

Control Architecture for the Deep Space Mission System (DSMS)

Wallace Tai, Peter Shames

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, MS 303-402, Pasadena, CA 91109, USA
Phone: (818) 354-7561, e-mail: wallace.s.tai@jpl.nasa.gov

AND

Richard Schell

Consolidated Space Operations Contract
595 Gemini Ave., Houston, TX 77058, USA
Phone: (281) 853-3050, e-mail: Rich.Schell@csoconline.com

Abstract

As NASA moves into an era of flying more missions at much lower cost and shorter development duration, the Deep Space Mission System (DSMS) has been redesigned to provide services to approximately 50 missions during the next 10 years. The DSMS is comprised of the present Deep Space Network (DSN) and Advanced Multi-Mission Operations System (AMMOS) managed by JPL. Fundamental to the DSMS redesign is a control architecture that will be incrementally implemented and deployed during the next decade. This new control architecture, jointly designed by JPL and the Consolidated Space Operations Contractor (CSOC) will reduce the service cost through consolidating systems, streamlining operations, and increasing the use of automation in routine operations. This paper summarizes how we established the key characteristics of the control architecture at different levels, i.e. subsystem, tracking station, deep space communications complex (DSCC), and overall DSMS levels. The monitor and control functions are then reallocated accordingly.

1. INTRODUCTION

The JPL Deep Space Mission System (DSMS) is a service system providing mission operations services to flight missions. It is a subset of the NASA's Integrated Operations Architecture (IOA). As an operational system, the DSMS is comprised of data system elements, i.e. hardware and software, and operational teams which are multi-mission in functionality. The DSMS includes both the present ground-based systems, i.e. the Deep Space Network (DSN) and the Advanced Multi-Mission Operations System (AMMOS), and the flight-based service elements being implemented by the Mission Data System (MDS) and Mars Network initiatives. A critical component of the DSMS is the control system. The "control system", in this context, is defined as the set of elements, embedded in the DSMS, performing the following two categories of functions: (1) Resource allocation and scheduling, (2) Asset configuration and control.

This paper gives a description of the new control system architecture for the DSMS designed jointly by JPL and CSOC during the period of May 1999 – March 2000. It is envisioned that this control architecture will be incrementally developed and deployed for operations throughout the next 10 years.

2. BACKGROUND

In 1995, an implementation effort, called Network Control Project (NCP), was initiated to modernize the control system in DSN. Key accomplishments of the effort include: (1) Replaced the antiquated computer equipment and monitor and control software at the 3 Deep Space Communications Complexes (DSCCs) with modern workstations and more sophisticated software. (2) Provided certain automated, albeit still limited, monitor & control capabilities at the DSCC to reduce the amount of manual operations conducted at the DSCCs. The task is scheduled to complete by the end of 2000. Since October 1997, two other initiatives, i.e. Network Simplification Project (NSP) and 26-meter Subnet Automation Task, have been in progress to consolidate and replace the aging telemetry,

tracking, and command (TT&C) subsystems at the various tracking stations in each DSCC. These new TT&C subsystems will be deployed for operations by October 2003 at 34-m/70-m stations and July 2000 at 26-m stations. While all these 3 initiatives represent significant steps toward an efficient and cost effective system, the DSMS still faces some key challenges in view of the increasing number of missions it has to support for the next decade and the declining budget profile for the system. Our strategic analysis concluded that an improved control system would be part of the solutions to these challenges. We further concluded that, since the DSMS is a subset of the NASA's IOA, the DSMS control architecture must be compatible with the technical objectives of the NASA's evolving Integrated Operations Architecture (IOA) being engineered by the CSOC. Chief among the technical objectives of the IOA are: (1) Provide high quality and reliable mission operations services at a significantly reduced cost. (2) Provide an integrated architecture that reduces overlap, and eliminates unnecessary duplication. At the present, software functions and operational approaches to control NASA's space operation systems are largely mission domain-unique (Earth orbiting, human exploration, and deep space respectively) due to (1) the lack of standard control architecture cross the three mission domains, (2) evolutionary nature of these systems. Toward the second IOA objective, the design of the DSMS control system must reflect a standard control architecture capable of accommodating all 3 mission domains.

3. GENERAL DESCRIPTION OF THE DSMS CONTROL ARCHITECTURE – PRESENT & FUTURE

Architectural Approach to Design

The design of the elements of a system ideally proceeds from a clear statement of the system architecture. Is it a single powerful central processor with simple terminals or a fully distributed set of powerful workstations working as peers, or are there workstations with servers which provide key data management and processing functions? The architecture is more than just the processors and their topology, it also includes allocation of processing functions, the physical network connections and bandwidths, service interfaces and protocols, user interfaces, failover strategies, operational activities, and assumptions about how faults will be recognized and handled, whether manually or automatically. The architecture of the system, made implicit during careful design, analysis, and validation, controls how the system works, where elements can be deployed, how the customers interact with it, and its responsiveness in the face of urgent demands for service, extending existing capabilities, meeting new requirements, and handling component failures.

In some cases the system architecture may not be explicit but must be discovered by investigation, either because it has not been clearly defined and described at the beginning of development or it has evolved over the years without having the documentation updated. See "Playing Detective: Reconstructing Software Architecture from Available Evidence", Kazman and Carriere, CMU-SEI 97-TR-010 for a very useful discussion of these issues. Whether explicit or implied, the architecture of the system defines and constrains the interfaces and functionality of all of the elements of the system. Thus, understanding the fundamental architecture of a system helps us to understand how it works and why it works the way it does. It is also useful to understand the business and economic drivers on a system, as these most often constrain many of the technical choices.

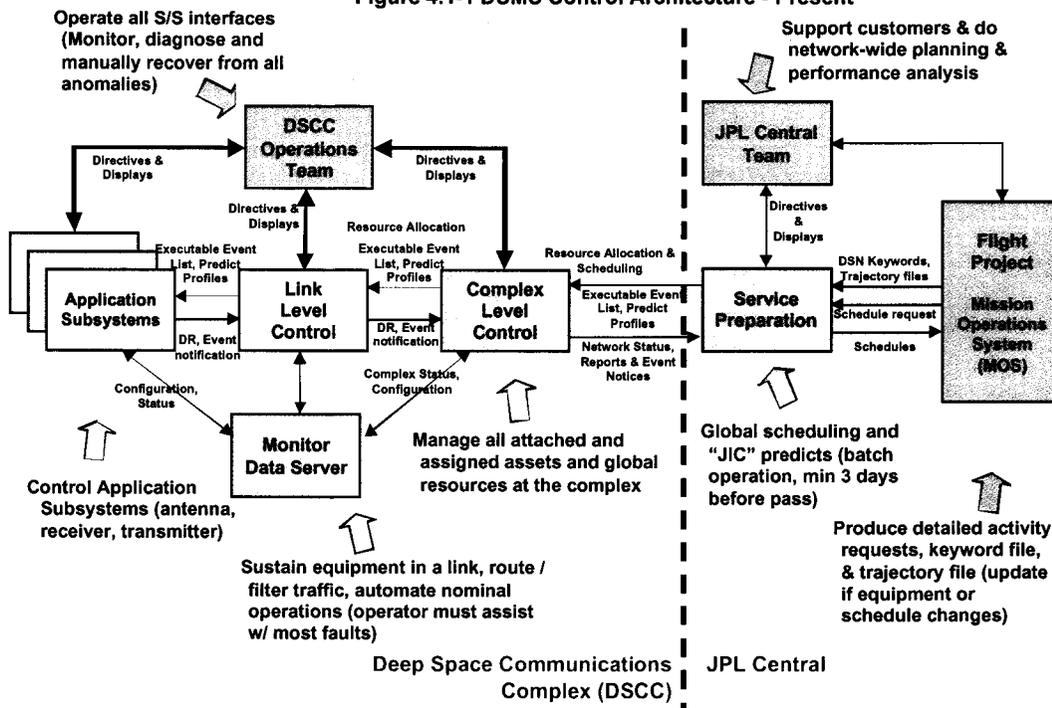
4. KEY CHARACTERISTICS OF THE DSMS CONTROL ARCHITECTURE – PRESENT AND FUTURE

4.1 General Description of the DSMS Control Architecture – Present and Future

The current DSMS control architecture is most easily characterized by two key assumptions:

- (1) The assumption that it was most cost effective to plan, schedule, configure and control the system from a single central location.
- (2) The assumption that some nominal operations can be automated but most off-nominal operations are best handled by a human.

Figure 4.1-1 DSMS Control Architecture - Present



The current control architecture is a highly centralized one, with all planning, scheduling, predicts generation, and configuration functions handled at the central site. Figure 4.1-1 depicts the present control architecture for the DSMS. Run-time operations at each of the complexes are also centralized at each complex, with relatively little responsibility allocated to any of the subsystems. Many of these planning and control functions require manual intervention and they operate in batch mode, rather than interactively. This is largely a legacy of earlier system designs that did, in fact, use central computers. Batch processing, done well in advance of real time, results in "just in case" planning for all possible eventualities, static plans, and less desirable responsiveness to last minute changes or exigencies. Where automation is employed only handling of nominal operations is expected, all off-nominal operations require human intervention, to reboot systems, switch connections, or to key in recovery actions. In this system mission users must explicitly know which individual pieces of equipment will be used to support their tasks, and must specify, in detail, the configuration of each element. Late changes in equipment or mission plans require recreation of these detailed ground system control scenarios. Managing all of these manually scheduled items and configuration is left to the mission user.

The future DSMS control architecture for the next decade is most easily characterized by two key assumptions:

- (1) The assumption that it is more cost effective to plan, schedule, configure and control the system as a set of hierarchical elements, each with limited span of control.
- (2) The assumption that all nominal and most anticipated off-nominal operations can be automated and that only unusual problems must be handled by a human.

The new control architecture is a highly distributed one, with planning and scheduling still done in one location, but predicts generation, configuration, and execution control functions done just-in-time, and at the lowest practical levels in the control hierarchy. Figure 4.1-2 shows the new control architecture. Run-time operations at each of the complexes are also distributed, with replicated, but separate, control logic for each station and for the central complex level functions. Most of these control functions are automated but all can be interactive when they need to be. This is a modern distributed system design that carefully allocates similar functions to appropriate classes of servers and

occurrences associated with the needed tracking passes. The requests for service must be augmented with further details (tied to assigned equipment) about a week before the scheduled support to a spacecraft. The service request details are specified in a keywords file (characterized as “sequence of events”) either provided by the flight project customers or generated by the DSMS operations. The DSMS control system then uses these keywords to drive the low-level activities to be performed at the DSCC and stations during the pass.

To flight project customers, this interface approach has the following ramifications: (1) The flight project MOS must possess detailed knowledge (as required for generating accurate keywords file) about the internals of the DSMS, thus defeating the purpose of counting on the DSMS as a service-providing system. (2) The tight coupling between service provider and service user tends to blur the line of service performance accountability. An inaccurate keyword generated by the flight projects could result in wrong predicts and low quality service instances, thus creating unnecessary question about who is accountable for the error.

The future control architecture calls for typical service requests to be specified in terms of types of service required and associated high level parameters. Unusual or critical mission events, e.g. encounters, maneuvers, can still be supported by more detailed parameters in the service requests. The service requests will then be used by the DSMS to generate schedule, event list, and predicts using a spacecraft dependent database and internal knowledge of the actual assigned equipment. Planned and scheduled service requests, and spacecraft characteristics, are kept in a service management database (SMDB) where they can be accessed and updated as needed. It is expected that the service requests will conform to the service package standards as specified by the CCSDS Panel 3 for Space Link Extension (SLE) Service Management.

4.3 Planning and Scheduling of Resources

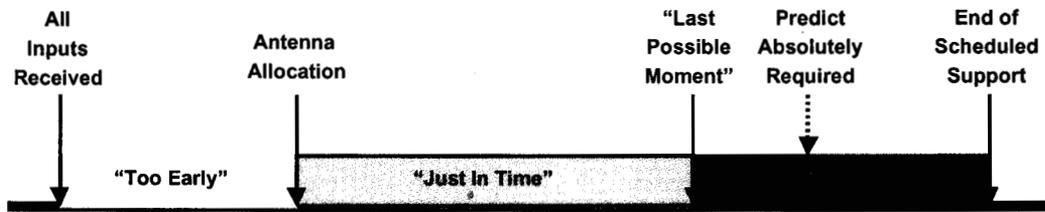
The current control system employs two distinct processes to support flight missions in planning and scheduling DSN resources: one for long-term planning and mid-range scheduling, the other for near-real-time and real-time scheduling. This is primarily due to the fact that historically the former process involves significant amount of engineering analysis in order to optimize the competing service requests from multiple flight projects, whereas the latter process is more a routine, repetitive task. Nevertheless, both are labor-intensive activities. The separation of these two processes over the past 20+ years has also resulted in two different scheduling tools used by the two teams. A direct effect of this is that, since each scheduling tool was developed to solve the problems prescribed by the corresponding process in isolation, the combined capabilities of both tools tend to be less than optimal. For example, the schedule conflict detection must be done in post-processing, i.e. after a preliminary schedule has been generated, simply because the detection rules and equipment status information are not an inherent part of the model and are not integrated with the scheduling tools.

Perhaps, the most pronounced trait in the present scheduling system is the serial relationship between schedule generation, schedule change request, conflict detection, and conflict resolution. All four steps involve batch-oriented data processing. All four steps require significant human intervention, e.g. data entry, by personnel on both DSMS and flight project sides.

The new control system features a single planning and scheduling process that is enabled by a single scheduling tool designed around the service management database (SMDB). Schedule generation, schedule change request, conflict detection, and conflict resolution are highly interactive to one another. Schedule conflict resolution will be the only activity requiring human intervention from the DSMS side. The interactive scheduling tool and SMDB will be directly accessible to the flight project mission engineers. In fact, the input to schedule generation will be based on service requests issued by the mission engineers eliminating even the need for any dedicated schedulers on the flight project side. Figure 4.3-1 depicts the planning and scheduling design in the new control system.

station level, as it will be able to dynamically respond to station reconfiguration and other changes. Figure 4.4-1 shows the concept of “just in time” predicts generation.

Figure 4.4-1 Concept of “Just-In-Time” Predicts Generation



- Too Early -
 - After all inputs have been received, but before allocation of the antenna to a support.
 - Only Just In Case predictions are possible.
- Too Late -
 - After the last possible moment to start predictions based on generation requirements and need time.
- Just In Time -
 - After allocation of the antenna to a support.
 - Before it's too late (i.e., up to required download time)
 - Earlier is better.
 - More chance to correct problems
 - Less impact of network outage

4.5 Automation in Service Execution

The current control system relies on centralized control with relatively passive end systems. It utilizes an “operator assist” approach to automating and orchestrating the activities performed by the various elements, i.e. subsystems and assemblies, at the DSCC during a tracking pass. The automation is accomplished through static scripts generated for nominal events spanning from pre-pass to in-pass and post-pass. These scripts, in form of script blocks, are tied together into a Temporal Dependency Network (TDN) based on their relationship in precedence. Figure 4.5-1 shows a high-level Temporal Dependency Network (TDN) and its script blocks generated to configure the various pieces of equipment during pre-pass. This approach represents a significant improvement, in degree of automation, from its predecessor, the “operator assist by macro” approach. It reduces operational costs. In the long run, it makes automated operations of nominal passes more repeatable and thus highly reliable. The challenges, however, are: (1) High development cost in order to accommodate new equipment and capability enhancement. New script blocks and updated TDN must be developed and tested, requiring costly knowledge engineering (2) Operators must be in the loop for anomaly and even minor exceptional conditions. This is because static scripts are inherently limited in adaptability. Developing scripts to handle recovery and unusual problems is a complicated effort. (3) TDN and static scripts are not opaque to operators and, as such, require additional training for operators and sustaining.

Figure 4.5-2 describes the design of automated service execution in the future control architecture. The automation design can be characterized as follows: (1) It applies a hierarchical control model. Control authority is formalized for each level of control elements. A new control element, i.e. the station controller, will be deployed at each station in a DSCC. Application subsystems, enabled by the Network Simplification Project (NSP) deployment by 2003, will be relatively smarter than they are today. (2) The control approach will be based on state-based, goal-oriented architecture championed by the JPL Mission Data System (MDS) initiative. (3) Common control applied recursively at complex, station, and subsystem level. (4) It delineates of control functions into state control (recovery), state determination (monitoring), and state knowledge (anomaly detection). (5) Automation in service execution will evolve with gradual increase in sophistication for each of the 3 functions. Some functions will continue to be manually executed in certain stages of development.

Figure 4.5-1 Automation in Service Execution: Present Scripts Scheme

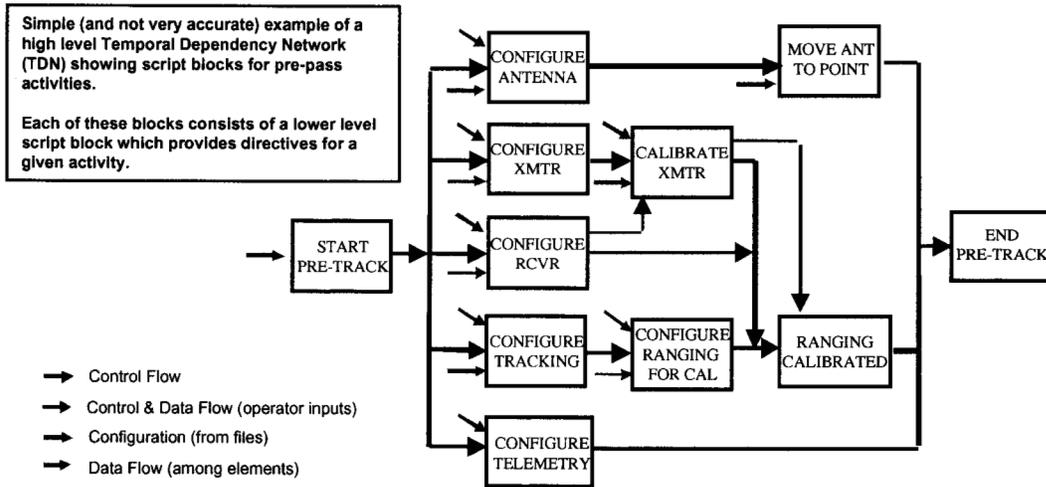
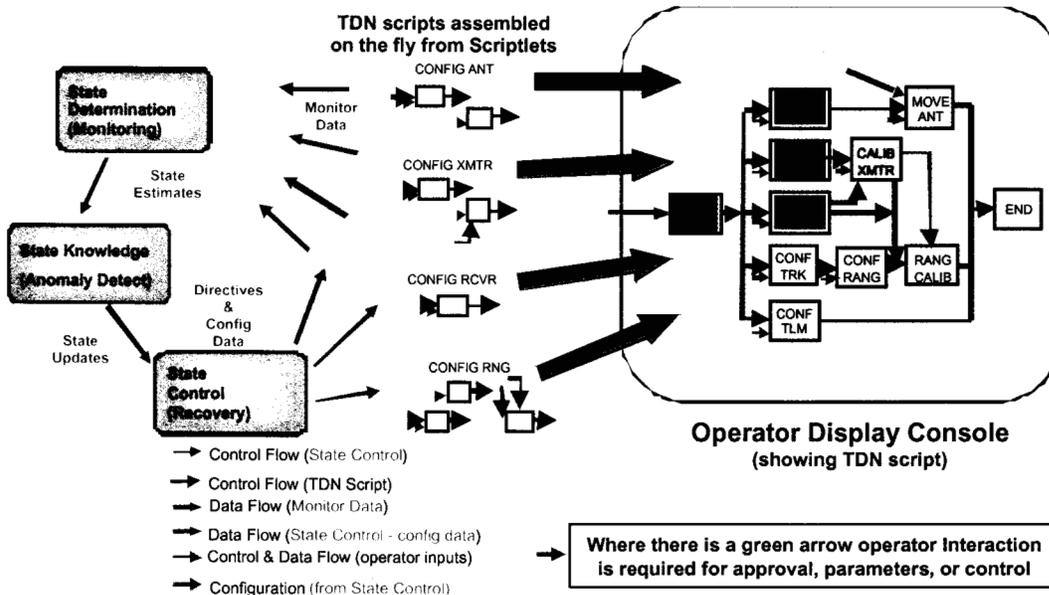


Figure 4.5-2 Automation in Service Execution: Dynamic Scripts Scheme



4.6 Fault Detection, Isolation, and Recovery (FDIR)

The current control system detects anomaly occurrences in the application subsystems based on the monitor data supplied by the subsystems. Determination of an anomalous condition relies on a set of threshold criteria preset by the operator. It requires operators to respond to a failure and take prescribed actions for recovery. The manual approach to FDIR is based on our belief that reliability takes precedence over automation. Under budget constraint, building a reliable system is far more important than automating fault detection, isolation, and recovery (FDIR). However, as mission demands increase and spacecraft for deep space missions become more intelligent, the DSMS control system design will have to be optimized to solve the following problems: (1) Human in the loop for FDIR is error prone and is known to suffer from unrecoverable science data loss. (2) It is a reactive measure correcting problems after the fact. Trending to prevent fault occurrences is needed. (3) It reflects a lack of systematic decomposition of functionality in FDIR.

Therefore, as shown in Figure 4.5-2, the new control architecture provides a framework for FDIR implementation: (1) Control engine responds to identified faults and uses dynamic scripts driven by extensible knowledge base. (2) FDIR use a combination of predictive and heuristic models. (3) Fault recovery is integrated with control elements and uses extensible knowledge base to select appropriate recovery procedures based on system state and required operational state. (4) The application subsystems, where many of the actions take place, will be more intelligent possessing certain built-in knowledge for internal fault detection and local recovery.

4.7 Visibility and Service Accountability

The current control system provides DSMS operators, engineers, and flight project customers visibility into the system via monitor data. Monitor data are in the form of detailed parameters (e.g. carrier SNR, Doppler residual, two-way coherence status, system noise temperature, etc.), associated with individual components of equipment at DSCC at any given time. They can be displayed in real-time allowing operators to obtain a snapshot of the status and behavior of the various subsystems. Interpreting many of these parameters requires detailed knowledge in not only the telecommunication link but also the equipment at DSCC, including their idiosyncrasies. Furthermore, to correctly determine the state of service execution, it is often necessary to correlate multiple types of parameters in the temporal domain. Thus, monitor data are of limited usefulness to operations and are not a cost-effective means to providing visibility into system state. The DSMS, as a service providing system, has to be performance accountable to flight project customers. For each instance of service or a tracking pass it provides, it must present its knowledge in data quantity, quality, continuity, and latency (QQCL), in an accountability report to flight projects. The current control system along with the application subsystems, e.g. telemetry subsystems, does not possess such knowledge, hence relies

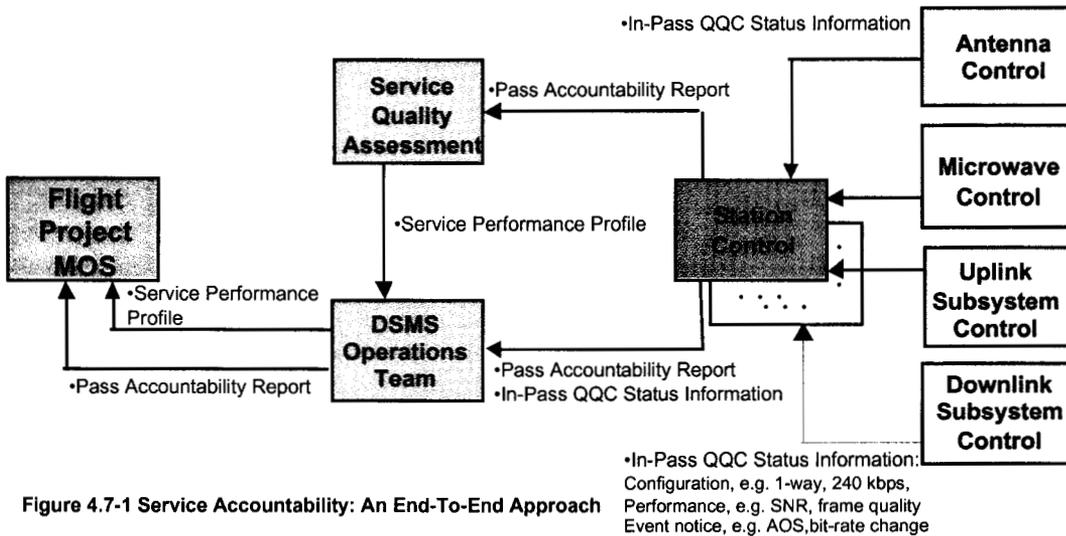


Figure 4.7-1 Service Accountability: An End-To-End Approach

heavily on customers to inform the DSMS of its own performance.

In the future control architecture, the service accountability data will be produced in real-time during the pass by the application subsystems. The in-pass service accountability data will be available to the DSMS control elements or operators to perform necessary actions in closed-loop control fashion. It will be accessible to flight project MOS for their visibility into service execution. A post-pass service accountability report will be generated to summarize the QQCL information for each service instance. It will be available to both DSMS operations and project MOS for accountability reporting. A new element, the Service Quality Assessment (SQA), of the control system will compile service accountability reports using monitor data, anomaly reports, operations logs, and spacecraft events, etc. to produce an updated performance profile and to analyze the causes of any data loss. Monitor data

will cease to be directly available to flight project MOS, but will continue to be used for DSMS internal purpose. Figure 4.7-1 depicts the end-to-end service accountability as the inherent attribute of the overall service quality assessment function.

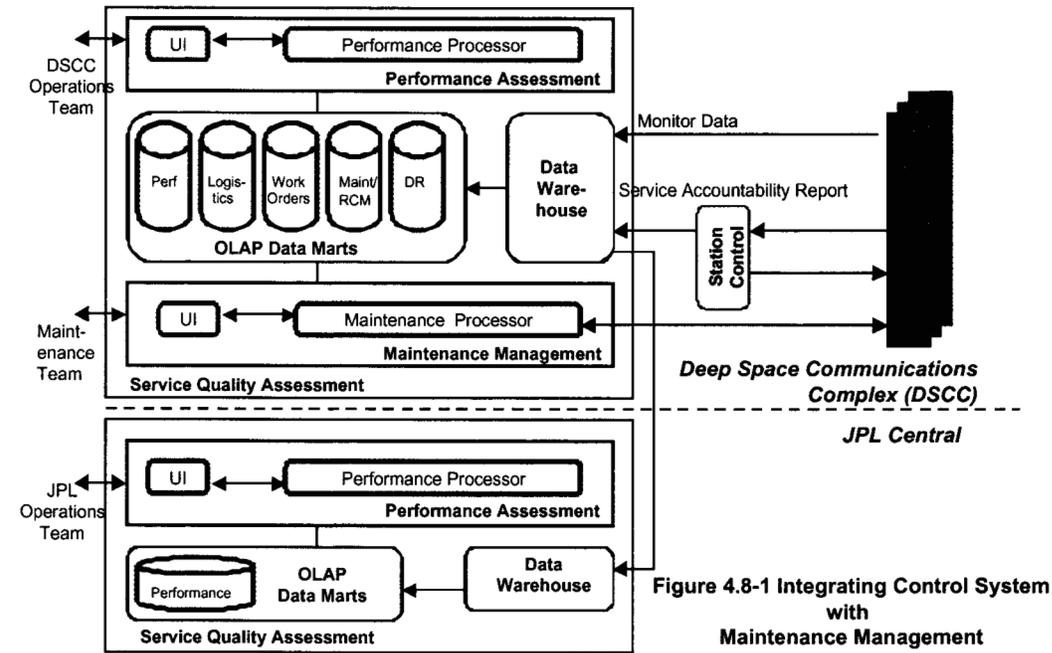


Figure 4.8-1 Integrating Control System with Maintenance Management

4.8 Relationship between Control System and Maintenance Management

At present, the connection between the control system and maintenance management is through discrepancy reports (DR) generated by the operations personnel. The maintenance management relies on the Reliability Centered Maintenance (RCM) tool to track the reliability of the equipment, determine maintenance needs, and provide an up-to-date maintenance database. Following up the DRs, opening a service record on a failed equipment, tracking the repair status and closing the DRs are all discrete activities performed by the maintenance personnel at the DSCC manually. We see two fundamental limitations created by the current control architecture in the maintenance management approach: (1) The DR-driven maintenance essentially reacts to equipment failures. It fixes problems after the fact. If the control system includes certain service performance analysis capability, then performance degradation and failures can be anticipated, thus allowing diagnostics and maintenance activities be initiated in a proactive fashion. (2) The lack of integrated data bases for the DRs, work orders, logistics, maintenance/RCM, and performance data results in a less efficient and less cost effective maintenance management process.

The new control architecture includes a new control element, the Service Quality Assessment (SQA), that will provide insight into the performance of the DSMS, track performance data, and predict future performance including anticipated failure. Thus, the control system can drive maintenance management to automatically initiate diagnostic action on potentially problematic equipment. In addition, the entire maintenance management becomes an integrated part of the SQA. All data bases mentioned above are integrated through a data warehouse tool. Figure 4.8-1 shows the functional design of the SQA.

4.9 Operator Interface

As shown in Figure 4.1-1, in the current control system there are three control paths from the operations team to the equipment at each DSCC. In this scheme, an operator controls the system by interacting with a complex-level control element, a link-level control element, and the application

subsystems. The operator issues directives to each of these elements and monitors the displays provided by them. Control flows involve passing through monitor data, e.g. status and configuration information, via a publish-&-subscribe mechanism, to the Monitor Data Server. Depending on the intelligence residing in the subsystems, some of the control flows are open loop, i.e. without positive confirmation to a directive. The display flows from all three elements to the operators are not filtered nor summarized. Clearly, the multiplicity of control paths from the operators reflects a control architecture that is centered on the operator's actions and is not based on a formalized control authority design. A formalized control authority design would ensure the optimization in control flows among the operators, complex-level control, link-level control, and application subsystems. That may mean more control authority has to be delegated to the other three elements for better operability and robustness in the control system.

In the new control system information from each of the subsidiary elements is summarized by the controlling element before being made available for display. Thus the station controller takes responsibility for summarizing the status of all of its controlled application subsystems. At the top-level display for the complex, the operator is presented with summary information from each station controller and from the global subsystem controller at the complex. This permits the operations staff to easily monitor the status of the overall complex. Of course, detailed access to all monitor data, and to a suite of analysis and trending tools, is available when necessary.

5. Summary and Conclusion

The future control architecture for the DSMS can be summarized as follows: (1) It is an explicit hierarchical control architecture. (2) It enables a more interactive and streamlined operational process from customer interface for service request to allocating and scheduling DSN resources and to predicts generation for configuring and controlling assets. (3) It increases the degree of automation in service execution. (4) It integrates monitor and control functions with maintenance management in DSN.

6. Acknowledgment

The authors would like to acknowledge the design work, visions, and ideas contributed by Joe Wackley, Gary Spradlin, Tim Pham, and Jay Breidenthal during the period of May 1999 – March 2000.

7. Bibliography

Wallace Tai is the System Engineering Manager for the Telecommunications & Mission Operations Directorate (TMOD) at JPL/Caltech. He joined JPL in 1981, managed data system development tasks, and was the system engineer in the mission operations systems for a few deep space missions including Cassini and Mars Pathfinder. Peter Shames is the Standards Office Manager at JPL/Caltech. He joined JPL in 1991 and was a TMOD Technology Manager, the System Architect for the multi-mission ground system. Previously he did system design for the Hubble Space Telescope Institute. Richard Schell is the Director of System Engineering & Integration for NASA's Consolidated Space Operations Contract. For many years he has managed a few Lockheed-Martin's data system development tasks at the Johnson Space Center.