DEPENDABILITY DESIGN FOR THE PLUTO FLYBY MISSION

Stephen B. Johnson, Research Associate
Larry P. Cooper, Director
Don D. Uhrich, Research Associate
University of Cincinnati Space Engineering Research Center
Cincinnati, Ohio

Abstract

The Pluto Flyby mission poses very stringent dependability requirements. These requirements drive a design that must optimize the reliability of the system over a decade, the availability of the spacecraft during the crucial flyby period, and the security of the science data during the long post-encounter playback period. This paper discusses the initial studies of the processes and techniques which will be used for the dependability of this difficult mission. These studies and initial analyses draw from dependability research and development at various institutions over the last two decades. After briefly outlining the major features of this prior work, their application to the Pluto mission will be discussed. A primary feature of the processes and techniques used is their application across all elements of the system, including the spacecraft hardware and software, the ground hardware and software, and the human operators.

Introduction

This paper begins with an overview of the Pluto Flyby mission, the current baseline spacecraft, operational strategy and managerial mechanisms. Next, methods developed at the Jet Propulsion Laboratory (JPL) and at other institutions (such as other NASA centers, the United States military, and the European Space Agency) for design and validation of highly dependable systems will be reviewed. After a more detailed look at the Pluto mission and system from the viewpoint of dependability, the major dependability issues facing the designers and operators of the system will be described. Finally, some preliminary results from the dependability analysis of the system will be presented.

The Pluto Flyby Mission and System

This section will describe the scientific motivation for the Pluto Flyby mission, the mission as it is currently envisaged, the institutional factors driving changes in the design and operation of this mission from previous planetary missions, and finally an overview of the spacecraft itself.

Scientific Motivation

Over the past decade, scientific knowledge regarding Pluto and its moon, Charon, has grown tremendously. As a target of scientific interest and opportunity, it has grown in importance correspondingly. Aside from the fact that it is the only planet not visited by any spacecraft, the Pluto - Charon system has been found to be of intrinsic interest in and of itself.

Pluto is the only rocky, solid outer planet. This, along with the highly eccentric orbit of Pluto suggests an origin different from that of other planets. Another unusual feature of Pluto is that its atmosphere forms and decays on a periodic basis. When Pluto is relatively close to the sun (as it is now), its nitrogen - methane atmosphere is in a gaseous state. However, when further from the sun, scientists predict that this atmosphere will condense and collapse onto the surface of the planet. Due to this unusual feature, and the relatively large size and close proximity of Charon, Pluto may well have landforms and dynamics unlike anywhere else in the solar system.[1]

The Pluto Flyby mission classifies its primary science objectives as 'Class 1A Science.' These are:

- Characterize Global Geology and Morphology
- Surface Composition Mapping
- Characterization of Neutral Atmosphere

To accomplish these primary objectives, the current candidate instruments include:

- Visual Imager
- IR Spectral Mapper
- UV Spectrometer
- Radio Science.

The Mission

Currently, discussions are under way with Russia over collaborative arrangements for the Pluto Flyby mission. At the moment, these discussions point to the possible contribution of two Russian launchers, along with two surface probes. Although these arrangements are not yet finalized, this paper will utilize this configuration as a baseline.

Two spacecraft will be launched to Pluto on direct trajectories (no gravity assist). Each will be launched on a Russian Proton 2-stage stack. The launch date is currently baselined for 2001. Cruise time, given current launcher and spacecraft mass estimates, will be approximately 9.2 years. It is expected that the two spacecraft will be launched from 6 to 18 months apart.

During the nine years between launch and encounter, there is no planned science activity. There will be very limited contact with the spacecraft, with only 4 hours of DSN tracking and interrogation per week planned. As the spacecraft's distance from earth increases, this will probably be increased.

All of the science acquisition occurs during the encounter period, which lasts from the time the spacecraft is able to acquire better data than is possible from earth, until soon after the flyby. Spacecraft science data should be better than earth-based data approximately 6 months prior to closest approach. It is expected that the science data will require roughly 1 Gigabit of memory. Since at Pluto distance the data rate is only 80 bits per second, all of the data will have to be stored on-board, and played back later over a period of roughly 6 months.

Imaging will begin roughly 4 to 6 months prior to closest approach. The flyby of the first spacecraft is designed to bring it to roughly 10,000 kilometers from Pluto. The second spacecraft is planned to flyby Pluto 6 months to 1 1/2 years after the first spacecraft. The second spacecraft will likely be targeted for a closer approach to Pluto and Charon.

Managerial and Operational Change

Despite its scientific value, a mission to Pluto could not be undertaken in today's fiscal environment unless mission costs can be considerably reduced from that of past planetary spacecraft. The current fiscal environment for science in general, and space science in particular, will not support a mission on the scale of Galileo or Cassini. The trend of bigger, more complex, more costly spacecraft had to be broken and reversed for the mission to have any chance at endorsement. This is a trend that can be seen in NASA's new 'Discovery' class missions. Yet, as the experience of the loss of the Mars Observer and the Galileo high-gain antenna deployment show, dependability concerns cannot be compromised if the mission is to succeed. In order to accomplish these goals, JPL's Pluto project management, system design and philosophy are significantly different from that of previous missions.

The Pluto Flyby mission is among the first planetary missions to employ a life-cycle cost accounting method. This is significantly different from previous projects which fragmented the budget into different accounts. This is a very positive feature, allowing operational considerations to play an essential role in the system design from the beginning of the project.

One of the primary ways to keep the system life-cycle cost low is to lower the weight of the spacecraft. This produces two significant effects. The first and most important is the use of a less powerful launch vehicle. The second factor is a reduction in flight time to Pluto, which translates into cost savings in operations. Originally, this operational savings was thought to be quite large. However, as operational costs have been reduced by methods discussed later in this paper, this savings has turned out to be

a less significant factor than originally anticipated.

Operational savings have been produced by the use of innovative management and automation, together allowing for large reductions in the mission operations (MOS) team for the duration of the mission. Previous JPL missions have usually been characterized by large operations teams which are dedicated to the mission. This has been a major factor in the flexibility and consequent success of prior missions. For the Pluto mission, reductions in team size are possible because of improvements in automation technology, and also because of the nature of the mission itself. Since during the 9-year cruise time there is little of scientific or engineering significance occurring (just trajectory corrections and general housekeeping), it is feasible to utilize a reduced MOS team without significant impact to the mission. For the Pluto mission, the dedicated MOS team is currently planned to consist of two engineers per subsystem, with further support called in on an as-needed basis to resolve anomalies, and for the Pluto Encounter.

Another major cost-reduction factor for the mission is a reduction in the quantity of Deep-Space Network (DSN) coverage as compared to previous planetary missions. Each of the two spacecraft will receive only 4 hours of coverage each week. Since the direct costs of DSN support are relatively large, this of itself is a large cost savings. In addition, since MOS teams generally grow around the analysis and production of data, a reduction in the data also promotes a reduction of the MOS workload. The reduced DSN coverage also necessitates a higher degree and a different kind of on-board autonomy than prior missions have utilized. The spacecraft itself will very likely have to make decisions regarding the telemetry that will be sent to the mission operations team. This is a significant change from previous systems, and one which must be carefully analyzed for its potential impact on mission dependability.

Another significant change in the JPL approach to the Pluto mission from prior planetary missions is a reduced reliance upon hardware and software with a 'space heritage.' JPL has been directed to utilize advanced technologies whenever possible to reduce costs. Additional risks will be tolerated if significant cost reductions are forthcoming. The substantial efforts at component miniaturization by the Department of Defense and NASA over the previous decade pay large dividends on this mission.

Another change for JPL is the greater use of external expertise than on prior missions. In particular, the Pluto Flyby project is working with many universities and small businesses in order to reduce costs. Both universities and small businesses can often deliver products and services at much lower costs than is possible under the usual prime-contractor or in-house build managerial styles. The key is to identify where this expertise lies, and to integrate products and services together with the overall goals of the mission and system.

The Spacecraft

Figure 1 shows the 1993 configuration of the spacecraft. It is a conventional planetary probe, but smaller and more autonomous than its predecessors.

Only a brief description of the spacecraft will be presented here. A more detailed description of the Pluto spacecraft baseline configuration is presented in Reference 2.[2]

The spacecraft is divided into seven subsystems, all traditional: Telecommunications, Electrical Power and Pyrotechnics, Attitude Control, Spacecraft Data, Structures, Propulsion, and Thermal Control. In addition to the candidate science instruments noted in the previous section, the spacecraft is also likely to include a Zond probe contributed by the Russians. This requires a spin table to spin up the probe, and an additional Radio Frequency (RF) Receiver. With these additions the current spacecraft mass is estimated around 180-200 kilograms, including propellant.

The Telecommunications Subsystem consists of a 1.5 meter diameter high gain antenna, with associated RF electronics, and an RF receiver for the Zond probe. Using 34 meter antennas, an 80 bps rate is expected at encounter. If 70 meter DSN antennas are used, a 160 bps rate is possible.

engineers is given the task of analyzing the interactions and coordinating the various subsystems.

The project fault protection group usually consists of one or more engineers operating at the system level, coordinating the activities of the command sequencing, data handling, attitude control, and power subsystems in terms of fault protection capabilities. There is usually one or more representative responsible for the fault protection embedded within the various subsystems that work with the systems engineer. These engineers have dual roles, corresponding to their subsystem role in embedding the fault protection into the subsystem, and the system role of helping the system fault protection engineer(s) design the system level fault protection.

Responsibilities of the fault protection group include analysis of the system design under fault conditions, preparation of the system validation plan and testing procedures, generating and tracking fault protection requirements and changes, coordinating system fault protection testing activities, and generating mission operations contingency plans. Since the fault protection engineers know best how the spacecraft behaves when anomalies occur, and also how best to recover the system, they are the best qualified to perform this MOS function.

Note that although the typical activities of the fault protection engineer moves beyond the flight system design (e.g. contingency planning), the JPL focus is on the spacecraft. As the focus of planetary system design moves towards the system life-cycle, more consideration needs to be given to mission operations. With Pluto's ten year operational life, this is appropriate. Since other organizations have developed mechanisms which concentrate more explicitly on the dependability of the operations phase, it is to these organizations that we can look for other approaches which will be of use for the Pluto mission.

Other Dependable System Design Processes

JPL is not the only organization that has successfully developed complex, highly dependable systems. Other institutions include other NASA centers, the US military, and the European Space Agency. These organizations have developed their own highly successful techniques, which are now beginning to find their way into various standards and publications. The common features of these new standards will be described, with an eye towards applying them to the Pluto Flyby system.

These new processes are not always easy to identify, because the names given to them vary with the particular organization and culture that generated them. Despite the apparent differences based upon the different names, each of these processes is dealing with many of the same issues as JPL's fault protection. For the purposes of this paper, we shall refer to the overall set of issues covered by these processes as *system dependability*, or dependability for short.

Generic Integrated Maintenance and Diagnostics (GIMADS)

This United States Air Force program based at Wright Research and Development Center created the Integrated Diagnostics military standard MIL-STD-1814. Within this document are references to other applicable military standards, and the various military procedures necessary for military weapon system acquisition. From the standpoint of dependability design processes, this standard focuses on the quantitative side of the dependability picture. Listed in this document are a number of quantitative measures of potential significance to space system dependability. These include: Mean time to diagnose, false detection and isolations, system checkout time, frequency of inspections, mean time to diagnose, diagnostic manpower, time to repair, reconfiguration time, servicing time, diagnostic mix, embedded fault coverage, and fault reporting latency.

This document was built around the need for diagnostic systems and requirements for Air Force operational systems such as fighter and reconnaissance aircraft. An important characteristic of this process which is different from other process documents is the break down of the system into its various missions and operational timelines, in order to then allocate mean time to diagnose,

isolate, and repair / reconfigure the system. False detections, isolations, retest OKs, and cannot duplicates (CNDs) thus play a major role in the large numbers of aircraft and missions typical for Air Force applications.

Space Station Work Package 2 Failure Tolerance and Redundancy Management Design Guide

This document, prepared by McDonnell Douglas Space Systems Company, outlines a process for the initial analysis and design of the failure tolerance mechanisms for Space Station Freedom (SSF), a major element of the dependability of that system.[6] After specifying various requirements for dependability (note it is not called dependability or fault protection in the document, although that is clearly what the subject matter is), section 4 of this document describes a process for failure tolerance analysis.

The initial step of the process is to define system end-to-end 'functions.' Functions are defined in an object-oriented manner. That is, functions are based on 'what' is to be done, not 'how' it is done. Once a function is defined in this manner, its criticality is specified. As an example, resupply of water to SSF is a function which can be assigned a criticality compatible with NASA criticality standards, whereas the process of transferring equipment or supplies to SSF cannot be assigned a criticality without specifying the item(s) being transferred.[7]

Once the functions are defined, they are then assigned criticalities based upon typical NASA standards as in JSC31000, Rev. E, Table 3-6.[8] These are interpreted in the space station document as follows:

- Criticality Category 1: A function essential for crew safety or Space Station Manned Base survival.
- Criticality Category 1S: A function essential for the detection monitoring, or control of hazards to crew safety.
- Criticality Category 2: A function essential to conduct mission operations with payloads or other vehicles.
- Criticality Category 3: All functions not in Criticality Category 1, 1S, or 2.

After criticalities are assigned, schematics of the system are generated, down to a level appropriate to analyze and assess block level redundancy within the system, typically Orbital Replaceable Units (ORUs) or Computer Software Configuration Items (CSCIs). Once these schematics are generated, then the system is analyzed for its fault tolerance capabilities. In particular, the timing constraints associated with FDIR are studied. These timing constraints include the period of criticality (when a particular function is active and critical), concurrency (which processes run in parallel, and can thus not share resources), time to criticality (the time duration between the time of the fault and its detrimental consequence), and the window of opportunity (the window of time in which the function can be initiated). These timing constraints affect the dependability design, in terms of which technologies are appropriate to respond to failures, what processes or procedures need to be used to recover the system, and so forth. This document may be the first to identify 'time to criticality' as a separate entity of import for system dependability. The design of the system not only accounts for these timing and criticality constraints to determine its redundancy management design, but also other factors such as weight, volume, power, and cost.

Assessment of the system can be greatly assisted by testability and fault tolerance analyses. Tools which aid these analyses are the Harris System Testability Analyzer (HSTA) and Digraph Matrix Analysis (the forerunner of FEAT). These tools assess ambiguity zones (where the fault cannot be isolated to one component with certainty) for testability and fault containment strategies for fault tolerance.

BSTS Fault Tolerance Guideline

This document was written by Fail-Safe Technology Corporation for Lockheed Missiles and Space Company for the Boost Surveillance and Tracking System program.[9] It presents the basic terminology, process, and techniques for design and analysis of fault tolerant systems and subsystems for BSTS. This document is written from the perspective of fault tolerant computing, as opposed to fluid, mechanical, or control

effects analysis, and fault tolerance analysis (fault trees, Markov models, etc.).

National Launch System System Health Management Design Methodology

This document was written by Martin Marietta Astronautics Group for the National Launch Systems (NLS) Advanced Development Program.[11] The goal of this study was to produce and document a process for the design of the Health Management System for NLS and other programs. It does not focus on technologies or tools for dependability, or create a candidate architecture. Rather it concentrates exclusively on the issues of designing a dependable system, and a number of processes which can help in the design of such a system. One of the primary purposes of this document was to describe and elaborate the issues behind the various requirements and processes necessary to develop a dependable system.

The basic assumption of this document is that a dependable system results from a consistent approach to understanding the implications of faults within the system, and designing the system to deal with them. A dependable system cannot be built by patching an already existing design which does not account for the possibility of faults.

Errors and faults spring from a number of sources, of which hardware failure is but one. Others include software bugs, human operator faults, faulty specifications, etc. The document discusses fault classification as a tool to simplify fault analysis. Fault classes are logical ways to group faults, which may or may not be useful for a given purpose. Thus the key to classification is to determine the uses of these classifications in analysis and design.

Error and fault containment are also major elements in the design process. Once an initial design has been determined, faults and their symptoms must be contained if the overall system is not to fail. Thus certain hardware mechanisms can be deployed in the design to provide redundancy in case of hardware faults, or lockout mechanisms to prevent human faults in commanding the system. Error containment mechanisms are not necessarily the same as fault containment mechanisms.

Another element in the design process is the time-to-criticality analysis. In order to contain, detect, isolate, and respond to faults, it is necessary to know the time between the onset of the fault and its detrimental consequence(s) to the system. This is the time in which the system (be it composed of hardware, software, or humans) has to respond. A comprehensive analysis based upon the time dynamics of faults can pinpoint problem areas in the design which can be fixed by designing out the fault, or deploying different mechanisms to detect and respond to the fault.

Another important element in the process is the use of qualitative requirements and features to validate a design. Quantitative analysis of a system is quite useful in the design process, to trade cost versus amount of redundancy, fault coverage with the cost of sensors, and so on. However, particular quantitative estimates of reliability or fault coverage cannot be proven quantitatively until the system is deployed (and many times not even then). Thus qualitative means must be used to validate the system. In particular appropriate testing using fault injection is the primary validation mechanism of the dependability of a system.

European Space Agency Failure Modes, Effects, and Criticality Analysis Requirements Document - Draft

This document, which is under development in the Product Assurance and Safety Support Division at the European Space Technology Center (ESTEC), develops a new standard for FMECAs within ESA. However, it is attempting to establish far more than the mechanics of FMECAs within ESA. This draft standard begins to establish a set of procedures very similar to United States Navy, NASA, and Air Force developments.

One of the novel sections of the document concerns the establishment of functional modeling and functional failure analysis. The purpose of functional modeling and functional failure analysis is "to characterize the various functions needed to perform a space mission so as to provide a basis for the

establishment of reliability requirements, including failure tolerance." In other words, fault analysis of the system is necessary during the conceptual development of the system to develop requirements for reliability and fault tolerance (i.e. fault protection).

The functional model breaks the system into its major functions, and then performs a top-down functional fault analysis. The analysis is documented using a function-versus-function matrix analyzing the interactions between subsystems.

The document next discusses development of the FMECA. This is a fairly standard treatment, where redundancy and criticality of the fault's effects are categorized. Near the end of this section (section 7), there is a discussion of the timing of failure effects, the detection, diagnosis, and response times. These timing effects are included in the FMECA.

The last section of the document puts forth a process for analysis of hardware / software interactions. This analysis uses the same support documentation and categorization as the FMECA.

Rockwell Integrated Vehicle Health Management Design Handbook

Rockwell International Space Systems Division is currently developing an IVHM design handbook for use throughout the division. For Rockwell, Integrated Vehicle Health Management involves the overall design of a dependable system. The handbook details the processes which can be used to achieve such a design.

Rockwell recognizes two paradigms for IVHM design, one stemming from a NASA tradition, the other from the Department of Defense. For the NASA-based tradition, they consider the Martin Marietta System Health Management methodology as a primary example. For the DOD-based tradition, they consider the Fault Tolerance Task for the System Utility and Survivability Evaluation as exemplary.[12] This contract is in the same tradition as the BSTS and US Navy documents discussed above, stemming primarily from the fault-tolerant computing community. Rockwell notes that the primary difference between the two traditions is one of degree rather than substance. The NASA-based process tends to be more qualitative than quantitative, and the DOD-based approach is somewhat more quantitative (reliability and availability play a larger role). For a given program, Rockwell leaves it as a purely pragmatic question as to whether qualitative or quantitative methods should be favored.

The Rockwell approach blends the two traditions. Both reliability estimation techniques and fault injection methods play a major role. Timing issues also play a major role, in time-to-criticality analysis and also for FDIR analysis of fault tolerance mechanisms. Fault and error containment regions are basic analytical building blocks for the system as a whole.

Dependable System Design Process Summary

As the discussions above make clear, there is considerable agreement from a number of sources on the types of techniques that should be utilized to make systems more dependable. Using the best elements of these documents and the JPL process, a recommended methodology can be summarized as follows:

- All parts of the system must be included in the dependability design: hardware, software, and people; flight and ground.
- Create a project position or team clearly responsible for system dependability. This must include active involvement in the system design, not just estimating reliability or performing after-the-fact FMECAs.
- Once a system concept has been defined, perform a functional analysis of the system. This analysis breaks down the system into functions, and then analyses these functions in terms of timing and criticality (time-to-criticality analysis). The result of this analysis includes a set of timing requirements for the system.
- When the system concept becomes an initial system design, fault types are identified and classified.
- The system is divided into fault and error containment regions. Within each region, each fault class is analyzed to

determine how it is contained, detected, isolated, responded to, its criticality, and timing.
- Validation of the system is performed primarily by qualitative means using fault injection. For this purpose, a fault set is defined from a preliminary FMECA, and fault injection tools must be developed for the system.
- Quantitative parameters such as Mean-Time-to-Repair, Reliability, and Availability can be estimated and analyzed using various tools, usually including Petri nets and Markov or semi-Markov chains.
- Either quantitative or qualitative approaches can be emphasized, depending upon the system and the customer.
- Pay particular attention to system-level interactions. These tend to be the most dangerous type of problem.

Pluto System Dependability Issues

The Pluto Flyby system, as with any other system, has its own unique characteristics. We must determine the major dependability issues which will be faced in the design of the Pluto spacecraft. This information is also critical in determining what areas must be investigated, and which methods are appropriate for this investigation. Some of the issues are common to all planetary probes, and others are unique to the Pluto mission.

Uniqueness

As is the case with other planetary probes, the Pluto mission is unique. Two of the spacecraft which will be built, following the strategy of the Viking and Voyager missions of the mid-1970s. This has significant implications for the dependability aspects of the design.

First, quantitative estimates of reliability or availability will be questionable, since many of the components will be unique. Even when specific components have prior heritage, they are combined in new ways. Because of this, the qualitative process which JPL has stressed for planetary probes will be more useful than the quantitative approaches more typical of DOD applications.

Second, the uniqueness of the hardware, software, and operational approaches make it very likely that design bugs will exist in the system, whether in the flight system, test system, mission operations system, or various procedures. This of course is one of the primary reasons why qualitative validation is so critical. But it also means that fault tolerance techniques, or on-board fault protection, must be built in a manner in which it can successfully cope with design faults. Recent experiences confirm this. For example, on the Magellan mission, a flight software operating system bug nearly caused loss of the spacecraft after Venus Orbit Insertion. Another example is on the Clementine mission, which was effectively put to an end because of a mismatch between flight and ground configuration tracking procedures. The on-board fault protection must be built in an extremely robust manner.

Third, there will be a small team building the system, knowledgeable about the system once it is completed. Unlike UNIX- or DOS-based computer systems, where there are many people who are knowledgeable about the system, the pool of resources is limited. If problems occur, as they invariably do, this same small group of people will have to be utilized to resolve the problem. This is exacerbated by the next issue.

Mission Length

The Pluto Flyby mission is a very long mission, much like the Voyager or Galileo missions. However, there is *no* significant science data gathering activity planned for the 9 year duration of the cruise to Pluto, other than possible checkout of equipment. This poses some very challenging problems.

Perhaps the most difficult problem is one of knowledge retention. Detailed knowledge of the Pluto spacecraft design must be retained for the entire duration of the cruise until encounter, even when much of the design is not used. The spacecraft *must* execute the science data collection correctly. Yet correct operation of the spacecraft is difficult to guarantee since many of its features are utilized for the first

time 9 to 10 years after the spacecraft design has been completed!

From the beginning of the design process until the end of the mission, 13 to 16 years will elapse. It is unrealistic to assume that all of the original designers of the system will still be available after such a long period. Knowledge of the design must be captured during the design process in a way that is less dependent upon the memories of individuals. This is true both because people will inevitably be unavailable, and because even if they were available, they are unlikely to remember all the relevant information nearly a decade after the design work was completed.

Communication and Decision Delay Times

All planetary probes have the characteristic that light-time communication delay times between the earth and the probe are on the order of minutes or hours, making real-time control impossible. Therefore all planetary probes have built in various mechanisms for autonomous operations. As discussed previously, on-board fault protection is one of these mechanisms.

Although the light-time communications delay is the characteristic that most often comes to mind, there is a second delay which planetary spacecraft share with many other systems. Whenever anomalies occur, analysis of the event takes place on the ground, and remedial action must be taken. The time that it takes from recognition of the anomaly on the ground until commands to recover the system are ready to be sent are typically on the order of hours to days. This delay time is equally significant for the design of planetary missions, for it usually defines the amount of time that the spacecraft must maintain itself in a safe configuration awaiting commands.

Limited DSN Coverage

Each Pluto spacecraft will receive about 4 hours of Deep Space Network coverage per week. As the distance between earth and the probes lengthen, the telemetry communication rates are estimated to decrease to 80 bits per second. With these low communication rates and limited time available, the Pluto spacecraft will have to be selective in the data in which it sends to the ground. This is particularly true in the case of anomalies.

Even though large amounts of memory (nearing 1 Gbits) may be available to store information on-board, sending this data back to earth will be extremely time-consuming. In the case of anomalies, a good design will make it likely that the data surrounding the initial anomaly is available in memory. The spacecraft itself will have to decide which data is relevant, and ship it to the ground first. This will likely entail the use of an on-board expert system.

There have been reservations about the use of on-board expert systems because of problems associated with validation of these systems. JPL, for example, places a premium on determinism in the software. Expert systems are viewed as non-deterministic, and hence not validatable. It is possible to reconstruct what the expert system *did*, but it is not so easy to figure out what it will *do* before the fact. The Pluto spacecraft will have to overcome the difficulties associated with validation of expert systems, because of the clear need for its use.

Reducing Costs

The Pluto project management has been given the directive to use new technologies and organizations. This is clearly a step in the right direction. However, there is a risk involved. How can all of the new organizations and new technologies be integrated and validated to the same standards as earlier planetary probes? This magnifies the one-of-a-kind problem discussed above.

The two Pluto spacecraft will include many new technologies. These in turn are built by organizations which may not be familiar with the standards and methods built up over the years at JPL. Additionally, JPL itself is trying to change its way of doing business in order to reduce costs, but without compromising the success of the missions. Finding a compromise between maintenance or improvement of dependability, while at the same time reducing costs is one of the primary issues throughout the aerospace

One of the perennial difficulties with designing systems is that the ground equipment to test the flight system must be ready before the flight system. It must be designed to interface with and test the flight system, and in some cases to mimic elements of the flight system. This is a problem because usually the flight system design is not yet frozen, and thus the test system must mimic or test a changing flight design. Fault protection testing also requires that the testing system provide certain capabilities. In particular, the test system must provide the capability to inject faults. This is an area of common agreement among the various dependable system design processes described previously. Since the simulated faults should mimic the behavior of real faults, something about real faults must be known first.

What can be done this early in the program? The primary consideration is to design the Ground Support Equipment (GSE) in such a way that fault injection capability can be built into the system when sufficient information is available to model fault behavior of a particular component. For example, the Pluto spacecraft will include a sun sensor. In order to test the fault protection software associated with the sun sensor, it will be necessary to model faults of the real flight sun sensor. This detailed information is not yet available, but the GSE architecture design is underway. It will be necessary to build a sun sensor interface, but leave the fault modeling capability in the GSE software. Thus when information becomes available regarding failure modes, the architecture will not prohibit the modeling of these failure modes.

Similarly, it is known that a system simulation will need to be built. Fault injection capability must be part of the planning for that simulation. The simulation will need to have the capability to insert simulated faults in the middle of a nominal set of events.

Knowledge Capture

It is already known that the Pluto Flyby mission has some unique problems associated with capture and retention of design knowledge. Yet sophisticated software for design knowledge capture does not seem to be feasible for cost reasons. In addition, it is clear that whatever scheme is used to capture design knowledge must aid, or at least not hinder the designer in his / her everyday work.

Various projects have tried to use integrated requirements and design knowledge capture tools, with varying degrees of success. If the project is large, and cost is less of an issue, these systems often prove their worth. But for a smaller project which is trying to be 'cheaper, faster, better,' it is not obvious that there is much of a cost savings to be gained. The philosophy of small projects is to minimize documentation and keep team sizes small. Pluto fits somewhere in between the large and the small project. In any case, the need for some sort of mechanism of knowledge capture and retention is clear.

One idea under consideration is to use a relatively simple scheme for capturing the design knowledge as it evolves. This would be used mainly to ensure that documents are consistently archived with a workable retrieval system. Since there are 9 years of cruise to work with, it is possible to work with this data and pick out the relevant details to ensure that the encounter sequence works properly. Thus the Pluto goals of working with a broader community can be met, along with the goal of improving the chances of mission success.

Another facet of the knowledge-capture issue is to consider the use (and non-use) of expert systems in previous spacecraft missions. It is clear that expert system technology is available. Yet the use of expert system technologies has not been widespread. In our view, a major factor in the under-utilization of expert systems has been the cost of capturing design knowledge. Expert system technologies are under-utilized because the cost of capturing the knowledge which must go into them is high. Thus, if expert system technologies are desired, then the design knowledge which must reside in them must be captured once *during the system design process* and then translated into mission operations-usable forms.

Modeling Strategies

The functional time-to-criticality analysis provides an initial set of information about the fault behavior of the system. It thus provides a simple test case of design knowledge capture and retention. Can this data be utilized for Pluto mission operations? If so, in what form is it captured, and into what form must it be transformed in order to be usable for mission operations?

There is at least one example of this sort of transformation available. Ames Research Center has created a set of tools to transform directed graphs or fault trees into common LISP. Since a fault tree or directed graph can be the core of a diagnostic engine, this is a very useful transformation. The time-to-criticality analysis is another example of a set of information that could be used in diagnostics. Thus a similar kind of transformation should be possible. Later on, when FMECAs and Fault Containment Region matrices are created, this information should also be transformed instead of recreated to form the Pluto ground (or flight?) diagnostic system.

Conclusion

Planetary probes have always posed a challenge to designers in terms of autonomy and dependability. The Pluto Flyby mission poses a an even more challenging set of requirements for system dependability.

In the past, JPL has met these challenges by developing a process for design and validation of on-board fault protection. Although very successful, this process has not been able to prevent a continual influx of changes from the fault protection design into the rest of the system. The Pluto Flyby mission challenges us to create a more cost-effective and systematic way to design a dependable system. Fortunately, over the last decade, new developments in the Department of Defense, other NASA centers, and the European Space Agency provide guidance for how to improve the fault protection design process.

The process to be used for the Pluto Flyby mission includes essential elements from the JPL and non-JPL design processes. Key features of this process include the performance of a functional time-to-criticality analysis, the creation of a fault protection group charged with system dependability, the development of error and fault containment strategies and corresponding analyses, and use of fault injection for system validation. The Pluto Flyby mission also puts a premium on improving methods for capturing and retaining knowledge. These are under investigation.

Using the best elements of design from JPL's planetary experience, and the design experience of other aerospace organizations, the demanding requirements of the Pluto mission can be met. It *is* possible to achieve high dependability for a reasonable cost.

Footnotes and References

[1] For further details on Pluto and the science mission for Pluto Flyby, see Stern, Alan, et. al., "The Highly Integrated Pluto Payload System (HIPPS), An Instrument Concept for the Era of Better, Faster, Cheaper," Proceedings from the 7th Annual AIAA / Utah State University Conference on Small Satellites, September 1994, Logan, Utah.

[2] Staehle, Robert L., et. al., "Pluto Mission Progress Report: Lower Mass and Flight Time Through Advanced Technology Insertion," Proceedings of the 44th Congress of the International Astronautical Federation, October 16-22, 1993, Graz, Austria, IAF-93-Q.5.410.

[3] SCL is a product of Interface and Control Systems, Incorporated, based in West Melbourne, Florida.

[4] Some readers may wonder why Mars Observer is not on this list. The Mars Observer project was a change from the existing design philosophy in that this system was designed from an earth-orbiting, commercial spacecraft tradition at General Electric Astrospace (now Martin Marietta Astrospace). It did not significantly follow the example of earlier fault protection designs from planetary probes.

[5] Kobele, P., *Maintainability of Unmanned Planetary Spacecraft: A JPL Perspective*, AIAA-89-5070, AIAA/NASA Symposium on the Maintainability of Aerospace Systems, 26-27 July, 1989, Anaheim, CA

[6] *Failure Tolerance and Redundancy Management Design Guide*, Document #MDCH4865, prepared for NASA contract NAS 9-18200, Work Package 2, by McDonnell Douglas Space Systems Company, Space Station Division, 1989

[7] Dausch, Robert J., *An Approach to Failure Tolerance Analysis for Space Station Freedom*, proc. of Digital Avionics Systems Conference, October, 1990, this paper summarizes the approaches contained in the Space Station document.

[8] *Space Station Projects Description and Requirements Document*, Volume 3, Revision E, JSC31000, 1 November 1989, Table 3-6

[9] *BSTS Fault-Tolerance Guidelines*, Prepared for Lockheed Missiles and Space Company, Sunnyvale, CA by Fail-Safe Technology Corporation, Los Angeles, CA, 1987, Document #FST87-206-1.

[10] *Draft Military Standard-Specification and Validation of Fault Tolerance in Electronic Systems Development*, Naval Air Warfare Center, Warminster, PA, 1993.

[11] Campbell, G., Johnson, S., Obleski, M., and Puening, R., *Final Report-SHM Design Methodology*, Denver: Martin Marietta Space Launch Systems Company 1992, document number MCR-92-5014, for Rocket Engine Condition Monitoring System (RECMS) contract, Purchase Order #F435025 of Contract F04611-89-C-0020, administered by United Technologies Corporation, Pratt & Whitney, Government Engines & Space Propulsion, West Palm Beach, FL.

[12] *Fault Tolerance Analysis Task for the System Utility and Survivability Evaluation (SUSE) Contract*. General Research Corporation, PO Box 6770 Santa Barbara, CA, 93160-6770. June, 1986, and March, 1987.