and economies of scale. These tools are available as the kinds of COTS hardware and software used in IMACCS. The use of COTS-based ground systems will expand as the need for low cost, easily used and automated systems continues to increase. Future missions must be flown economically, which requires that all phases in the mission life cycle must be considered for development cost reduction and operational enhancement. Future extensions of IMACCS will address all phases of the spacecraft lifecycle, from system concept to end of life (from design to debris).

Within GSFC there are efforts being made to support these causes. The Landsat 7 mission is now pursuing the use of state modeling to support the automation efforts of the ground system using COTS tools validated on IMACCS and its extensions. NASA Goddard has also accepted a proposal to replace the current Upper Atmosphere Research Satellite (UARS) control center with a COTS-based system. The use of automation will be the responsibility of the flight operations team. This team, with support from the members of the IMACCS team will develop the state models to support the monitoring of the UARS satellite and develop the pre-pass planning scripts that will be used to automate routine commanding of the UARS satellite.

## REFERENCES

Bracken, M. A., Hoge, S. L., Sary, C. W., Rashkin, R. M., Pendley, R. D., & Werking, R. D., "IMACCS: An Operational, COTS-Based Ground Support System Proof-of-Concept Project," 1st International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations, Rutherford Appleton Laboratory, Chilton, Oxfordshire, U. K. - September, 1995.

Butler, M. J. et al., "A Revolutionary Approach for Providing Low-Cost Ground Data Systems," 4th International Symposium on Space Mission Operations, Munich, Germany - September, 1996.

Crowley, N., "Multimission Advanced Ground Intelligent Control Program," National Security Industrial Association Symposium, Sunnyvale, CA - August, 1995.

Klein, J. R. et al., "State Modeling and Pass Automation in Spacecraft Control," 4th International Symposium on Space Mission Operations, Munich, Germany - September, 1996.

Lin D. C., Klein, J. R., Pendley, R. D., & Hoge, S. L., "Use of Xrunner for Automation," 4th International Symposium on Space Mission Operations, Munich, Germany - September, 1996.

Malina, R.F., "Low-Cost Operations Approaches and Innovative Technology Testbedding at the EUVE Science Operations Center," 45th Congress of the International Astronautical Federation's Symposium on Small Satellite Missions, Jerusalem, Israel - October, 1994.

Montfort, R., "Center for Research Support, A New Acquisition Management Philosophy for COTS-Based TT&C Systems," National Security Industrial Association Symposium, Sunnyvale, CA - August, 1995.

Pendley, R. D., Scheidker, E. J., & Werking, R. D., "An Integrated Satellite Ground Support System," 4th Annual CSC Technology Conference. Atlanta, GA - June, 1994.

Pendley, R. D., Scheidker, E. J., Levitt, D. S., Myers, C. R., & Werking, R. D., "Integration of a Satellite Ground Support System Based on Analysis of the Satellite Ground Support Domain," 3rd International Symposium on Space Mission Operations and Ground Data Systems. Greenbelt, MD - November, 1994.

Stottlemyer, A. R., Jaworski, A., & Costa, S. R., "New Approaches to NASA Ground Data Systems," Proceedings of the 44th International Astronautical Congress, Q.4.404. Graz, Austria - October, 1993.

Stottlemyer, A. R. & Hassett, K. M., "Renaissance Generic Architecture, Version 2.0," Mission Operations and Data Systems Directorate, NASA Goddard, 504-REN-96/003 - May, 1996.

Werking, R. D. & Kulp, D. R., "Developing the CSC Integrated Ground Support System," 7th Annual AIAA/Utah State University Symposium on Small Satellites, Logan, UT - September, 1993.

Wheal, C. A., "Application of State Space Modeling Techniques to Satellite Operations," Altair Aerospace Corporation, Bowie, MD - 1993.

Zillig, D. J. & Land, T., "GN Advanced Receiver Prototype II (GARP II) A Charged-Coupled Device Programmable Integrated Receiver," Goddard Space Flight Center - 1995.