

11172

31992

TRANSPORTABLE PAYLOAD OPERATIONS CONTROL CENTER REUSABLE SOFTWARE:
BUILDING BLOCKS FOR QUALITY GROUND DATA SYSTEMS

p. 9

Ron Mahmot¹, John T. Koslosky¹, Edward Beach², Barbara Schwarz³

¹ NASA/Goddard Space Flight Center, ² Computer Sciences Corporation (CSC), ³ Integral Systems Inc.

ABSTRACT

The Mission Operations Division (MOD) at Goddard Space Flight Center builds Mission Operations Centers which are used by Flight Operations Teams to monitor and control satellites. Reducing system life cycle costs through software reuse has always been a priority of the MOD. The MOD's Transportable Payload Operations Control Center development team established an extensive library of 14 subsystems with over 100,000 delivered source instructions of reusable, generic software components. Nine TPOCC-based control centers to date support 11 satellites and achieved an average software reuse level of more than 75%. This paper shares experiences of how the TPOCC building blocks were developed and how building block developer's, mission development teams, and users are all part of the process.

1. INTRODUCTION

The TPOCC is a control center architecture which takes advantage of workstation based technology to improve mission operations and reduce development costs for Payload Operations Control Center's and Mission Operations Center's in GSFC's MOD. The TPOCC architecture is characterized by it's distributed processing, industry standards, commercial off-the-shelf (COTS) hardware and software products, and reusable custom software. The reusable TPOCC software is integrated with mission applications to provide

the health and safety monitoring, and commanding capabilities for various NASA satellites. This includes the capability to process and display telemetry, build and send commands, and perform special processing. In addition, TPOCC provides a graphical user interface and a procedural command language that automates ground system and spacecraft control.

The TPOCC development team established an extensive library of 14 subsystems with over 100,000 delivered source instructions of reusable, generic software components. By encapsulating the basic control center functionality into reusable building blocks, the control center design and implementation becomes a much easier task. The nine TPOCC-based control centers to date support 11 satellites and achieved an average software reuse level of more than 75% (see Table 1). The challenges involved in establishing and maintaining a cost-effective library of software components for ground system development include: making the initial investment, getting mission teams to adapt a reuse paradigm, configuration control, and staying current with technology.

This paper describes the MOD's approach to establishing and maintaining the TPOCC reusable software.

2. BACKGROUND

NASA's Goddard Space Flight Center (GSFC) Mission Operations Division (MOD) provides

Ground Support Systems for a variety of scientific satellites. The MOD designs, implements, tests, and delivers control centers - both traditional Payload Operations Control Centers (POCCs) and the newer, functionally expanded Mission Operations Centers (MOCs) - that provide command management and mission planning functions. Using an operations control center, the Flight Operations Team (FOT) provides around-the-clock support to its spacecraft, typically establishing communications contact every few hours for a period of about 20 minutes. During these brief contacts, the control center system functionality allows the FOT to quickly evaluate the current condition of the spacecraft and transmit the commands necessary to maintain its health and control the spacecraft and its science instruments.

Recognizing the many similarities among missions, the MOD always has made software reuse a priority. Prior to TPOCC, the most successful reuse effort was the Multisatellite Operations Control Center (MSOCC) Applications Executive (MAE), which

achieved a maximum software reuse level of about 40%.

The TPOCC project began in 1985 as a Control Center System Branch research and development effort examining the new workstation based technology to further reduce mission development cost and improve operations support. The ICE / IMP Control Center was successfully prototyped using a TPOCC system in little more than six months. Following this proof of concept POCC software was developed for the SAMPEX, WIND / POLAR, and ICE / IMP missions simultaneously with the implementation of the TPOCC core reusable software. The TPOCC core reusable control center software was completed with the delivery of TPOCC release 6.3, in July 1992, and successfully supported the SAMPEX launch. In addition to the numerous MOD control center applications which TPOCC now supports other non-control center applications utilize TPOCC software to reduce their development costs (see Table 1). TPOCC software continues to evolve based on mission needs and new technology opportunities.

TABLE 1 - APPLICATIONS INCORPORATING TPOCC

CATEGORY	APPLICATION
NASA/GSFC Control Centers:	International Cometary Explorer (ICE) / Interplanetary Monitoring Platform (IMP) Payload Operations Control Center (POCC), Interplanetary Physics Laboratory (WIND) POCC, Polar Plasma Laboratory (POLAR) POCC, Solar and Heliospheric Observatory (SOHO) POCC, Solar Anomalous & Magnetospheric Particle Explorer (SAMPEX) POCC, Fast Auroral Snapshot Explorer (FAST) POCC, Submillimeter Wave Astronomy Satellite (SWAS) POCC, X-Ray Timing Experiment (XTE) Mission Operations Center (MOC), Tropical Rainfall Measuring Mission (TRMM) MOC, Advanced Composition Explorer (ACE) MOC
Alenia Spazio Control Center:	X-SAR Mission Planning Operations System
NASA/GSFC Spacecraft Operations Tools:	Shuttle Payload Interface Facility (SPIF),

NASA/GSFC Spacecraft Operations Tools:	Generic Trend Analysis System, Generic Spacecraft Analyst Assistant, Ground Operations Technology Testbed, TPOCC Orbital Signature Analyzer, Space Views, FAST Level Zero Processor (LZP), Real time Attitude Display System (RTADS), Heads Up Display (HUD)
NASA/MSFC Spacecraft Operations Tools:	Gravity Probe B Project
NASA/GSFC Spacecraft Simulators:	TPOCC Advanced Spacecraft Simulator (TASS) (TASS systems have been tailored and delivered to WIND, POLAR, SOHO, FAST, SWAS, XTE, and TRMM) SOHO Simulator, XTE Simulator
NASA/GSFC Science Center:	SOHO Experimentor Operations Facility
Cal Tech Science Center:	ACE Science Operations Center (SOC)
NASA/GSFC Spacecraft Integration and Test (I&T) Facilities:	SAMPEX I&T, FAST I&T, XTE I&T
Martin Marietta Spacecraft I&T Facility:	WIND/POLAR I&T
Johns Hopkins University Applied Physics Laboratory (APL):	ACE I&T

3. BASELINING THE BUILDING BLOCKS

Prior to baseling TPOCC software and committing to mission deadlines an investment was made in a system development concept, technology evaluation, and prototyping POCC subsystems. The System development concept resulted in a series of mandates for both the architecture and the management approach for mission's which will use the TPOCC architecture.

The management approach taken on TPOCC was to institutionally fund and manage the TPOCC reusable software development and architecture. Each mission would then fund an effort to build a dedicated control center based on TPOCC building blocks integrated with mission specific software. Enhancements to the core reusable software would be evaluated on a case by case basis using project funding when necessary.

There were four key TPOCC architecture mandates: distributed processing; widely accepted industry standards; reusable software

with clearly defined interfaces; and commercial off-the-shelf (COTS) products.

TPOCC adapted distributed processing as a means to scale hardware and software to the requirements and complexity of a specific mission. Implemented through a client/server method, functional data processing capabilities are developed as independently executing software tasks distributed across a group of networked inexpensive heterogeneous computers.

Although all TPOCC software components can execute on a single workstation, the current generation of TPOCC based control centers are hosted on a heterogeneous set of processors to meet mission data throughput needs. The main hardware components are a front end processor (FEP) and multiple UNIX workstations connected to an Ethernet-based local area network (LAN). The FEP, a VME chassis containing one or more single-board computers using a real-time UNIX-like operating system, performs the time-critical control center functions. The VME bus

provides high data throughput between the processes hosted within the FEP, while reducing data volume on the external LAN. The FEP also contains mass data storage, as well as additional NASA communications (NASCOM) interface hardware for receiving spacecraft telemetry data and for sending commands and data to the spacecraft.

Another major contributor not only to software reuse across control centers but also to the extendibility of TPOCC to other non-control center applications is its clearly defined and tightly controlled interfaces. Because all TPOCC based systems produce software that complies with these interfaces, capabilities developed by mission development teams can be easily shared, either by subsuming the new capability into the TPOCC reusable software base (if the capability is truly generic) or directly into other applications that require the specific capability. These well defined interfaces also make it possible for TPOCC compliant applications to share data. TPOCCs defined data service protocol allows a loosely coupled application to link into a TPOCC based system, access its telemetry stream, receive updated parameter values in real-time, act on that data in its own software, and return data to the TPOCC based system for display.

Adhering to widely accepted industry standards was seen as an essential part of TPOCCs approach to building a software library of components which could be compatible with advances in hardware technology. The standards used for TPOCC hardware components include VME, Ethernet, RS 232, RS422, and SCSI. TPOCC software standards adheres to open systems communications standards such as the Transmission Control Protocol/Internet Protocol (TCP/IP), external data representation (XDR), and network file system (NFS). TPOCCs user interface standards

include the X-Window System and the Open Software Foundation(OSF)/MOTIF software.

Following widely accepted industry standards also puts TPOCC in a position to maximize the use of COTS products wherever possible. Graphical User Interface, database, development tools, and network management are a few of the areas where TPOCC has utilized COTS products in lieu of costly in house development efforts. As a result, system enhancements have become more dependent on COTS vendors, making vendor reliability an important aspect of technical evaluations.

A significant amount of time was spent on doing technical evaluations and prototyping in the early phases of the TPOCC project. High risk requirements were prototyped and workstation based hardware and software components were evaluated for its application to the control center environment. Prototyping continues to play an important role in evolving the TPOCC reusable software base with technology.

4. ESTABLISHING A REUSE PARADIGM AMONG MISSION TEAMS

System development based on TPOCC software building blocks and TPOCC architecture introduced technical and management challenges. Initially, TPOCC development was localized within a research group and focused primarily on developing concepts, identifying applicable technology, and prototyping high risk areas. Mission POCC development teams were accustomed to developing, implementing, and testing each POCC individually using FORTRAN on a IBM or DEC like minicomputers. Both the TPOCC development group and mission development teams needed to change their perspective. Mission developers could no longer view themselves as developers of separate and

unique control centers but rather as contributors to the larger body of common control center software.

The first step toward this paradigm shift was to introduce the mission development teams to the principles and mindset necessary to make large-scale subsystem reusability a reality. The TPOCC development group conducted classes on the TPOCC development methodology, system design, and system capabilities. Also development teams attended vendor training classes on the new technology and software implementation environment. The third method of training was having the mission teams build prototype POCC systems utilizing configured TPOCC generic subsystems and adding their mission specific code. The complete telemetry processing path was prototyped for SAMPEX & WIND/POLAR in approximately two months .

Communication and feedback between the TPOCC staff and mission development teams becomes increasingly important as the number of TPOCC-provided capabilities and the number of TPOCC-based applications increase. A Configuration Review Board (CRB) and working group, with a variety of representatives including mission development teams (contractors and GSFC personnel for each mission), the TPOCC staff, users, and representatives of other (i.e. non-control center) TPOCC based systems was established. The working group ensures that everyone is kept abreast of the efforts, needs and schedule of the collective group. As mission development teams identify candidate capabilities for generic implementation, the CRB determines which candidate capabilities are truly generic, schedules their implementation, and provides configuration control.

As the number of missions increase, requirements for new capabilities grew but the

TPOCC budget has remained constant. In order to make this effort successful while controlling cost, temporary teams with members from the TPOCC group and mission development teams are formed to design and implement new generic software. In addition to providing additional staff to implement generic capabilities, TPOCC staff gets insight into what portions of a capability are generic, and help identify likely differences among missions. For the missions, these teams provide insight into designing software for reusability and experience in differentiating the mission-unique elements of a problem from the generic.

Using TPOCC software and architecture framework, mission development teams have also been able to support unique needs of individual spacecraft requirements. The WIND/POLAR POCC, supporting two spacecraft scheduled to launched within a year, handles two physical channels of time division multiplexed (TDM) telemetry, while other TPOCC missions use the CCSDS standard. The SOHO POCC, scheduled to launch in mid-1995, must handle a unique combined TDM and CCSDS telemetry format.

To date, six TPOCC-based applications systems have been configured to support eight different spacecraft. The first three POCCs developed for the Small Explorer (SMEX) program, SAMPEX, FAST, and SWAS, also serve as a case study in determining the levels of reuse that can be attained with the TPOCC architecture within a series of missions. The SAMPEX POCC consists of COTS products, 67% TPOCC-generic software and 33% mission-specific software (see Table 2). By reusing both SAMPEX mission-specific software and the expanded TPOCC-generic software, the FAST POCC achieved an 81% reuse level. The SWAS POCC, expanding this

reuse base to include FAST mission specific software, achieved a reuse level of 92%.

Two latter missions, XTE and TRMM chose a TPOCC-based-MOC approach, integrating the traditional POCC, mission planning, and command management functions together into one system by sharing functionality and reusable software which had previously been implemented separately. Since the TRMM and XTE spacecraft have significant similarities,

much of the mission specific code in XTE will be reusable in TRMM.

Another indicator of TPOCC reuse across missions is the number of new generic capabilities asked for by mission development teams and users. Totals by mission of new generic capability requests, documented as Internal Configuration Changes Requests (ICCRs), are listed in Table 2.

TABLE 2 - NASA/GSFC Control Center TPOCC Reuse Data

Project Name	Mission Specific DSI*	Reused DSI	Total System DSI	Percent Reuse	Total*** Generic Change Requests(ICCRs)
ICE/IMP**	25407	72552	97959	74%	17
ISTP series: WIND	43372	81895	125267	65%	23
POLAR	7414	124934	132348	94%	0
SOHO	43690	94840	138530	68%	8
SMEX series: SAMPEX	38308	77125	115433	67%	16
FAST	22707	96534	119241	81%	11
SWAS	9800	114434	124234	92%	1
XTE	133340	128520	261860	49%	14
TRMM	14450	235360	249810	94%	2

*DSI = Delivered Source Instructions

**Following its initial prototype, ICE/IMP was completely rehosted

***Number of approved ICCRs from TPOCC Release 2 through Release 12

5. CONFIGURATION CONTROL

The TPOCC CRB oversees the reusable software configuration management as defined in the TPOCC Project Support Plan. Because the TPOCC Reusable Software forms the core of several control centers, configuration control is necessary across development efforts and it is essential that:

- Proposed changes [i.e., internal configuration change requests(ICCRs)] to the TPOCC software be visible to TPOCC missions before implementation

and that TPOCC applications be given the opportunity to analyze the proposed changes and assess the impact to the TPOCC applications

- Proposed changes [i.e., configuration change requests (CCRs)] to mission software be visible to the TPOCC project and other application teams for analysis, and possible reusable implementation recommendations.
- Problems detected during testing of the TPOCC release be documented as TPOCC internal discrepancy reports

(IDRs) and made visible to all TPOCC applications

- Problems detected during testing of TPOCC applications be documented as discrepancy reports (DRs) and made visible to all TPOCC applications
- DRs written against TPOCC applications are analyzed; assigned to either generic TPOCC or a mission development team based on the results of that analysis; and if assigned to the generic TPOCC, made visible to all TPOCC applications.
- The reusable TPOCC capabilities are accurately incorporated in the TPOCC Capabilities Document

5.1 Configuration Baselines

Two levels of software baselines are established for TPOCC reusable software. The TPOCC Capabilities Document, analogous to a typical system requirements document, contains a structured list of TPOCC capabilities. A build release plan maps the capabilities to specific releases. This permanent baseline will be updated only in response to ICCRs written against TPOCC software or CCRs written against a TPOCC application but implemented as a generic TPOCC capability.

The second configuration baseline documents the as built TPOCC release. This release product baseline consists of the software release, design documentation, the TPOCC implementation guide, generic user's guide sections, test procedures, data, and results.

The TPOCC Detailed Design Specification describes the current design of each generic TPOCC subsystem and includes a summary of each subsystems functions, a subsystem architecture diagram, high level descriptions of each task, unit prologs, a calling hierarchy within each task, network interface

specifications, file definitions, and data structure specifications.

The TPOCC Implementation Guide is the programmers manual for using and maintaining TPOCC generic software subsystem. It includes a description of the fundamental UNIX concepts that are used in building a TPOCC application system, a description of the functions available in the TPOCC software library, and a description of how to build and use each generic TPOCC subsystem.

The generic user's guide sections describe operational aspects of the TPOCC reusable software which are incorporated into the system user's guides of each TPOCC mission. Also the TPOCC Display Page User's Guide is provided which describes the methods of creating display pages with each release.

5.2 Software Support Policy

The TPOCC approach used in software support is similar to that of many operating system vendors. The development team makes each release of reusable software available to application development teams for incorporation into their deliverable. If an application group elects not to incorporate the new TPOCC release then software support from the TPOCC development team will not be guaranteed. Any reusable software changes made by the application teams without following the CRB standard procedures for configuration control will nullify TPOCC's support agreement with that application.

New capabilities are scheduled in TPOCC releases based on mission need dates, with an average of two software releases a year. Frequently the TPOCC development team is over booked with new capabilities and something has to give. Based on priority and the required expertise, some ICCRs get

delegated to application development teams. ICCRs that can wait get deferred. An applications implementation of a generic capability is initially delivered as part of its mission specific software and then folded into the TPOCC release at a latter time. Larger subsystems such as the NASCOM interface, packet processing, or Network Control Center (NCC) Interface has been implemented with TPOCC and mission support staff.

New TPOCC releases are tested in a mission independent testbed environment and then made available to mission development teams for integration with mission specific software. A test cycle is completed when an independent acceptance test team tests the complete application system delivery. Problems identified in the application may, after analysis, be found to reside in either the application software or in TPOCC. The CRB reporting mechanism makes generic problems known to all applications.

Changes to generic TPOCC get initiated as an Internal CCR on TPOCC reusable software itself or as a CCR written against the missions requirements document and presented to the Configuration Control Board. Generic TPOCC ICCRs are reviewed by both the generic TPOCC project and the application groups for technical feasibility, cost, and schedule impact. Having the user's who initiate the requirements attend the CRB has been an important part of the success enjoyed thus far.

6. TECHNOLOGY INFUSION

New technology advances are incorporated into the TPOCC architecture by: integrating new/upgraded COTS products, integrating new ground system tools/components developed by NASA organizations external to the MOD, and enhancing TPOCC software. An institutional prototyping and technical

evaluation group looks at various products from each area for it's applicability to the TPOCC architecture based on mission requirements and industry trends. As members of the TPOCC Working Group, users and mission development teams provide significant input into the prototyping and technical evaluation activities. New systems for trend analysis, s/c subsystem monitoring, and s/c visualizing have been integrated in TPOCC's loosely coupled architecture. The TPOCC User Interface continues to be enhanced by taking advantage of new MOTIF GUI capabilities.

The workstation era has accelerated the obsolesce of computer systems. A characteristic of the mainframe & minicomputer era was to replace systems every five years which effectively reduced any reusable software base to zero due to software dependencies on the hardware. TPOCC software has transitioned between two generations of FEP computers and ports to Sun, HP, IBM, and VAX workstations.

7. CONCLUSION

Averaging 75% of a missions control center deliverable, TPOCC software has reduced cost and shortened the control center development life cycle. Established standards both in industry and spacecraft development have increased the reuse factor to well over 90%. A successful process for evolving the TPOCC software with new capabilities and new technology is in place. As projects "Re-Engineer" the development life cycle by relying more on the TPOCC software maturity schedules will continue to be shortened and cost reduced.

REFERENCES

- NASA/GSFC, MOD, CCSB, 511-1PSP/0190, CSC and ISI, TPOCC Project Support Plan (Revision 4),
December 1993
- , 511-4SSD/0393, CSC and ISI, TPOCC Implementation Guide for Release 11, July 1994
- , 511-4DDS/0193, CSC and ISI, TPOCC Detailed Design Specification for Release 10, February 1994
- , 511-4SUG/0594, CSC and ISI, TPOCC Display Page Users Guide for Release 11, June 1994
- , 511-4SUG/0494, CSC and ISI, TPOCC Generic User's Guide Sections for Release 11, July 1994

- , 511-4SSD/0290, CSC and ISI, TPOCC Reusable Subsystem Capabilities (Revision 4), April 1994
- , CSC and ISI, "TPOCC: A Satellite Control Center System Kernel That Fosters High Reuse and Lower Cost"
- , CSC, "TPOCC Task Labor Projections Through FY95"

ACKNOWLEDGMENTS

We wish to extend special thanks to Carroll Dudley, Dolly Perkins, Judy Bruner, Dan Mandl, Mike Rackley, Jack Lauderdale, Dave Johnson, and Bill Stratton for their ideas which went into this paper.