Deep Impact Sequence Planning Using Multi-Mission Adaptable Planning Tools With Integrated Spacecraft Models

Steven Wissler[*], Pierre Maldague[†], Jennifer Rocca[‡] and Calina Seybold[§]
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California, 91109

The Deep Impact mission was ambitious and challenging. JPL's well proven, easily adaptable multi-mission sequence planning tools combined with integrated spacecraft subsystem models enabled a small operations team to develop, validate, and execute extremely complex sequence-based activities within very short development times. This paper focuses on the core planning tool used in the mission, APGEN. It shows how the multi-mission design and adaptability of Apgen made it possible to model spacecraft subsystems as well as ground assets throughout the lifecycle of the Deep Impact project, starting with models of initial, high-level mission objectives, and culminating in detailed predictions of spacecraft behavior during mission-critical activities.

Nomenclature

APGEN	=	Activity Plan Generator
C&DH	=	Command and Data Handling
CDR	=	Critical Design Review
DI	=	Deep Impact
DSN	=	Deep Space Network
FB	=	Flyby Spacecraft
FP	=	Fault Protection
FSW	=	Flight Software
GDS	=	Ground Data System
HRI	=	High Resolution Instrument
HRI_IR	=	High Resolution Instrument Infra-Red
IMP	=	Impactor Spacecraft
ITS	=	Impactor Targeting Sensor
MMPAT	=	Multi-Mission Power Analysis Tool
MMTAT	=	Multi-Mission Telecom Analysis Tool
MOS	=	Mission Operations System
MRI	=	Medium Resolution Instrument
NAIF	=	Navigation and Ancillary Information Facility
NVM	=	Non-Volatile Memory
OD	=	Orbit Determination
PDR	=	Preliminary Design Review
SCU	=	Spacecraft Control Unit
S/C	=	Spacecraft
SubPDNO	=	Subcarrier Power to Noise Ratio
TCM	=	Trajectory Correction Maneuver
TOI	=	Time Of Impact
TOFI	=	Time of Final Imaging
V&V	=	Verification and Validation

[*] Principle Engineer, Planning & Execution Systems Section, 301-250D
[†] Senior Engineer, Planning & Execution Systems Section, 301-250D
[‡] Senior Engineer, Flight Systems Engineering Section, Mail-Stop 301-490
[§] Senior Engineer, Flight Systems Engineering Section, JPL M/S 264-850

I. Introduction

THE Deep Impact mission was ambitious and challenging. JPL's well proven, easily adaptable multi-mission sequence planning tools combined with integrated spacecraft subsystem models enabled a small operations team to develop, validate, and execute extremely complex sequence-based activities within very short development times. This paper introduces the core multi-mission tool set, with a particular focus on the activity planning tool APGEN. In addition, the Deep Impact sequence planning process is discussed in detail, including the extensive use of key technical resource modeling, for both flight and ground assets, to enable successful activity design within tight constraints. Subsequently, model validation, predictions, and in-flight results are summarized. In this section, the focus remains on the importance of key technical resource models and how they were leveraged to achieve positive in-flight performance. Finally, the operations team experience is highlighted for several major mission phases, including lessons learned with regard to the multi-mission tool set used throughout Deep Impact operations.

II. Multi-Mission Planning & Execution Architecture

APGEN is a Multi-Mission tool, indicating its adaptability to many different space missions without software changes. The multi-mission approach has demonstrated significant cost savings over the previous approach, in which each mission (re-)designed its own suite of Ground Data System uplink tools. This section introduces APGEN, explained in the context of the general planning process for space missions. In general, the use of sophisticated software to assist (or, in the case of autonomous space missions, replace) human planners is an area that continues to change rapidly, with considerable interest from both mission engineering and computer science specialists. Due to the rapidly evolving state of this paradigm, in addition to the APGEN approach, several alternative architectures and a few speculative ideas about the future evolution of APGEN and other such tools are also offered for consideration. Reader feedback and ongoing development work are encouraged, and will hopefully be more focused, given the discussion herein.

APGEN is a powerful tool due to its portability for a wide variety of space mission designs. As stated, the multi-mission approach affords considerable cost savings over the traditional approach. It should be noted, however, that these tremendous cost savings also come at the expense of some organizational complexity. unlike a mission-specific tool (designed and developed by mission personnel and fit naturally into its unique operational environment), a standard multi-mission tool only becomes useful for a specific mission once it is "adapted" for that mission. The adaptation process consists of capturing all relevant mission-specific information (such as activity types, consumable and non-consumable technical resources, and flight and mission rules) in a form the tool can understand, (i.e. adaptation files expressed in a suitable tool-specific language). Consequently, APGEN must interface with three very different elements of the mission team: APGEN users, APGEN adapters, and APGEN developers. APGEN users are typically operations personnel who create activity plans and command sequences to be executed by the S/C. APGEN adapters are GDS software personnel who create the adaptation APGEN uses to implement mission-specific features. APGEN developers are GDS software administrators who maintain and upgrade APGEN. To discuss the complexity of APGEN in a universally understandable way, this paper borrows terms from ordinary English (i.e. Plans, Activities, Resources, Models, etc.). APGEN is defined as an "Activity Plan Generator." In the next few sub-sections, the terms: *activity* and *plan*, will be clarified as they relate to the current implementation of APGEN.

It is relevant to note that specific to the domain for space missions, many activities that execute onboard a S/C are usually the result of many years of very careful planning. An operational S/C in a position to make observations of distant astronomical objects, then relay these observations back to Earth, is an extremely precious resource. As a result, many S/C activities are proposed, debated, modeled, engineered, traded-off, refined, and finally translated into executable sequences of commands over a period of time that can span several years and involve several hundreds of people. APGEN helps to conserve these precious resources, and is often a mission-enabling tool.

A. What is an *Activity*?

1. The Basics

> Activity: (...) natural or normal function: as **a**: a process (as digestion) that an organism carries on or participates in by virtue of being alive **b**: a similar process actually or potentially involving mental function; (...)

There is, of course, some circularity involved in defining an activity as a process, but one can tentatively state that *an activity is a process with a specific purpose.* To turn this statement into something useful, the definition

needs further refinement. In particular, focus must be afforded to expand on both *process* and *purpose,* and which is more important during activity development.

2. Process vs. Purpose

APGEN follows an approach to the planning problem that focuses on the process. For example, much of the work that must be performed when adapting APGEN to a specific space mission consists of specifying algorithms for turning high-level activities into lower-level ones. Another approach to planning, favored by the research community, is to focus on the purpose or goal of each activity in the plan, and to let a computer program (the 'planner') develop an algorithm, or plan, that will satisfy the high-level goal. Each approach has its advantages and disadvantages (Table 1). There has also been a hybrid planner, designed and used by the MER mission, merging the capability of APGEN with those of an AI-based application (Europa).[6]

Table 1. Algorithmic vs. Goal-Oriented Approaches to Planning

Approach	***Advantages***	***Disadvantages***
Algorithmic	• Designing an algorithm is like writing a program. • Efficient algorithms are available for many different tasks; high performance can be achieved easily.	• Understanding an algorithm is not always simple. • When the system runs into unforeseen circumstances, it has no way of figuring out what to do.
Goal-Oriented	• The basic concepts (specifying goals, making rules, and delegating tasks to subordinates) are simple. • To the extent that a S/C or robot "understands" the purpose of its activity, it can select alternatives when things don't go as anticipated.	• Goal-based systems must search for solutions among many possibilities. Brute force is generally out of the question; other methods may not work. • Heuristic methods that can bring search times down to realistic levels are not easy to develop.

APGEN follows the algorithmic approach simply because it was deliberately based upon empirical methodology found to work well in historical Planning and Sequencing activities at JPL. However, the APGEN Software Requirements Document (SRD) did not preclude inclusion of some elements of a goal-oriented approach; in fact, one requirement states explicitly that APGEN shall interface with an automatic planning tool. Details were unspecified, but the intent was to offer future APGEN users the benefits of both approaches. Although not much has been done so far to implement this requirement, funding has been provided (February 2001) to explore the possibility of infusing technology from the Deep Space 1 Remote Agent into APGEN (work to be performed by ARC, JPL, and SRI).

3. Hierarchy: Activities Recursively Defined

A well-known property of dictionaries of the English language is that they are generally helpful in spite of the fact that most definitions are circular (useless from a theoretical point of view). This is possible because when the user looks up a definition in a dictionary, he is not starting from a state of complete ignorance of the language. There is an implicit ranking of the terms which a dictionary defines, so that definitions tend to use lower-ranking terms (the ones which the user is hopefully already familiar with) to describe the higher-ranking ones (whose meaning is not always clear to the user). Similar to the dictionary, an activity could be defined as a hierarchy of simpler activities. Although recursive, this definition is valuable. It often happens in the planning process that merely exhibiting the hierarchical nature of the plan is all that is needed. APGEN allows users to give activities any name and it is therefore possible, by choosing names that are meaningful to the intended audience, to produce meaningful plans with APGEN using nothing more than a hierarchy of generic activities.

The purpose of introducing the recursive definition of 'activity' is, to some extent, reflective of the fact that space mission activity planning in its early stages is fairly circular. As an example, the overall plan may consist of a pre-launch activity, a cruise activity, an orbit insertion activity, and a mapping activity. Similarly, the cruise activity is also defined as the second part of the overall plan. 'Cruise' refers to a specific phase in the life of an interplanetary mission, and there is nothing artificial or recursive about it. However, from a formal perspective, the cruise activity *as defined in the plan* has no substance other than sitting between the pre-launch activity and the orbit insertion activity. Circularity and self-reference are an inherent part of the early design process for just about any product. A product can not provide its function until all its parts have been defined, and the parts can not be fully designed until the product can be described to some extent. This act of 'creating something out of nothing' is familiar to all designers, at least as the first step of a new design. Planning for a new mission is similar to designing a new

product. In the early stages, it can be valuable simply to exhibit a hierarchy of tasks or activities that reference each other. Seeing the hierarchy can be helpful in identifying the structure of the emerging mission. Developing the correct structure is a difficult task, and the ability to display and manipulate hierarchies of yet-undefined activities is an important requirement for any mission planning tool.

4. Adding Depth: Activities That "Do Something"

As discussed above, the hierarchical nature of an activity by itself already provides useful information about the activity. At some point, however, the user will want to add depth to activities beyond their relative placement within the hierarchy. At this point, the most relevant question for the user is, what can an activity do in addition to being part of the hierarchy? There are two answers to this question: an introspective answer that stays within the confines of APGEN, and a big-picture answer that looks at the entire uplink process for a space mission. To illustrate the difference, consider an activity such as a S/C maneuver. From the microscopic view of the APGEN display, a maneuver activity has a start time, a duration, and is displayed with a specific pattern and color on a specific legend of the display. From the macroscopic point of view, a maneuver (i.e. a Trajectory Correction Maneuver) is dictated by the need to maintain the actual S/C position and velocity to within a specified navigational tolerance of the ideal trajectory as defined by the mission plan. Deciding whether a maneuver is necessary requires analyzing telemetry and exercising navigation software. The actual execution of a maneuver involves an intricate ballet of actions that cause the S/C to turn off subsystems as needed, adopt the appropriate attitude, activate one or more thrusters for a specified length of time to achieve the desired velocity change, and resume nominal cruise operation. Verifying success of a maneuver requires still more telemetry and analysis. APGEN is a small cog on a big wheel--in order to explain the function of the cog, the function of the wheel must also be explained. Drawing on practical experience, the APGEN 'cog' fits into the 'wheel' of the overall space mission as follows:

1) <u>Mission Scenario Visualization Tool</u>

During mission operations, the sequences of commands executed by the spacecraft have been scrutinized, and finalized (cast in stone) prior to execution. To visualize what is taking place in space, data and commands embedded in the sequence must be extracted and displayed for the flight team on the ground. Special tools (including SEQ tools like SEQ_REVIEW) are available for that purpose. In the early phases of a mission the challenge is to create and display planning information in an intuitive manner. APGEN has been used by several missions in this capacity. Sometimes APGEN is used to display the contents of an existing integrated planning document, for instance the Cassini two-year cruise plan. Alternatively, APGEN can be used to express the contents of just one spreadsheet (e.g. Excel), which is of interest for Mars missions.

2) <u>Constraint Visualization Tool for Detailed Science Planning</u>

APGEN is not only used to display activities visually, but also to model impacts of those activities on resources of critical importance to science planners (on-board data storage, ranges to observation targets, electrical power, fuel, etc.) APGEN is very useful in this arena due to its (1) ease with which an adapter can express the impact of an activity on a resource, (2) effectiveness of the display in conveying a sense of where the plan is highly constrained and where it is flexible, and (3) ease with which a user can make changes in the activities and evaluate resource impacts.

3) <u>Sequence Building Tool to Support Mission Operations</u>

In this capacity, there is overlap between the roles of APGEN and SEQGEN (the main SEQ tool for sequence modeling and validation currently in use at JPL). From an adapter's perspective, there is not a big difference between a model (SEQGEN) that verifies the sequence conforms to all mission and flight rules, and a model (APGEN) that exhibits the effects of the sequence on mission resources. Some missions (MPL, for example) choose to extend the use of APGEN from planning into operations, and to only use SEQGEN as a batch-mode validation check after the sequence design has been finalized using APGEN. For such missions it would be beneficial if the adaptations of APGEN and SEQGEN were consolidated; at present, this remains a future prospect.

4) <u>Resource and Constraint Analyzer for Interactive Planning</u>

This method is implemented in the MER 3-D visualization system. In this mode, another application is in charge of defining the main elements of the plan (based on information that APGEN is not able to display). APGEN is invoked as a planning server with the primary function of evaluating impacts of the evolving plan on critical resources. This is a new development in the use of APGEN.

Now that the context in which APGEN is used has become more explicit in terms of the big picture, it is possible to delve further into the more introspective, APGEN-centric point of view. It is clear from the above examples that one of the most useful things APGEN does is to document, in an easily-viewable format, impacts of a proposed set of activities on S/C resources. This clarifies the characterization of APGEN as a "resource-based planner." In addition to displaying a timeline and letting the user interact with it, APGEN also displays impacts of the activities in the plan on important resources such as power, fuel, downlink opportunities, and data storage. In fact, the main difference between APGEN and a pure graphics package is that APGEN does modeling work to evaluate the impacts on the resources defined in the adaptation, while a graphics package only displays activities as symbols.

It should also be clear from the examples above that most activities relate to objects that lie outside the scope of APGEN. In other words, APGEN is not used in a vacuum. Well before an activity is dropped into the APGEN timeline, a lot of up-front work may already have occurred. At the other end of the process, the life of an activity extends well beyond its APGEN manifestation. For example:

1) Background Information

   Activities that convey useful background information to mission planners are usually imported from applications other than APGEN. Examples include DSN allocations, which indicate to the planner when a downlink could take place; activities that contain navigational information in the form of SPICE kernels; and activities that are extracted from spreadsheets.

2) Output Files

   Many activities need to be output in a format that is not APGEN's "native" APF format. The SASF format used by SEQGEN is an example. Such conversions are not automatic, but require careful coordination between the adaptations of the tools that are exchanging data.

3) Socket Communications

   More recently, users have been exercising APGEN's ability to communicate with other applications through sockets. There is not yet a standard in such communications, which tend to be based on the APGEN command language (ACL). In the future, applications (such as XML) would make socket-based applications much easier to develop and maintain.

B. What is a *Plan*?

1. Planning: The Small Picture

In the APGEN-centric context of activity planning, the "Plan" is whatever the user builds using APGEN. A key requirement on APGEN is that users have the ability to save the current instantiation of the plan, shut down APGEN, and then return later to open and update the plan. APGEN provides several ways of saving the current plan for later use/rework.

2. Planning: The Big Picture

> Plan: (...) **a**: a method for achieving and end **b**: an often customary method of doing something; PROCEDURE **c**: a detailed formulation of a program of action **d**: GOAL, AIM (...)

Stepping back to adopt a big-picture context, the "Plan" is a lot bigger than a simple user's file. In the very definition of the word *plan*, note the insightful reference to procedural and goal-oriented approaches. Clearly, a plan is oriented towards the future, probably contains instructions about how to get there, and is a reflection of a stated purpose. APGEN's relationship to the plan can be described in the following ways:

- □ Showing in explicit, clearly understandable form what has been planned so far,
- □ Importing files/data products from other applications to provide users with useful context for planning,
- □ Evaluating and displaying implications of the current plan,
- □ Capturing results of the plan in files/data products for export to other applications for further processing,
- □ Exercising algorithms for decomposing high-level activities into lower-level ones, and
- □ Scheduling activities based on availability of resources, as dictated adaptation scheduling algorithms.

3. Future Considerations: Automation

Activity planning is an area that undergoes rapid change due to its high interest amongst mission engineers and computer scientists. Using autonomous software to assist or replace human planners is a key element of interest in this changing field. In the future, it is likely that significant advances will occur in three major arenas: increased connectivity/automation of ground-based tools, increased on-board automation, and/or better validation tools.

Onboard automation continues to advance as space hardware follows the same expansion curve as the desktop in terms of performance and memory size. In addition, new tools such as Virtual Machine Language (VML) are becoming available to make S/C sequences smarter. From a GDS perspective, increased onboard automation will make it more difficult to predict what/when activities will happen. Event-driven sequencing, in particular, makes it extremely complex to enumerate all possible ways in which onboard events will unfold. The future likely holds one of two possibilities:

possibility A: Full-fledged validation of each S/C sequence will no longer be provided by the GDS. A few representative cases will be tested as a sanity check, but maintaining S/C safety will become the responsibility of onboard C&DH and FP subsystems. New validation techniques will be used to certify that the onboard software will maintain S/C safety in all circumstances.

possibility B: Enhanced ground-based validation tools will learn to deal with the uncertainty introduced by event-driven sequences. In particular, uncertainty in both timing and parameter values will be accommodated. Event-driven sequences will continue to be certified "free of violations" with respect to flight and mission rules. Planning and Sequencing Engineers will, over time, learn how to limit the uncertainty in S/C sequences to make complete validation possible.

These outcomes will impact ground software such as APGEN in very different ways. Possibility A could make the GDS much simpler, to the extent that the Science team could concentrate on giving the S/C high-level goals and letting the S/C handle the details. It remains to be seen if/how the S/C team could quantify for the Science Team just how hard the S/C would try to achieve these goals (Would any probably of success significantly lower than 100% be reasonable? Would “all possible outcome” scenario testing still be so important? Would any rational team really be able to trust the S/C to just do the right thing for one-shot critical mission events?). Today, whenever a S/C system is made more complex, the complexity of the GDS required to support it goes up, not down. Possibility B seems more realistic. The big unknown, however, is how much flexibility is left for event-driven commanding when one imposes the requirement of full validation. Finally, it is safe to assume that future missions will continue to insist on more collaboration, not just between planning and sequencing tools, but also among external systems such as web servers and relational databases. Science teams are almost always distributed geographically, many times in different countries. The file-based methods that in use today will need to be replaced by more sophisticated tools for archiving, updating, and distributing information about planning and sequencing data. Small steps have been taken along this path with SEQtalk, a socket-based mechanism for inter-process communications between APGEN and other applications. Significant development work is still needed in the activity planning paradigm.

III. Deep Impact Sequence Planning

THIS section introduces the Deep Impact Mission’s approach to sequence planning, development, validation, and uplink. The small budget, short timeline, and severely constrained engineering resources required the use of many existing multi-mission and activity planning tools. APGEN was used extensively throughout the project lifecycle, by the entire flight team.

A. Planning and Sequencing Challenges

Deep Impact was a challenging technical mission on a shoestring budget. Many spacecraft resources were over-subscribed. Power, thermal, telecom, and instrument sun avoidance constraints made attitude changes for TCMs and instrument calibrations difficult to implement. Navigation accuracy for targeting the spacecraft for a comet collision required trajectory correction maneuvers to be done with a very short development time, with execution shortly before impactor separation. The quantity of science and navigation imaging during encounter, encounter approach, and instrument calibrations required accurate modeling of instrument timing, image data storage, and downlink throughput.

B. Design Approach

The Deep Impact uplink process was based on the Multi-Mission process developed and used by a variety of missions operated by JPL’s Multi-Mission Operations organization. These missions included Mars Global Surveyor,

Mars Climate Orbiter, Mars Polar Lander, Stardust, Mars Odyssey and Genesis. This system has been proven to be highly functional and reliable, and can be adapted to new missions at a low cost. Deep Impact did not have the budget to build completely new Planning and Execution tools and processes.

To meet these challenges, it was necessary to use an existing sequence planning tool in a unique way which allowed the system engineers developing the sequences to have modeling of all major spacecraft subsystems available within the planning tool. APGEN was chosen as the sequence planning tool due to it's powerful, easy to adapt modeling capability and the ability to include external model libraries. JPL's Modeling and Simulation Technologies group provides multi-mission, reusable subsystem models which are easily integrated into APGEN. Deep Impact utilized the Power modeling library (MMPAT) and the Telecom Link Analysis modeling library (MMTAT). JPL's NAIF group provided multi-mission ephemeris evaluation tools used for calculating geometric quantities for space and ground assets. Adaptation and use of APGEN for Deep Impact began early in the mission development process. The earliest models implemented and used for design validation included C&DH data flow and attitude modeling. As the spacecraft design matured, models were updated to reflect spacecraft capabilities and were tied to actual spacecraft commands. This allowed for transitioning from high-level planning and mission/spacecraft design validation to modeling and validation of sequences used in the System V&V program. These sequences ultimately evolved into flight sequences. The DI process (Fig. 1) allowed for a smooth transition into Phase E flight sequence processing.

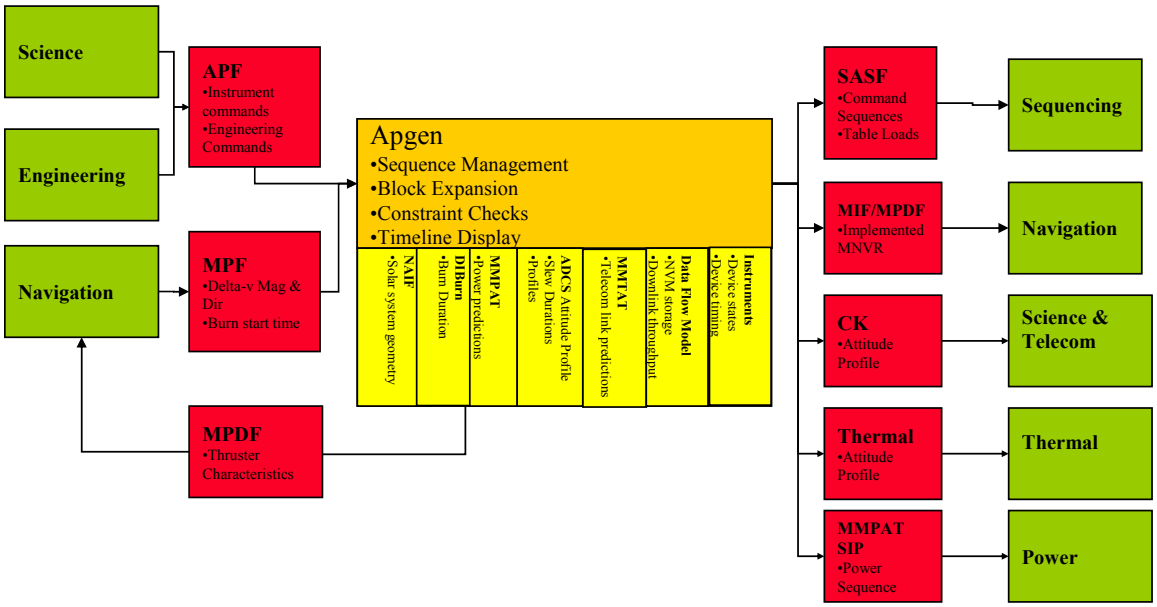

Figure 1. Deep Impact Sequence Planning Architecture

1. Attitude Modeling

The Flyby spacecraft attitude was modeled using APGEN's modeling adaptation language with algorithms supplied by the Deep Impact Attitude Control Team.[2,3] Attitude modeling was essential for Deep Impact in order to properly model power, HGA pointing, gimbal stop constraints, telecom link margin, star tracker constraints, and instrument bore sight constraints. Spacecraft position within the solar system was generated using the NAIF spice library and trajectory spice kernels generated by the Deep Impact Navigation Team. Attitude information was output in C-Kernel format and transferred to the Thermal Subsystem analyst and the Telcom Subsystem analyst for use in their subsystem modeling.

2. Power Modeling

Power modeling in APGEN was performed using the Multi-Mission Power Analysis Tool (MMPAT) modeling library. The MMPAT API was originally developed by the Modeling & Simulation Group with input from the Planning & Execution Group. MMPAT is easily integrated into sequence based modeling simulators such as SEQGEN and APGEN. MMPAT has been used successfully by Deep Impact and the Mars Exploration Rovers. Having MMPAT integrated into APGEN for Deep Impact allowed for a number of improvements in the pre-launch operations development process and flight sequence development. When MMPAT is run through APGEN, Power subsystem characteristics are immediately available to the Systems engineer generating the sequence plan (Fig. 2).

This allows the sequence planner to validate power subsystem operations without having to iterate with the Power subsystem engineer. After the plan is complete, APGEN, using the MMPAT API, generates a power sequence input file which can be transferred to the stand-alone MMPAT tool and processed by the Power subsystem analyst to validate the operational scenario.

Deep Impact had sufficient solar array power for virtually all mission phases. Once the launch sequence was complete and the solar arrays were deployed, power was never a major concern for operations planning and could generally be simplified in terms of limiting spacecraft attitude during off-nominal operations, such as Trajectory Correction Maneuvers (TCMs). The MMPAT runs through APGEN were useful as a final validation of the sequence design. The global characteristics of the encounter attitude design were fixed early in pre-launch development and power was never a concern during post-launch encounter operations planning.

3. Telecom Link Modeling

Telecom link modeling in APGEN was performed using the MMTAT modeling library. This library has a similar API to the MMPAT power modeling library. This made it relatively simple to integrate with APGEN. The adaptation for MMTAT was generated from telecom performance estimates provided by the spacecraft contractor. MMTAT proved to be very useful, as it is a sequence driven model, and can react to changes in configuration for both the spacecraft and the DSN.

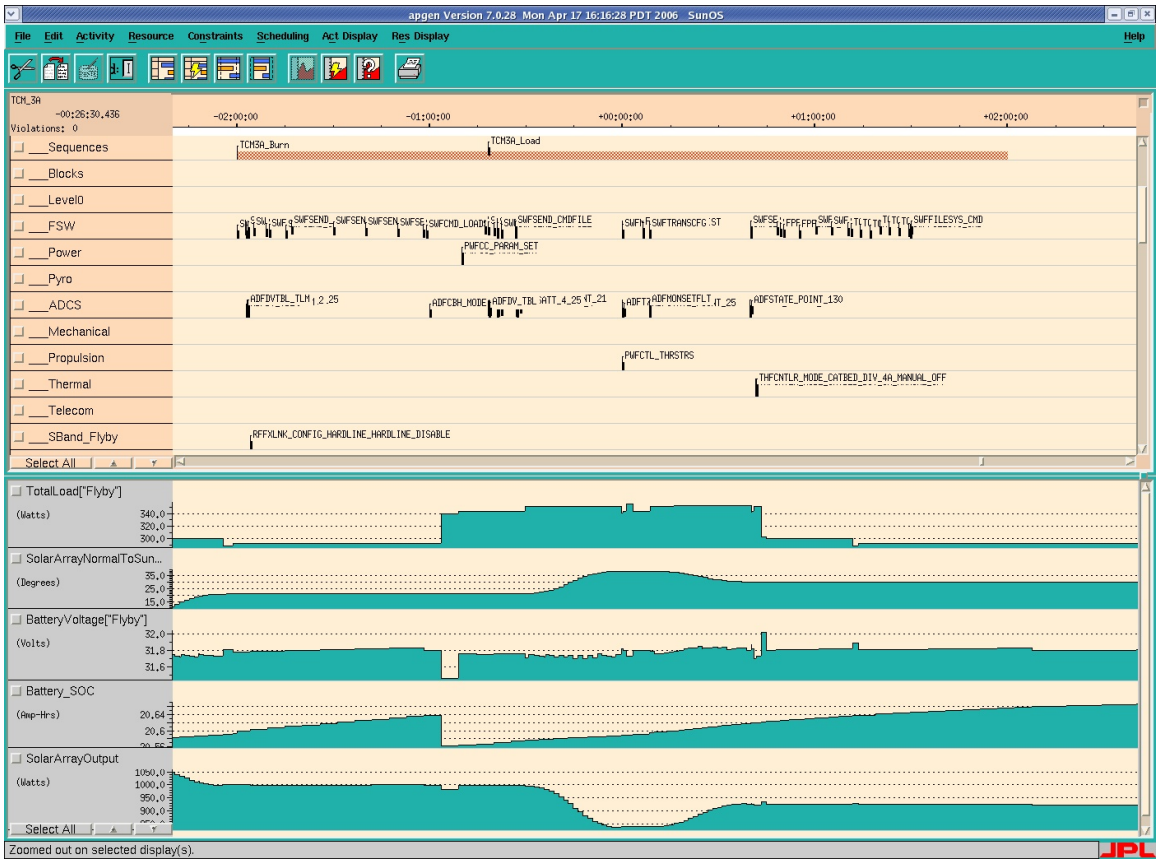

Figure 2. TCM-3A Power Profile

4. Data Flow Modeling

The data flow model was especially important for Deep Impact, due to the large amount of science and navigation imaging required for mission success. In addition, the oversubscription of critical resources such as non-volatile memory and downlink bandwidth, and the necessity to share the imagers between science and navigation were also major concerns. Figure 3 represents the flow of data through the Deep Impact flyby spacecraft. The data flow model implemented in APGEN accurately modeled each point in this process. Early use of the model during S/C development resulted in predictions that the flight software buffer scheme would run out of image buffers during nominal encounter imaging operations. This resulted in a change in the flight software buffer management design. Accurately modeling the data flow to this fidelity proved to take too long modeling large imaging sequences,

and the fidelity was reduced somewhat to improve performance. Modeling of the data at the buffer level was removed after the new flight software buffering algorithm was implemented and tested with flight like sequences. The modeling of image data volume and end-to-end throughput (all the way to the ground) was maintained. The ability to predict NVM usage, D/L queue depth and the location of an image file through the system was critical for validating the complex encounter and encounter approach sequences prior to validation on the testbed. It was impractical to run the entire encounter approach sequence on the testbed, due to its 8-day duration. The encounter sequences were revised hundreds of times prior to the final flight versions, and each of these revisions needed to be executed on the testbed. The fidelity of the data flow modeling allowed for validation of the imaging sequence design prior to spending the 30+ hours on final testbed validation. Figure 4 is an APGEN screenshot of the encounter data flow.

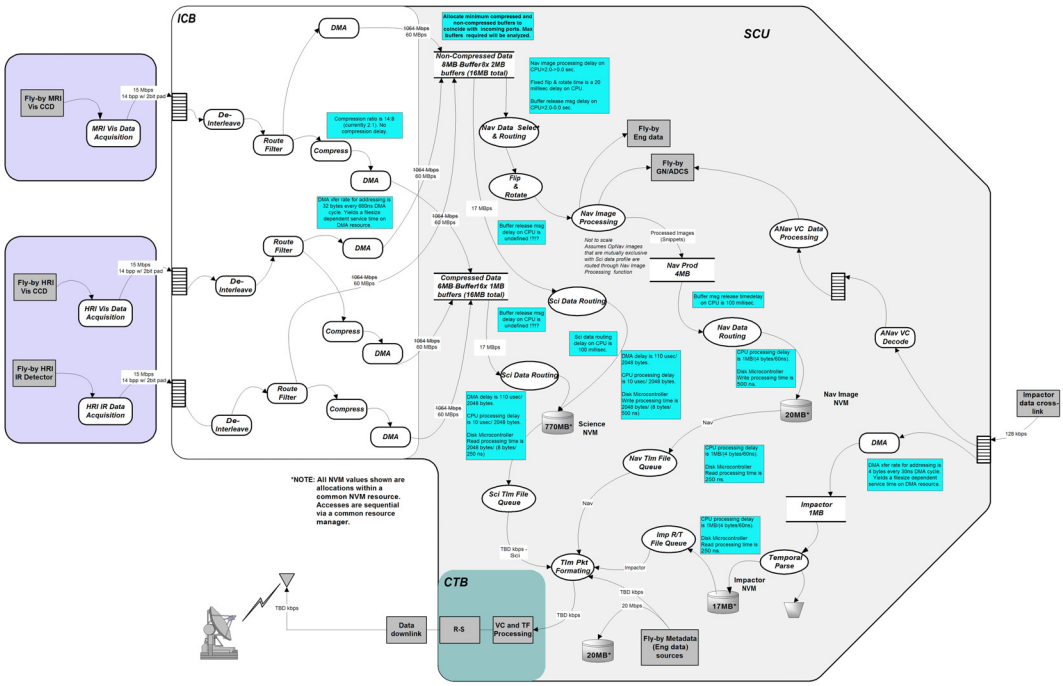

Figure 3. Science data flow through Deep Impact C&DH

5. Ground Station and Operations Process Modeling

Ground Station state and resource modeling was generally performed internally using APGEN's modeling adaptation language. DSN and optical observatory positions are generated using SPICE kernels and libraries through APGEN's User-Defined Library interface. During early mission development, standard DSN products used in operations planning are not always available. A multi-mission APGEN model of the DSN was generated for Deep Impact which could be used to generate DSN viewperiods and station allocations. These could then be used to generate DSN activities such as uplink and downlink signal acquisitions, loss of signal events and doppler mode changes. Combined with a modeled representation of the Mission Operations Process, these proved useful for generating operational scenarios for PDR and CDR, as well as planning uplinks for critical activities. Figure 5 shows an APGEN screenshot from the MOS CDR presentation on encounter operations scenarios.

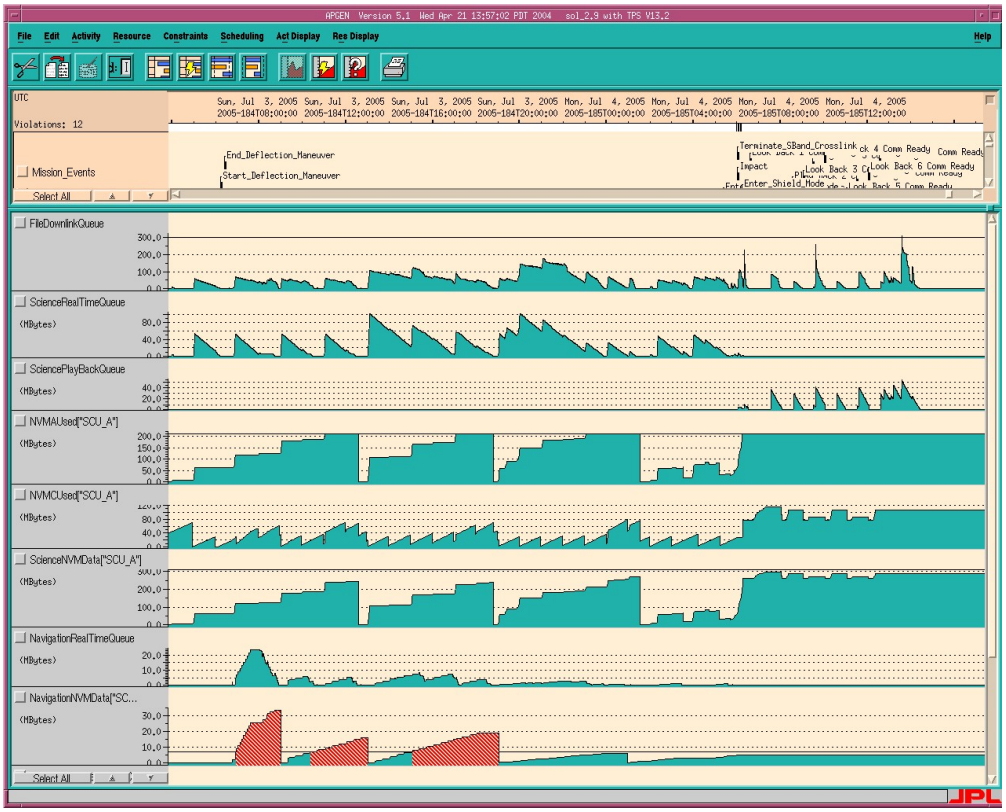

Figure 4. Apgen screenshot of encounter data flow

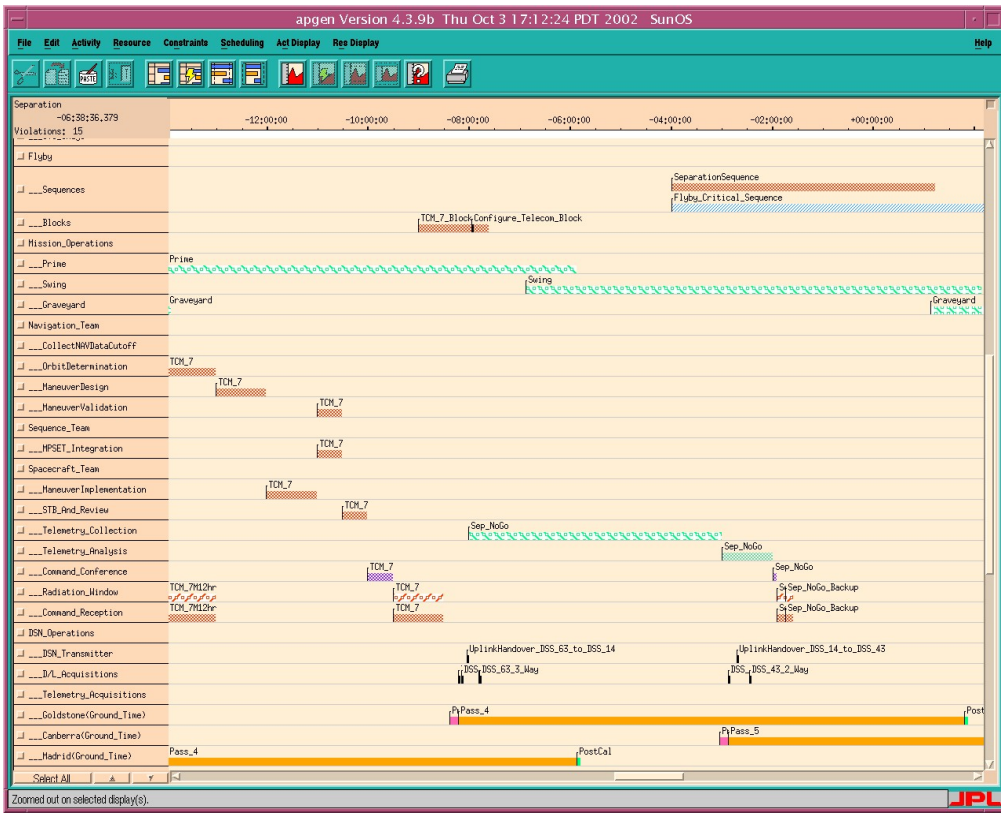

Figure 5. Separation Operations Timeline for MOS CDR

6. Trajectory Correction Maneuver Design

Deep Impact's requirements for delivery accuracy of the Impactor to Tempel-1 (30 KM B-Plane) at time of release placed unusual time constraints on the Maneuver Design Process. The maneuver execution time was 6 hours before Impactor release, and the tracking data cutoff to Navigation team was 11 hours prior to maneuver execution. This short timeline for maneuver design was the driving factor for having a sequence planning tool with all required S/C models for sequence validation. This allowed the systems engineer responsible for designing and building the sequence products to build a sequence that would not violate any mission constraints, without time-consuming iteration with each subsystem engineer's stand-alone analysis tools. Figure 6 represents the operation process timeline for the maneuver design process leading up to impactor release.

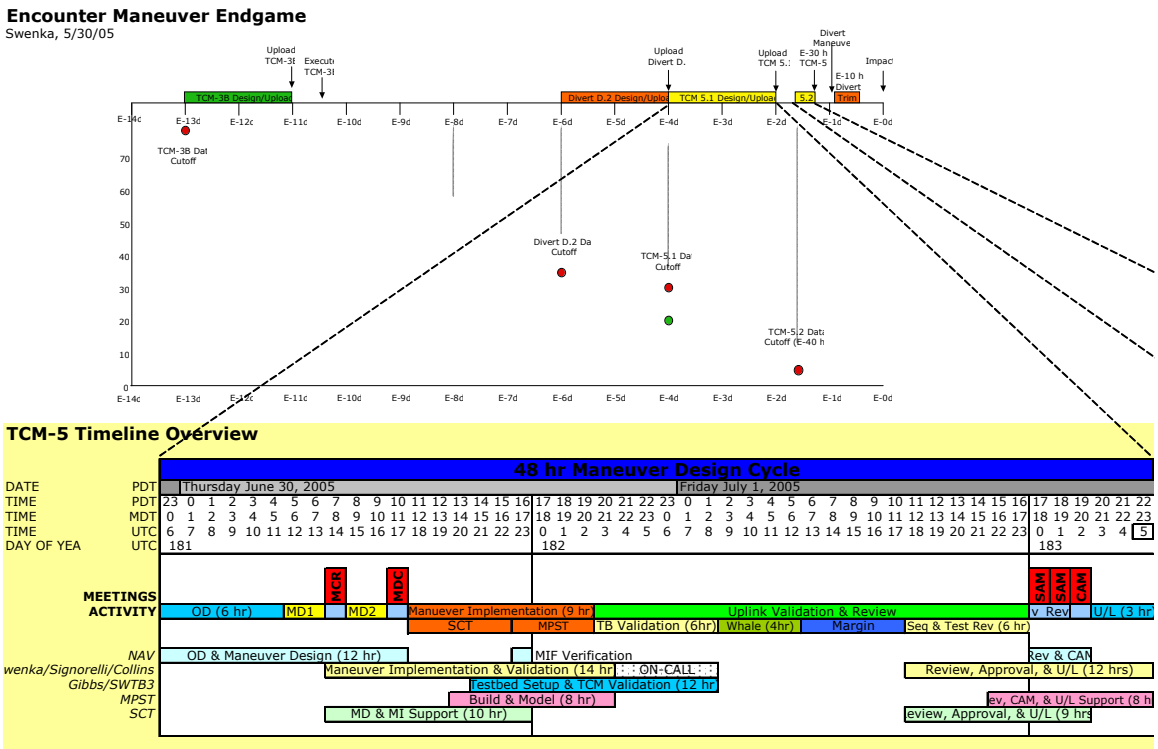

Figure 6. TCM-5 Maneuver Design Timeline[5]

C. Model Initialization & TestBed Set Up

Most Deep Impact sequences needed to be run on the hardware testbed prior to uplink to the spacecraft. The testbed initialization process involved powering on the testbed, initializing the dynamics simulation, executing the pre-launch commanding procedure, running the launch sequence, and then running a sequence to take the testbed to a flight-like state appropriate for the sequence to be simulated. This initial-conditions sequence was generated using APGEN as a by-product of getting the APGEN simulation into the correct state to model the sequence. To facilitate the generation of this conditions sequence, a re-usable parameterized APGEN activity was built which the systems lead building the sequence could use to initialize the APGEN model and to generate the testbed conditions sequence. The two conditions activities were: (1) a generic activity that included configuration parameters; and (2) an activity which took a single mission phase parameter, and supplied a canned set of parameters to the generic activity.

IV. Model Validation and Results

DEEP Impact made extensive use of GDS modeling tools in the sequence planning and validation process. This section illustrates how well the S/C performance, as predicted by models, tracked to the actual in-flight performance for all key technical resources.

A. Attitude Model

The following plots of model vs. spacecraft telemetry show the attitude model to be more than accurate enough for sequence planning purposes. Of key importance is the accuracy to check slew durations and instrument line-of-

sight sun avoidance constraints. In Figs. 7 and 8, the body rate prediction vs. performance during the slews is very good. The rate excursions during the burn in Fig. 7 can also be seen. Figures 9 and 10 show high-gain antenna gimbal angle prediction vs. performance. This model allowed the systems engineer planning the sequence to determine if the high gain antenna could track the Earth, or if a switch to the low gain antenna was required.

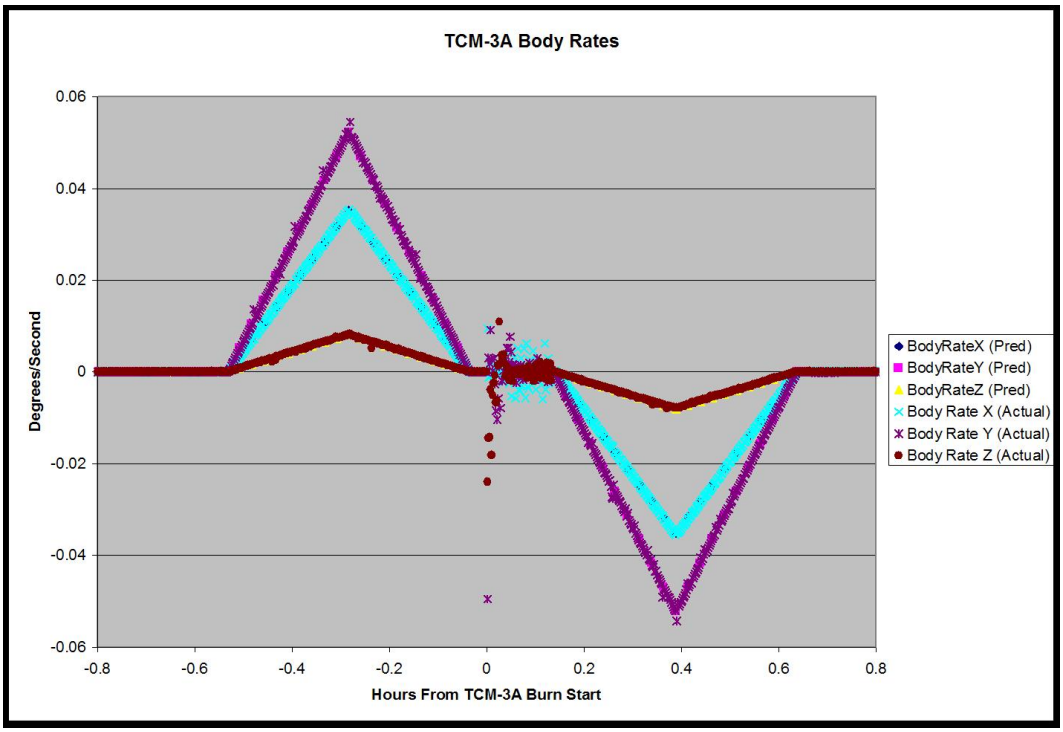

Figure 7. S/C Body Rate modeling performance for TCM-3A

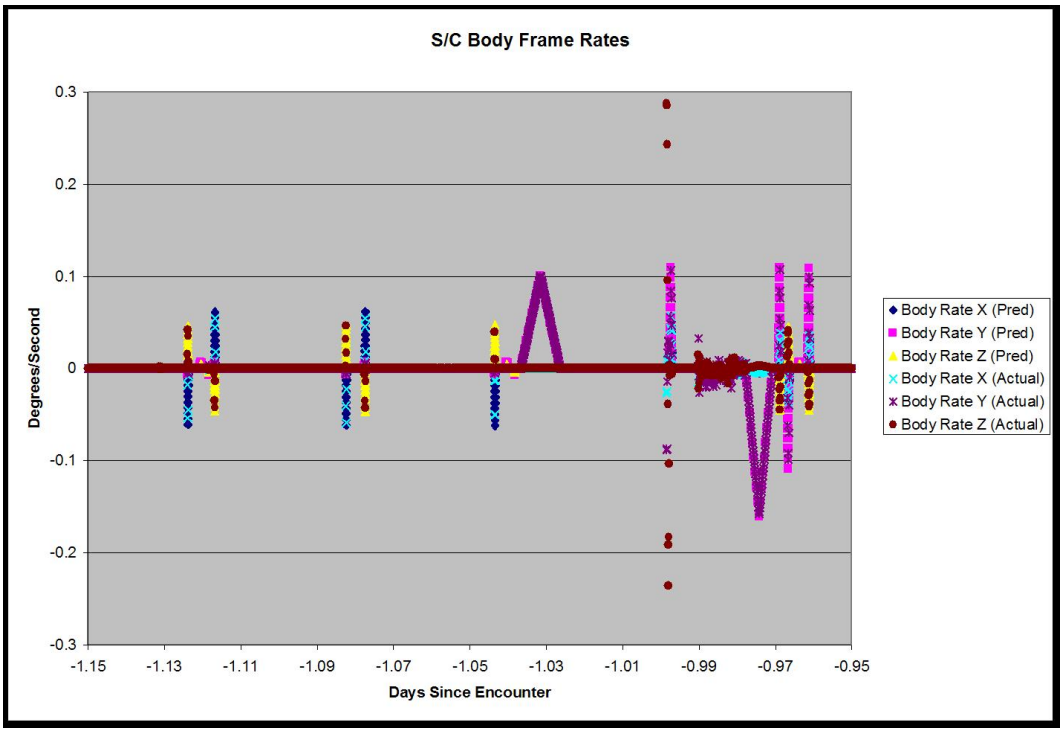

Figure 8. S/C Body Rate modeling performance for Encounter (Final 2 hours)

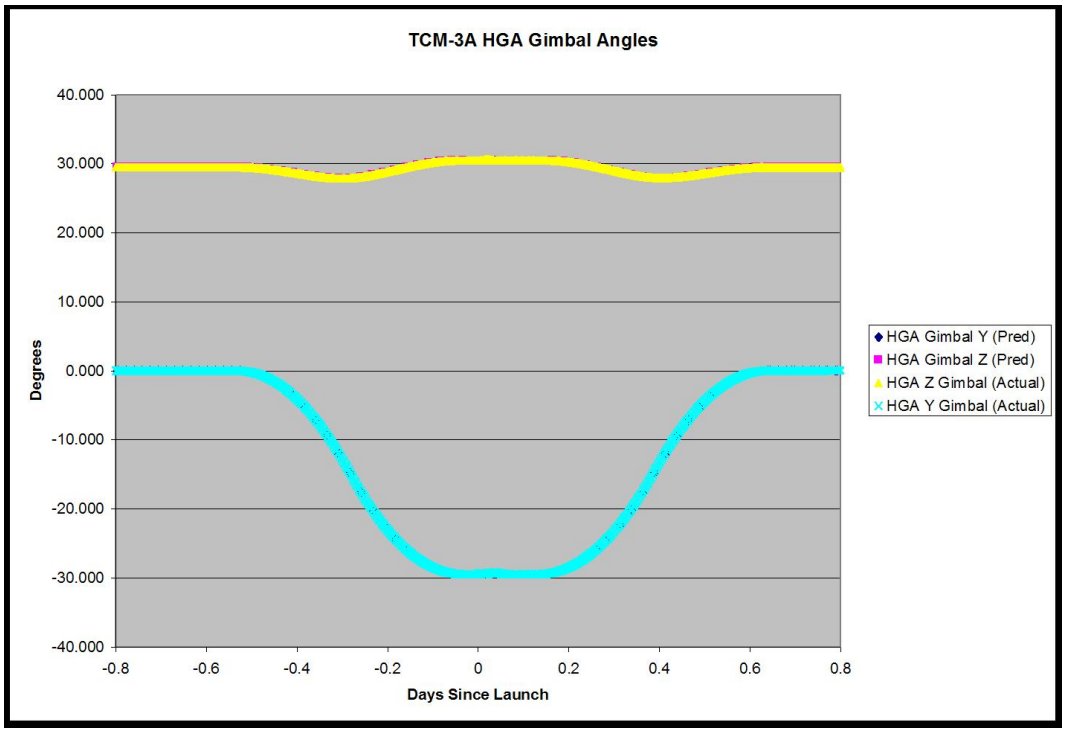

Figure 9. S/C HGA Gimbal Angle modeling performance for TCM-3A

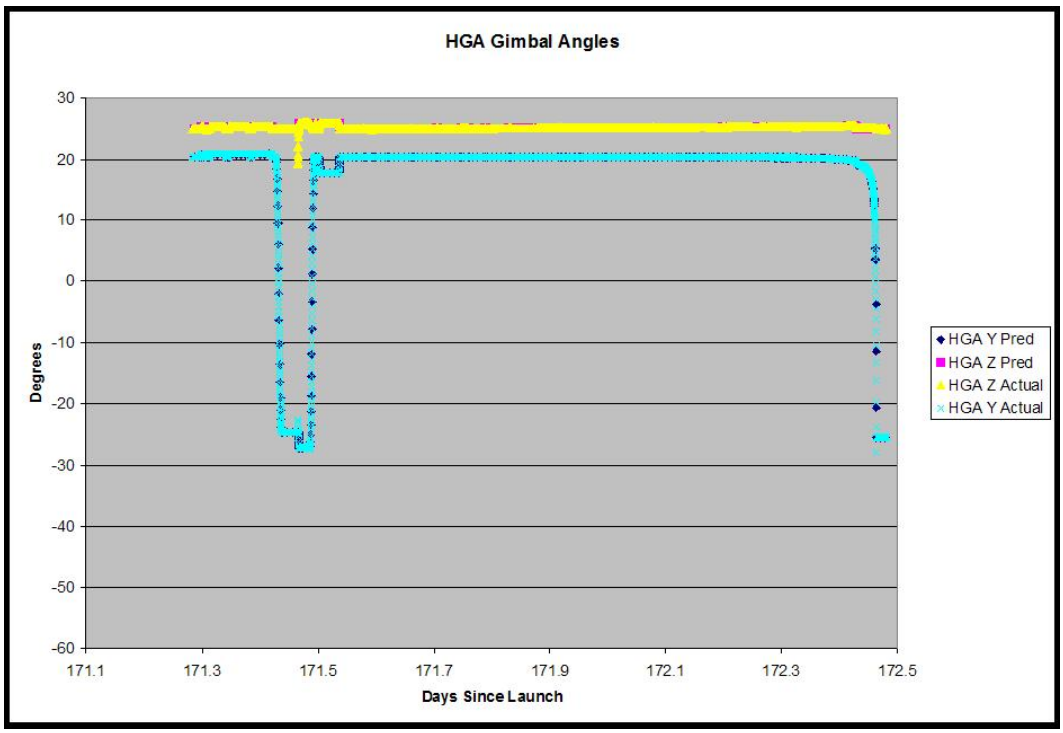

Figure 10. S/C HGA pointing modeling performance for Encounter

B. Telecom Link Model

Figure 11 shows a plot of the MMTAT performance for the first 60 days of the mission using subcarrier power to noise as the figure of merit. The predicted SubPDNO is compared to DSN monitor data collected during tracking passes. The overall performance was good, predicting a slightly higher signal level, until adjustments to the model

were made post-launch, based on actual telecom performance. Large variations in performance were tracked to such events as a snow storm at the Madrid tracking station which caused a large reduction in the X-Band signal strength.

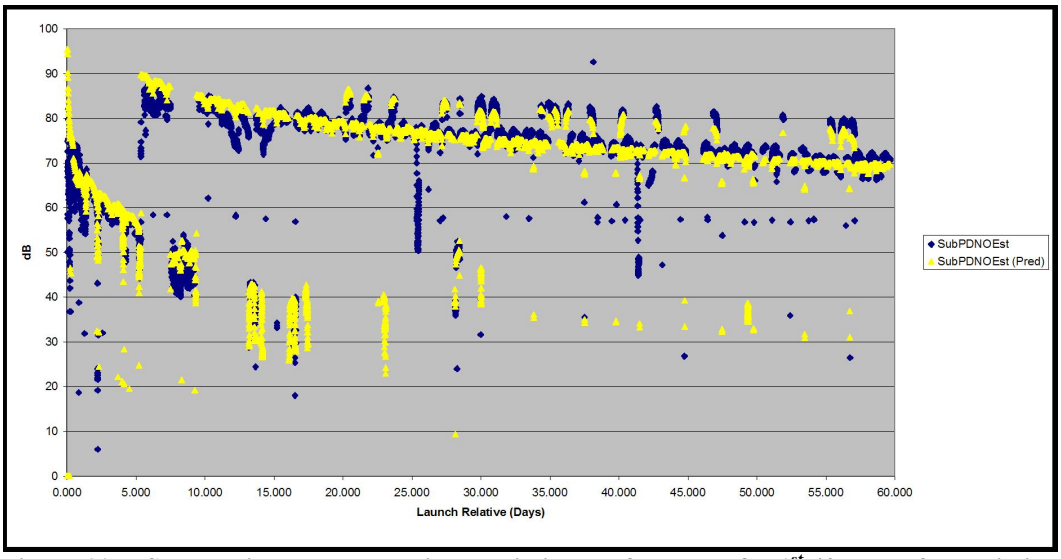

Figure 11. Sub carrier Power to Noise prediction performance for 1st 40 days of the mission

C. Data Flow Model

Figures 12 and 13 represent a key data flow parameter, the number of files in the downlink queue, a good overall representation of the accuracy of the data flow model. It can be seen that the model predicted value matches telemetry from the testbed run, and the S/C telemetry from encounter. A FSW error resulted in missing data from the testbed run between about 23 hours to 15 hours before encounter. There were some short periods as seen in figure 15 where the model and telemetry deviated. No root cause for this discrepancy has been identified, but the problem self-corrected in a short amount of time.

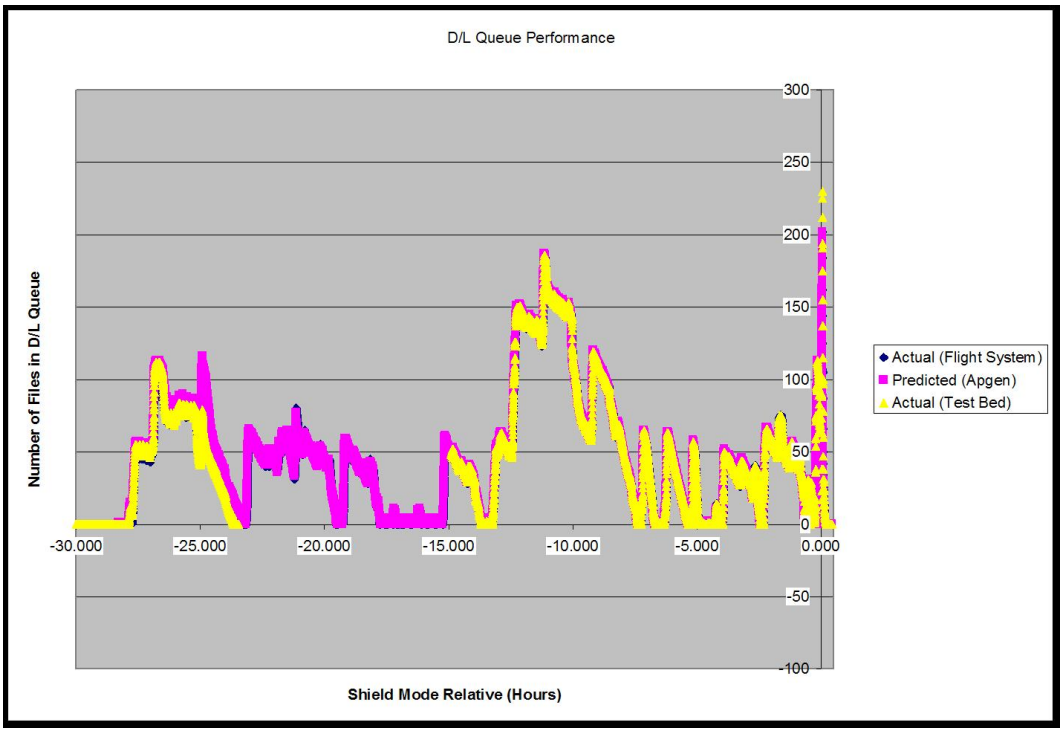

Figure 12. Downlink Queue Performance for Final 30 hours before Impact

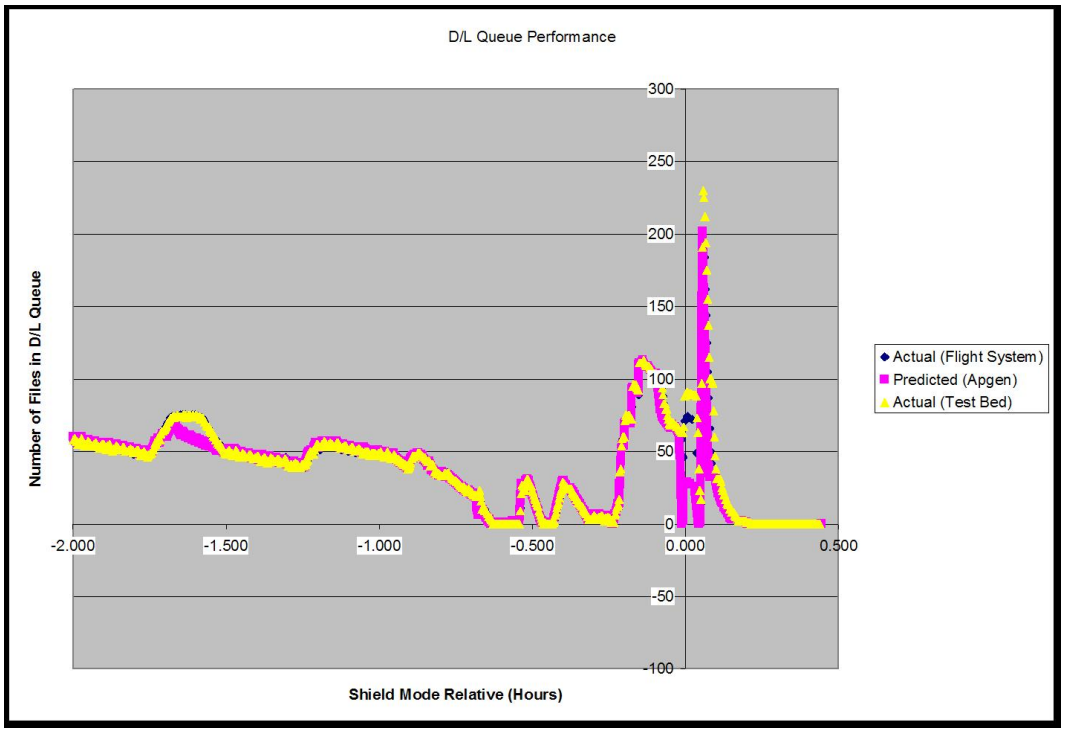

Figure 13. Downlink Queue Performance for Final 2 hours before Impact

V. Operations Experience

THE operations phase for Deep Impact spanned a very short 6 months from launch to end of mission. For each of the critical mission phases, multi-mission tools, as adapted for DI, were used extensively in the planning, development, validation, and execution of all in-flight activities. This section considers each key mission phase in turn, from launch through encounter. The experiences of the operations team, from concept to execution is addressed, including discussion of key lessons learned.

A. Launch

Deep Impact began its journey to comet Tempel 1 on January 12, 2005. This section introduces the first of DI's operational phases, including discussion of the operations team experience during the development, test and execution of the first two days of the Deep Impact mission.

1. Mission Phase Description

The Deep Impact launch phase[4] began during countdown when the spacecraft transitioned to internal battery power and ended when the spacecraft achieved 3-axis stabilized control ("Point State"), one day after launch. The major activities of this mission phase included liftoff and boost of the launch vehicle, insertion into a circular park orbit, a coast period followed by a two-step hyperbolic injection burn, separation of the spacecraft from the launch vehicle, solar array deployment, detumble, initial acquisition by the Deep Space Network (DSN), establishing 2-way telemetry, and playback of the launch data. Launch of the Deep Impact spacecraft was executed by a Boeing Delta II 7925 launch vehicle from Space Launch Complex 17 at Cape Canaveral Air Force Station in Florida. The launch period for the Deep Impact mission spanned from 12 January 2005 to 28 January 2005. There were two launch opportunities each day of the launch period.

Figure 14 presents a timeline of the spacecraft launch sequence, which initiates post-separation spacecraft events. At separation, breakwires provide the signal to the Flyby flight software to initiate the launch sequence without ground command. The Deep Impact launch sequence is classified as a critical sequence, indicating that fault protection can autonomously initiate numerous retries of the sequence and reconfigurations of redundant hardware until a proper safe, power-positive, and stable spacecraft state is achieved.

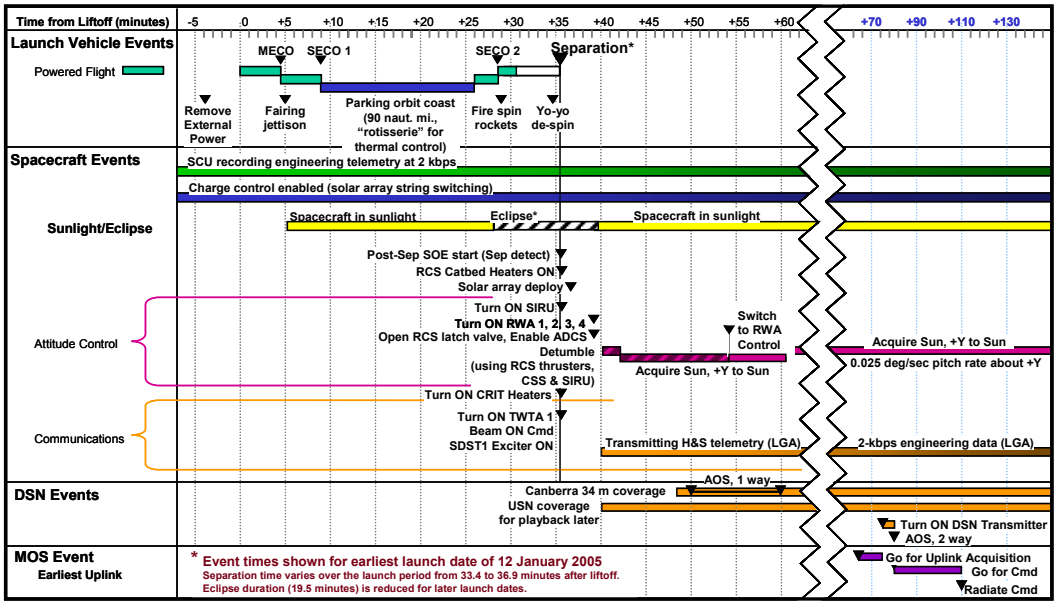

Figure 14. Launch Sequence Timeline

At the end of the launch sequence, about 25 minutes after separation, the flight system coasts away from the Earth with the solar array deployed, transmitting engineering telemetry at 2 kbps to the Earth over each of the low-gain antennas, and rolling slowly about the sunline (+Y axis) at a rate of 0.025 deg/s. To establish reliable ground commanding of the spacecraft, an incremental initial checkout of telecom systems is performed, culminating in 2-way coherent downlink, with ranging modulation, and command modulation on. Once basic commandability is verified, playback of launch telemetry begins, followed by turn-on and checkout of the star trackers. On the morning of the day after launch (L + 1 day), "Go To Point" attitude control activities begin to bring the spacecraft under 3-axis stabilized control. This procedure takes 6 to 8 hours to complete and involves assessing the performance of the star trackers, monitoring convergence of the attitude estimator, validating the orbit state, and turning to the first 3-axis stabilized attitude (0° tip angle). Upon confirmation of acceptable ADCS behavior, the spacecraft is commanded to "Point State, Target 130" (30° tip angle), a "preferred" 3-axis attitude. Successful achievement of Target 130 marks the completion of the launch phase, as shown in Fig. 15.

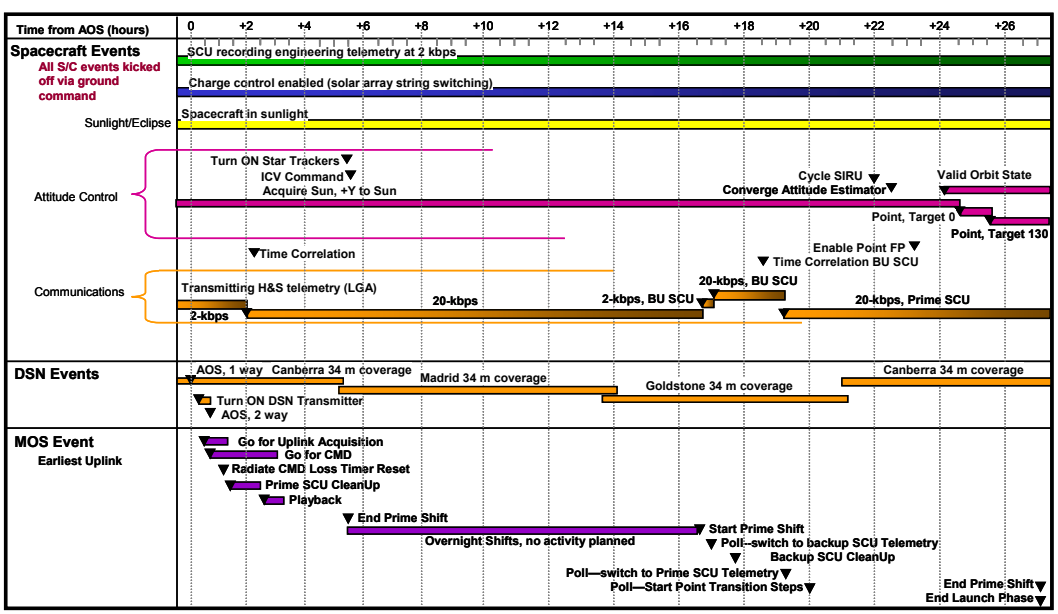

Figure 15. Count-Up and Point State Acquisition Timeline

2. Operations Experience

The launch sequence served as a pathfinder for Deep Impact. This sequence was the first to execute on the testbed, the first to use APGEN for its design, the first to fully enable fault protection in modeling or in test, and the first to run on the backup flight computer. Many of the processes and tools were vetted during the development of launch, which was the most tested sequence on the project.

Modeling tools proved essential during launch phase development, as a major concern was uncovered leading to key design changes. APGEN attitude modeling, with constraint checking enabled, showed that the planned launch vehicle trajectory could place the spacecraft such that the instrument bore sights could point directly at the sun. This would have prevented mission success. The modeling run uncovered this issue far ahead of standard engineering analysis of test runs, allowing re-design early on. The modified design was also modeled, showing quick resolution of the concern once a candidate solution was identified. Of key concern during launch phase development is the available time after launch that the spacecraft can remain powered via its internal battery in the event of a fault. The integrated MMPAT model allowed APGEN to predict the available time, without testbed execution and standard engineering analysis. This proved extremely valuable, as several key project reviews leading up to launch requested additional analyses in this area—all of which were easily accommodated with the available tool set. In addition, as late as the launch readiness review (just days prior to the first launch date), changes to the launch sequence products were required for every opportunity due to a slightly heavier final launch vehicle mass. With multi-mission tools in place, the flight team was able to modify, regenerate, and model sequence products for all opportunities in just a few days.

Deep Impact launched successfully on its first attempt, 12 January 2005. During execution of the launch sequence, a 'false alarm' Remote Interface Unit fault occurred due to improperly specified Catbed heater sensor settings. The improper setting caused fault protection to execute hardware swaps, which interrupted thruster activity during detumble. As a result, spacecraft safing occurred during the launch sequence. Due to rapid modeling and test execution prior to launch, this scenario had been fully executed by the flight team during an Operational Readiness Test multiple times, and the in-flight anomaly was smoothly resolved. The ability of the team to quickly identify, propose a solution, and test it was enhanced by the multi-mission tool set and experience with those tools for DI. The full set of planned activities, ending at Point Target 130, were completed only a few hours later than expected, even with the safing event.

B. Instrument Calibrations

Calibration activities of the four instruments (HRI, MRI, HRI_IR, and ITS) began in mid-January and ended in mid-July, spanning all primary mission phases except Launch. The calibration activities fell into four categories – science, autonomous navigation (autonav), optical navigation (opnav), and engineering (such as star-tracker-to-instrument alignments). This discussion focuses on the science calibration activities performed during the Commissioning phase, although the experiences were common throughout all phases.

1. Mission Phase Description

The Commissioning phase was formally defined in the Mission Plan[4] as ending at Launch + 30 days. Originally, Deep Impact was to have been a 1.5 yr mission that included an Earth flyby, so that only spacecraft engineering activities, such as subsystem checkouts and a TCM, were in the first month after launch. At liftoff, however, the Deep Impact primary mission was only seven months in duration with no Earth flyby. The effect of this was to move several of the instrument calibrations into Commissioning, particularly those using the Moon as a target. As a result, the Commissioning timeline was fairly dense, with activities extending to about Launch + 33 days (Fig. 16). Due to unforeseen events (Table 3), several commissioning science calibrations were actually performed well into Cruise.

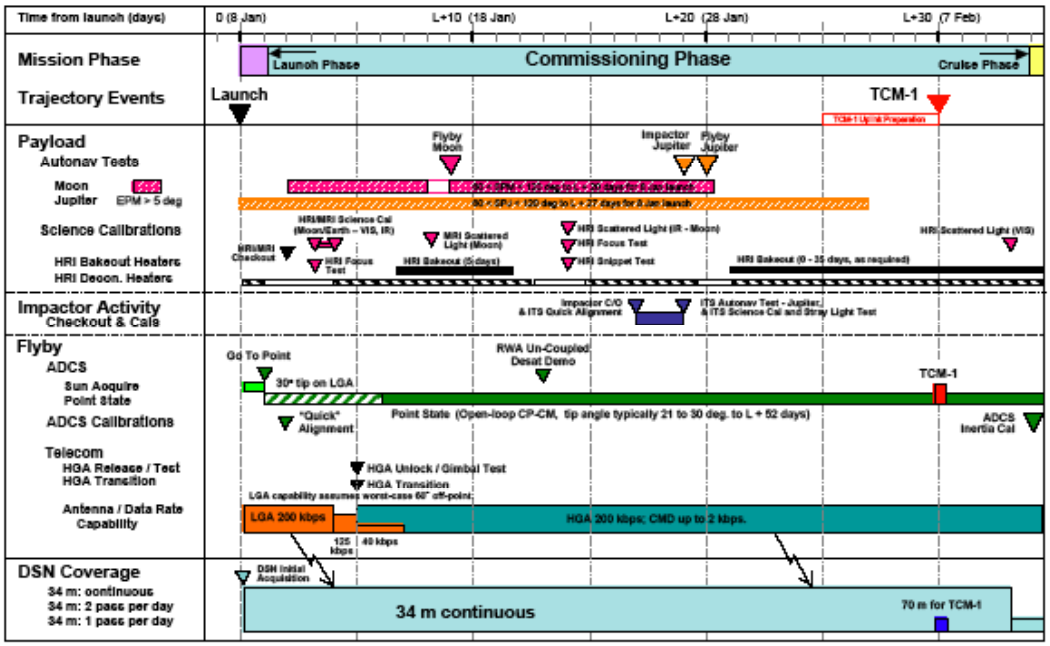

Figure 16. Deep Impact Commissioning Phase Timeline

2. Operations Experience

To provide the maximum amount of flexibility, an early decision was made to perform all science calibration activities as relative-timed sequences instead of absolute-timed sequences. This turned out to be a useful decision as shown in Table 3, which lists the first 30 days of the mission as executed and as planned. In the table, the following cell color scheme is used: blue - launch phase, yellow – margin, green – reaction wheel desaturation opportunity, and orange – anomalies. Science calibration activities are highlighted in blue text. The three days of margin leading up to the end of Commissioning were used to prepare for a series of unplanned activities to try to diagnose and correct the HRI focus. The first activity in this set went from initial idea to spacecraft execution, performing all testing and review requirements, in seven days. It executed on the spacecraft 34 days after launch. The HRI focus problem was ultimately diagnosed, but the corrective action is being performed now in imaging post-processing by the science team (no correction was possible on the spacecraft).

In preparation for performing science calibrations on the spacecraft, APGEN and other tools were relied on extensively to create the SASF files needed by sequencing to generate flight products and to provide the sub-system leads with the appropriate input products for their review tools. The science team provided the initial science calibration inputs in Excel files. These files were very complex, with multiple worksheets (one per instrument used in the calibration and one for pointing at a minimum) and 84 columns and dozens to hundreds of rows per worksheet. Each worksheet could contain commands for several different sequences (5-10 was the norm for imaging worksheets). The process for converting the raw input from Excel to the AAF or APF files that APGEN needs for input involved several tools developed for this purpose. Once APGEN was properly set up for an activity (a multi-step process in and of itself), then the tool was run to model the activity and generate SASF files for the sequencing team. Once a clean modeling run was achieved, data products for sub-system review were generated using additional tools.

Because the DI adaptation for APGEN was primarily set up to support Encounter, there were a few limitations in its capability to support activities in other phases of the mission. Two of them were particularly cumbersome, but with ultimately no effect on activity success, when working on the science calibration sequences:

- When generating relative-timed sequence SASF, APGEN would use a variable (related to the epoch provided for the modeling run) plus the relative-time offset in place of an absolute-time. This output format did not impact the generation of the flight sequences themselves, but it was incompatible with the PEF-generation tool - and all of the subsystem engineers were required to review PEFs for each activity. The solution to this problem involved a minor modification of the PEF-generating tool by the sequencing team

and quick, specific hand-editing by the activity lead for each affected sequence. It was an inelegant but sufficient work-around for such a short mission.

- APGEN could not distinguish between data downlink commands for SCU A and SCU B. Since all of the data-intensive science calibrations involved storing and downlinking non-redundant data from both SCU's, it was not possible to use the data flow modeling feature. As a work-around, outside calculations were performed so that the backbone sequences were timed with sufficient margin to allow data playback before the next segment of the calibration would begin.

Even with the limitations, it would not have been possible to perform the number of science calibrations desired without the capabilities that APGEN provided. An example of this is in the use of APGEN to reduce required testbed time. The Deep Impact project was burdened by an inefficient testbed architecture that added 2-6 hrs of setup time to every test. This time had to be added to the 10-36 hrs typically required for the science calibration sequences to run. Unfortunately, competition to get on the schedule for one of the four testbeds was intense, especially with both flight activities and Encounter development activities vying for large blocks of time. Using the APGEN flight rule and constraint checking was particularly useful for identifying problems in the few minutes of a modeling run, instead of the several hours it would take on a testbed. As a result, most of the science calibration sequences had only one run on a testbed – the required "run for record" for sub-system review. This not only enabled several calibrations to meet their spacecraft execution deadlines, but also freed up significant testbed time for development work on the critical Encounter sequences.

Table 3. Deep Impact Commissioning Activity Timeline – Actual vs. Planned

Date	Days from Launch	Actual	Pre-Launch Plan
1/12/05	0	Launch Day 1, Safe Mode Entry	Launch Day 1
1/13/05	1	Safe Mode Recovery, Go to Point	Launch Day 2 - Go to Point
1/14/05	2	HRI/MRI Checkout & Quick Alignment	HRI/MRI Checkout & Quick Alignment
1/15/05	3	Prep for Science Cal	HRI/MRI Science Moon/Earth Cal (VIS & IR) Part 1, HRI Focus Test
1/16/05	4	HRI/MRI Science Moon/Earth Cal (VIS & IR) Part 1, HRI Focus Test	HRI/MRI Science Moon/Earth Cal (VIS & IR) Part 2
1/17/05	5	HGA Gimbal Unlock & Checkout, Transition to HGA	HGA Gimbal Unlock & Checkout, Transition to HGA
1/18/05	6	Margin	Margin
1/19/05	7	FSW Patch 6.2.4, Solar Flare-Induced Safe Mode Entry	Margin
1/20/05	8	Safe Mode Recovery	MRI Stray Light Test
1/21/05	9	Safe Mode Recovery	Flyby Autonav Test - Moon
1/22/05	10	Margin	Margin
1/23/05	11	Margin	Margin
1/24/05	12	HGA Gimbal Anomaly	Margin
1/25/05	13	MRI Stray Light Test - MRI Turn-Off Anomaly	RWA Uncoupled Desat Opportunity
1/26/05	14	Flyby Autonav Test - Moon	HRI Focus, IR Scattered Light, HRI Snippet Test
1/27/05	15	Margin	Margin
1/28/05	16	MRI Stray Light	Margin
1/29/05	17	HRI Focus, IR Stray Light, HRI Snippet Test, IR Lunar Scan	Impactor Checkout, ITS Quick Alignment
1/30/05	18	Margin	Margin
1/31/05	19	Margin	ITS Autonav Test - Jupiter,ITS Science Cal & Stray Light Test
2/1/05	20	Impactor Checkout - Lite	Flyby Autonav Test - Jupiter
2/2/05	21	RWA Uncoupled Desat	Margin
2/3/05	22	Delta DOR	Margin
2/4/05	23	IR Lunar Scan	Margin
2/5/05	24	Flyby Autonav Test - Jupiter	Margin
2/6/05	25	Margin	Margin
2/7/05	26	Margin	TCM-1 Preparation
2/8/05	27	FSW Patch 6.3.0 Flyby File Uplink & SCU-B Install	TCM-1 Preparation
2/9/05	28	FSW Patch 6.3.0 Flyby SCU-A Install & SM Recovery	TCM-1 Preparation
2/10/05	29	TCM-1 Uplink Window	TCM-1 Uplink Window
2/11/05	30	TCM-1 Execution	TCM-1 Execution
2/12/05	31	Margin	TCM-1B Execution (if necessary)
2/13/05	32	Margin	Margin
2/14/05	33	Margin	HRI Stray Light Test

C. Encounter Approach

The final week prior to comet encounter was designated as the Encounter Approach Phase. This densely packed week consisted of the first long-duration parallel operation of both Flyby and Impactor spacecraft, several mission enabling maneuvers, and an ambitions imaging campaign including a mission critical science calibration. This section presents a short description of this phase. In addition, flight team experience building and flying the longest week of DI's mission is discussed.

1. Mission Phase Description

The week of Encounter Approach activities spanned from Impact minus eight days to Impact minus 28 hours. The last week prior to encounter was one of the most challenging of the mission due to the vast scope of activity that had to be accomplished so close to the impact event.[1] Key elements of Encounter Approach included: activation of the critical encounter sequence, maneuvering to continuous comet-pointing attitude, extensive optical navigation and science imaging sequence execution, divert maneuver design and uplink, final science point spread function and infrared mini-calibrations, Trajectory Correction Maneuver design and uplink, and final preparation of the Impactor spacecraft for separation and its subsequent free-flight to Tempel 1. In addition to the major elements noted above, regular maintenance activities were also performed during Encounter Approach, including regular science and navigation image downlinks, Impactor Targeting Sensor (ITS) imaging and downlinks, delta-DOR, and Impactor engineering telemetry playback. Figure 17 depicts the phase timeline for Encounter Approach.

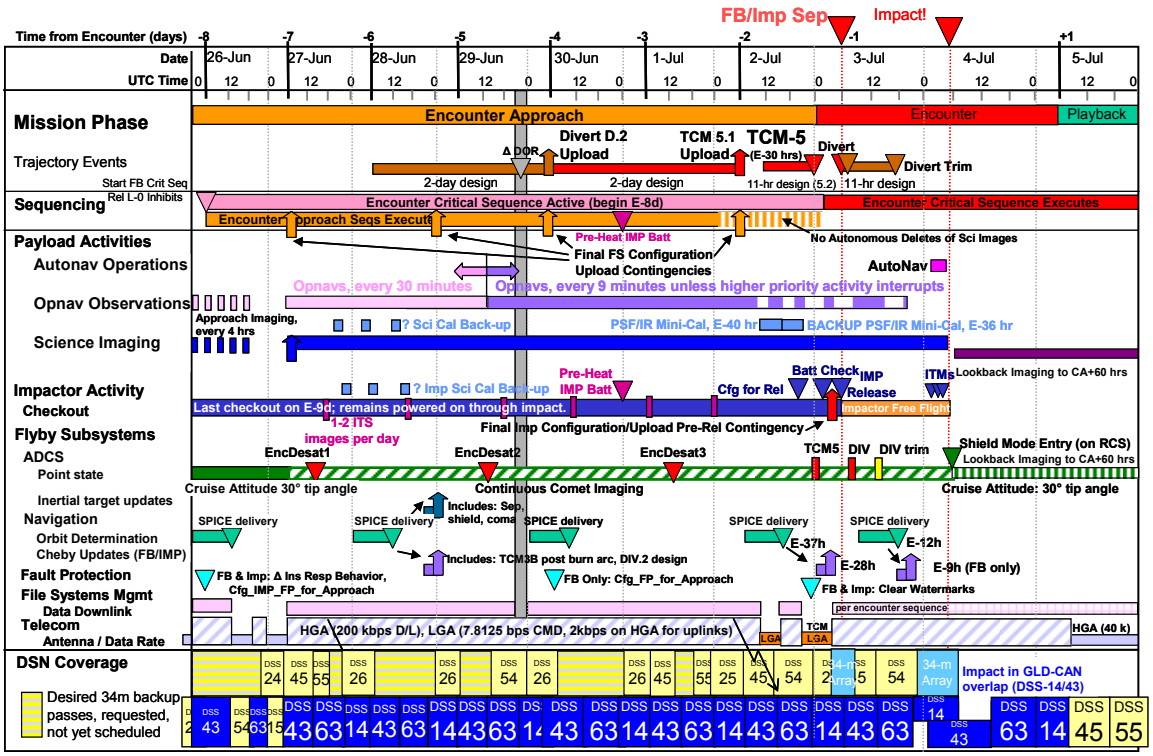

Figure 17. Encounter Approach Phase Timeline

All encounter approach activities were designed to execute on-board via sequences, with the only ground commands issued to kick off the backbone sequence. Continuous 70-m antenna coverage and nearly continuous 34-m antenna back-up coverage provided by the Deep Space Network were required to support this period of Deep Impact's mission. During this final week, both spacecraft were fully powered, with all instruments and engineering subsystems in operation. This week represented the first extended period of continuous Impactor spacecraft operation outside of checkout/calibration/demonstration activities performed during cruise. It was discovered during early commissioning that Deep Impact's High Resolution Imager was out of focus. The science team devised a plan to use deconvolution on images taken with this telescope. In order to perform successful deconvolution, a set of mini calibrations needed to be performed within 48 hours of impact. These calibration images were labeled as critical to mission success, as they would enable the science team's ability to post-process the HRI images and achieve improved resolution compared to the out-of-focus images.

2. Operations Experience

Activity design for Encounter Approach was severely constrained. To begin the design effort, several guiding principles were employed: minimize real-time commanding, limit uplinks to pre-selected windows, design all sequences to be self-deleting, complete all downlinks prior to end of phase, and delete all images and sequences not needed for encounter at completion of phase. Fault protection design during this phase was similar to that used during cruise, with the addition of instrument fault protection. All sequences were non-critical, meaning that if fault

protection interrupted them, they would not be autonomously restarted. Elements with time constraints were placed into the timeline first: TCMs, calibrations, Impactor preparations for separation, and uplinks. Following the placement of time critical activities, other required activities were placed on the timeline. The imaging campaign was the most challenging element to accommodate in the timeline. This imaging sequence, spanning seven days, was by far the largest sequence designed for Deep Impact.

The data flow aspect of Encounter Approach was problematic with thousands of images being taken for both science and navigation. Due to the overwhelming size and time duration of the encounter approach sequence, the operations team relied heavily on sequence modeling tools to validate the sequence prior to uplink. In fact, the sequence was so large that is was never tested on the testbed in its entirety—only small, critical portions of the sequence were fully executed. Instead, the team used the APGEN tools, processes, and models to build the large sequence, then model its performance against the numerous tight constraints. This included data flow, attitude control, power performance, and Impactor crosslink models, all of which were required to fully understand the design space for encounter approach. With very little design space, modeling was essential. Iteration with this model (and varied sequence timing) finally produced a sequence design that was predicted to meet all constraints. Many iterations of the integrated model allowed navigation throughout the design space, minimizing the required testbed execution cycles (and durations). Without the sequence development tools used for Deep Impact, the final week before encounter would not have executed on the spacecraft successfully.

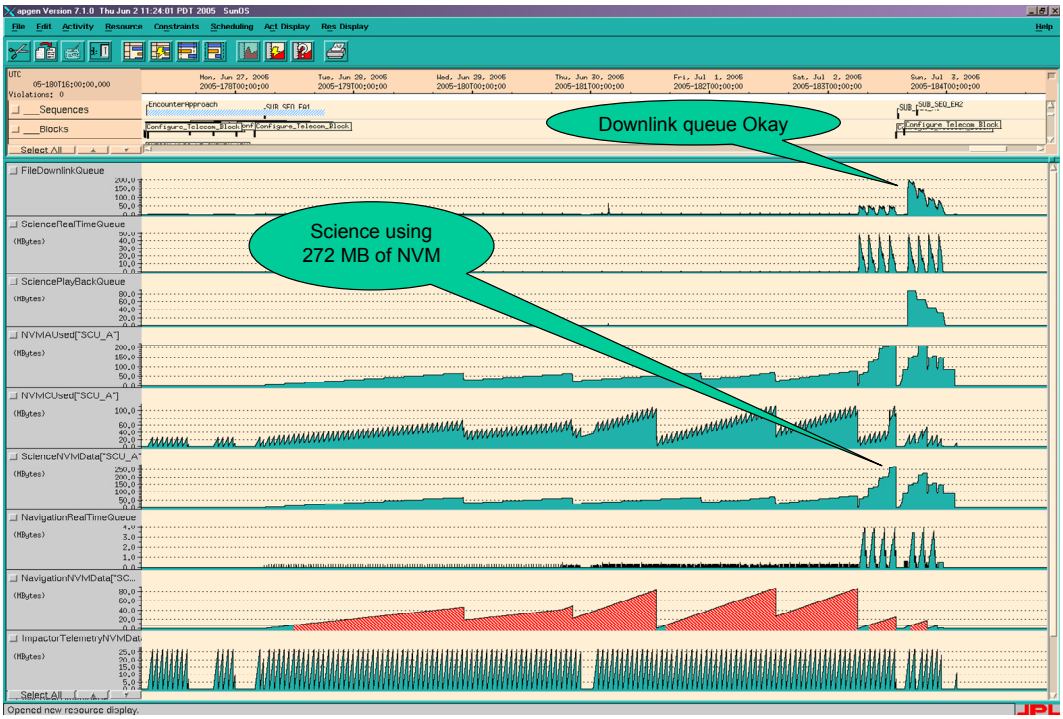

Figure 18. Encounter Approach Phase Integrated Data Flow Model

Figure 18 shows the final successful APGEN model output for Encounter Approach. To run a full iteration of the sequence using flight software and the testbed would have taken over eight days. A single iteration of the APGEN model took only 90 minutes. This drastic time savings was a key element in the tight timeline for Encounter Approach development. Initial design inputs from the science (thousands of commands in a spreadsheet format—nearly impossible to get 100% correct), navigation, and engineering teams resulted in a non-workable sequence design—many images would have been deleted prior to downlink, Impactor telemetry would have been lost, and flight system computational performance would have suffered. The modeling tools used allowed rapid identification of specific problem areas, and provided capability to try out suggested changes without wasting valuable testbed time.

In flight, the Encounter Approach models were validated—real performance tracked well with that of the models. Predictions indicated that a small number of navigation images would be deleted prior to downlink (a remarkable prediction given the starting inputs and constraints on the design). Of the 4389 science and navigation images taken during encounter approach, 127 did not reach the ground in flight. All 127 images lost did not impact mission success. The lost images were deleted prior to downlink by the on-board sequence. It was found that a

single over-subscription in real-time telemetry was the cause for the bandwidth overflow. This over-subscription *fully accounts* for the mismatch in performance to predictions. All performance metrics for each subsystem, the flight team, and the science team, were met or exceeded for the final week prior to impact, *as predicted*. Without comprehensive testbed execution, the sequence performed exactly as modeled. Obviously, these tools were enabling for this mission phase.

D. Encounter

The comet encounter with Tempel 1 was the final critical event for the Deep Impact mission. This section discusses the most important, most challenging, and most spectacular day of Deep Impact. Also captured herein are lessons learned and operational experiences of the flight team for this historic event.

1. Mission Phase Description

The encounter phase for the flyby starts at initiation of the critical sequence prior to impactor separation. Impactor separation occurs 24 hours prior to impact and is accomplished by targeting the flyby spacecraft to hit the comet, releasing the impactor and then performing a divert maneuver of 101 meters/second to slow the flyby spacecraft and target the flyby to pass the comet at a 500 km altitude. Slowing the flyby allows imaging of the comet for 800 seconds following the impactor hitting the comet. The flyby captures science and optical navigation data periodically up to about 2 hours before impact, where science imaging frequency increased and autonomous navigation imaging begin.

The *a priori* error in the time of impact (TOI) at the time the imaging sequences were generated was +/- 3 minutes. The high-rate imaging sequence for impact needed to be started with an accuracy of around 3 seconds. The final imaging sequence for the high-resolution crater imaging had a similar timing requirement. The image sequences around impact and shield mode, were relative timed sequences spawned from the critical sequence. AutoNav could adjust when these sequence start by calculating a TOI and time of final imaging (TOFI) update. These updates were calculated by AutoNav, based on current OD and sent to the command manager flight software for adjusting the sequence timing. The TOFI update also controlled the timing of the transition to shield mode. Shield mode is an attitude which kept the solar arrays edge-on to the dust flow, and put the spacecraft –X axis in the direction of the dust flow. This maximized the effectiveness of the dust shields. This transition was timed to occur when the Flyby spacecraft was 700 KM from Tempel 1 and 13 minutes 22 seconds after nominal Impactor impact. The Flyby spacecraft stayed in this attitude for 27 minutes, until it was through the worst of the dust environment, and then transitioned back to image Tempel 1. In the look back attitude, the High Gain Antenna could not be pointed at the Earth. The look back imaging was done on the Low Gain antenna at a downlink rate of 10 bits/second. The Flyby spacecraft was sequenced to periodically switch between an attitude which allowed for playback of stored images, and the imaging attitude. Figure 19 is a high-level representation of the encounter timeline.

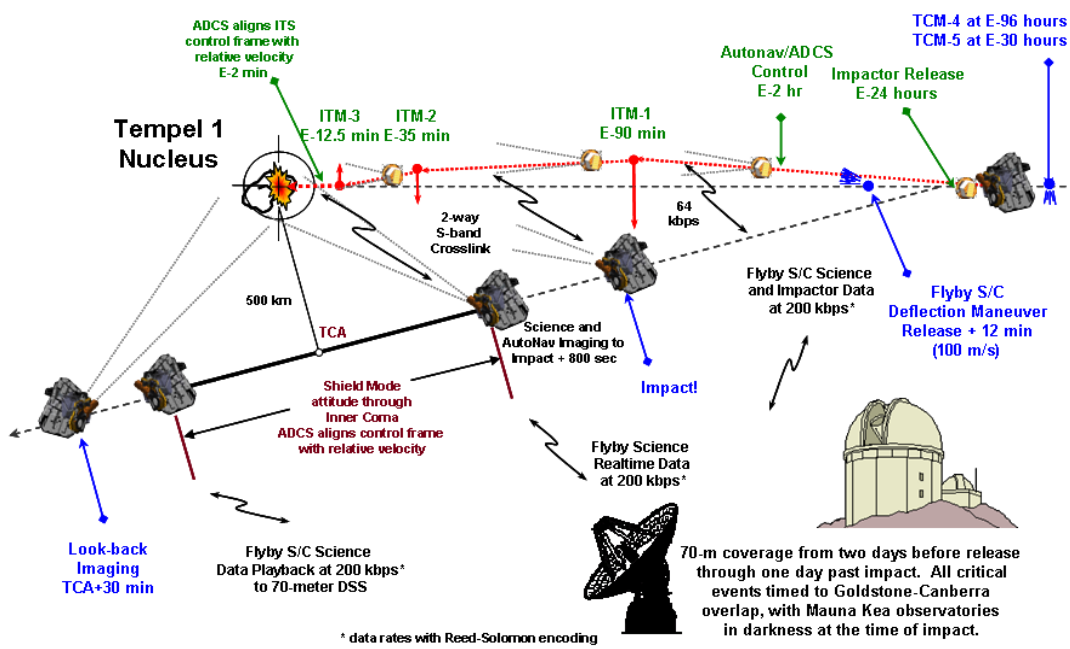

Figure 19. Encounter timeline

2. Encounter Sequencing Strategy

Deep Impact used a "mark and rollback" capability for critical sequences which allowed fault protection, upon detection of a fault, to pause sequence execution, repair the fault, and then resume the sequence by either falling back to a mark point and re-executing parts of the sequence, or to roll forward and skip parts of the sequence to catch up with the critical timeline.

3. Encounter Data Capture Design

The Deep Impact Flyby Spacecraft stored engineering telemetry and instrument image files on a file system in non-volatile memory (NVM). There were two 256 MB NVM devices available. The first device had 2 dedicated drives for Flight Software and FSW parameter tables. This first device also had a general purpose data storage drive, which contained sequence and parameter files uplinked from the ground, all engineering telemetry files and "overflow" instrument image files. The second device was dedicated to instrument image files. Table 4 shows NVM Allocations for encounter.

Table 4: Encounter Flyby S/C NVM Allocations

Data Type	Allocation
Science	309 MBytes
Navigation	6.95 MBytes
Impactor Telemetry	30 MBytes

The Instruments could send images to either or both of the flight computers. The backup flight computer's file system was primarily used for redundant storage, however, due to the descope in file system space, some images were written non-redundantly to only one flight computer in order to preserve some of the original science data. This significantly complicated the sequencing and data return strategy. During playback, it was necessary to swap to the backup flight computer to retrieve images. Swapping telemetry streams was always a risky proposition because the backup didn't have an active 1553 interface. This meant no available engineering telemetry from the active spacecraft hardware when viewing telemetry from the backup.

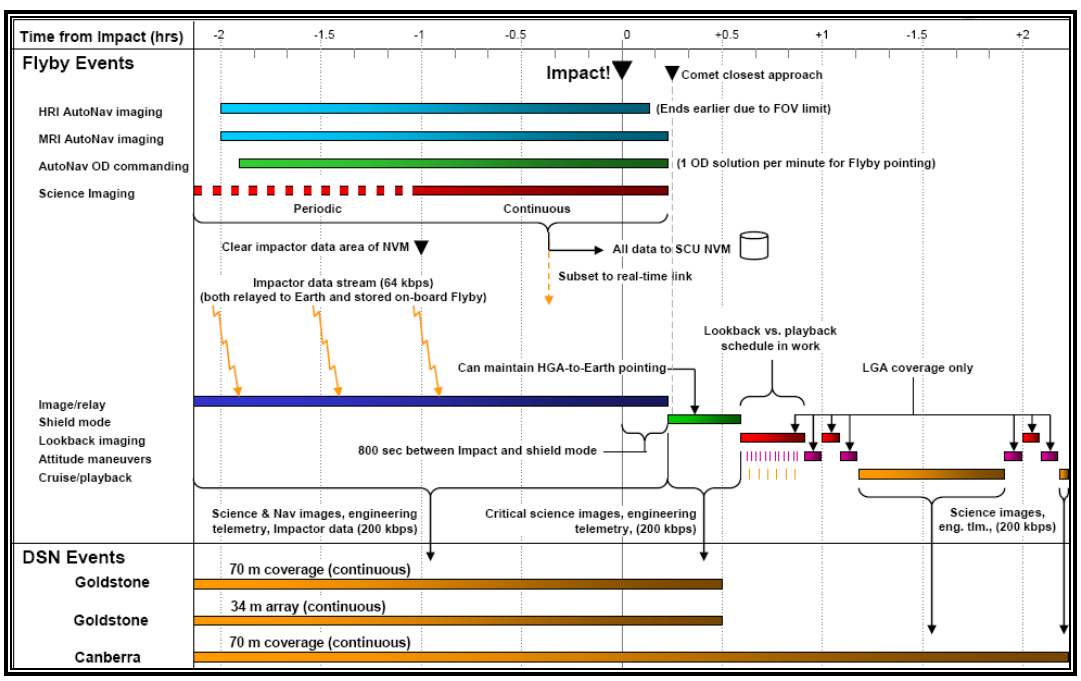

Figure 20: Encounter Flyby Spacecraft Capture Flow Overview

Instrument image files could be queued for downlink to the ground in real-time (via correct parameter in imaging command), or sent to the ground later (via a file downlink command). All images prior to Impact minus 5 hours were sent to the ground in real-time. These images were periodically sequenced for deletion prior to ground confirmation of receipt in order to clear NVM for new image acquisitions. Within 4.5 hours of impact, most science images were kept in NVM storage until after the playback phase completed and all images were confirmed received

on the ground. NVM was nearly filled at the time imaging was completed at shield mode entry. The look back imaging, which occurred after closest approach, used the remainder of the available NVM. Figure 20 depicts encounter data capture characteristics for the flyby spacecraft and Fig. 4 shows key characteristics of the APGEN data flow model. C&DH throughput performance was a continual problem throughout most of the encounter test program, with critical impact and crater images lost on nearly every test run on the spacecraft pre-launch and testbed runs pre- and post-launch. The throughput performance fell off from predictions during periods of heavy imaging and file system usage. The performance was improved with Flight Software updates throughout the test program, and appeared adequate for flight in the final 2 months prior to encounter.

The Impactor spacecraft communicated with the Flyby spacecraft via an S-band radio. Commands and file transfers to the Impactor were processed on the ground into a file of bits to be radiated, then uplinked to the Flyby spacecraft as a file transfer to the Flyby's on-board file system. A command was then sent from the ground for the Flyby to send that file using the s-band link to the Impactor, which saw it as either a real-time command or a file transfer. Telemetry data received on the Flyby from the Impactor S-Band was stored in 128 Kbytes NVM files and automatically queued up for D/L. When the files were received on the ground, they were automatically read and processed into Impactor telemetry frames for real-time display and storage. The collection and storage of this data on the Flyby could be turned on or off by command. There was sufficient data allocation on the Flyby NVM to store 30 MB of Impactor telemetry. This equated to just over 1 hour of data at 64 Kbps. The Impactor data collection strategy was to turn on collection, collect data for 1 hour, turn off data collection and delete the files in NVM. This sequence was repeated until one hour before impact. The last hour was collected and remained in NVM post impact. This strategy ensured nearly continuous Impactor telemetry to the ground, but only stored the last hour of telemetry on the Flyby for post encounter playback. Any telemetry not received on the ground prior to 1 hour before impact was un-recoverable if initial transmission attempt failed. Due to descopes affecting downlink data rate, there was major contention between science image data and Impactor telemetry in the available downlink bandwidth. The sequences and downlink priorities had to be carefully orchestrated in order to avoid latency in either science images or Impactor telemetry. Flyby S-Band RF reception was turned on at Separation plus 30 seconds. S-Band RF transmit was turned on at Separation plus 6 minutes to ensure the RF level was low enough to avoid damage to the Impactor S-Band receiver. There were 6076 Impactor telemetry files (about 760 MBytes) collected by the Flyby and relayed to the ground during the encounter critical sequence.

Due to the Flight System Testbed being a critical, oversubscribed resource, it was essential to have a high-fidelity data throughput model within the sequence planning toolset. Many thousands of hours of testbed time were saved by validating the data flow characteristics of sequences prior to running them on the testbed.

4. Operations Experience

APGEN proved invaluable in validating key characteristics of encounter related to science data storage and throughput, attitude constraints and restrictions and managing the large number of sequences necessary to implement the fault protection strategy. Science and navigation image commanding requests were received as excel spreadsheets. These spreadsheets frequently had numerous errors which were difficult to detect visually by inspecting the spreadsheets. APGEN was used to detect most of these errors and provide quick feedback to correct the problems. Use of APGEN to model and constraint check the sequences saved tremendous resources of testbed time by having a quick validation of these key sequence characteristics. This allowed for the use of the testbeds to manage the complex interactions between attitude control with autonomous navigation and fault protection.

VI. Conclusion

HAVING a standard, multi-mission based tool set, in general, is an enabling technology for low cost mission operations. Using APGEN as a sequence planning and validation tool, with integrated subsystem modeling allowed the DI MOS to operate complex, critical mission phases with minimum workforce for sequence planning and execution, as well as having short turn around times for events such as instrument calibrations and TCMs. APGEN also allowed for a significant reduction in testbed executions for instrument calibrations, encounter approach, and final comet encounter. For Deep Impact's spectacular success at Tempel 1, APGEN proved critical.

Acknowledgments

The work described by this paper was performed at JPL, managed by The California Institute of Technology, under contract to the National Aeronautics and Space Administration. The Deep Impact spacecraft was designed and built by Ball Aerospace & Technologies Corporation. Deep Impact was a PI-led NASA Discovery Mission, with Principal Investigator Dr. Michael A'Hearn from the University of Maryland.

References

Proceedings

[1]Wissler, S, Rocca, Jennifer and Kubitschek, D., "Deep Impact Comet Encounter: Design, Development, and Operations of the Big Event at Tempel 1," Space Systems Engineering Conference, Georgia Institute of Technology, Atlanta, Georgia, GT-SSEC.C.2, http://hdl.handle.net/1853/8032, 10 November 2005.

Reports, Theses, and Individual Papers

[2]Hughes, M., "Deep Impact Slew Tool Algorithm Design and Test Results for APGEN, Rev A", JPL IOM DI-341-02-119-A, 5 December 2002.

[3]Hughes, M., Speckman, K., Schira, C., "ADCS Target Table Description", DI-SC-ACS-055A, 20 March 2003.

[4]Blume, W.H., and Wang, K.C., "Deep Impact Project Mission Plan," December 2004.

[5]Swenka, E., "Endgame Maneuver Timeline, Rev A", 23 June 2005.

Electronic Publications

[6]Mitchell, A., Bresina, J., et al, "MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission," *IEEE Intelligent Systems* [online journal], IEEE1094-7167//04, pp8-12, URL: http://ieeexplore.ieee.org/iel5/9670/28315/01265878.pdf?arnumber=1265878 [cited 10 May 2006].