

# Zero to Integration in Eight Months, The Dawn Ground Data System Engineering Challenge

Lydia P. Dubon\*

*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, City, CA 91109, USA*

The Dawn Project has presented the Ground Data System (GDS) with technical challenges driven by cost and schedule constraints commonly associated with National Aeronautics and Space Administration (NASA) Discovery Projects. The Dawn mission consists of a new and exciting Deep Space partnership among: the Jet Propulsion Laboratory (JPL), responsible for project management and flight operations; Orbital Sciences Corporation (OSC), spacecraft builder and responsible for flight system test and integration; and the University of California, at Los Angeles (UCLA), responsible for science planning and operations. As a cost-capped mission, one of Dawn's implementation strategies is to leverage from both flight and ground heritage. OSC's ground data system is used for flight system test and integration as part of the flight heritage strategy. Mission operations, however, are to be conducted with JPL's ground system. The system engineering challenge of dealing with two heterogeneous ground systems emerged immediately. During the first technical interchange meeting between the JPL's GDS Team and OSC's Flight Software Team, August 2003, the need to integrate the ground system with the flight software was brought to the table. This need was driven by the project's commitment to enable instrument engineering model integration in a spacecraft simulator environment, for both demonstration and risk mitigation purposes, by April 2004. This paper will describe the system engineering approach that was undertaken by JPL's GDS Team in order to meet the technical challenge within a non-negotiable eight-month schedule. Key to the success was adherence to an overall systems engineering process and fundamental systems engineering practices: decomposition of the project request into manageable requirements; definition of a structured yet flexible development process; integration of multiple ground disciplines and experts into a focused team effort; in-process risk management; and aggregation of the intermediate products to an integrated final product. In addition, this paper will highlight the role of lessons learned from the integration experience. The lessons learned from an early GDS deployment have served as the foundation for the design and implementation of the Dawn Ground Data System.

## I. □ Introduction

The NASA Discovery Program funds innovative science investigations designed to push the limits of solar system exploration through low-cost, schedule-challenged projects such as Dawn<sup>5</sup>. Dawn is the ninth Discovery project. Its mission is to travel to the two oldest and most massive asteroids in the solar system, Vesta and Ceres, study their geophysical properties, and thus increase our understanding of the conditions and processes behind the creation of the solar system<sup>1</sup>. To accomplish the mission, the Dawn spacecraft relies on ion propulsion technology to orbit both asteroids. This feat is unique to Dawn. The Dawn implementing organizations are: the Jet Propulsion Laboratory (JPL); Orbital Sciences Corporation (OSC); the German Aerospace Center, Deutsches Zentrum fuer Luft-und Raumfahrt (DLR), and the Max Planck Institute for Solar System Research (MPS); and the Italian Space Agency, Agenzia Spaziale Italiana (ASI). Dawn's payload consists of 2 Framing Cameras provided by DLR and MPS; a Mapping Spectrometer provided by ASI; and a Gamma Ray and Neutron Detector provided by Los Alamos National Laboratory (LANL). Dawn's Principal Investigator is Dr. Christopher T. Russell (UCLA). The Dawn spacecraft is expected to launch June 2007.

---

\* Dawn GDS System Engineer, and Project GDS Engineering Group Supervisor, Integrated Ground Data Systems, 4800 Oak Grove Drive/MS 301-285.

Strategic partnerships are key to achieving low-cost, innovative scientific missions. The above participating organizations formed a new strategic partnership. DLR and ASI are instrument donors. OSC builds the spacecraft, and is responsible for flight system test and integration and for launch operations. The Dawn Science Center (DSC) at UCLA is responsible for science planning and science operations. JPL is responsible for project management. JPL also provides the ion engines and the Ground Data System (GDS) to conduct Deep Space mission operations.

Strategic partnerships also had to be formed at the team level in order to meet project risk-mitigating milestones. One such partnership involves the JPL GDS team and the OSC Flight Software (FSW) team. During the first Technical Information Meeting, August 2003, the GDS team was briefed of the project commitment to provide the instrument teams with a spacecraft simulator environment for early instrument engineering model test and integration, by April 2004. The spacecraft simulator (S/C Sim) environment would be provided by OSC, and should include flight-like GDS interface components provided by JPL. This requirement emerged in response to concerns regarding the use of two different ground systems: OSC's for flight system test and integration and JPL's for flight operations. The lack of an existing interface between these two systems heightened the concerns.

## **II. □ The Dawn GDS System Engineering Challenge**

Dawn GDS system engineering activities started August 2003. These focused on readiness for the Ground Segment (GS) Preliminary Design Review (PDR). The PDR was scheduled for early December 2003, and the first GDS delivery for summer of 2004, per the baseline schedule. A "bare bones" GDS 0.0 was inserted into the schedule to address the S/C Sim commitment to the instrument teams. The Dawn GDS system engineering challenge was to transition from a state of zero design and implementation and zero experience with OSC, to a state of ground integration into OSC's S/C Sim environment, within 8 months. This was to be accomplished within budget and schedule, without jeopardizing PDR readiness and the integrity of the GDS design. The objective of this paper is to describe the GDS systems engineering approach and process, which enabled the GDS team to successfully meet the integration challenge.<sup>†</sup>

## **III. □ The Dawn GDS System Engineering Approach**

The Dawn Ground Data System inherits JPL's multi-mission ground system, which is known as the Advanced Multi-Mission Operations System (AMMOS). The Dawn GDS is a Software Intensive System (SIS), which requires software system engineering to be integrated into the overall system engineering process. The integration of these two disciplines is a trend in Software Intensive Systems<sup>2</sup>. The flavor of this integration was key to the success of GDS 0.0. The GDS Systems Engineering approach provided a structured development framework which was used to guide, monitor and control the Agile<sup>6</sup> software engineering efforts that were adopted to develop critical GDS 0.0 software components. Three medium-scale critical software engineering tasks were identified as candidates for Agile-type approach: implementation of a new Dawn command generation toolkit; implementation of the command and telemetry database translation suite, which translates OSC's command and telemetry database specifications into corresponding AMMOS command and telemetry database products; and implementation of a new tool referred to as Test Conductor Assistant with the initial purpose of flight and ground software integration and test. Agile methods are best known for their use by projects with uncertain requirements<sup>6</sup>. In the GDS 0.0 case, the adherence to fundamental systems engineering practices mitigated the uncertainty of requirements. The Agile emphasis was on the skill level of the development teams and on their interactions. This proved to be a successful approach by emphasizing the discipline of process at the GDS System Engineering level and using it to guide the engineering design and development. Figure 1 depicts the overarching GDS System Engineering process. This is the same process that was in place as part of the roadmap to the Dawn Ground Segment PDR. It was important that the PDR not be jeopardized and that the GDS 0.0 approach be aligned with the overall GDS design effort.

---

<sup>†</sup> It should be noted that other Systems Engineering domains such as Payload System Engineering and Mission Operations System Engineering are key contributors to the overall success of the project. The intent of this paper, however, is to focus on the GDS domain.

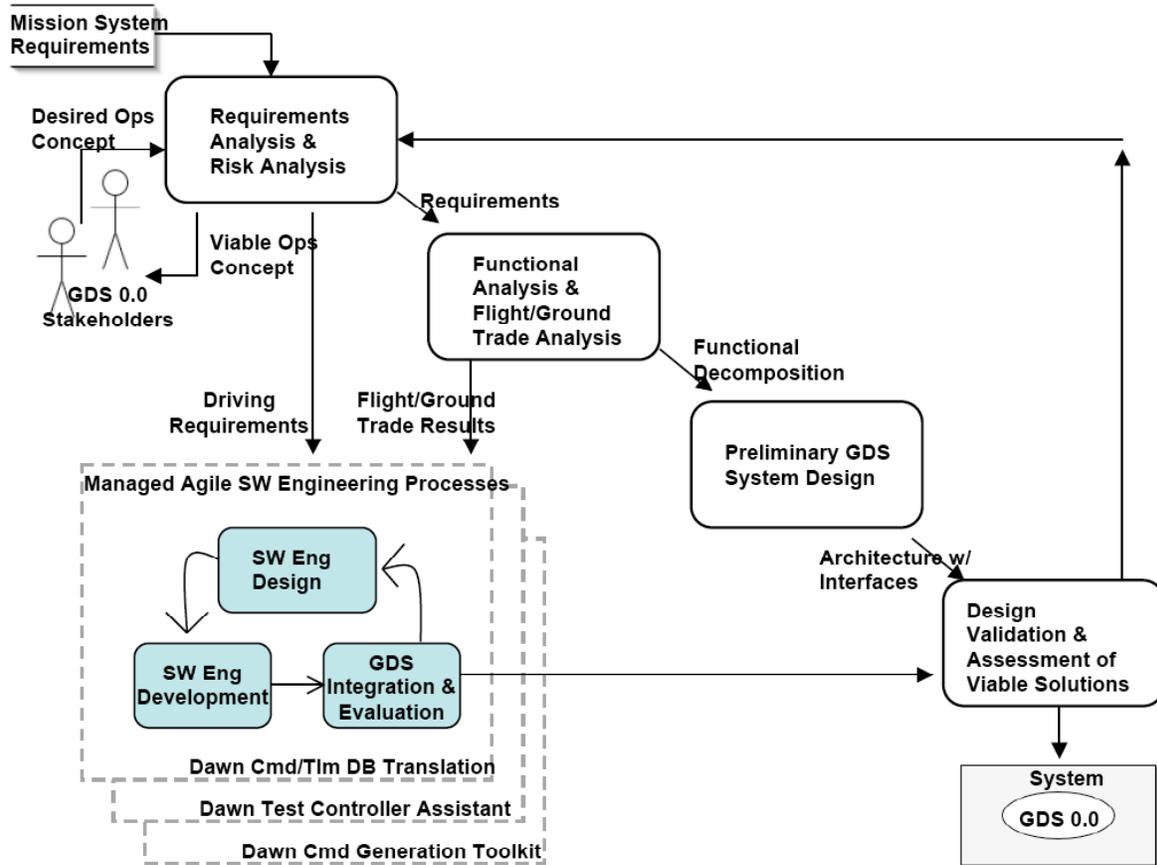
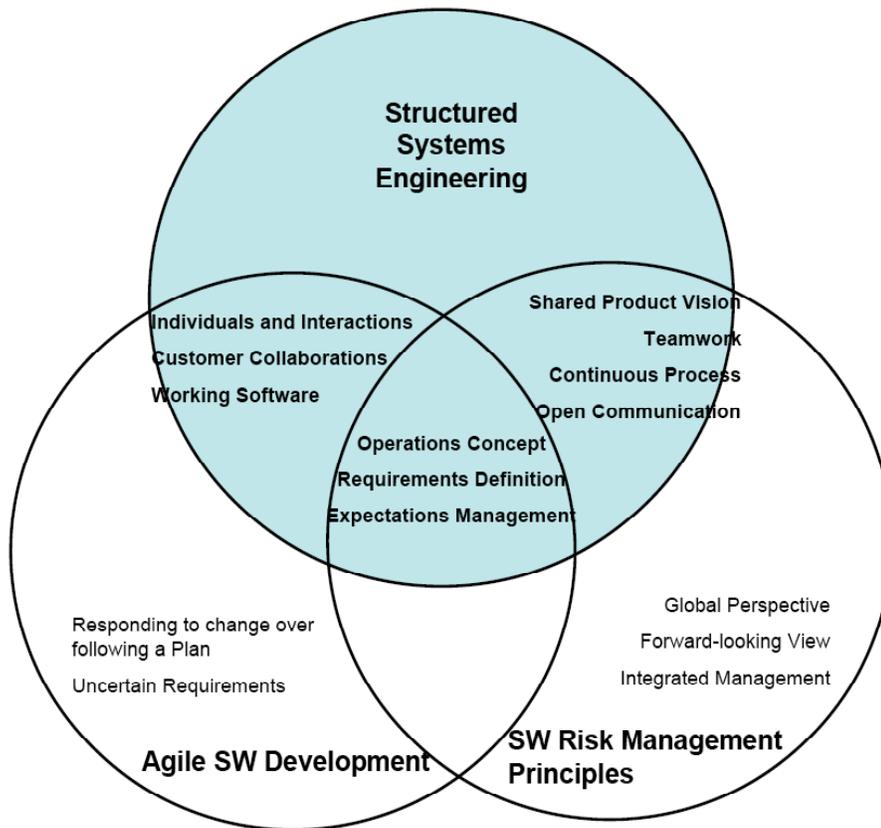


Figure 1. Dawn GDS System Engineering Process

A formal and disciplined systems engineering approach enabled the definition of and adherence to a risk management plan. A simple operations concept and a firm set of driving requirements were key to managing stakeholders' expectations, and thus bounding the GDS 0.0 solution to fit within a fixed schedule. The process<sup>4</sup> depicted in figure 1 was also important in the management of interfaces. A formal GDS 0.0 system concept resulted in early understanding of both internal and external interfaces. The Dawn DGS System Engineering process facilitated the coordination and management of interfaces during GDS 0.0 development, which contributed to system stability and integrity.

### A. The Structured GDS System Engineering Approach to Development

An important component of the systems engineering process is the set of practices which, at the aggregated system level, provided an overall structured development approach. This paper demonstrates that this was the correct methodology in response to a non-negotiable schedule and no room for failure. The key practices were: integration of multiple disciplines and experts into an Agile team effort; decomposition of the S/C Sim integration project requirement into a manageable set of key functional requirements aligned with GDS system driving requirements; in-process risk management plan; and aggregation of the intermediate Agile products into an integrated final product. The structured GDS System Engineering method provided a common ground where both Agile principles and Software Risk Management principles<sup>3</sup> came together to meet the GDS 0.0 challenge. This is depicted in Figure 2.



**Figure 2. Structured Systems Engineering Development Approach**

The practices, software development strategy and risk strategy emphasized in Figure 2 were possible because of the caliber of the team that was assembled. The Structured Systems Engineering approach also enabled software developers to see the GDS system as more than a sum of the parts. It kept the teams focused on a common GDS vision, which contributed to the system-level integrity during development. The risk strategy focused on the following principles<sup>3</sup>: Shared Product Vision, Teamwork, Continuous Process and Open Communication. These aligned very well with following Agile development principles: Individuals and interactions over processes and tools; customer collaboration over contract negotiation; and working software as a measure of progress.

*1. Integration of Multiple Disciplines and Experts into a GDS Team Effort*

The emphasis on highly skilled individuals and their team interactions led to the successful development of the three critical software tasks: Dawn Command (Cmd) and Telemetry (Tlm) Database Translation toolkit; Dawn Command Generation Toolkit; and Dawn Test Controller Assistant (TCAssist) Tool.

- Dawn Command and Telemetry Database Translation Toolkit

Two JPL domain experts were recruited: the authority on the AMMOS multi-mission telemetry standard; and the authority on the AMMOS Command Definition Language (CDL). In addition, the Mars Exploration Rover dictionary software architect and JPL Agile method practitioner, was enlisted to develop the Dawn Telemetry and

Command Database Translation and Management software. These three experts worked side-by-side for approximately six months in the quest to automate the translation of the OSC command and telemetry databases into their AMMOS counterparts, which were then integrated into the Dawn GDS. As an example of the Agile nature of the teams and work, initially, the CDL expert manually generated a Dawn CDL which was needed as input to the Dawn Command Generation toolkit development effort. Both the CDL expert and the GDS software architect analyzed the OSC command specification and database Interface Control Document and defined algorithms that led to full automation of the command database translation. The structured GDS system engineering process guided these team activities by managing requirements and controlling the risk.

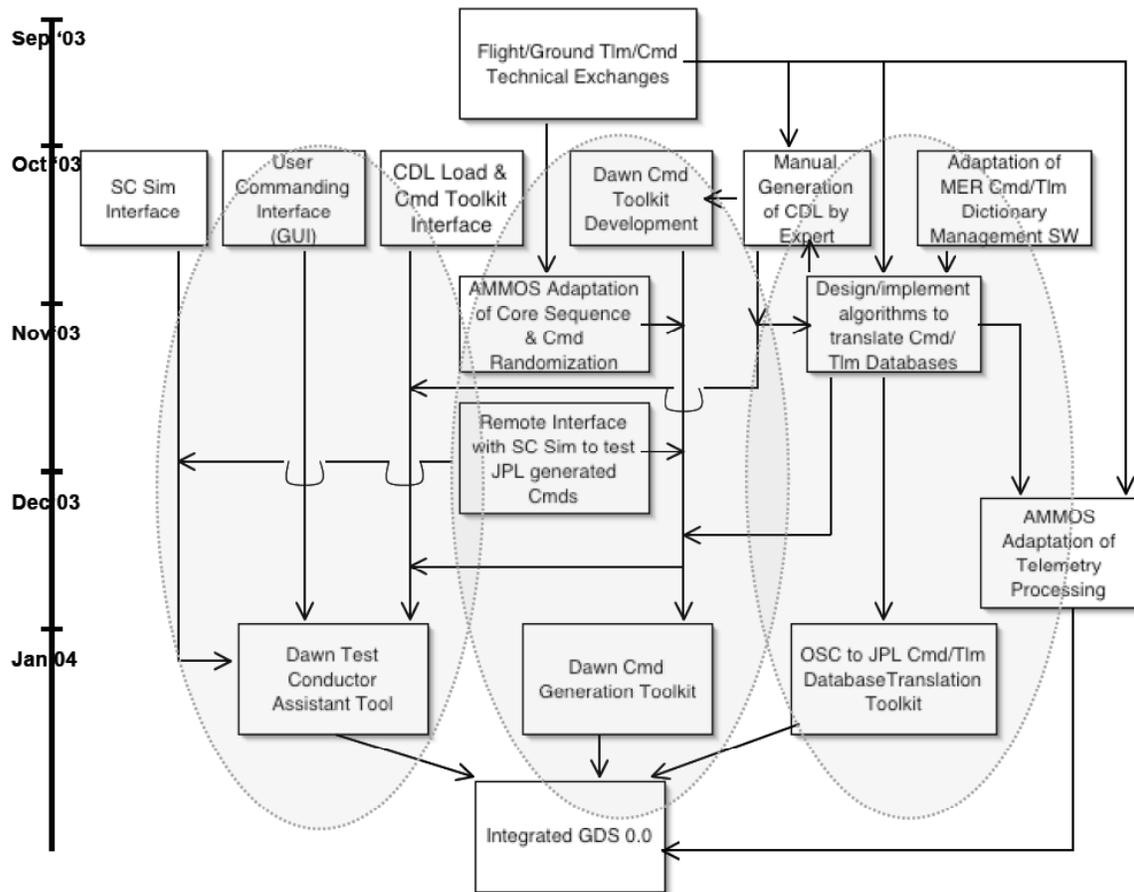
- Dawn Command Generation Toolkit

An experienced flight project sequence adaptation developer led the Cmd Generation Toolkit effort. He immediately partnered with the command side of the database translation software effort. The first GDS 0.0 intermediate milestone was to have the OSC S/C Sim successfully process commands generated by the JPL Cmd Generation Toolkit. This required thorough understanding of Dawn's command packet format; remote connectivity to an OSC S/C Sim. These were non-existent as of September 2003. Manually generated CDLs from the database translation effort were first used by this effort to iterate through the Dawn command packet generation process.

- Dawn Test Controller Assistant Toolkit

The TCAssist effort required integration of the new Dawn Cmd Generation Toolkit, and an early version of the Dawn CDL in order to design and implement the capability to dynamically generate Dawn commands. This was a driving requirement from GDS 0.0 stakeholders. An experienced flight/ground software integration engineer led the TCAssist effort. An underlying goal of the TCAssist effort was to have the OSC FSW team embrace AMMOS, which they had not been exposed to before. The TCAssist lead was instrumental in making this happen.

Figure 3 depicts the activities and dependencies among the three Agile software development efforts. As the development efforts gained momentum, the need for an overarching structured system engineering approach became even more evident if the schedule was to be met with an integrated, working GDS. There was no room for creeping requirements nor for uncontrolled risks. System-level schedule and milestones were common to all the parallel development efforts.



**Figure 3. Parallel Agile Software Development Efforts Leveraging from Team Expertise**

*2. Decomposition of the Project S/C Sim Integration Request into Functional Requirements Aligned with System Driving Requirements*

Early on, a concise set of verifiable uplink and downlink requirements were defined for the Agile teams.

On the uplink side:

- Provide GUI that provides for selection of instrument commands and parameters
- Generate flight-like AMMOS command files containing instrument commands
- Extract Command Link Transmission Units (CLTUs) from command files and sent to S/C Simulator

On the downlink side:

- Process frames from S/C Sim containing instrument packets
- Packetization and instrument telemetry channel extraction
- Display of telemetry channels defined in the instrument telemetry database

GDS System Engineering determined that the above “bare bones” capabilities would meet the S/C Sim integration goal within schedule and cost. At first glance the above requirements appear straightforward. However, the above requirements had to be aligned with the GDS system-level driving requirements (1, 2, 4, 7) in Table 1. This was necessary in order to conform to the overall systems engineering process, which was working towards PDR.

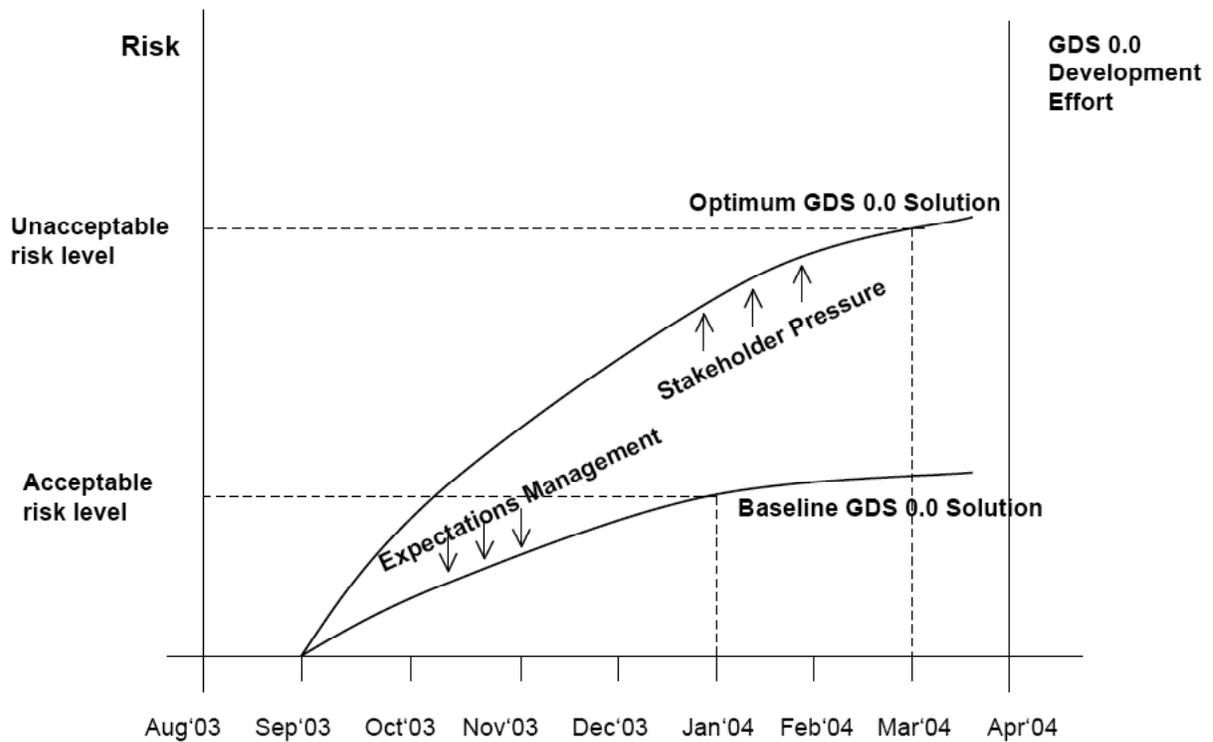
<b>1. Advanced Orbiting System (AOS) Frame with Turbo Encoding</b>
<b>2. Translation of OSC Command and Telemetry Dictionaries</b>
3. Delivery of Virtual Machine Language (VML) C structures to FSW
<b>4. Support CCSDS Standards which are part of OSC's heritage architecture</b>
5. Support Instrument and Spacecraft file loads without on-board file system
6. Support Distributed GDS Architecture (JPL, Orbital and UCLA)
<b>7. Support ITAR view of Ground/Mission data for foreign Instrument Teams</b>

Table 1. Dawn GDS Driving Requirements

As depicted in Figure 2, the overall system-level structured GDS system engineering approach aligned risk management principles with the need for an Agile software development approach to meet the April '04 deadline. Managing stakeholders' expectations was necessary to adhere to the above set of requirements, and key to the adopted risk management strategy.

### 3. Risk Management

The Risk Management strategy consisted of the following: 1) Definition of the risk as the uncertainty of not meeting a key project milestone due to the "zero" implementation state of the ground system; 2) Continuous assessment of this risk during the GDS 0.0 development cycle; 3) Management of expectations in order to control the understanding of the problem to be solved by the development teams, and to avoid uncertainty of requirements; and 4) Focus on team-oriented software risk management principles as depicted in Figure 2. The overall risk management strategy is captured in Figure 4.



**Figure 2. Risk Management Strategy: Adhere to Baseline Solution through Expectations Management**

Firm management of expectations was necessary to counteract stakeholder pressure for more capabilities within the non-negotiable schedule. The GDS System Engineering team determined that the solution desired by GDS 0.0 stakeholders (FSW and Payload teams) presented an unacceptable level of risk. Analysis of the development trade space concluded that baseline capabilities should be implemented by Jan '04, and allow for GDS 0.0 system level integration and test to begin.

The risk was assessed continuously. The following milestones were included in the schedule as necessary to mitigate the risk:

- Operational remote socket interface between JPL Dawn Command Generation toolkit development environment and OSC S/C Simulator Environment November 2003.
- Delivery of Dawn Telemetry Input Subsystem by multi-mission AMMOS team December 2003.
- Commanding of S/C Sim using Dawn Command Generation Toolkit January 2004.
- GDS Telemetry processing of S/C Sim downlink data January 2004.
- Test Conductor Assistant able to accept commands and parameters and generate command files dynamically by January 2004.
- GDS hardware and software deployment to OSC March 2004.
- On-site GDS training for OSC FSW team March 2004.
- On-site GDS support for Instrument Team S/C Sim Training April 2004.

As each milestone was reached, comprehensive tests were executed to ensure that any software re-factoring done as a result of the Agile mode the teams were in, did not jeopardize the integrity of the system interfaces and system capabilities.

#### 4. Aggregation of the Intermediate Products into a Final GDS 0.0 Solution

The formal systems engineering approach used to plan, track and control all GDS 0.0 fast-paced and aggressive development activities was effective in maintaining system-level integrity when all efforts and products were aggregated to a final GDS 0.0 solution. By managing expectations, requirements, and risk through a disciplined GDS System Engineering approach, a simple yet complete GDS 0.0 solution was designed, implemented and deployed to OSC in time for the April '04 instrument team S/C Simulator training. OSC FSW and JPL ground software integration was achieved in zero to eight months. Figure 5 depicts the final GDS 0.0 solution, which successfully met project stakeholders needs and, in the end, their expectations.

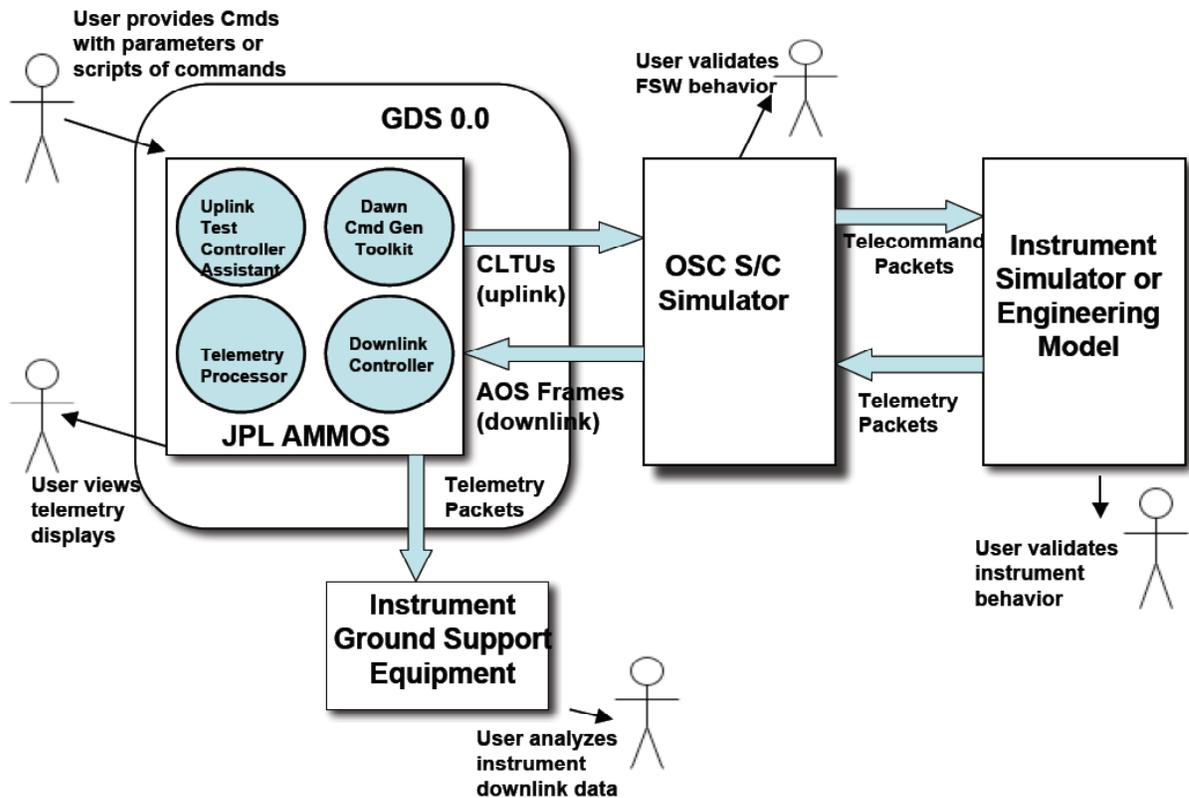


Figure 5. GDS 0.0 Final Solution to the Integration Challenge

#### IV. □ The Dawn Ground Data System Lessons Learned

The importance of a flexible systems engineering process was a key lesson learned. The GDS system engineering approach in support of GDS 0.0 focused on a key set of practices, which were not only streamlined to meet the schedule and budget, but also to capitalize upon the expert and skilled nature of GDS 0.0 contributors. By maintaining a system-level focus, the overall systems engineering process unified team members with a common goal: the success of the ground data system as a whole and not just the success of their individual expert contributions. Incorporation of Agile-type development efforts was aligned with a risk management strategy based on team-oriented principles, thus achieving a more stable baseline solution.

GDS 0.0 also encapsulates technical lessons learned. These have been fed into the Dawn GDS design and delivery process. The GDS preliminary design, and flight/ground telemetry and command trade-offs were validated with GDS 0.0. The automation of the command and telemetry and database translation served as the foundation for the future automation of the database processes involving both OSC and JPL. With GDS 0.0, the gap between OSC's ground system and JPL's ground system was breached. The gap had been a concern of the Dawn standing review board. In addition, driving requirements played a key role in GDS 0.0. Four out of the seven key driving requirements were addressed in GDS 0.0. This resulted in a very stable GDS foundation to build upon. For example, a GDS 0.0 byproduct is a rigorous command and telemetry regression test suite. Also, of equal importance is the strong technical partnership that emerged between the GDS Team and the FSW Team as a result of GDS 0.0. In conclusion, the Dawn GDS Team met the integration challenge in zero to eight months by using a structured systems engineering approach, which guided and cared for multiple threads of software development efforts with an emphasis on people, skills, and partnerships.

### Acknowledgments

*The GDS work described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The author would like to acknowledge and thank key JPL Dawn GDS 0.0 contributors: Magdy Bareh (Test Conductor Assistant Software Lead; Jesse Wright (Software Architect for Database Translation and Management and Agile Method expert); James Goddard (Dawn Command Generation Toolkit and Sequence Adaptation System Engineer); Dung C. Doan (Command Definition Language expert); Betsy Wilson (AMMOS Telemetry Expert); Carol Glazer (Telemetry, Tracking and Command System Engineer); Eric Tauer (Dawn Project Database Management); Robin O'Brien (ATLO GDS); and Dipak Achhanni (Dawn GDS I&T). The author would also like to acknowledge the Orbital Sciences Dawn Flight Software Team, led by Dave Termohlen; and the systems engineering support provided by the Dawn Ground Segment Manager, Ray Morris, the Mission Operations System Engineer, Kathryn Schimmels, the Science Operations System Engineer, Carol Polansky, and by the Payload System Engineer, Betina Pavri.*

### References

#### *Proceedings*

<sup>1</sup>Rayman, M. D., "Dawn: A Mission in Development for Exploration of Main Belt Asteroids Vesta and Ceres," *55<sup>th</sup> Astronautical Congress*, 4-8 October 2004, Vancouver, Canada, Paper IAC-04-Q.5.05

#### *Journal*

<sup>2</sup>Boehm, B. "Some Future Trends and Implications for Systems and Software Engineering Processes," *The Journal of The International Council on Systems Engineering*, Vol. 9, Number 1 Spring 2006, pp. 1-19.

#### *Books*

<sup>4</sup>Kossiakoff, A., and Sweet, W. N., *Systems Engineering Principles and Practice*, John Wiley & Sons, Hoboken, New Jersey, 2003, Chaps. 3, 7-8.

#### *Electronic Publications*

<sup>3</sup>Higuera, R. P., Haimes, Y. Y., "Software Risk Management," CMU/SEI-96-TR-012, June 1996. URL: <http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.012.htm> [cited 12 February 2005]

<sup>5</sup>NASA Discovery Program, URL: <http://discovery.nasa.gov/dawn.html> [cited 28 March 2006].

<sup>6</sup>Manifesto for Agile Software Development, URL: <http://www.agilemanifesto.org/> [cited 28 March 2006].