VALIDATION OF FAULT-TOLERANT PLANS FOR EUROPA CLIPPER

Steve Schaffer[1], Steve Chien[1], Eric Ferguson[1]

[1]*Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109, USA, E-mail:* firstname.lastname@*jpl.nasa.gov*

ABSTRACT

The Europa Clipper mission will explore that icy moon in a series of brief flybys through the Jovian radiation belts. A single event upset in the spacecraft flight computer during these critical scientific periods could jeopardize the success of the mission. Rather than safing and awaiting operator intervention, the Clipper mission envisions limited onboard autonomy that can restore spacecraft state sufficiently to resume the encounter observation plan as rapidly as possible. The contingency plan is contained in an Activity Restart Timeline (ART) that is transmitted in parallel with the nominal plan, which must be co-validated jointly against all spacecraft state and resource constraints amid unpredictable fault timing. A prototype validation tool was built that leverages declarative spacecraft models and automated search techniques to find such potential inconsistencies in the unified contingent mission plan. Early validation results within motivating scenarios are presented.

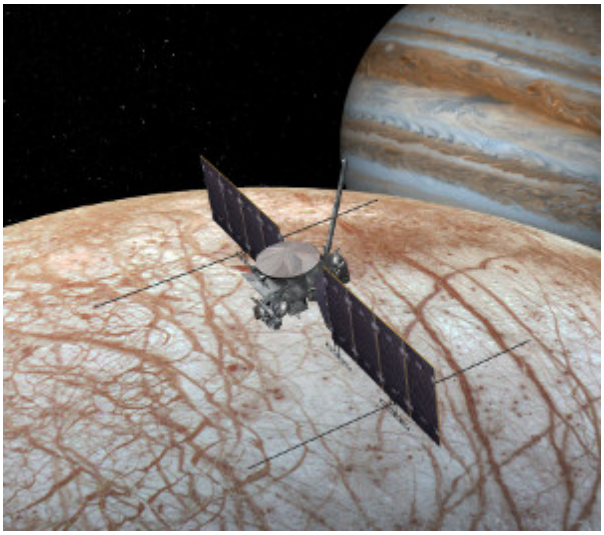

Figure 1: Artist's impression of the Europa Clipper spacecraft during a critical brief flyby of Europa.

1. INTRODUCTION

The Europa Clipper mission [1] will explore the potential for life within the icy moon's sub-surface ocean. The environment nearby Europa is dominated by the intense Jovian radiation belts, which would continuously bombard a Europa orbiter mission with disabling fluxes of high-energy charged particles. Instead, Clipper uses a complex mission trajectory around Jupiter [2] that provides for many brief close flyby encounters of Europa that dive through the radiation bands, interleaved with prolonged retreats to the relative safety of a high apoapsis. This minimizes the total radiation dose received by focusing the exposure within the narrow encounter periods, which are unfortunately also the most scientifically critical to the study of Europa's surface and composition [3]. There is attendant risk that the flight system will suffer an upset or reboot during or immediately prior to a flyby. The traditional spacecraft safing approach that overrides the nominal mission plan and awaits operator intervention for recovery is ill-suited to address the opportunity cost of missing scientific observations during the limited encounters.

Instead, the Europa Clipper mission is evaluating the application of limited onboard autonomy that could respond immediately after an upset to resume functional scientific operations as soon as possible. While some science observations may still be unavoidably lost concurrent to the fault and recovery itself, and others precluded by pointing drift or other persistent effects, the remainder of the encounter science value might be reclaimed by swift onboard response. The relevant automatic recovery efforts that should be invoked change throughout the flyby, and are thus pre-loaded within an Activity Restart Timeline (ART) onto the spacecraft in parallel with the corresponding nominal mission sequence. The ART recovery actions bring the spacecraft back onto the negotiated nominal mission plan as best as possible. An alternative approach wherein the spacecraft determines the best in-situ recovery strategy and subsequent plan on its own initiative was also studied [4], but a pre-established contingency script provides the operability benefit that it may be pre-validated and approved by mission stakeholders. Such ART validation is the focus of the present work.

The ART must be developed carefully in coordination with its matched nominal sequence to ensure that all mission safety constraints are unambiguously upheld in the face of unpredictably timed spacecraft faults, while also salvaging as much science value as possible. This requires reaffirmation of mission flight rules not only during a posited fault and immediate recovery, but also

throughout the rest of the downstream nominal mission plan, which may be impacted by residual effects from the fault (or the recovery). Prediction of the interrelated effects and plan conflict identification are enabled by a declarative model of the spacecraft states, resources, and commands that is encoded in the Automated Scheduling and Planning Environment (ASPEN) tool [5]. In turn, that model was constructed in reference to a detailed discrete event simulation of the spacecraft within the Activity Plan Generator (APGEN) framework [6], which is used in ongoing model-based system engineering and mission design efforts [7][8].

The ASPEN planning architecture provides timeline-based search capabilities that allow efficient evaluation of when faults may legally occur within a unified nominal and contingent plan. This allows for immediate provable validation at a level of confidence matching the credence given to the spacecraft model used by the tool. In contrast, approximation techniques such as Monte Carlo discrete stochastic sampling of fault timings within long-running simulations only slowly builds confidence in a given plan, and requires considerably more computational power.

A prototype ART validator, ARTcritic, was constructed to leverage the declarative model and search capabilities into efficient assesment of Europa Clipper fault-tolerant mission plans. The prototype focuses on an small assortment of spacecraft instruments, states, and resources selected to span the spectrum of the full mission system. Evaluation of the tool within several motivating scenarios demonstrated its capability to identify latent problems within seemingly reasonable unified mission plans, illustrating subtle complexities that must be accounted for when operating the ART-based control system.

2. APPROACH

The fault-tolerant operation of the Europa Clipper spacecraft is enabled by an onboard time-indexed table of fault response actions, the Activity Restart Timeline (ART). As shown in Fig 2, the nominal mission command sequence is uplinked along with a corresponding ART as part of the unified control program for the spacecraft. The nominal mission activity plan represents considerable collaborative effort by the mission science and engineering teams to determine the best balance among the challenging scientific objectives of the mission. That diligently constructed plan is endangered in the event of a spacecraft fault, as my occur due to a radiation induced single event upset. The main processor could be reset and the flight software rebooted during the critical brief period near periapsis with Europa when the majority of close-range science observations are planned.

A traditional orbiter might enter safe mode and await manual diagnostic and recovery steps by mission controllers back on Earth, but this is not feasible for Clipper. First, the elliptical nature of Clipper's orbit around Jupiter means that the spacecraft is traveling fastest nearby Europa and has precious little time (≈4 hours) before the flyby is over. Even if the human operations team were somehow able to respond immediately, the round-trip light time between Jupiter and Earth is ≈1.5 hours. Second, unlike circular orbiting mapping missions that can make up for missed science targets on the next repeat overflight, the Clipper spacecraft must wait at least 14 days to complete another orbit, and even then may never have another geometric opportunity for specific targets. Third, the trajectory includes only a limited number (≈40) of flybys before the mission ends in controlled disposal (to prevent potential contamination of Europa), so each encounter represents a significant portion of the mission's success criteria.

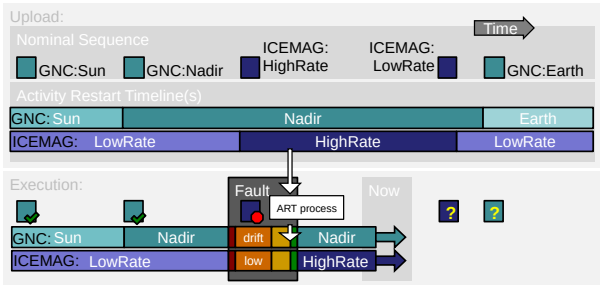

Figure 2: The Activity Restart timeline (ART) consists of target states that should be restored in order to allow rapid return to the nominal command sequence after a spacecraft fault event, even if some commands are missed during the fault.

Instead of safe mode, the ART is invoked to autonomously restore a suitable spacecraft state that allows resumption of the nominal sequence. The ART consists of a time series of specific states that each component of the system should be restored to, covering the entire period over which the ART might be invoked. As show in Fig. 2, the ART target states will typically mirror the predicted states of a fault-free execution of the nominal mission plan, but may also differ in important ways. In particular, the timing of state transitions may be subtly different or some groups of nominal transitions may be entirely absent in the ART, for example when a fault precludes completion of a complex observation pattern. Each ART target state is backed by a spacecraft command sequence that is able to bring about the intended state from any of the expected post-fault states, and as such might involve conditional logic After the ART restoration tasks complete successfully, it is safe for the flight software to transition back to the nominal mission plan and undertake any subsequent observations.

2.1. Spacecraft Model

Validation of a paired nominal sequence and corresponding ART table, refereed to jointly as a control program, requires understanding several facets of the spacecraft system behavior. Foremost, predicting fault-induced violations of flight rules within the nominal sequence requires details of the states, resources, commands, and constraints in the system. Fortunately, this detailed modeling effort has been ongoing since the early mission design phases of the mission, and is currently encoded in an APGEN adaptation used in generating proposed nominal mission plans for study. The Europa Clipper spacecraft comprises a number of instruments that span a spectrum of sensing and commanding modalities. Fixed in-situ detectors have relatively simple control strategies compared with scanning imaging instruments, though they interact with each other through shared spacecraft states. Some instruments are directly commanded from the flight computer itself, and are thus highly susceptible to faults that cause a nominal sequence command to be missed. The ART response for these instruments typically requires reissue of any missed commands, as long as the delayed execution does not cause further problems. Other instruments are driven by their own built-in command tables or macros, and are mostly immune to an untimely reset of the main computer. Even so, the ART responses must still ensure that these external tables are properly populated and that the ongoing instrument execution does not interfere with other shared spacecraft resources. The prototype validation tool leverages the prior modeling work, focusing initially on four representative instruments (ICEMAG, REASON, MISE, SUDA) that span the major categories, along with relevant states and resource fluents. The APGEN imperative discrete event simulation model for these instruments, was translated into a declarative activity model more suited to automated search within the ASPEN architecture.

2.2. Fault Model

The next major model component is the failure itself, diagrammatically shown in Fig. 3. During the reset and reboot of the flight software, the spacecraft is not under active high level control, much less following along the nominal mission command sequence. Some individual system components, in particular some instruments, may continue actuating according to last commands from the flight software, but any coordination among components will fail without the flight software messaging bus. These surviving components will realize the outage via the absence of clock synchronization messages, and may eventually take internal steps to safe their subsystems (e.g. closing instrument shutters), but the responses are highly instrument specific. Since the instrument internal fault protection and other blissfully unaffected instrument macros can influence states that may matter to

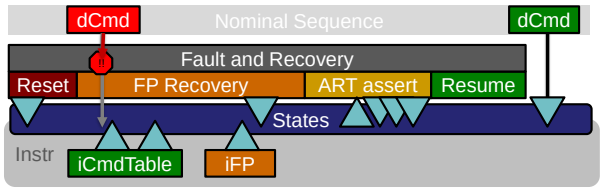

Figure 3: A fault event comprises the processor reset, flight software fault protection recovery, ART reassertion of states, and finally resumption of the nominal sequence. Flight software commands sequenced during the event will have no effect, but external instrument subsystems may continue to modify the shared states.

the ART validity, it is important to include a model of those behaviors. At the present early phase of instrument implementation, these behavior details are largely unavailable and so were approximated with the empty response. Absent active central control, global spacecraft states will likely drift out of their intended assignments, most significantly the spacecraft pointing (since the uncommanded reaction wheels will slowly despin and the star tracker will be offline). A precise model of the effects of the reset itself is also difficult to construct due to the unpredictable influences on the spacecraft while the flight computer is offline, but some approximations are possible (e.g. a maximum pointing drift given expected prior reaction wheel rates).

After the reboot completes successfully, the flight software fault protection module will activate in order to recover any mission safety critical spacecraft states. Fault protection will re-engage active control of e.g. spacecraft attitude and heating, but will not yet contemplate the nominal command sequence, which will still be ignored during this period. In order to accurately predict possible interactions, a detailed model of the flight software fault protection module is also desirable in a full validation system, which is naturally hard to ascertain before its requirements are finalized and code is actually written. The validation prototype assumes only the most rudimentary state recovery by fault protection, setting them to a designed unknown-but-safe value. Altogether, the reset and low-level recovery may take several minutes.

After the spacecraft is cleared for continued operations by the fault protection module, it will invoke the ART module to try and reassert high-level science states and return to the nominal plan. The ART module looks up the current time in the uplinked ART table to determine the proper set of state reassertion sequences to trigger. Since the ART issues a (possibly conditional) series of its own commands during this reassertion phase, the nominal plan remains suspended. Any commands sequenced from the first reset through to the end of the ART reassertion will not be executed, and thus must not

factor into the state predictions of the validator. The prototype disables the effects of any nominal commands during all of the fault and recovery phases via a simple conditional check wrapping each command. The model of what states the ART process will effect itself is precisely that derived from the reassertion command sequences attached to each target state. Since the ART invokes normal flight software behaviors, the existing internal models of those commands can be reused.

Finally, after the ART module is done reasserting all of the relevant target states, the nominal command sequence is resumed and the command effects are re-enabled. The nominal commands may interact with any of the states left over by the previous fault event phases, and so must be modeled out to the end of the planning horizon during validation. The ASPEN framework accomplishes this propagation internally via a causal graph through the activities and related timelines. The fault model thus consists of four phases: reset of the flight computer, recovery by fault protection, reassertion of ART states, and resumption of nominal commanding. Each of these should be modeled as deeply as possible in order to allow the validator to detect any adverse interactions, though the prototype model primarily focuses on the ART's own effects.

2.3. Fault-Sensitivity Search

With models of the spacecraft behavior under both nominal and fault conditions in hand, the question remains as to how to use them to efficiently validate a given ART. The prototype validator accomplishes this task by again leveraging the fast re-prediction of plans and conflict detection provided by the ASPEN framework. A hypothetical composite reset-recover-reassert-resume fault event activity is created and then used to probe each of the relevant time ranges in the nominal plan for any induced conflicts, as shown in Fig. 4 The time ranges of interest are delineated by a change in any subsystem's target ART state, so each different conjunction of ART responses results in the creation of a unique composite fault probe activity. The probe fault is then tested at critical points within the applicable time window by temporarily placing the fault event activity in the schedule and propagating out the remainder of the plan to elicit any immediate or downstream conflicts. Thanks to change detection in the ASPEN causal dependency graph, the forward propagation short-circuits when values stabilize at the frontier. Fore example, prior state changes would not need to propagate past the next unconditional state transition, and data volume changes would not need to propagate past the next full drawdown of the data buffer.

The critical points at which to test the probe fault action at least include the times at which the probe activity's state effects and resource reservations interact with those from the nominal sequence. In a full validator implementation, these points could be determined by walking backward through the causal network of nominal activities and states to identify time points where the probe activity may change its net effects on the plan. In turn, those probe-changing criteria require examination of the modeled input and output states of the probe fault activity, with special attention to any conditional relationships among them (though many commands have very straightforward unconditional effects). The prototype validator instead uses an interim binary search approximation to find the critical points. The time range of constant ART response is divided into contiguous spanning sub-ranges, and the probe fault tested for conflicts at each boundary. The prototype assumes that there are no undetected transitions in conflict behavior between like-determined boundaries, but dives deeper into the sub-regions that are bracketed by different conflict determinations. The process repeats until the actual critical point is isolated within some tolerable threshold. Conflicts that are revealed represent incipient fault sensitivities in the combined control program, and indicate that either or both of the ART and nominal sequence should be revised.

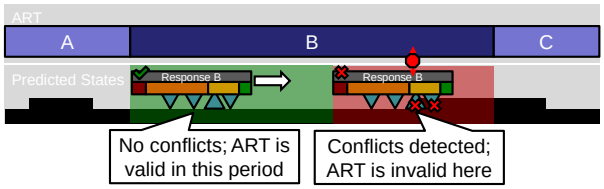

Figure 4: An ART is fully fault-tolerant only where the combination of fault reset, recovery, and ART-invoked reassertion is consistent with the nominal plan. Conflicts may occur immediately, or appear much later in the plan.

The finally determined probe conflict transition point is reported up to the user as the endpoint of a fault-sensitive region of the plan. According to the modeled interactions, a fault event (including as-specified ART response) within any of these disjoint regions would eventually lead to some kind of flight rule violation. Any fault-sensitive regions in the plan are unacceptable for a fully fault-tolerant mission plan. The user interface reports the detected fault-sensitive regions of the plan and allows the user to reconstitute the fault probe within any of them so that they can explore the implications of that fault. Once again, the ASPEN framework assists tracing back from conflicts through their proximate causes and eventually to all contributing causes. Users can use this guidance to tweak their proposed mission control program before resubmitting it for validation.

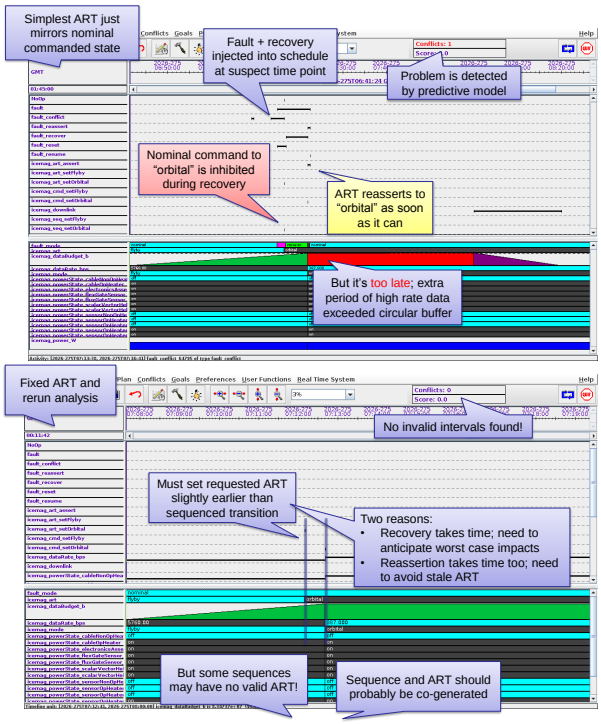

Figure 5: The validation tool finds periods fault-sensitivity, and then allows users to explore the impact of specific faults and eventually revise the unified control program to be fully fault-tolerant.

3. RESULTS

The prototype validator was tested within several specially designed scenarios that demonstrate the capabilities of the model-based conflict search approach. Only an excerpt of the the full APGEN spacecraft model was translated into declarative form for the validator, including several representative instruments, resources, commands, and states. Furthermore, the model was modified so as to exaggerate the resource contention and the ART responses in the mission; these fictions allow more expeditious testing of the features of the validator. The workflow through two specific scenarios is described below.

3.1. ICEMAG Scenario

The first scenario centers on the ICEMAG magnetometer instrument, which is an in-situ sampling instrument tuned to study the interior of Europa via magnetic induction. The ICEMAG is commanded directly from the nominal mission plan: separate commands are sequenced at the right times to change the instrument between its default low-rate collection mode used throughout the majority of th orbit and the special high-rate collection mode used only within the close flyby. The mode commands are scheduled in the nominal mission plan to correspond to relevant altitudes predicted from the spacecraft trajectory. A later downlink activity moves the data off of the instrument's limited buffer and eventually back to Earth, freeing the memory consumed since the last downlink. The initial ART provided for validation parallels the simple nominal sequence for ICEMAG exactly: ART will reassert the default low-rate mode by default, but will assert the high-rate mode between the scheduled high-rate command and matched low-rate command.

The nominal sequence is validated to be conflict free by itself in a no-fault scenario, but when combined with the simple ART and potential faults, the validator tool reports two periods of fault-sensitivity – even with such a simple control program. Fig. 5 shows the conflict induced when a probe fault is created by the user to examine the reported sensitivity at the very end of the flyby: the memory buffer is overflowed with excess high-rate data. Note that the memory limit was much exaggerated in order to exhibit this fault (the actual spacecraft has copious storage and ICEMAG is a small contributor to data volume). The extra data that drives the conflict-free nominal plan into over-subscription after a fault is accumulated while the reset is occurring and the expected command to low-rate does not go through. The ART correctly reasserts to the low-rate mode as soon as it can after that, but it is too late, and the memory is already overflowed.

There are several corrective resolutions that the user may choose from in this case. First, the nominal sequence could be modified so that the low-rate mode transition near the end of the flyby occurs earlier, avoiding the maximum margin of overflow possible during a fault event. Second, the nominal sequence's initial high-rate transition could be pushed forward so that there is less data in the buffer, similarly avoiding the overflow margin during the fault. Third, if this problem was identified during mission design, the instrument data buffer could be re-sized to accommodate the extra margin. Fourth, the ART transition to low-rate mode could be divorced from its corresponding nominal command and moved early enough that the ART essentially anticipates the upcoming mode transition.

The second fault-sensitivity detected in the ICEMAG scenario has to do with the ART asserting a stale state. Since the ART processing itself is not instantaneous, a nominal command may be missed between reading the ART table and finally resuming the nominal sequence. The most direct way to correct for this possibility is to shift the ART transition slightly earlier than the nominal sequence transition time (by at least the duration of the ART processing loop). Similarly to above, this allows the ART to anticipate transitions and assert the upcoming state in case of faults rather than a state that will be stale by the time it takes effect. This adjustment is a general requirement when using the ART scheme, and is

magnified when the ART processing step takes significant time (e.g. if it contains conditional delays).

Since both problems with the ICEMAG control program can be solved by moving the ART transition earlier, the user selects that approach. The bottom of Fig. 5 shows the zoomed-in corrected plan, with the ART transition slightly offset from its corresponding nominal sequence transition. No further fault sensitivities are found in the control program, meaning that this combination of nominal sequence and ART is indeed fault-tolerant to the modeled level of detail. Note that there may still be side-effects of a fault: for example a fault that is recovered by transitioning to a low-rate mode early would be forgoing that period of high-value data, and this must be balanced against the risk of violating flight rules by overflowing buffers. Indeed, the corrective adjustments made to the nominal schedule to ensure complete fault-tolerance may introduce inefficiencies into the nominal plan, which will be suffered regardless of if a fault is actually encountered or not. This is the nature of fault-tolerant plans without futher onboard decision making.

3.2. REASON Scenario

The second scenario examined focuses on the REASON ice penetrating radar instrument. REASON is commanded in mixed mode: direct commands are issued from the nominal plan in advance of the flyby in order to load configuration and macro tables into the instrument, but the instrument controls its own data acquisition modes during the actual flyby. The power-on, warm-up, and shutdown commands are also direct commanded. A simple ART that exactly mirrors these nominal transitions encounters the same stale ART fault-sensitivity noted above for ICEMAG, so the first attempt at a REASON ART includes those shifts already.

Running the validator shows two fault-sensitive period: one during the configuration commands, and one during the self-running instrument macros. The first configuration conflict is caused by an overly simple state assertion sequence in the ART: since the instrument must cycle through its boot-up and table load phases in sequence, it is not sufficient to jump directly to table load after a fault event that spans the boot-up period. This is corrected by increasing the conditional complexity of the sequence invoked by the ART to recover to configured state: if the boot-up has not completed, do that first, and then proceed with the configuration load. This increases the predicted worst-case duration of the ART response, and so requires further shifting of the ART transitions to avoid stale ART conditions.

Unfortunately, the second REASON fault-sensitivity found during the instrument internal flyby macro execution is caused by exactly that kind of anticipation shift. A fault near the end of the macro period will anticipate and assert the upcoming transition to the off state in case of a fault. However, the instrument is still running according to its built-in macro and cannot be safely switched off before the end of the flyby period. This is exactly the opposite of the problem seen in the ICEMAG scenario, and the fix is to remove the anticipatory shift introduced to the ART off mode transition. Fortunately, there are no downstream resource conflicts with a slightly over-running off transition for REASON like there were for ICEMAG memory, and this resolves the fault-sensitivities. If there were conflicts of both kinds, the nominal sequence (or spacecraft design) would have to be adjusted to leave enough margin for the over-run.

4. FUTURE WORK

Most of the problems identified in the scenario control programs were caused by missed nominal commands: those that should have taken effect, but were blocked during the reset and recovery process. The ART module attempts to patch up the holes left in the plan when a command is missed, but it requires special care to ensure the precise timing is still right. A significant advantage is seen in the REASON commanding mode whereby all of the time critical transitions are handled internally by the instrument, which invites future instrument designs to prefer this modality.

The validation tool demonstrates the significant advantage of granting real insight into what kinds of contingency failures may occur during ART fault responses, and into the need for validation at all. Traditional mission planning only predicts one nominal path through a plan, relying heavily on low-level fault protection and spacecraft safing in case anything goes awry. This work has shown that it is quite instructive to explore the various ways a plan may break when exposed to generalized fault scenarios, and will hopefully inform future mission operations practice.

The occurrence of multiple faults within a single planning period was not explored in the present work, and presents a significant future challenge. The mission has adopted the stance that two faults during a single flyby warrants retreat to safe mode.

The planning engine used for validation is directly suited to assist in simultaneous co-generation of the unified control program, which would save on separate revision cycles repairing validation issues. A unified workflow of planning and validation would be a boon to the mission operations teams.

Other areas for future improvement already mentioned include: increased breadth and depth of the declarative model of the spacecraft, detailed modeling of actual instrument and flight software fault protection behaviors, and causal graph informed search for critical fault time points.

5. CONCLUSION

A prototype software tool was developed to validate fault-contingent mission plans for Europa Clipper by

leveraging a declarative spacecraft model and search-based automated planning techniques. The tool is able to efficiently detect latent inconsistencies between declared spacecraft constraints and potential fault scenarios that may occur anytime during the plan. The conflicts are detected in both immediate interactions as well as downstream effects, and for both the fault as well as the recovery operation itself. The prototype was demonstrated with a representative excerpt of the complete spacecraft model within several motivating scenarios, revealing unexpected complexities in designing unified mission control programs. The success of the prototype attests to the power of software models and planning tools to assist in the careful validation process required for Europa Clipper fault-tolerant planning.

6. ACKNOWLEDGMENTS

Portions of this work were performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1] Phillips CB and Pappalardo RT (2014) Europa Clipper mission concept: Exploring Jupiter's ocean moon. In: *Eos, Transactions American Geophysical Union*, 95(20):pp.165–167.

[2] Campagnola S, Buffington BB and Petropoulos AE (2014) Jovian tour design for orbiter and lander missions to Europa. In: *Acta Astronautica*, 100:pp.68–81.

[3] Pappalardo RT, Senske DA, Prockter L, Hand KP, Goldstein B et al. (2016) Science of the Europa Multiple Flyby Mission. In: *AAS/Division for Planetary Sciences Meeting Abstracts*, volume 48.

[4] Verma V, Gaines D, Rabideau G, Schaffer S and Joshi R (2017) Autonomous Science Restart for the Planned Europa Mission with Lightweight Planning and Execution. In: *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*, Pittsburgh, PA.

[5] Chien S, Rabideau G, Knight R, Sherwood R, Engelhardt B, Mutz D, Estlin T, Smith B, Fisher F, Barrett T, Stebbins G and Tran D (2000) ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling. In: *Proceedings of 2000 International Conference on Space Operations (SpaceOps)*, Toulouse, France.

[6] Maldague PF, Wissler SS, Lenda MD and Finnerty DF (2014) APGEN scheduling: 15 years of Experience in Planning Automation. In: *SpaceOps 2014 Conference*, p.1809.

[7] Dubos GF, Coren DP, Kerzhner A, Chung SH and Castet JF (2016) Modeling of the flight system design in the early formulation of the Europa Project. In: *Aerospace Conference, 2016 IEEE*, IEEE, pp.1–14.

[8] Bayer T, Buffington B, Castet JF, Jackson M, Lee G, Lewis K, Kastner J, Schimmels K and Kirby K (2017) Europa mission update: Beyond payload selection. In: *Aerospace Conference, 2017 IEEE*, IEEE, pp.1–12.