TRADEOFFS IN FUNCTIONAL ALLOCATION BETWEEN SPACECRAFT AUTONOMY AND GROUND OPERATIONS: THE NEAR (NEAR EARTH ASTEROID RENDEZVOUS) EXPERIENCE

S. C. Lee and A. G. Santo
Johns Hopkins University Applied Physics Laboratory,
Johns Hopkins Rd.,
Laurel, Maryland 20723

Abstract

Today's modern spacecraft often fly a computing power equivalent, or nearly equivalent, to the computing power available to the ground operations team. This enables the spacecraft to perform many functions autonomously that previously could only be planned and carried out from the ground. In some cases, this increased computing power is required to perform functions that must be carried out on the spacecraft. For example, fault detection and correction must be carried out on the spacecraft when the time scale of critical faults is shorter than the time between ground contacts. In many given function. In implementing these functions, tradeoffs between ground operations and spacecraft autonomy must be considered.

With a maximum time of 12 days out of ground contact and a round-trip light time as high as 56 minutes, NEAR requires a moderate degree of onboard autonomy to react to faults and safe the spacecraft. Beyond the basic safing requirements, many additional functions can be carried out onboard. For example, momentum management, center-of-mass management during velocity change maneuvers, and optical navigation are all functions considered for onboard autonomy on NEAR. The allocation of these functions to onboard software or to ground operations involves tradeoffs such as development time for onboard software versus ground software, uplink/downlink bandwidth, resource management, life cycle costs, and spacecraft safety.

I. Background

The NEAR mission takes advantage of an unusual opportunity that occurs only once every seven years to reach to the Near-Earth asteroid 433 Eros. It will make the first comprehensive, spatially resolved measurements of the geology, mineralogy, and elemental composition of an asteroid.

Mission Design

The spacecraft launched from the Eastern Test Range in Cape Canaveral, Florida in the late afternoon on February 17, 1996. This was the second day of the sixteen day launch window. Figure 1 shows an ecliptic plane view of the mission trajectory[1]. During the 36-month cruise to Eros, the spacecraft passes within 1200 kilometers of the main belt asteroid 253 Mathilde, on June 27, 1997. The Mathilde encounter will produce the first close up images of a C-class asteroid.

One week after the Mathilde encounter, a large trajectory correction maneuver is executed to target an Earth swingby. This will be the first use of the bipropellant system. When the spacecraft passes close to the Earth in early 1998, the Earth's gravity bends the trajectory to match Eros's orbital plane. This sets up the optimal geometry needed for the slow approach to Eros in 1999. At that time, several rendezvous maneuvers insert the spacecraft into orbit around Eros. This allows intensive study of the asteroid for up to a year.

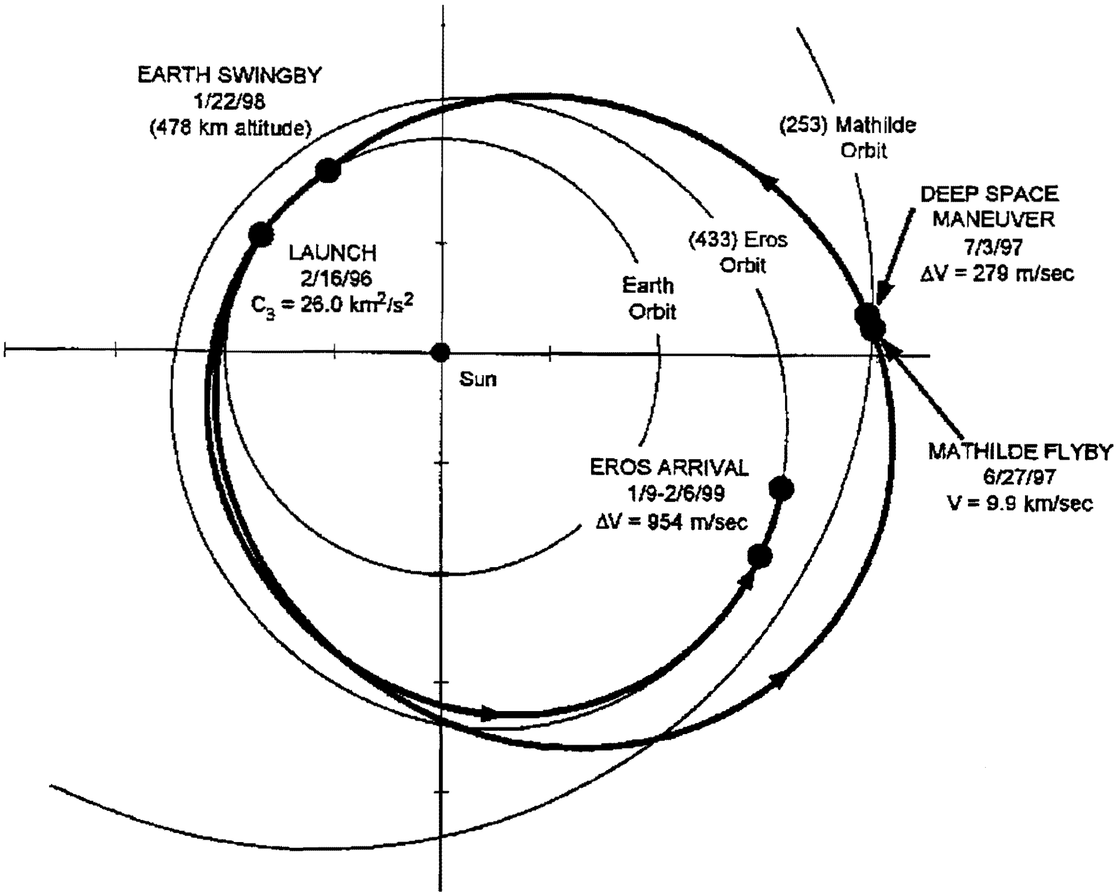

Figure 1 Mission Trajectory

Spacecraft Design

Figure 2 shows the spacecraft in the deployed flight configuration[2]. A 1.5 m high gain antenna (HGA) and four solar panels are mounted on the outside of the forward deck. Most electronics are mounted on the forward and aft decks. The science instruments, except for the magnetometer, are hard-mounted on the outside of the aft deck with co-aligned fields- of-view. The magnetometer is mounted on the HGA feed. The interior of the spacecraft contains the propulsion module. The solar panels, the HGA, and the instruments are all fixed in place. The entire spacecraft must be rotated to point various components, such as the HGA at Earth or the solar panels at the Sun. The spacecraft is designed with a distributed architecture where subsystems do not share common hardware.

The telecommunication subsystem is an X-band system capable of simultaneously transmitting telemetry data, receiving spacecraft commands and providing a frequency coherent ranging capability. Besides the HGA, a medium gain fanbeam antenna is mounted on the forward deck, and two low gain hemispherical antennas are mounted to the forward and aft decks. Redundant, unswitched transponder/command detector unit (CDU) pairs are connected to the antennas through a coaxial switching network, allowing two separate command reception paths at independent data rates. Two data rates are supported: a 7.8 bps rate used for emergency communications and a 125 bps rate is for normal communications. The redundant telemetry conditioning units (TCUs) are cross-strapped to the transponder exciters. Eight downlink data rates are used from 9.9 bps to 26.5 kbps. The selected rate is a function of the downlink coding scheme, the NASA Deep Space Network asset, the spacecraft to earth distance, and the Sun Spacecraft-Earth geometry. The 9.9 bps rate is used for emergency recovery.

The power system comprises four gallium arsenide solar panels, a 9 amp-hour super NiCad battery, and the power system electronics. The spacecraft bus is regulated at 33.5 +/-0.5V when the solar array power is adequate to supply the load and battery charge power. The bus follows the battery voltage whenever the battery is in discharge.

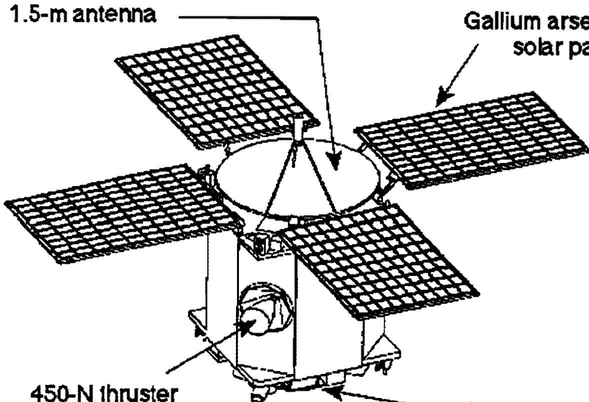

Figure 2 Flight Configuration

The interior of the spacecraft contains the dual mode propulsion module. The propulsion module contains the three propellant and two oxidizer tanks, 11 hydrazine monopropellant thrusters grouped into six different pods, and the 450N bipropellant thruster. The location of the propulsion tanks is selected to maintain the spacecraft's center-of-mass along the thrust vector of the 450N thruster throughout the mission as the bipropellant is depleted.

The Command and Data Handling (C&DH) subsystem comprises redundant command and telemetry processors, redundant solid state recorders, a power switching unit to control spacecraft relays, and an interface to a redundant 1553 standard bus for communicating with other processor-controlled subsystems. The redundant components are cross-strapped among themselves, and among the redundant uplink chains of the telecommunications subsystem. The functions provided by the C&DH subsystem are command management, telemetry management, and autonomous operations.

The Guidance and Control (G&C) subsystem is composed of a suite of sensors for attitude determination, actuators for attitude corrections, and processors to provide continuous, closed loop attitude control. In operational mode, the attitude is controlled to a commanded pointing scenario. In safe modes, the G&C maintains the solar panels pointed to the Sun for maximum power, and attempts to place the Earth within the medium-gain antenna pattern to establish ground communications.

The facility instruments carried on the spacecraft are a visible-light imager, an IR spectrograph, a 3 axis magnetometer, an x-ray/gamma-ray spectrometer, and a laser rangefinder.

II. Spacecraft Autonomy and Mission Operations Tradeoffs

There are a number of issues to be considered in trading off between spacecraft autonomy and ground operations. The first issue is time criticality of operational and safing events. In some instances, ground control of events is not possible, because of round-trip-light-time (RTLT) and ground contact frequency considerations. For example, an asteroid flyby may require autonomous closed-loop pointing, if the uncertainty in the asteroid location does not allowed a pre-planned sequence and the RTLT does not allow ground intervention. For safing, both the RTLT and the frequency of ground contact enter into the tradeoff. Even if a safing event occurs during a pass, the damage may be done before the RTLT allows the ground to learn of and correct the failure. Further, failures that occur between ground contacts must be handled autonomously if the spacecraft cannot survive with the failure for the longest possible time between contacts. For example, if a battery goes into unexpected discharge, the battery may be depleted before the next ground contact when Mission Operations could correct the source of the power drain.

In-flight performance is another tradeoff area. Often, on-board software can increase spacecraft performance and reduce required spacecraft resources. For example, using on-board accelerometers and gyroscopes for closed-loop control of a change in velocity maneuver (deltaV) can increase burn efficiency and reduce the required amount of on-board propellant. If the spacecraft mass margins are sufficient, however, ground-calculated open-loop deltaV's reduces the complexity of the on-board software.

Increasing the complexity of the on-board software increases both the development time and the chances that a bug exists that may risk the mission. On the other hand, human error is often a source of failure. The tradeoff between these two sources of risk must be considered in allocating functions between spacecraft autonomy and mission operations. For a mission critical risk, measures may be implemented both on the ground and on the spacecraft to check for a fault. To assess the relative risk of human error versus a flight software bug, many factors must be taken into account. The maturity of the software development process, the flight software team experience, the mission operations team experience, and the complexity of the planned operations all must be considered.

Schedule risk can also be an important driver of tradeoffs between spacecraft autonomy and mission operations. The spacecraft development is driven by a launch schedule. Functions that can be done on either the spacecraft or the ground and are not needed until late in the mission cannot be allowed to drive the spacecraft development schedule. For an interplanetary launch, there may be a very restricted launch window, but a long (years) cruise period before a science encounter. During that cruise period, ground software and procedure development can continue, but the flight software generally must be completed before launch.

Resource management is a fourth issue when deciding tradeoffs. Personnel resources are seldom unlimited. Often, the talents of certain key personnel are needed in several areas (both within and among programs). Many tradeoffs are decided on the availability of a key personnel resource. For example, if the flight software development is behind schedule, and additional personnel with the right qualifications cannot be added, then a flight software function, such command syntax checking, could be moved to the ground software. The planned quality of personnel is an additional consideration. If the Mission Operations team is to consist primarily of individuals with little technical knowledge, then more automation and correctness checking on the spacecraft allows that team to carry out operations.

Other resources may also enter the equation. For example, computer resources are another area that can force a tradeoff. Only limited types of processors are available for flight use, and their number on-board may be limited as well. At some point, the computer resources on-board may not be able to handle all the desired tasks, and some may have to be placed on the ground.

A fifth important tradeoff is uplink and downlink bandwidth. Many functions can be moved to the spacecraft that are done on ground computers, the only benefit being a reduced uplink or downlink bandwidth. For example, by flying a command sequence generator, the uplink bandwidth requirement is reduced. On-board lossless data compression can save downlink without sacrificing any information.

Finally, mission life cycle costs must be considered in making on-board autonomy versus ground operations tradeoffs. The question is: where can a requirement be met with the least cost? The cost of developing and testing flight software is considerable, when appropriate quality assurance procedures are followed. The cost of developing ground software or procedures may be less, but the recurring cost of ground execution over the mission lifetime may exceed the flight software development costs. For example, the more ground contacts that are planned and paid for every day, the less on-board autonomy is required. In addition, tradeoffs may be made to minimize organizational costs over multiple missions. For example, an organization might chose to implement a standard spacecraft command language. This would move software traditionally done on the ground to the spacecraft with an additional cost to the first spacecraft, but with potentially lower costs for future programs.

III. NEAR Tradeoffs

With a 27-month development time and a fixed 16-day launch window, schedule was the primary driver of NEAR spacecraft design decisions. This consideration drove the development team to minimize the requirements for the on-board autonomy. Only when another consideration, such as time criticality, performance or risk overrode the schedule was on-board autonomy used.

An example of an operational tradeoff driven by time criticality is an autonomous algorithm to detect depletion of the bipropellant oxydizer. To meet mass margins, NEAR planned no margin in the amount of oxydizer it carried. The spacecraft has two oxydizer tanks. Because oxydizer use cannot be completely equalized between the tanks, two depletion burns were anticipated where one tank would be emptied at a time. If a bi-propellant burn is carried on more than 6 seconds after oxydizer depletion, the thruster is damaged. At the time of the planned oxydizer depletion burns, the RTLT is 40 minutes. To assure that the engine survives the first depletion burn, an on-board algorithm is needed that detects oxydizer depletion and shuts down the burn within 6 seconds. With the extra 6 kg of fuel allowed by the final spacecraft dry mass and the extremely low Delta launch errors, NEAR may have sufficient oxydizer margin that this algorithm will not be used.

The majority of the on-board safing is also driven by time criticality and time between ground contacts[3]. A critical aspect of spacecraft safety is power. Because the spacecraft can operate on battery power only for a short period (<2 hours), keeping the fixed solar panels pointed towards the sun and the spacecraft load below the solar panel output is the first priority of the safing design. The NEAR mission design includes one 12-day, no-contact passage behind the Sun, so the safing was designed for spacecraft survival over a 12-day unattended period. Any serious fault affecting a critical spacecraft subsystem results in remedial action (such as turning off a primary system and bringing a backup on-line) and entry into one of the safe modes. All safe modes cause the spacecraft to point the solar panels at the Sun. Table 1 lists events that are detected by the on-board safing algorithms and result in safe mode.

Table 1
Safe Mode Events

G&C Computer Reset
Switch to Backup G&C Computer
DeltaV Abort
Autonomous Thruster Use
Star Tracker Failure/ No Star ID
Digital Sun Detector Failure
Switch to Backup Gyroscope
Bus Regulator Failure
Battery Charger Failure
Battery Discharge
Battery Over-temperature
Solar Array Lockup
Fuel Tank Overpressure
C&DH 1553 Bus Failure
Loss of Mission Time
Low Voltage Sense (26V)
Bus Voltage<23 V
Command Lost Timeout
Last Resort Timer Reset
Sun Keep-in Violation

While correct operation of the G&C subsystem is time critical, on-board processor resources limited the amount of on-board safing that could be performed for the G&C. The ultimate check on the accuracy of the G&C is to compare expected to actual sensor input for a given control output. The expected sensor input would be calculated by a simulation of the spacecraft dynamics (a truth model). If the difference between the expected and actual inputs fall outside a tolerance, fault correction actions could be initiated. The processor resources on NEAR were insufficient to run an on-board truth model. Checks on the G&C operations were limited to simple reasonableness checks on mode, sun angle limits, and sensor input.

A ground/flight software tradeoff based on performance involved on-board closed-loop deltaV control. One way to achieve a given deltaV is to calculate the correct pointing vector and burn time on the ground. After directing the spacecraft to point correctly, the ground simply commands the needed thrusters on for the burn time. Some NEAR burns can and will be performed in this fashion. During the asteroid phase of the NEAR mission, small, extremely accurate deltaV's must be performed weekly to maintain the spacecraft orbit about the asteroid. The accuracy of these small burns cannot be achieved open-loop from the ground. To realize the required performance, NEAR needed accelerometers and a closed-loop algorithm on-board to control these burns.

When time criticality and performance did not require on-board software, functions were allocated to the ground. One operational issue on NEAR was the requirement to balance the fuel use among the three fuel tanks so as to control the spacecraft center-of-mass within certain tolerances. Controlling the spacecraft center-of-mass is necessary because the thrust vector of the large bipropellant thruster is fixed. The misalignment between the thrust vector and the spacecraft center-of-mass must produce less torque than the small monopropellant thrusters can offset. At first, it was thought that fuel tank switching must be an autonomous function carried out by the G&C during the course of a burn to keep the spacecraft from tumbling. Analysis showed, however, that the center-of-mass could be controlled to a factor of 10 better than the requirement simply by using pre-planned ground commands to switch the tanks. The choice was made to leave tanking switching as a ground-commanded function due to limitations in flight software personnel resources (schedule).

Memory management is another example of a NEAR requirement that is allocated to the ground operations. The NEAR on-board processors have the capability of storing sequences of commands fro execution at a future time. The memory areas where these command 'macros' are written must be carefully managed to avoid overwriting portions of macros with new macros, etc. The on-board software could have contained tables designating used memory areas and executed commands to protect and release various areas. These features would prevent inadvertent ground corruption of on-board macros. This function was allocated to the ground because of schedule, and also personnel resource restrictions.

Early in the C&DH software development, the Mission Operations team requested that the C&DH implement a command to save a time and pointer to the current solid state recorder location, to make management of downlink data easier. In retrospect, this feature is easy to implement and would have been of great benefit to Mission Operations. At the time, the schedule risk to the C&DH team seemed higher than the benefit gained by Mission Operations, however, and the request to add this new requirement was turned down.

Command verification is a function with many components. Checksums or Cyclical Redundancy Checks can be used to check for errors introduced by the uplink. Commands can be checked for illegal opcodes, or out-of-limit parameters. Finally, command sequences can be checked against operational constraints. For example, a NEAR operational constraint is to turn off the power amplifier before changing the position of any RF coaxial switch. For NEAR, the choice was made to check for transmission errors and command syntax on-board, but flight software personnel resource constraints forced the operational constraint checking to be left to Mission Operations.

Consideration of risk, however, did require on-board command verification affecting the critical spacecraft systems after they are executed. For example, on-board safing algorithms maintain the minimum complement of spacecraft subsystems at all times. Because a faulty deltaV maneuver can easily cause mission failure, all burn parameters are checked both on the ground and by the on-orbit software before a deltaV is performed.

Risk assessment also factored into the functional allocation of momentum management between on-board algorithms and ground based operations. The NEAR spacecraft is normally controlled (except during deltaV maneuvers), by four reaction wheels. External torques (for example solar radiation pressure) can eventually build up the spacecraft system momentum to the point that wheel control is no longer possible. Before this point, system momentum must be dumped by the application of an external torque - either using the on-board thrusters, or solar radiation pressure. Planned momentum management from the ground is both feasible and preferable. Using solar radiation torques, momentum can be dumped along two of three inertial axes with no fuel use. Even if thrusters are required to dump the momentum, doing so under ground control while in contact with the spacecraft is safer than autonomous thruster use by the spacecraft.

On the other hand, should momentum build up faster than anticipated by the ground, or if ground contacts are delayed longer than expected, a high system momentum would cause mission failure. Therefore, a backup, autonomous momentum dump algorithm is implemented on-board. If system momentum exceeds a critical, programmable threshold, the on-board algorithm uses the thrusters to dump momentum to a safe level. Checks on the use of the thrusters are divided between two of the on-board processors, so that a single errant processor cannot inadvertently trigger autonomous thruster use.

Another area where on-board software is used to reduce the potential for human error is the calculation of spacecraft pointing vectors. The on-board software uses a generic pointing definition when the ground needs to control attitude to any orientation in inertial space. This generic facility could be used for all attitude maneuvers, even those used frequently such as pointing the HGA at the Earth. To use the generic scenario for this purpose however, requires the Mission Operations team to compute the specific parameters for the pointing definition with each use. To lower the risk of human error, several "canned" pointing scenarios are implemented on-board, using on-board spacecraft knowledge, such as point HGA at Earth and point solar panels at Sun.

Some on-board software was developed for NEAR to reduce downlink requirements. On-board data compression of images is implemented on the imager data processing unit. The basic science requirements are met without on-board data compression, but an improvement in performance can be realized with it. Due to schedule pressure and personnel resource limitations, data compression was made a goal, not a requirement, for the imager software. Fortunately, the imager development allowed the addition of several data compression algorithms.

Another type of data compression was not implemented on NEAR. The NEAR telemetry system samples and records or transmits housekeeping data using a simple time commutation system. This is an inefficient use of bandwidth, because very slowly changing data is replicated for weeks on end. A more efficient scheme uses on-board intelligence to only downlink data when it has changed sufficiently to merit ground controller attention. Due to personnel resource limitations, no such feature was implemented on NEAR. To allow the ground controllers a "quick look" feature, a data summary table, giving the highest and lowest value of each housekeeping parameter, with a time for each, is implemented. Using this feature, Mission Operations can scan this table and decide if any playback of recorded housekeeping data is necessary.

The tradeoff driver that had the least impact on the NEAR allocation of functionality between the spacecraft autonomy and Mission Operations is mission life cycle cost or organizational cost. The fixed and extremely short development time (27 months) made schedule the overriding consideration. For a short development schedule, the incremental costs of adding personnel were insignificant. Early during the NEAR concept development, consideration was given to putting optical navigation capability on-board for both cruise and asteroid operations. This feature would have reduced mission operations cost by eliminating the navigation team. However, in contrast to ground-based navigation which is well-understood and commonly performed, on-board navigation algorithms needed to be developed and verified over a considerable time period. Therefore ground navigation is baselined to reduce the schedule risk.

Still, some additional on-board software was written to reduce the Mission Operations load.

After launch, when schedule pressures were reduced, an algorithm for autonomous momentum management using solar radiation pressure was developed. This algorithm is incorporated in an upload (required for other reasons) planned for the near future. This feature will be tested in flight, and if it works as planned, may be incorporated in future missions, reducing the cost of Mission Operations for these missions.

Even with the considerable effort that went into allocation of functions between the ground-based operations and the on-board software, lessons are always learned during the actual operation of a spacecraft. Part of the on-board autonomy on NEAR is implemented as a series of rules and command macro responses that are reprogrammable from the ground. While this added some complexity to the on-board software, it allowed the flexibility to move some functions from the ground to the spacecraft as the operations team gains experience with the spacecraft. For example, this feature is used by Mission Operations for data recording and downlink management, replacing the need for the on-board data management command Mission Operations had requested and was denied during development. This use for autonomy rules was never considered by the flight software designers. Autonomy rules are also being used to time repetitive commands, a function that was planned to be accomplished by pre-calculated timetagged commands uploaded from the ground. While providing flexibility generally implies more software complexity and risk, its benefits far outweigh the cost of the development and test time needed to provide it safely.

IV. Future Trends

If past trends continue, the demand for data downlink will continue to outpace the improvements in RF capability. Fortunately, the computer resources that can be flown will also increase rapidly. Finally, pressure for reduced life-cycle costs, particularly Mission Operations costs, will force standardization and use of more off-the-shelf technology. These trends will probably result in the following changes:

• The uplink and downlink interfaces to spacecraft will be standardized within an organization. This will allow reuse of Mission Operations tools and reduce training costs. In the future, "driving" a spacecraft will be analogous to driving a car, where experience with one model allows operations of other models without knowing the internal details. This trend is beginning in the standardization of command languages (e.g., SCL), but no progress has been made on standardizing downlink.

• Autonomy will be increased on-board, to compensate for reduction in the frequency and duration of expensive ground contacts. For deep space missions, on-board navigation can really reduce the need for ground contact during the cruise phase.

• On-board data management will increase in sophistication. Recorders will become random access devices. Data compression and even on-board data analysis will increase to reduce the requirement for downlink. Data sharing among subsystems will be possible as networking architectures are implemented on spacecraft. Selective downlink will be a reality.

• Reduced costs will also force increased code reuse and off-the-shelf hardware. When forced to use standard components, the ability to optimize any particular application will be decreased. In the future, spacecraft will be produced to give the best result at a fixed cost, rather than optimal performance at any price.

V. References

[1] FARQUHAR, R.W., DUNHAM, D.W., and McADAMS, J.V. "NEAR Mission Overview and Trajectory Design," *Journal of Astronautical Sciences*, Vol. 43, No.4, 1995, pp. 353-371.

[2] SANTO, A.G., LEE, S.C., and GOLD, R.E. "NEAR Spacecraft and Instrumentation," *Journal of Astronautical Sciences*, Vol. 43, No.4, 1995, pp. 373-398.

[3] LEE, S.C., SANTO, A.G. "Near Earth Asteroid Rendezvous (NEAR) Spacecraft Safing Design," IAA-L-0517, *Proceedings of the Second IAA Conference On Low-Cost Planetary Mission Conference Procedings* (to be published)