Workflows, User Interfaces, and Algorithms for Operations of Autonomous Spacecraft

Federico Rossi Tiago Stegun Vaquero Marijke Jorristma Ellen Van Wyk Bennett W. Huffman
Dan A. Allard Nihal N. Dhamani Scott Davidoff Ashkan Jasour Anthony C. Barrett
Rashied Amini Mathieu Choukroun Raymond Francis Mark Hofstadter Michel D. Ingham
Vandi Verma Rebecca Castano

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena (CA) 91109
{*first.m.last*}@jpl.nasa.gov

***Abstract*— Autonomous planning and scheduling is a key enabling technology for future robotic Solar System explorers: as missions venture farther in the Solar System, light-speed delays and low available bandwidth make on-board autonomy increasingly attractive to maximize science returns and enable otherwise-infeasible observations of transient phenomena, e.g. storms on gas giants and plumes on icy worlds. However, ground operations of future autonomous explorers will require a paradigm shift, moving from the current practice of specifying timed sequences of commands to specifying high-level goals that on-board autonomy should elaborate based on the spacecraft's state and on the sensed environment. In this paper, we explore the problem of adapting ground operations processes, roles, and tools to accommodate on-board planning and scheduling. We design and prototype a framework of user interfaces and algorithmic tools to support uplink and downlink processes of future autonomous spacecraft. The framework's goals are to allow scientists and engineers to both convey their desired intent to the spacecraft in a format compatible with the on-board planner, and reconstruct and explain the decisions made on-board and their impact on the state of the spacecraft. We assess the performance of the framework through a design simulation where JPL scientists and operators simulate realistic operations of an Ice Giant multi-flyby mission concept, aided by the proposed framework. The design simulation confirms that the proposed approach holds promise to enable operators to interact with on-board autonomy, and suggests a number of recommendations for the next generation of operations tools supporting autonomous spacecraft.**

TABLE OF CONTENTS

1. INTRODUCTION

On-board spacecraft autonomy is key to enabling high-priority scientific investigations of fast-changing phenomena such as storms on gas giants and plumes on icy worlds. The temporal variability of these phenomena, combined with high light-speed latency and very limited downlink bandwidth, precludes traditional human-in-the-loop operations, requiring spacecraft to make autonomous system-level decisions without consulting scientists and operators on the ground.

A number of on-board autonomy capabilities for planning and scheduling [1], [2], [3], [4], [5], event detection [6], autonomous orbital [7] and surface [8], [9] navigation, and fault management [10] have been demonstrated to high technology readiness level. Some of these capabilities have been already adopted as part of routine operations workflows in planetary missions, such as autonomous event detection and navigation for surface exploration missions. However, the problem of designing mission operations to accommodate on-board *planning and scheduling* remains an active area of research.

Autonomous planning and scheduling allows a spacecraft to plan and execute activities opportunistically, based on previous observations and on availability of on-board resources; for instance, an autonomous planning module can decide to perform follow-up observations of a detected event of interest (say, a transient plume) if sufficient power, thermal, and data volume resources are available, potentially increasing science returns compared to pre-planned sequences, and providing access to otherwise-unobservable short-duration phenomena. However, this concept of operations requires a radical rethinking of ground operations [11]. During uplink operations, scientists and operators must provide the on-board planner with their *intent*, represented as a prioritized set of high-level activities (each characterized by pre-conditions and expected effects) that the spacecraft should attempt to accomplish, and assess the likely outcome of such intent on the spacecraft state — a radical departure from current operations, where operators compose a sequence of commands, each with a prescribed start time and maximum duration, and the spacecraft executes the sequence. During downlink operations, in turn, engineers must reconstruct the planner's decisions, assess their impact on the spacecraft state, and identify any anomalies that may be masked by the presence of on-board autonomy — a significantly more complex task compared to the current practice of confirming nominal execution of a prescribed sequence, whose impact on the spacecraft state is comparatively more predictable in advance.

978-1-6654-9032-0/23/$31.00 ©2023 IEEE.

In this paper, we present the results of a two-year effort to develop and test workflows, user interfaces, and tools designed to bridge this gap, supporting operators of future autonomous robotic explorers.

State of the Art

Operations of conventional spacecraft—Current deep space explorers and Earth observation spacecraft are typically operated through a ground-in-the-loop process where operators uplink *sequences* of commands that the spacecraft performs at prescribed times (or they command staged on-board sequences to be executed); an on-board executive may also verify that specific conditions are met before a task is executed [12]. Most of the existing work on the uplink operations has focused on the traditional paradigm of generating sequences of commands (with significant margins to cope with uncertainty) on the ground. The process of generating sequences is done either manually (i.e. operators think through the desired commands and their timing, and explicitly encode a sequence in a format that the spacecraft can execute), or using ground-based planning tools that develop and validate a sequence based on high-level goals specified by the operators [13], [14], [15], [16]. In the latter case, operators have to specify not only a dictionary of commands and how they should be executed, but also high-level activities and tasks (with their respective set of pre-conditions, effects, resource constraints, and temporal constraints) and goals that need to be satisfied by the ground-based automated planners when generating a sequence.

These automated planners traditionally have high computational cost, which makes on-board deployment impractical for most planetary missions, where computational resources are limited. As planning algorithms become more efficient and available on-board computational resources increase, the planning process is likely to gradually move on-board. In this case, we will no longer uplink sequences but rather intent expressed as goals – a path that has been pursued in Mars 2020 rover operations with the on-board planner [17], [18]. However, this new paradigm creates challenges for the uplink process with respect to capturing operators' intent and creating goals, commands, activities, constraints and spacecraft models: if these elements are not well specified, critical science opportunities and observations may be lost, potentially jeopardizing the achievement of primary mission objectives. These challenges are amplified for spacecraft that need to be operated further into the Solar System, such as missions to the Ice Giants, where the mission has limited communications bandwidth, short-duration science opportunities, and large uncertainty related to the environment and target science observations. Uplink operations of future autonomous spacecraft will require an iterative design process of intent that helps operators not only build the right set of goals, but also understand the possible outcomes, and trust that the on-board planning and scheduling software will achieve the desired intent. To our knowledge, no existing tools or framework fully addresses this iterative uplink process specifically for on-board autonomy. Moreover, it is common to see tools developed in an ad-hoc fashion that do not provide a fully-integrated experience for the uplink-downlink cycle.

For downlink, in turn, spacecraft typically transmit three classes of data products to ground operators: (i) channelized data, which provide time-series measurements of key variables representing the state of the spacecraft; (ii) event notifications [commonly referred to as event records, or EVRs, at the Jet Propulsion Laboratory (JPL)], short time-stamped messages reconstructed to a text string on the ground, produced by on-board flight software when prescribed conditions are satisfied; and (iii) binary data products, which encompass a variety of files including instrument data, engineering images, and any other file produced by the on-board software for downlink. Downlink operators examine the downlinked channelized data, EVRs, and data products to assess whether the spacecraft correctly executed the sequence that was uplinked, whether any errors were raised by the flight software and reported in EVRs, and whether the spacecraft state (as measured by the channelized data) fits within nominal bounds. A variety of tools have been developed to assist operators in this effort, allowing ground operators to automatically verify whether the downlinked data satisfies user-prescribed rules (ranging from simply checking whether a value lies within bounds, to assessing whether the sequence of EVRs received matches a prescribed pattern), including AMPCS [19] and VISTA [20], part of NASA's Advanced Multi Mission Operations System (AMMOS) [21]. Tools such as Jupyter notebooks [22], [23] are also routinely used to examine downlinked data in detail through custom Python scripts. Further, a new system called Rounds is in use by several JPL missions to perform more complex automated analysis of telemetry than was possible via legacy channel alarm checks. However, critically, existing tools and workflows assume that the spacecraft's nominal course of action is known in advance (namely that, in the absence of failures, the spacecraft will fully execute the uplinked sequence), and focus on identifying and highlighting deviations from this expectation; in contrast, with on-board autonomy, the spacecraft's decisions are *not* fully known in advance, which introduces a major challenge for downlink operators accustomed to monitoring nominal sequence execution.

On-board planning and scheduling—Several on-board planning and scheduling tools for autonomous spacecraft have been demonstrated to high technology readiness levels, including the CASPER on-board planner for the EO-1 spacecraft [24], the Remote Agent Planner/Scheduler for the DS-1 spacecraft [1], the MEXEC planner demonstrated on the Asteria spacecraft [3], and the onboard planner currently under development for the Mars 2020 Perseverance rover [4], [5]. Operational tools such JPL's Crosscheck [25] have been developed to help operators interact with such planners when they are used *on the ground* (i.e., when the planner is used by operators to create a sequence that is then uplinked to the spacecraft). However, no tools are currently available to interact with *on-board* planners—a critical gap which this effort aims to overcome.

Contribution

Our contribution is threefold.

First, building upon our prior work [26], we assess required changes to operations workflows to support autonomous spacecraft. We identify new required roles, including an "autonomy engineer" dedicated to supporting the translation of operators' intent into goals understandable by on-board autonomy software and interpreting the on-board software's decisions. We also assess the need for new software tools to support these roles: in particular, operators will need user interfaces and algorithms to support the capture of scientific and engineering intent; simulate and assess the likely outcomes of their intent, both at the tactical level (i.e., what tasks will be executed on board) and at the strategic level (i.e., how the tasks will help achieve high-level scientific requirements); and reconstruct decisions made by on-board autonomy software, understand their rationale, and assess their impact on on-board resources.

Second, we design and prototype nine user interfaces and two sets of algorithmic tools designed to fulfill these requirements and increase operability and trust amongst operators.

Finally, we conduct an in-depth design simulation [27] where JPL spacecraft operators enact the proposed workflow and interact with the tools we designed, simulating the operations of an autonomous spacecraft in the Neptune-Triton system across multiple flybys; the design simulation assesses the suitability of the proposed workflow and tools, and identifies critical directions for future development.

Organization

The rest of the paper is organized as follows. In Section 2, we formally define the problem of operating autonomous spacecraft, and lay out our assumptions. Section 3 discusses the proposed workflows and roles for operations of autonomous spacecraft, and highlights key differences with respect to traditional, ground-in-the-loop operations. Section 4 describes the user interfaces and algorithmic tools developed in this effort, and the design thrusts motivating their selection and development. In Section 5, we describe how the performance of the proposed workflow, tools, and algorithms was evaluated through a design simulation; the findings and recommendations stemming from the design simulation are reported in Section 6. Finally, in Section 7, we summarize our results and lay out directions for future research.

2. PROBLEM STATEMENT

A number of autonomous capabilities have been developed for robotic Solar System explorers, including autonomous planning and scheduling, autonomous locomotion, autonomous failure detection, identification, and recovery (FDIR), and autonomous detection of targets of scientific interest. In this work, we focus on developing operations tools for on-board autonomous *planning and scheduling*. We also assume the availability of on-board autonomous event detection (e.g., [6]), but we do not specifically focus on operations of such tools, except to the extent that they interact with on-board planning and scheduling (e.g., a detection of an event might trigger scheduling of follow-up observation activities).

We define the autonomous on-board planning and scheduling problem as follows.

Problem 1 (On-board planning and scheduling) Consider a spacecraft characterized by a set of states that affect autonomy decisions and that can be measured on-board (e.g., available power, temperature, and availability of certain instruments, or whether a scientific phenomenon of interest has been detected). Ground operators provide the spacecraft with a list of high-level activities that should be executed, ordered according to their priority; each activity can only be executed if certain state constraints are satisfied (for example, an instrument may only be turned on if the temperature falls within specific ranges; or a follow-up observation may only be executed if a phenomenon of interest has been detected). The expected impact of executing an activity on the spacecraft states is also assumed to be known (e.g., turning on a heater is expected to increase both temperature and power draw by a known amount). The data structure encoding the tasks and their priority, impacts, and constraints is denoted as a *task network*. The *on-board planning and scheduling problem* consists of selecting which activities should be executed, and when, so as to maximize a given utility (e.g., the the number of observation activities that are executed), while ensuring that spacecraft state and temporal constraints are satisfied.

On-board planning and scheduling systems that solve the planning and scheduling problems (referred to as "planners" in this paper) have been developed to high technology readiness level [3], [5], and demonstrated in spaceflight. However, to date, the problem of how ground operators and scientists will interact with such on-board autonomy largely remains an open question.

Uplink operations for future autonomous spacecraft will have to solve the *intent capture* problem: that is, how to capture and translate operators' latent intent, and the many trade-offs between science and engineering goals inherent in spacecraft operations, into a specification that can be used by on-board autonomy software, as opposed to a scripted sequence of commands. Downlink operations, in contrast, will focus on reconstructing and explaining the spacecraft's autonomous decisions, and assessing the spacecraft's state and health—a problem made significantly more complex by the presence of on-board autonomy.

We define the problems of *uplink operations for autonomous spacecraft* and *downlink operations for autonomous spacecraft* as follows:

Problem 2 (Uplink operations of an autonomous spacecraft) Consider a spacecraft equipped with on-board autonomous planning and scheduling capabilities. Develop a workflow, user interfaces, and algorithms to assist scientists and spacecraft operators in (i) defining a *task network*, i.e., a set of prioritized tasks, associated with state constraints and state impacts, that can be used as an input by the spacecraft's on-board autonomy to generate plans that capture the intent of scientists and ground operators, and (ii) understanding and evaluating the possible outcomes given the task network, the on-board autonomy capability, and uncertainty about the spacecraft and its environment.

Problem 3 (Downlink operations of an autonomous spacecraft) Consider a spacecraft equipped with on-board autonomous planning and scheduling capabilities, and a task network provided to the spacecraft by uplink operators. Develop a workflow, user interfaces, and algorithms, and identify data products that should be downlinked so that operators can (i) understand what tasks were executed on board, (ii) understand why the on-board autonomy software scheduled these tasks (and not others) for execution, and (iii) reconstruct and assess the state of the spacecraft, including any uncertainty associated with it.

In the rest of this paper, we propose and evaluate workflows, user interfaces, and algorithms to address these challenges.

3. WORKFLOWS FOR AUTONOMOUS SPACECRAFT OPERATIONS

In this section, we describe a proposed workflow for autonomous spacecraft operations, and highlight new roles and tools that will be required to support such operations. A preliminary version of the workflow was presented in [26].

Most missions plan across several planning cycles with varying time horizons. We refer to longer planning cycles as "strategic" and shorter cycles as "tactical". While significant planning occurs at a strategic level, starting prior to launch,

we envision that the majority of the key operational changes due to autonomy will occur at the tactical level; accordingly, we focus on the design of a tactical level planning workflow incorporating data collected in prior downlinks, where intent is revisited and tasks are created, updated, or deleted for the next planning cycle.

Uplink Operations

The proposed workflow builds upon conventional uplink operations consisting of three core operator groups: science and instruments (consisting of scientists, instrument engineers, and the science operations working group, or SOWG), spacecraft engineering (including the spacecraft engineer and the data management engineer), and cross-cutting (including mission planners and autonomy engineers). Each group must capture their intent, in the form of a task network (defined in Problem 1), so it can be conveyed to a spacecraft with on-board autonomy. This operations concept builds on iterative planning approaches used in conventional operations. Operators infer the impact of their intent in a model-predict-adjust process [26], integrating intent capture, modeling, outcome prediction, and assisted explanation of predictions. This process aims to build operator trust that the on-board autonomy will achieve their goals.

Workflow Process—Engineering teams review initial conditions (i.e., the expected spacecraft state at the beginning of execution of the next plan) at the beginning of a tactical planning cycle, while the science team identifies new desired science observations or updates to existing observations. Scientists and engineers then have an opportunity to change the specified intent, in the form of a task network, and receive near-instantaneous assessment of its possible impacts on the mission in the "predictions" phase (described below) using simulations.

Scientists may change their intent specification with new goals and prioritize them in negotiation, resulting in different spacecraft behavior. We note the autonomy engineer role as a key operator responsible for owning system-level autonomy, ensuring sound autonomous algorithms and behavior, and troubleshooting undesirable outcomes from simulation to maximize performance.

The team executes simulations to predict how the task network will be executed on-board (increasing coverage of uncertainty elements and fidelity of simulation as needed), analyzes the results, and adjusts intent accordingly. Inspection of simulation results (either individually or as clusters of likely outcomes) may lead the team to adjust or remove goals.

New Tooling—This highly iterative process calls for new tooling to support the proposed collaboration between teams, and software capable of (i) capturing intent from different groups and representing it in the form of task networks, (ii) running Monte Carlo simulations, and (iii) visualizing and analyzing predictions of possible outcomes from the simulations. Tooling should also afford the ability to capture environmental, science, and spacecraft performance variability in order to capture a realistic distribution of possible outcomes.

Downlink Operations

In downlink operations for current flight projects, several downlink engineers typically monitor and review spacecraft systems and reports to feed forward data for uplink planning using dashboards, scripts, reports, graphics, and 2D/3D views, and interactive and exploratory tools.

Workflow Process— The science team must analyze downlinked data to understand whether their intent has been achieved. Because the uplinked plan may embody a wide range of possible outcomes, downlink operators must reconstruct which tasks were executed, and understand why those tasks (and not others) were selected by the on-board planner. The spacecraft team must also analyze the health and safety of spacecraft subsystems, and reliably estimate the spacecraft's state. An autonomy engineer with in-depth knowledge of on-board autonomy can help explain behavior and connect uplink predictions to the spacecraft's reconstructed behavior. All teams must converge on shared understanding of downlinked data, and build models to stage initial conditions for uplink planning.

New Tooling— The process described above imposes two needs with respect to tooling. First, tools must reliably reconstruct on-board state and plan execution, possibly in the face of limited downlinked data. This reconstruction, denoted as "actuals", must be compared to the set of possible outcomes from uplink planning, and connected to the operators' intent. Second, tools must enable investigation such that operators can interpret *why* the on-board autonomy software made decisions, despite potentially limited information from the planner due to limited bandwidth. To address this, we propose building user interfaces that enable users to observe correlations between different data sources (e.g., executed tasks, spacecraft states used by autonomy, and EVRs produced by flight software), so as to increase the operators' situational awareness, and provide them with all relevant information to form a mental model of the autonomy's decisions—all the while avoiding to overwhelm them with superfluous or redundant information.

4. USER INTERFACES AND ALGORITHMS

We are now in a position to describe the user interfaces and algorithmic tools developed to address the operators' needs outlined in Section 3. All user interfaces were designed in detail, and selected interfaces were also implemented as interactive browser-based applications; implementation of the remaining user interfaces is ongoing.

Uplink User Interfaces

Science Planning— The Science Planning tool (Figure 1), tailored to operations of orbiters and flyby missions, supports adding and updating science goals and related activities in the task network. It gives scientists visibility into what science goals are active during a selected time period, where along the trajectory a desired observation can be performed, and how the observation conflicts with other possible activities. To edit the task network, scientists can view and update the priority of observation tasks, view and update the logic driving conditional execution of tasks, and preview what outcomes may happen as a result of modifications to the task network. 2D or 3D graphical views of the target system provide a preview of the observation geometry. Upon modifying the goals, the team can see how the changes impact predicted progress towards the goals through simulations. The Science Planning tool leverages concepts from Science Opportunity Analyzer [28] (e.g., opportunity search and design with preview graphics) and VERITaS [29] (e.g., science requirements modeling and measurement).

Task Network Editor—In this work, intent is ultimately represented as a *task network*, described in Problem 1. This particular representation is the foundation of timeline-based

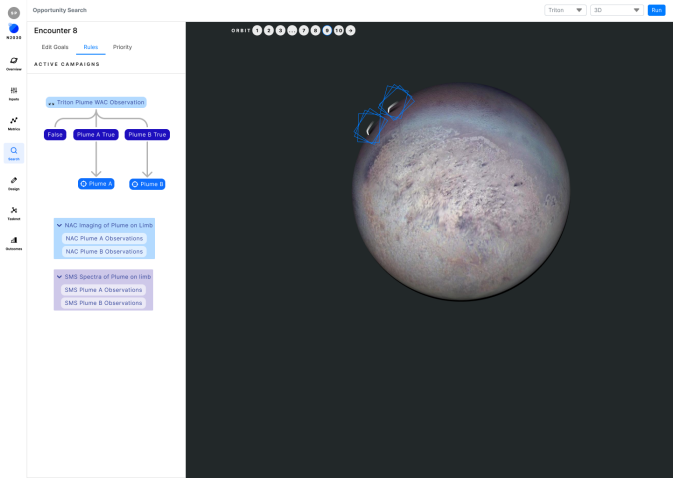


Figure 1. Science Planning tool: visualizes science data, aids identification of science opportunities, and allows scientists to add and update science goals.

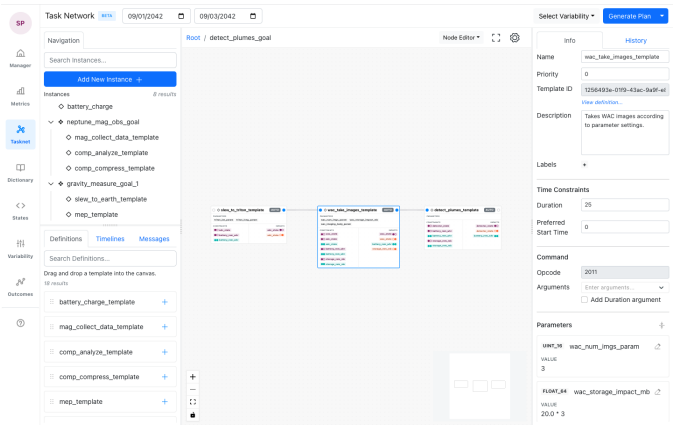


Figure 2. Task Network Editor: enables operators to create and edit task networks. This tool is implemented as a browser-based application.

temporal planning and Hierarchical Task Network planning. While our proposed framework is general, our current implementation uses MEXEC [3], [30] as the core planning and execution system on-board the spacecraft. Therefore, capturing intent follows the task network formulation described in [3], meaning that goals are expressed in the form of tasks, including their pre-, post- and maintenance conditions, impact constraints, temporal and resource constraints, and priorities, as well as ordering constraints and hierarchical decomposition of tasks into sub-tasks.

The Task Network editor shown in Figure 2 is a tool for creating and visualizing task networks. Engineers, autonomy experts, mission planners and operators may create, update, delete, and validate tasks either from scratch or starting from templates, and preview simulation outputs running a planner such as MEXEC. The tool provides a high-level view of science campaigns, and lets users create and inspect subtasks.

This tool centralizes intent capture and representation from different teams. For example, the observational goals provided through the Science Planning tool are added as tasks in the task network representation managed by the Task Network editor, i.e., goals are merged and represented as a task network. Such a representation matches semantically with MEXEC's input; the tool provides a translation process from the task network graphical representation to the input format required by MEXEC.

The Task Network Editor's design was informed by planning and sequencing tools including SEQGEN [31], APGEN [32], and COCPIT [33] (e.g., scheduling, validation, and plan timeline visualization), as well as Crosscheck [25](e.g., explainability, visualizations of planning cycles).

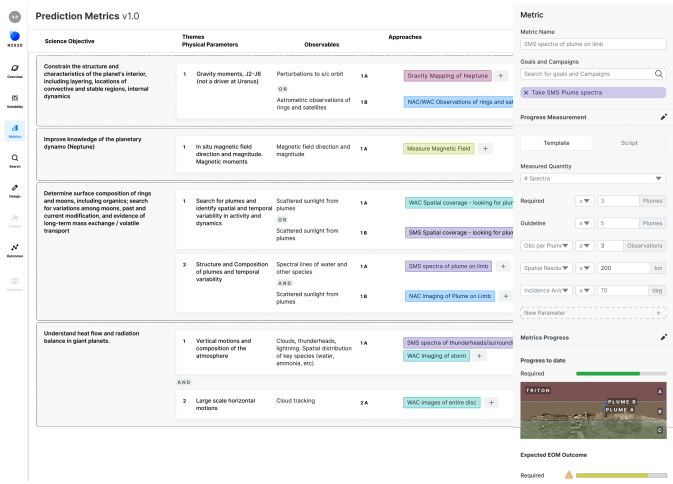


Figure 3. Metrics tool: traces mission science objectives to operational requirements. This tool is implemented as a browser-based application.

Metrics—The Metrics tool (Figure 3) captures the traceability of mission science objectives all the way from Level 1 (L1) requirements (requirements from the customer) [34] down to performance metrics as expressed in operational requirements. The original version of the tool was presented in [26]; in this work, we have added explicit connections between L1 requirements and metrics. Metrics define the quantity and quality of science data that need to be collected in order to answer a science question. L1 requirements can be satisfied with combinations of metrics, and users can encode the logic relating them in the Metrics tool. While users can review completed and projected metric success within the Metrics tool, the metrics data is also integrated into other tools to support tracking of the mission progress and making trade-offs between different version of the task network. The Metrics tool builds off of the Clipper M-STAF/P-STAF [34] [35], specifically its structure and use of a science traceability framework, as well as VERITaS [29] and CLASP [36], both tools for modeling of science requirements and measurements.

Variability—The Variability tool, shown in Figure 4 and originally proposed in [26], allows a user to specify a desired level of uncertainty for environmental and spacecraft parameters to be used in Monte Carlo simulations. It captures both engineering and science variability, where engineering variability describes the spacecraft performance (e.g., task duration, power draw, and size of data products) when executing tasks, and science variability describes the environment (e.g., how many features of interest are present, the size of such

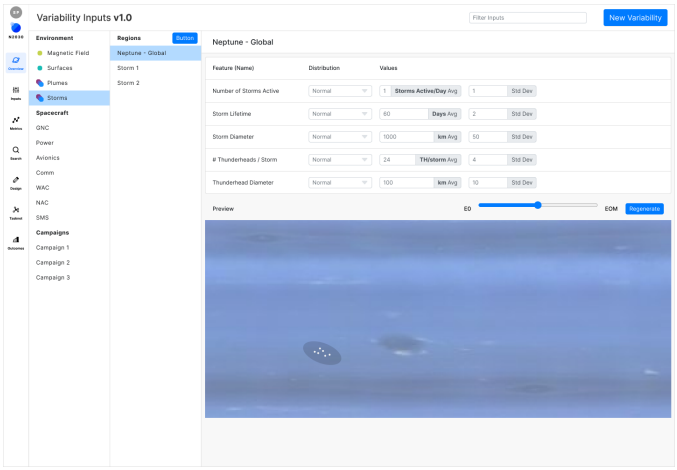


Figure 4. Variability: captures spacecraft and environmental variability used in simulations. This tool is implemented as a browser-based application.

features, and start and end times of time-varying features). Users can specify different types of parameterized probability distributions, and set the distribution's parameters. Monte Carlo simulations using variability drawn from the user-specified distributions help operators identify edge cases and potentially unfavorable outcomes, as well as exceptionally favorable outcomes. For science variability, operators can validate their inputs with visual previews. The tool leverages concepts from the Mars 2020 planning tools [37] and work on probabilistic assessment of failure risk of the Europa Clipper spacecraft due to radiation [38].

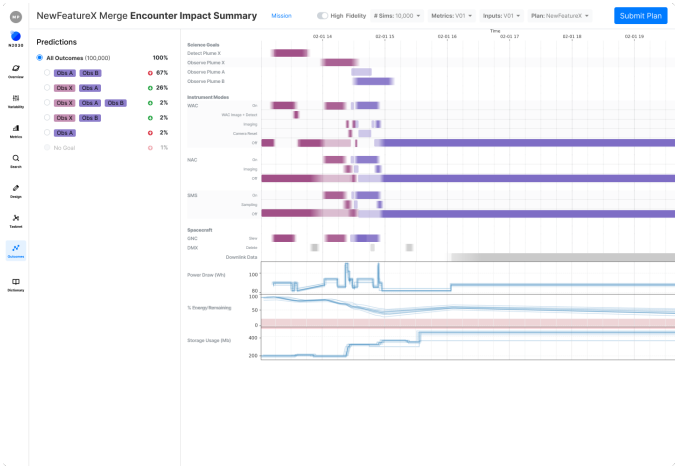


Figure 5. Prediction: Supports analysis of Monte Carlo simulation results. This tool is implemented as a browser-based application.

Prediction—The Prediction Outcomes tool (Figure 5) supports visualization of the wide range of executions produced by Monte Carlo simulations. It allows operators to see, at a high-level, the distribution of executed and skipped tasks, and the related spacecraft states, observed in the simulations. Operators can inspect the performance of individual simulations and understand the conditions that led to them, helping inform decisions to edit the task network. Executions and outcomes can be clustered based on goals achieved and tasks executed, and also on the presence of anomalies. The likelihood of each cluster may help scientists identify whether they might achieve the science they need. The tool leverages concepts from planning and scheduling tools including Raven [39] and Copilot (Mars 2020 Rover) [37], as well as VERITaS [29].

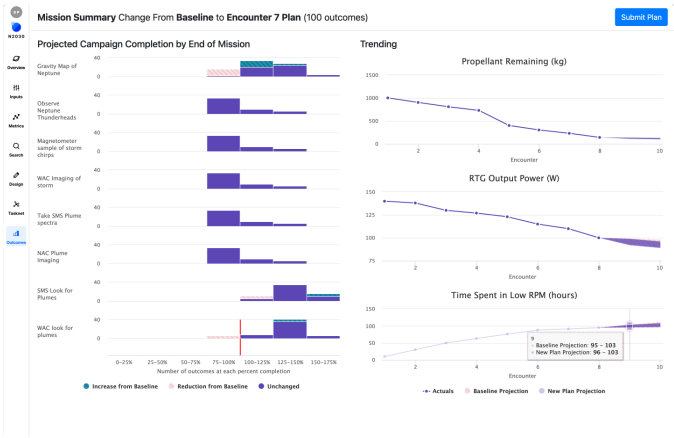


Figure 6. Mission Outcome: communicates mission-level impacts of Monte Carlo simulations.

Summary (Mission Outcome) —The Mission Outcome tool (Figure 6), originally proposed in [26], shows the results of many Monte Carlo simulations for the duration of the mission to support analysis of changes to intent from the strategic perspective. It highlights the difference between a new proposed task network and a baseline one in terms of key trends, campaign success rate (as measured by the metrics specified in the Metrics tool), and likely outcomes. This tool helps users identify whether a change to the task network has unintended impacts, or if that change supports campaign progress. The tool leverages modeling concepts from VERITaS [29] and CLASP [36], as well as work on probabilistic assessment of failure risk of the Europa Clipper spacecraft due to radiation [40](e.g., visualization of possible mission outcomes).

Downlink User Interfaces

Plan Reconstruction —The Plan Reconstruction tool (Figure 7) plays back what the spacecraft planned to do based on estimated state and resource values in incremental steps, and shows what tasks were actually executed, so that downlink operators can assess what the on-board planner did and determine the cause of its decisions [41]. To alert users to the status of executed goals during playback, indicators show whether a task is "complete," "in progress," "incomplete," or "waiting" to be scheduled. The reconstructed plan can be compared to clustered predicted outcomes shown in the Prediction Outcomes tool. When users roll over executed tasks that captured images or science measurements, they can view the related downlinked data products (e.g., captured images), helping situational awareness. The tool also contains state estimation features (described next) that allow the reconstruction of on-board states based on (possibly incomplete) telemetry and on models of the spacecraft. The uncertainty of the estimate is also visualized to ensure the operator is not overconfident in the estimates. The design of the tool was inspired by existing downlink subsystem dashboards, and by JPL's SEQGEN [31]

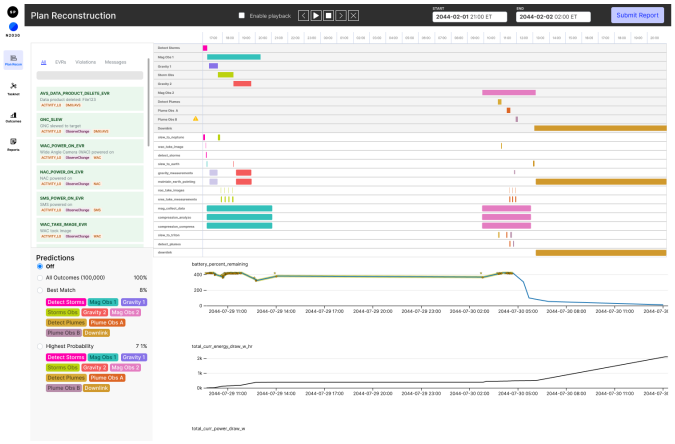


Figure 7. Plan Reconstruction: Step-by-step planner playback and comparison to prediction clusters.

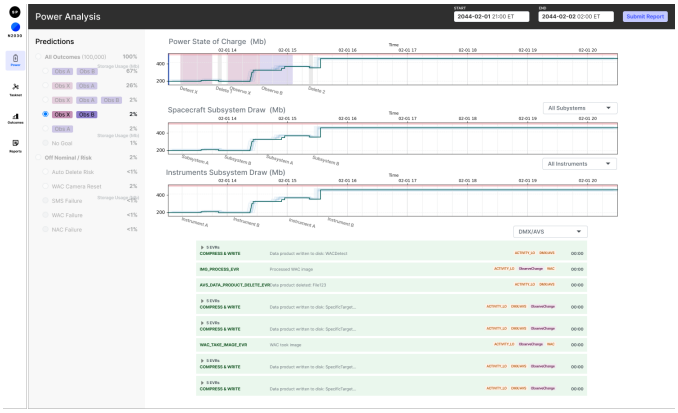


Figure 9. Power Analysis: Example of a subsystem specific dashboard that utilizes outcome prediction and EVR table.

and Crosscheck [25].

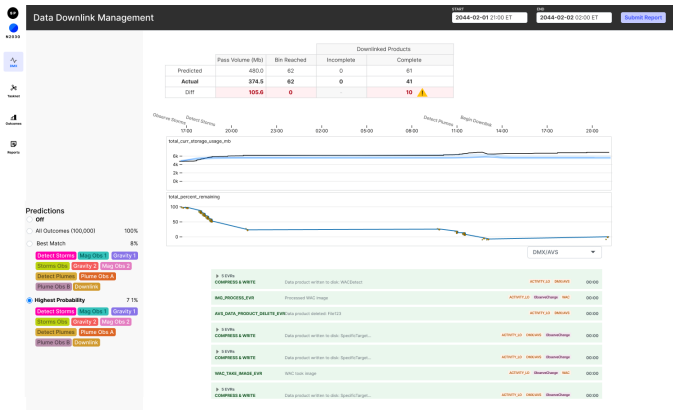


Figure 8. Data Management Dashboard: Displays predicted and actual downlinked products.

Data Management Dashboard— The Data Management Dashboard (Figure 8) displays information relevant to the Data Management Engineer. Specifically, it includes a table of predicted and actual data products that were downlinked, resource views containing the total current storage and total storage percent remaining, and resource prediction overlays organized by cluster. A table displaying EVRs is filterable by subsystem or instrument.

Power Dashboard—The Power Dashboard (Figure 9) is an example of a subsystem-specific dashboard that utilizes the outcome prediction component, curated resource views, and a filterable EVR table. Specific to this dashboard is the display of Battery State of Charge, Spacecraft Subsystem Data Draw, and Instruments Subsystem Draw. Overlaid on the resources are indicators of when tasks impacting the spacecraft's power state were executed.

Algorithms

Prediction Engine— The prediction engine (illustrated in Figure 10) allows operators to assess the range of possible outcomes for a given task network and a given level of uncertainty (specified in the Variability tool). The prediction engine allows to run large-scale Monte Carlo simulations, sampling stochastic conditions according to the Variability tool's inputs, and outputs the empirical distribution of outcomes of the simulations. As such, the Prediction Engine is responsible for three main steps: (i) sampling variability distributions, (ii) running a simulation of the spacecraft and its environment with the sampled variability as an input, and (iii) recording all pertinent outputs and resulting state information into a shared database that can be queried by operators.

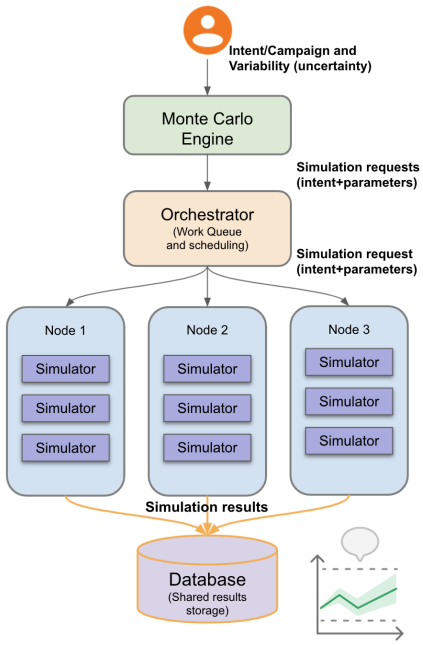


Figure 10. Overview of the architecture of the Prediction Engine.

In order to sample the variability distributions, we use a Monte Carlo approach [42]. Such an approach has been used previously on the M2020 Perseverance Rover mission to predict and visualize rover plan execution with the Copilot

tool [37]. Variability distributions defined in the Variability tool are used as an input into the Prediction Engine, using the distributions to produce a series of sampled inputs into the simulator. Since the Monte Carlo method needs a large number of sampled cases to be effective, a key design driver is the ability to efficiently run hundreds to thousands of high-fidelity simulations (which can require tens of hours of computation each) in parallel. While simulations can run faster than real-time, the speedup is limited, and requires intensive computing resources. In order to more easily deal with these requirements, we built the Prediction Engine on Kubernetes, an open-source container orchestrator, to give us the ability to easily scale resources and achieve parallelism in our simulations [43]. Not only does Kubernetes enable efficient orchestration and dynamic scaling, but it also easily allows the deployment of the engine on a cloud provider to help meet the computing requirements of our simulation. The output of the simulations are stored in a PostgreSQL database [44], also hosted in the cloud. With Kubernetes, the Prediction Engine is designed to be highly configurable, not only in terms of specifying simulation parameters such as the number of runs, but also in terms of configuring the degree of parallelism and the amount of computing resources available.

Since each simulation run in Monte Carlo is treated independently, we selected a generator-worker approach to infuse parallelism into the Prediction Engine. In this approach, the generator is responsible for performing the sampling over the variability distribution, converting the samples into the input format required by the simulator, and adding this input onto a shared work queue, repeating this process for the number of simulations specified by the configuration. Meanwhile, many workers are tasked with taking input requests from the shared work queue, running a simulation based on the sampled input, storing the simulation output in the shared database, and shutting down if no items remain on the work queue. While the number of workers that can be run in parallel is compute-limited, we were able to achieve a speedup of roughly 400x when comparing against sequential real-time simulations. Figure 10 shows the high-level workflow of the Prediction Engine, from sampling to storing outcomes.

State Estimation—We have developed and implemented a set of algorithmic tools to model continuous spacecraft states and their relationships with on-board measurements. With these models, downlink operators can estimate and infer states based on received telemetry even if the data is sparse, noisy, or corrupted. In this project, we use three different sets of tools including factor-graph-based and task network-based estimation algorithms for continuous states, and Hidden Markov Model-based algorithms to estimate discrete states.

• *Factor Graph-Based Estimation* — This set of tools uses factor graph representations of the spacecraft and environment [45]. We use factor graphs to model probabilistic relationships between states of the spacecraft and its environment, and the spacecraft's observations. As shown in Figure 11, in factor graphs, vertices represent spacecraft states at a point in time, and edges (denoted as factors) represent the probabilistic relationships between the states and the measurements received on the ground. The goal of factor graph-based estimation is to identify the most likely set of states given the measurements received and the known probabilistic relationships between states. Nonlinear optimization tools are used to find the maximum-a-posteriori (MAP) estimate of the most likely states, e.g., the set of state variables that best explain the spacecraft's observations.

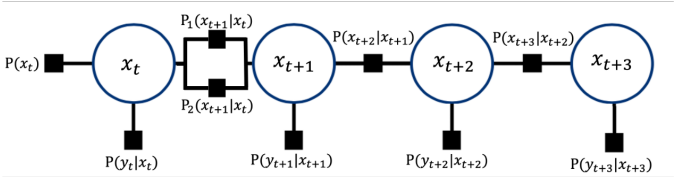


Figure 11. An example of a factor graph that shows the probabilistic relationship between state x and measurement y at different time steps. In this example $\mathrm{P}(x_t|x_t+1)$, $\mathrm{P}(x_t|y_t)$, and $\mathrm{P}(x_t)$ represent the probabilistic relationship between state x at time steps t and $t+1$, the probabilistic relationship between state x and measurement y at time t, and the initial distribution of state x at time t, respectively. The edge between the states x_t and x_{t+1} is multi-modal with two hypotheses. Also, multi-modal distributions can be used to represent the probabilistic relationships.

Factor graphs can capture multi-modal models and multi-modal distributions of states and measurements. In multi-modal models, multi-hypothesis factors are used to represent separate hypotheses, e.g., the presence or absence of a phenomenon of interest, or the presence of a fault in a spacecraft sensor. In this case, factor graph optimization looks for the hypothesis which best explains the measurements. Multi-modal distributions such as non-Gaussian distributions can also be used to represent nonlinear process and measurement models and uncertain data associations, e.g., the possible geographical location of a phenomenon of interest.
We use open source libraries to construct and solve the factor graphs. More precisely, we use GTSAM [46] for standard factor graphs, MH-iSAM2 [47] for factor graphs with multi-hypothesis factors, and MM-iSAMv2 [48] for factor graphs with multi-modal distributions. The key advantage of factor graph-based estimation is its ability to accommodate complex spacecraft and environment models with arbitrary relationships between state variables. However, the price of such flexibility is both significant computational complexity, and a high modeling effort to construct the factor graphs.
• *Task Network-Based Estimation for Continuous States* — A key difficulty in the use of factor graph-based tools is the need to develop models of the spacecraft and its environment as factor graphs—a complex undertaking that can be daunting for operators. To address this, we developed a second set of tools designed to leverage existing spacecraft and environment models used by on-board autonomy software, specifically, the MEXEC planning and execution system [3], [30].
The developed state estimator is a Kalman filter [49]. It uses state and task impact models of the task network used by MEXEC to construct a linear model of the spacecraft and its environment; it reconstructs what tasks were executed by querying EVRs downlinked by the spacecraft; and it uses downlinked channelized data as measurements for the estimator.
While MEXEC only uses deterministic models for tasks, the Kalman filter accommodates probabilistic models, with user-defined covariances, to account for execution and measurement noises. More precisely, the developed state estimator performs the following steps:
1. Task Network-to-Model: The estimator reads the task network and builds the Kalman filter matrices accordingly. Probabilistic state projection models are built using the task impact models of the task network, and additive Gaussian noises (specified separately) account for execution noises.

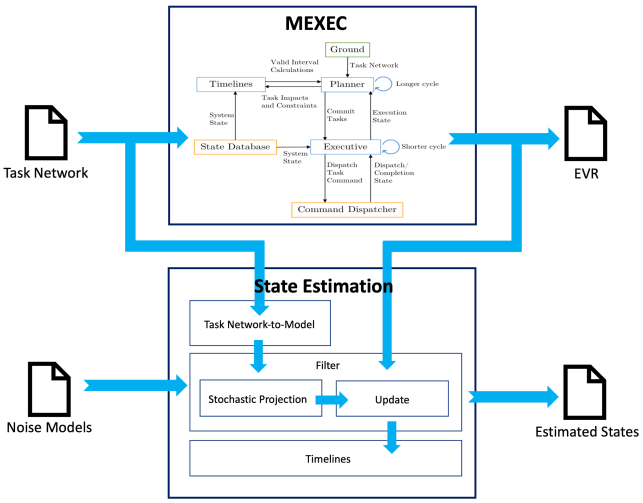


Figure 12. Task Network-Based Estimation.

Also, linear measurement models with additive Gaussian sensor noises are used to create the probabilistic measurement models.

2. Stochastic Projection: The constructed probabilistic state projection models and the sequence of the executed tasks inferred from EVRs are used to obtain predicted state values, as well as the associated uncertainties due to execution noise.

3. Update: At this step, on-board state measurements are combined with the stochastic projections to compute the state estimates and estimation uncertainties. More precisely, in this process, the estimator compares the actual on-board measurements with the predicted measurements obtained from the probabilistic measurement models, and then uses the optimal Kalman filter gain to update the states and the associated uncertainties from the stochastic projection step.

Figure 12 shows the components of the designed estimator.

- *Input-Output Hidden Markov Models for discrete states* — We also developed a set of algorithmic tools to model discrete spacecraft states and their relationships as an Input-Output Hidden Markov Model (IO-HMM) [50], [51], and estimate their value based on telemetry. The proposed approach allows operators to describe the high-level behavior of a spacecraft as a set of interconnected components, each characterized by inputs, outputs, and a set of discrete states, or modes. Each component's internal state characterizes the relationship between the component's inputs and outputs; the goal of the estimation process is to estimate the state of each component. For example, the high-level behavior of a very simple attitude control system (ACS) can be represented as follows:

```
varType('Boolean : False,True')
varType('Command : Idle,Track,None')

componentType('siderostat',
    inputs = 'Command:i',
    outputs= 'Boolean:valid',
    modes= {'Tracking' : '["True"]',
            'Idling' :   '["False"]',
            'Error':   {'["True"]':0.5, '["False"]':0.5}},
    transitions = ["Tracking -> Idling : 0.99 : i == 'Idle'",
                   "Idling -> Tracking : 0.99 : i == 'Track'",
                   "* -> Error        : 0.01"])
```

The ACS takes as inputs the commands "Idle" and "Track". The ACS can be in one of three states, "Tracking", "Idling", or "Error". The module outputs "True" if it is in tracking mode, "False" if it is in idling mode, and a uniform random value if it is in "Error" mode. When receiving an input command of "Idle" or "Track", the module switches to the corresponding internal state with 99% probability; and switches to state "Error" with 1% probability.

The modeling language is designed to be intuitive, and easy to use for operators with training in the tool, but no background in state estimation.

Critically, multiple components can be combined to model a complex spacecraft system, connecting the outputs of one component to the inputs of another. The resulting system can be modeled as input-output HMM, and state estimation algorithms such as the Viterbi algorithm [52] and the forward-backward algorithm [53] are employed to estimate the value of the states of each component, given the inputs provided to the components (which are known from EVRs downlinked from the spacecraft) and selected measurements of their outputs (also available as EVRs or channelized data). The Viterbi algorithm only reconstructs the most likely state of each component; in contrast, the forward-backward algorithm is able to reconstruct the marginal distribution of each state, at the price of slightly increased computational complexity.

The resulting estimates are integrated in the Plan Reconstruction tool, described above, in order to assist operators in estimating the behavior of spacecraft components such as onboard detectors (and specifically, to identify the presence of false positives) and other state-machine-based flight software functions. Figure 13 shows the output of the tool, estimating the likelihood of the state of each spacecraft component over time, and allowing operators to correlate it with autonomy decisions and measured spacecraft states.

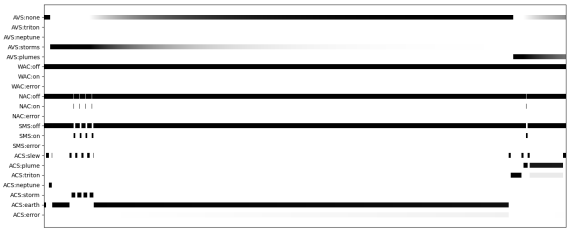


Figure 13. Output of the IO-HMM state estimation tool. The forward-backward algorithm is used to estimate the probability distribution of the discrete states of multiple spacecraft components. The horizontal axis denotes time; the color of each row denotes the likelihood that the spacecraft is in the state indicated by the row.

5. Design Simulation

Study Design

In order to assess the suitability of the proposed framework, including its workflow, user interfaces, and algorithms, we performed an in-depth design simulation [27] with engineering and science teams from JPL's operations community. In the design simulation, eight participants were asked to role-play uplink and downlink roles and enact two full uplink-downlink cycles for a simulated spacecraft in a realistic scenario including multiple instruments, conflicting science

goals, and unexpected anomalies.

The scenario we used reproduced the operations of a notional orbiter studying the Neptune-Triton system and, in particular, exploring the presence and evolution of plumes on Triton and storms on Neptune, and the gravity and magnetic fields of the two bodies. The spacecraft carried non-gimbaled wide-angle and narrow-angle cameras, a sub-millimeter spectrometer, and a magnetometer; gravity investigations were performed by monitoring the orbiter's location through two-way communication with the Deep Space Network. MEXEC was used for on-board planning. In the scenario, MEXEC planned and scheduled activities based on the task network provided by operators, the state of the spacecraft, and its observations (e.g., when plumes or storms were detected, the planner accommodated follow-up observation tasks with different instruments).

Due to conflicting pointing requirements and to the lack of a gimbal, camera observations and gravity measurements were generally mutually exclusive. In addition, low available downlink bandwidth required strict prioritization of on-board data products, making it impossible to readily downlink all the observations collected by the spacecraft. To further exercise the proposed workflow and tools, two anomalies were introduced: first, a noisy reading of the spacecraft's bus voltage caused the on-board planner to conservatively remove some follow-on observations from the schedule; and, second, bad weather at a downlink station prevented downlink of all engineering data, hindering the operators' situational awareness and encouraging the use of state estimation tools.

Participants

Participants in the design simulation included four JPL scientists and four experienced spacecraft operators. The scientists focused respectively on investigating storms on Neptune, plumes on Triton, the magnetic field of the system, and its gravity field. The operators represented the roles of autonomy engineer, mission planner, instrument engineer, and data management engineer. Participants were introduced to the scenario used in the design simulation, including the instrument suite and autonomy capabilities of the spacecraft; they also received a short, one-hour training on the framework proposed in this work.

Procedure

The participants simulated two "day-in-the-life" uplink-downlink cycles by role-playing operations according to the workflow discussed in Section 3 and interacting with prototypes of the tools presented in Section 4. The user interfaces used in the simulation were a mix of interactive, clickable tools (implemented as browser-based applications) and "Wizard-of-Oz" [54] prototypes where skilled facilitators manually modified the user interfaces (using the Figma tool) in response to users' inputs. The spacecraft and its environment were simulated in an ad-hoc mixed-fidelity simulator that faithfully captured the spacecraft's power state, attitude, data management, and communications, and used simplified models for the spacecraft's thermal state and for its instruments. Facilitators supervised the operations, providing information and context to the users where needed, and incorporated user inputs into the tools that were not implemented as prototypes. Each session of the design simulation was recorded and transcribed, allowing experimenters to analyze the operators' interactions with the tools in detail; operators were also asked to provide feedback both in daily written "diaries", and in debrief sessions with facilitators.

6. FINDINGS AND RECOMMENDATIONS

We grouped the research questions explored in the design simulation, as well as our findings and recommendations, into three categories: those that apply to uplink operations, those that apply to downlink operations, and those that are cross-cutting and apply to both. Cross-cutting findings look at the roles, processes, and overall tooling that we designed to support autonomous spacecraft operations. Uplink findings focus on intent capture and outcome prediction, and downlink findings focus on plan reconstruction and state estimation.

Cross-cutting findings: Roles, Processes, and Tools

Observations and surveys were collected during the design simulation across both uplink and downlink sessions to test whether the design of roles were realistically scoped, and ensure that the processes and tools were sufficient for operators to do their jobs. We found that the majority of operators reported that their roles and the processes within the design simulation were realistically scoped. However, given the scaled-down version of the design simulation as compared to real operations, the autonomy engineer noted that *"on a sufficiently complex mission with more complex plans, just understanding what the autonomous scheduler/planner did and why would be enough work for one person, with tuning/assessing the image-processing algorithms (cloud/plume detectors) as a job for a second person."*

Within the scale of the design simulation, the tool set was sufficient for operators to perform the jobs. One operator reported: *"I found the tools to be adequate and intuitive. I was able to do my 'normal job' with next to no training which I think is a huge accomplishment!"*

Beyond the scope of the design simulation, our team noted that additional work should be done to identify what tools and processes may be needed, if any, by the uplink team to address the continuously evolving probable outcomes of the spacecraft state, which evolve as additional information is received and processed by downlink operators.

To continue evaluating the benefit of these tools in real autonomous spacecraft operations, we recommend additional work be done to increase the breadth of use cases that operators perform, by giving them enough detail in the resource timeline projection to evaluate planning impacts beyond the upcoming flyby. We also recommend that the tools incorporate more granular details, such as instrument- and subsystem-specific effects and constraints in the Science Planning tool.

Uplink findings: Intent Capture

To test the effectiveness of the Intent Capture tools, including the Metrics and Science Planning tool suite for scientists and mission planners, and the Task Network editor for engineers, we sought to determine whether the tools fit within operators' mental models, whether the tools were sufficient for operators to express their planning intent, and whether the tools supported negotiations in the operations process.

Based on surveys completed by participants, we found that the majority of participants found the concepts introduced in the tools to be similar to ones that they had experienced on other missions – an encouraging result which suggests that the tools are a good match for the operators' mental models. Based on this finding, we infer that the Intent Capture tools contain enough similarities to existing operations paradigms to support user adoption of the parts of the tools that are

new and innovative within autonomous spacecraft operations. However, during the design simulation, we observed that scientists did not use the tools to explore opportunities to optimize science return without guidance from the facilitators. In contrast, engineers sought out opportunities to use on-board autonomy to maximize engineering and science tasks, and even requested additional features such as data downlink management modeling to pursue further optimization.

When surveyed as to whether their needs were reflected in the science planning tool and in task network, 100% of participants replied in agreement. However, given the limitations of the design simulation, it was not possible to re-run simulations and explicitly assess the impact of changes to the task network, a key functionality of the proposed workflow; this was noted as a limitation by some of the participants, who would have liked to fully evaluate the plan based on their proposed changes to the task network.

The Metrics tool, coupled with predictions of the likelihood of mission success, was observed and reported as effectively supporting negotiations among scientists in cases where scientists had to decide which science measurement should take priority. One scientist noted that *"The modeling was useful [in that] if you make a decision and set a new goal, [you will see] how much it advances or doesn't advance the science objectives."* Several participants noted that they would have liked to see the simulated impacts of the proposed updates as soon as they made changes, a feature that is planned but was not available in the limited prototype used during the design simulation. Since the design simulation scenario did not require engineers to negotiate resources, no data was collected on the usefulness of the Task Network tool in negotiating resources.

To improve upon the success of the Intent Capture tools, we recommend follow-up user research sessions be done with scientists to identify opportunities to communicate the benefits and methods for optimizing science return through on-board autonomy.

Within the entire suite of Intent Capture tools, we recommend ongoing testing with increasingly realistic use cases to ensure that the tools can support the majority of planning use cases. Additionally, we recommend ongoing user sessions in which interactive simulation results based on user feedback can be delivered within the duration of the planning session.

Uplink findings: Outcome Prediction

To test the effectiveness of the proposed design, we sought to determine whether the Variability tool, Outcome Prediction tool, and Summary tool fit within operators' mental models, helped users come to a conclusion about why different outcomes could occur onboard the spacecraft, and supported decision-making in uplink planning.

Observations collected during the design simulation confirmed that the designs' constructs, such as prediction clustering and variability inputs, fit within the operators' mental model. One example of this was seen when the mission planner used prediction outcomes to search for possible "showstoppers" in the most likely plan cluster (i.e., search for problems where the team would unequivocally decide not to uplink the task network as-is, for example due to failure to schedule important tasks, planning behavior deviating substantially from expectations, or no progress towards objectives), and then reviewed science gains and losses in terms of tasks executed and progress in other outcome clusters. In another example, the autonomy engineer used outcomes prediction to identify missing precedence constraints and revise the task network accordingly: specifically, a missing ordering constraint that required a slewing task to occur before collecting imaging data was identified and added to the task network. Finally, operators demonstrated a requisite understanding of the tools and their purpose during one uplink group session, where the majority of the time was spent discussing the relationship between outcomes and variability inputs.

It was also observed that, with limited training, participants used the Outcome Prediction tool to inform the majority of the planning process. Across sessions, the autonomy engineer identified improper task constraints based on predicted outcomes during initial planning. In uplink, the team used outcome prediction to frame discussion, and the downlink team used output from the tool to identify possible anomalies.

With on-board autonomy, operators will likely experience increased uncertainty concerning what will happen on-board as compared to current missions, in which command sequences are generated and uplinked from the ground. Despite this, operators reported high confidence in predicting what the spacecraft would do. Specifically, half reported that they were 80% to 100% confident in what would happen on-board, with the other half reporting at 60 - 79% confidence. The majority of the participants reported that they found that prediction clustering in the Outcome Prediction tool increased their confidence in what would happen onboard.

Participants' confidence in the design of the Outcome Prediction tool was demonstrated in their use of it to evaluate, discuss, and make decisions about different planning options. For example, the Mission Planner reported that they used Outcome Prediction to *"see if there were any obvious outliers or something that disqualifies this seemingly best-case scenario"*, and inspected lower-probability outcomes to see what additional science goals could be met.

Feedback on the design included multiple requests for visualizations that indicate whether environmental or variability inputs impacted the outcome prediction clusters. We also observed during uplink processes that operators spent an inordinate amount of time discussing low-probability outcomes, which we noted could have negative effects on the overall workflow.

In addition, it was noted that Outcome Prediction contributed significantly to downlink engineers' analysis of what happened on-board the spacecraft, in comparison to what was likely to happen. Across all downlink sessions, discussions focused on mismatches between predicted outcome and actuals, with operators correctly identifying that, in one orbit, there was a mismatch between the expected number of NAC imaging activities in the prediction compared to the actual observation window.

The Outcome Prediction tool proved highly valuable in helping the operations team understand the range of outcomes that could result from variations in the model or environment. Recommendations for future work include designing a way for users to identify whether environmental or variability inputs have greater impact on a specific outcome, and investigating the impact of operators' time spent on low probability outcomes.

Downlink findings: Plan Reconstruction

To test the performance of the Plan Reconstruction tool, we assessed how well the Plan Reconstruction tool helped operators correctly interpret what was autonomously detected, planned, and executed by the spacecraft, and whether they could determine what informed the outcome of the plan.

Based on the data we collected, we conclude that the Plan Reconstruction feature helped operators assess what was executed on-board the spacecraft, and identify the conditions that informed the on-board planner to generate the plan. Specifically, we observed operators using the tool to identify whether the spacecraft performed as expected. One engineer reported that *"The plan reconstruction made it very clear what was scheduled on-board and at what time. I was able to clearly confirm whether each of the instruments performed all of their tasks required to make detections and observations."* Additional design features that called operators' attention to incomplete tasks were also noted positively, with one operator reporting that *"Seeing the goals at the top of the plan reconstruction provided further confidence that the instruments behaved nominally. Seeing a warning indicator at the Plume B observation made it obvious that the NAC / SMS were not able to make the observations at Plume B. In being able to check that the tasks 'slew to Triton' and 'detect plumes' executed nominally, I was able to rule out that this issue was a result of failure during the previous subtasks."*

While the data pointed to the usefulness of the Plan Reconstruction feature in determining the actions of the spacecraft, we also noted that design simulation participants referred to their knowledge of the last uplinked model to contextualize the plan that was received. Due to the fact that participants in the design simulations acted as both uplink and downlink engineers, and therefore were aware of the current state of the task network model, we recommend that further research be done to evaluate the benefit of situational awareness of the model on the effectiveness of the playback feature, and on any additional state estimation features used to determine the state of the spacecraft.

Additionally, we recommend that further work be done to explore how the current design would scale to a more complex task network model with increased dependencies between goals and resources. Specifically, we are interested to learn which information will become most valuable to operators in their assessment of the on-board planner and autonomous detectors.

Downlink findings: State Estimation

To test the success of the proposed state estimation tools and interfaces, we observed and surveyed whether operators used them to accurately and confidently estimate the state of the spacecraft.

We found that, in cases where only partial data was available, that operators found state estimation to be valuable. One engineer reported that *"For the times where telemetry was not available, the uncertainty visualization helped me to understand the possible state of the spacecraft resource. Instead of being left completely in the dark or needing to make rough estimates on my own, the state estimation provided me with possible states for that resource usage based on the partial data that was able to be downlinked. This analysis provided me with confidence that the issue of the missing NAC image and SMS measurements was not likely to be due to on-board storage reaching capacity or battery usage being drained too quickly."*

However, in cases where data was not missing, we noted that availability of the estimation tools did not dramatically alter the operators' analysis process. The process relied on comparing predicts to raw spacecraft measurements to decide the next step in their analysis. For example, when viewing the estimated resource data, one downlink engineer asked to see what data was missing to inform where a problem might exist.

We note that operators' use of the state estimation features may have been impacted by insufficient exposure to the new tools prior to the design simulation. With just 30 minutes to review the latest telemetry and assess the state of their system, operators performed their analysis as efficiently as possible, possibly prioritizing tools and techniques that were familiar to them. We recommend that future user testing be done with additional training on the state estimation tools, to better reflect the operators' deep familiarity with their tools in actual operations.

To explore opportunities for increasing the usefulness of the state estimation features within downlink operators' primary analysis workflow, we also recommend additional user research and iterative design sessions with operators.

7. CONCLUSIONS

We study the problem of operations for autonomous spacecraft: that is, what workflows and tools will be necessary to enable operators of future autonomous missions to interact with on-board autonomy, with a focus on on-board planning and scheduling. We designed a set of user interfaces and algorithmic tools to address this challenge, and assessed their effectiveness through a design simulation with JPL operators and scientists. The design simulation largely confirmed that the proposed approach is effective in capturing uplink operators' intent and allowing downlink operators to understand the spacecraft's decisions and assess its state, effectively addressing the problem of operations of autonomous spacecraft. A number of recommendations for improvements to the tools were also gathered, and will be implemented in future work.

In addition, a number of directions for future research are of interest.

First, we will continue the implementation and maturation of the proposed tools, and their integration with existing frameworks used for mission operations such as AMPCS.

Second, we plan to assess the scalability of the proposed tools to larger datasets. A typical mission on the scale of Mars 2020 or Europa Clipper has tens of thousands of different data types that are used in downlink analysis, including tens of thousands of telemetry channels, thousands of predicted channels, tens of thousands of flight software parameters, and thousands to tens of thousands of EVRs. We plan to ensure that both user interfaces and algorithms are able to scale up to the needs of flagship-class missions.

Third, we will continue exploring the role of state estimation algorithms in the downlink analysis process. State estimation holds promise to help bridge the gap between unorganized EVRs and a full representation of the spacecraft state (or even a representation of spacecraft state from the perspective of the on-board planner); however, additional work is required to make sure that state estimation is provided to operators at

junctures when it is most useful.

Fourth, we plan to propose standards for onboard data types (e.g., the format of specific autonomy-related EVRs, or common representations of the spacecraft's state) to make it possible to design general-purpose ground analysis tools that can be used across missions, without requiring major adaptations for each subsequent project. This effort will likely include both collections of EVRs and dedicated autonomy engineering data products.

Finally, we plan to pursue infusion of the proposed tools in future autonomous technology demonstration missions (starting with integration and testing operations and operational readiness tests), in order to raise the technology readiness level of the proposed framework, identify additional use cases, and ensure that the proposed approach is fully able to support operators of future autonomous spacecraft.

ACKNOWLEDGMENTS

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

REFERENCES

[1] D. Bernard, G. Dorais, E. Gamble, B. Kanefsky, J. Kurien, G. Man, W. Millar, N. Muscettola, P. Nayak, K. Rajan, , N. Rouquette, B. Smith, W. Taylor, and Y.-W. Tung, "Spacecraft autonomy flight experience - the ds1 remote agent experiment," in *AIAA Space Technology Conference and Exposition*, 1999.

[2] D. Tran, S. Chien, R. Sherwood, R. Castano, B. Cichy, A. Davies, and G. Rabideau, "DEMO: the autonomous sciencecraft experiment onboard the EO-1 spacecraft," in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 163–164.

[3] M. Troesch, F. Mirza, K. Hughes, A. Rothstein-Dowden, R. Bocchino, A. Donner, M. Feather, B. Smith, L. Fesq, B. Barker *et al.*, "Mexec: An onboard integrated planning and execution approach for spacecraft commanding," in *Workshop on Integrated Execution (IntEx)/Goal Reasoning (GR), International Conference on Automated Planning and Scheduling (ICAPS IntEx/GR 2020)*, 2020.

[4] J. Agrawal, W. Chi, S. A. Chien, G. Rabideau, D. Gaines, and S. Kuhn, "Analyzing the effectiveness of rescheduling and flexible execution methods to address uncertainty in execution duration for a planetary rover," *Robotics and Autonomous Systems*, vol. 140 (2021) 103758, 2021.

[5] D. Gaines, S. Chien, G. Rabideau, S. Kuhn, V. Wong, A. Yelamanchili, S. Towey, J. Agrawal, W. Chi, A. Connell, E. Davis, and C. Lohr, "Onboard planning for the Mars 2020 Perseverance rover," in *16th Symposium on Advanced Space Technologies in Robotics and Automation*, June 2022.

[6] R. Francis, T. Estlin, G. Doran, S. Johnstone, D. Gaines, V. Verma, M. Burl, J. Frydenvang, S. Montaño, R. Wiens *et al.*, "AEGIS autonomous targeting for ChemCam on Mars Science Laboratory: Deployment and results of initial science team use," *Science Robotics*, vol. 2, no. 7, 2017.

[7] S. Bhaskaran, "Autonomous navigation for deep space missions," in *SpaceOps 2012*, 2012, p. 1267135.

[8] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, "Global planning on the Mars Exploration Rovers: Software integration and surface testing," *Journal of Field Robotics*, vol. 26, no. 4, pp. 337–357, 2009.

[9] A. Rankin, M. Maimone, J. Biesiadecki, N. Patel, D. Levine, and O. Toupet, "Mars Curiosity rover mobility trends during the first 7 years," *Journal of Field Robotics*, vol. 38, no. 5, pp. 759–800, 2021.

[10] R. Mackey, A. Nikora, C. Altenbuchner, R. Bocchino, M. Sievers, L. Fesq, K. O. Kolcio, M. J. Litke, and M. Prather, "On-board model based fault diagnosis for cubesat attitude control subsystem: Flight data results," in *2021 IEEE Aerospace Conference (50100)*. IEEE, 2021, pp. 1–17.

[11] D. D. Dvorak, M. D. Ingham, J. R. Morris, and J. Gersh, "Goal-based operations: an overview," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 3, pp. 123–141, 2009.

[12] T. Uhlig, F. Sellmaier, and M. Schmidhuber, *Spacecraft operations*. Springer, 2015.

[13] J. Barreiro, M. E. Boyce, M. B. Do, J. D. Frank, M. Iatauro, T. Kichkaylo, P. H. Morris, J. C. Ong, E. Remolina, T. B. Smith, and D. E. Smith, "EUROPA: A platform for AI planning, scheduling, constraint programming, and optimization," in *International Conference on Automated Planning and Scheduling (ICAPS) – International Competition on Knowledge Engineering for Planning and Scheduling*, 2012.

[14] G. Bernardi, A. Cesta, A. Finzi, and A. Orlandini, "A knowledge engineering environment for P&S with timelines," in *Workshop on Knowledge Engineering for Planning and Scheduling (KEPS), International Conference on Automated Planning and Scheduling (ICAPS)*, 2013.

[15] G. Verfaillie and C. Pralet, "A timeline, event, and constraint-based modeling framework for planning and scheduling problems," in *Workshop on Knowledge Engineering for Planning and Scheduling (KEPS), International Conference on Automated Planning and Scheduling (ICAPS)*, 2020.

[16] M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, J. Yglesias, B. Chafin, W. Dias, and P. Maldague, "MAPGEN: Mixed-initiative planning and scheduling for the Mars Exploration Rover mission," *IEEE Intelligent Systems*, vol. 19, pp. 8–12, 01 2004.

[17] G. Rabideau, V. Wong, D. Gaines, J. Agrawal, S. Chien, S. Kuhn, E. Fosse, and J. Biehl, "Onboard automated scheduling for the Mars 2020 rover," in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation for Space*, ser. i-SAIRAS'2020. Noordwijk, NL: European Space Agency, 2020.

[18] A. Yelamanchili, J. Agrawal, S. Chien, J. Biehl, A. Connell, U. Guduri, J. Hazelrig, I. Ip, K. Maxwell, K. Steadman, and S. Towey, "Ground-based automated scheduling for operations of the Mars 2020 rover mission," in *Proceedings Space Operations 2021*, May 2021.

[19] W. L. Quach, L. DeForrest, A. T. Klesh, and J. School-

craft, "Adapting a large-scale multi-mission ground system for low-cost cubesats," in *SpaceOps 2014 Conference*, 2014, p. 1634.

[20] J. Trimble and G. Rinker, "Open source next generation visualization software for interplanetary missions," in *14th International Conference on Space Operations*, 2016, p. 2348.

[21] A. Ko, P. Maldague, D. Lam, T. Bui, and J. McKinney, "The evolvable advanced multi-mission operations system (AMMOS): making systems interoperable," in *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, 2010, p. 2303.

[22] F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 21–29, May 2007.

[23] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. development team, "Jupyter notebooks - a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Scmidt, Eds. Netherlands: IOS Press, 2016, pp. 87–90.

[24] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, R. Lee, D. Mandl, S. Frye *et al.*, "The EO-1 autonomous science agent," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. Citeseer, 2004, pp. 420–427.

[25] J. Agrawal, A. Yelamanchili, and S. Chien, "Using explainable scheduling for the Mars 2020 rover mission," in *Workshop on Explainable AI Planning (XAIP), International Conference on Automated Planning and Scheduling (ICAPS XAIP)*, October 2020.

[26] R. Castano, T. Stegun Vaquero, F. Rossi, V. Verma, E. Van Wyk, D. Allard, B. Huffmann, E. M. Murphy, N. Dhamani, R. A. Hewitt, S. Davidoff, R. Amini, A. Barrett, J. Castillo-Rogez, M. Choukroun, A. Dadaian, R. Francis, B. Gorr, M. Hofstadter, M. Ingham, C. Sorice, and I. Tierney, "Operations for autonomous spacecraft," in *IEEE Aerospace Conference (AERO)*, Big Sky, MT, Mar. 2022.

[27] C. Blackwood, E. M. Murphy, M. Le, G. Pyrzak, S. Y. Kim, S. L. Laubach, G. Tan-Wang, and S. Davidoff, "Ecosystem prototyping at scale with design simulation," in *The ACM Conference on Designing Interactive Systems*, in preparation.

[28] C. A. Ortega, C. A. Polanskey, M. Llopis, P. Rosemurgy, C. R. Lawler, E. K. Alonge, M. Dailis, and L. T. Elkins-Tanton, "Psyche science planning with the Science Opportunity Analyzer," *International Conference on Space Operations*, 2021.

[29] E. W. Ferguson, S. S. Wissler, B. K. Bradley, P. Maldague, J. Ludwinski, and C. R. Lawler, "Improving spacecraft design and operability for Europa Clipper through high-fidelity, mission-level modeling and simulation," in *International Conference on Space Operations*, 2018, p. 2469.

[30] M. S. Feather, B. Kennedy, R. Mackey, M. Troesch, C. Altenbuchner, R. Bocchino, L. Fesq, R. Hughes, F. Mirza, A. Nikora *et al.*, "Demonstrations of system-level autonomy for spacecraft," in *IEEE Aerospace Conference (AERO)*. IEEE, 2021, pp. 1–18.

[31] B. Streiffert and T. O'Reilly, "The evolution of Seqgen-a spacecraft sequence simulator," in *SpaceOps 2008 Conference*, 2008, p. 3523.

[32] P. F. Maldague, S. S. Wissler, M. D. Lenda, and D. F. Finnerty, *APGEN Scheduling: 15 Years of Experience in Planning Automation*. SpaceOps Conference, 2014.

[33] I. Deliz, A. Connell, C. Joswig, J. J. Marquez, and B. Kanefsky, "COCPIT: Collaborative activity planning software for Mars Perseverance rover," in *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 01–13.

[34] L. Jones-Wilson and S. Susca, "A framework for extending the science traceability matrix: Application to the planned Europa mission," in *2017 IEEE Aerospace Conference*, 2017, pp. 1–14.

[35] L. Jones-Wilson, S. Susca, and K. Reinholtz, "Project-domain science traceability and alignment framework (P-STAF): Analysis of a payload architecture," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–16.

[36] M. Troesch, S. Chien, and E. Ferguson, "Using automated scheduling to assess coverage for Europa Clipper and Jupiter Icy Moons Explorer," in *International Workshop on Planning and Scheduling for Space (IWPSS 2017)*, 2017.

[37] W. Chi, J. Agrawal, S. Chien, E. Fosse, and U. Guduri, "Optimizing parameters for uncertain execution and rescheduling robustness," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, no. 1, pp. 501–509, May 2021.

[38] K. J. McCoy, M. DiNicola, C. Everline, H. Burgoyne, K. Reinholtz, and B. Clement, "Europa Clipper planetary protection probabilistic risk assessment summary," *Planetary and Space Science*, vol. 196, p. 105139, 2021.

[39] E. W. Ferguson, S. S. Wissler, B. K. Bradley, P. F. Maldague, J. M. Ludwinski, and C. R. Lawler, "The power of high-fidelity, mission-level modeling and simulation to influence spacecraft design and operability for Europa Clipper," in *Space Operations: Inspiring Humankind's Future*. Springer, 2019, pp. 195–231.

[40] Y. Ding, S. Duggan, M. Ferranti, M. Jagadpramana, R. Rege, Y. Zhovtobryukh, and M.-E. Paté-Cornell, "Probabilistic assessment of the failure risk of the Europa Clipper spacecraft due to radiations," *Risk Analysis*, vol. 40, no. 4, pp. 842–857, 2020.

[41] A. Yelamanchili, G. Rabideau, J. Agrawal, V. Wong, D. Gaines, S. Chien, E. Fosse, J. Biehl, S. Kuhn, A. Connell, J. Hazlerig, I. Ip, U. Guduri, K. Maxwell, K. Steadman, and S. Towey, "Ground and onboard automated scheduling for the Mars 2020 rover mission," in *International Workshop on Planning & Scheduling for Space (IWPSS)*, July 2021.

[42] S. Raychaudhuri, "Introduction to monte carlo simulation," in *2008 Winter Simulation Conference*, 2008, pp. 91–100.

[43] "Production-grade container orchestration." [Online]. Available: https://kubernetes.io/

[44] PostgreSQL Global Development Group, "Postgresql." [Online]. Available: https://www.postgresql.org/

[45] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[46] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep. GT-RIM-CP&R-2012-002, 2012. [Online]. Available: http://hdl.handle.net/1853/45226

[47] M. Hsiao and M. Kaess, "MH-iSAM2: Multi-hypothesis iSAM using Bayes Tree and Hypo-tree," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1274–1280.

[48] "Caesar.jl, v0.10.2," 2021. [Online]. Available: https://github.com/JuliaRobotics/Caesar.jl

[49] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[50] Y. Bengio and P. Frasconi, "An input output hmm architecture," *Advances in neural information processing systems*, vol. 7, 1994.

[51] ——, "Input-output hmms for sequence processing," *IEEE Transactions on Neural Networks*, vol. 7, no. 5, pp. 1231–1249, 1996.

[52] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[53] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[54] J. F. Kelley, "An iterative design methodology for user-friendly natural language office information applications," *ACM Transactions on Information Systems (TOIS)*, vol. 2, no. 1, pp. 26–41, 1984.

BIOGRAPHY

Federico Rossi is a Robotics Technologist at the Jet Propulsion Laboratory, California Institute of Technology. He earned a Ph.D. in Aeronautics and Astronautics from Stanford University in 2018. His research focuses on optimal control and distributed decision-making in multi-robot systems, with applications to planetary exploration and coordination of fleets of self-driving vehicles for autonomous mobility-on-demand.

Tiago Stegun Vaquero is a technical group lead in the Artificial Intelligence, Integrated Planning and Execution Group at the Jet Propulsion Laboratory, California Institute of Technology. He holds a B.Sc., M.Sc., and Ph.D. in Mechatronics Engineering from the University of Sao Paulo, Brazil. Tiago previously held a MIT research scientist position and a joint Caltech/MIT research position where he worked on Resilient Spacecraft Systems and Risk Sensitive Planning/Scheduling algorithms. At MIT, Tiago also worked on Risk-aware Planners and Executives for Autonomous Underwater Vehicles and Autonomous Cars. Tiago also held a research position at the University of Toronto where he worked on Multi-Robot Planning and Coordination. His research interests include knowledge engineering for autonomous vehicles, and automated planning and scheduling for single and multi-robot missions.

Marijke Jorritsma is a User Experience Designer at the NASA Jet Propulsion Laboratory, where she designs operations software for robotic space exploration. Her work includes designing a constraint-based automated scheduler for the Europa Clipper Mission, as well as work on other tools including a 3D targeting tool for Mars2020 (ASTTRO), an augmented reality tool for CAD model visualization (Protospace), and a science planning visualization tool (SOA). She holds a M.S. in Integrated Digital Media from NYU's Tandon School of Engineering, and a BFA in Film from the San Francisco Art Institute.

Ellen Van Wyk has a Master's in Information Management and Systems from the University of California, Berkeley, and a Bachelor's in Neurobiology from the University of Washington. She has researched science and engineering practices and designed new approaches and technologies for improving their workflows for the Jet Propulsion Laboratory since 2017.

Bennett Huffman graduated with a B.S. in Information Systems and Human-Computer Interaction in 2022 from Carnegie Mellon University. He is a User Interface Developer at the NASA Jet Propulsion Laboratory, where he focuses on front-end tooling, data visualization, and development.

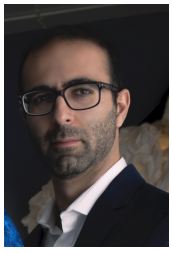
Dan Allard has a Bachelor's in Engineering Physics from Tufts University, Medford MA. He is currently a software development manager of ground software for Europa Clipper and a system engineer for the Mars network relay planning system. He has worked as a ground software developer and manager for missions including Cassini, MSL, SMAP, and Mars 2020.

Nihal Dhamani graduated with a B.S. in Computer Science and B.S.A in Astronomy in 2019 from the University of Texas at Austin. He is currently a Data Scientist in the Artificial Intelligence Group at NASA's Jet Propulsion Laboratory, where he works on various projects related to automated planning and scheduling.

Scott Davidoff is the Manager of the Human-Centered Design Group, and the Data to Discovery Program at JPL. He is also the Data Visualization Lead for the PIXL Instrument on on the Perseverance Rover. His research in Human-Computer Interaction focuses on how to help scientists and engineers understand complex data, and how to figure out what to build. Dr. Davidoff has a Ph.D. in Human-Computer Interaction, and MS degrees in Computer Science and Human-Computer Interaction, all from Carnegie Mellon University.

Ashkan Jasour is a Robotics Technologist at NASA's Jet Propulsion Laboratory (JPL), California Institute of Technology. Before joining JPL, he was a research scientist at the Massachusetts Institute of Technology (MIT). His research focuses on risk-aware planning, control, and optimization under uncertainty for robotic and autonomous systems, with an emphasis on non-Gaussian nonlinear problems. In 2016, he received his PhD in Control Systems/Electrical Engineering and PhD minor in Mathematics from the Pennsylvania State University. He was also a Postdoctoral Associate at MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL).

Anthony Barrett is a member of the Mission Control Systems & Deep Learning Technologies Group at the Jet Propulsion Laboratory, California Institute of Technology. He earned a Ph.D. in Computer Science from the University of Washington in 1997. His research focuses on diagnosis, planning, scheduling, and knowledge compilation applied to spacecraft autonomy. He was also a tactical uplink lead for the MER rovers.



Rashied Amini is a systems engineer at Jet Propulsion Laboratory, California Institute of Technology working in mission formulation and in research of autonomous technologies. He is currently a concept lead for astrophysics and planetary missions and was the Habitable Exoplanet Report Manager and Galaxy Evolution Probe Study Lead submitted to the 2020 Astrophysics Decadal Survey. As a result of his formulation work, he is interested in supporting the maturation of technologies critical to science exploration. He received a Ph.D. in Physics from Washington University in St. Louis.

Mathieu Choukroun received a Ph.D. in Earth and Planetary Science from Université de Nantes, France in 2007. He is currently is a planetary scientist at the Jet Propulsion Laboratory, California Institute of Technology. His primary research aims at better understanding the exchange processes that take place between the interior and the surface (or atmosphere/coma) of icy worlds and comets. This research involves experimental investigation of the physical and chemical properties of icy materials, and thermodynamic and geophysical modeling of icy worlds and cometary environments to apply the experimental results.

Raymond Francis holds a B.A.Sc. in Mechanical Engineering from the University of Ottawa (2006), an M.Sc. in Physics (Space Science) from the Royal Military College of Canada (2010) and the Ph.D. in Electrical and Computer Engineering from the University of Western Ontario (2014). He is currently a Science Operations Engineer at NASA Jet Propulsion Laboratory, where he is Science Operations Deputy Team Chief for NASA's Perseverance Mars rover, and a member of the SuperCam instrument operations team. He has worked on Curiosity rover operations since 2012, including as a member of the ChemCam instrument operations team. He was lead system engineer for the deployment of the AEGIS autonomous targeting software on Curiosity, and Science Team Training System Engineer for the Mars 2020 project. Outside of operations, his current work focuses on the development of techniques for autonomous science and the deployment of autonomous capabilities into operational exploration missions.

Mark Hofstadter is a scientist at the Jet Propulsion Laboratory, California Institute of Technology. He holds a Ph.D. in Planetary Science from Caltech. He uses both space- and ground-based radio telescopes to study our solar system, with an emphasis on giant planets. He was the PI of the MIRO instrument which flew on the Rosetta spacecraft to a comet, and was instrument scientist on the Earth-observing AIRS/VisNIR instrument.

Michel Ingham is the Chief Technologist of JPL's Systems Engineering Division. Since he joined JPL in 2003, he has worked as a software systems engineer and system architect on a variety of projects, including the Mars Science Laboratory rover, and the Europa Clipper mission. He has led several NASA, JPL and DARPA research and development activities in the areas of model-based systems and software engineering, software architectures, and spacecraft autonomy. Dr. Ingham received his Sc.D. and S.M. from MIT's Department of Aeronautics and Astronautics, and his B.Eng in Honours Mechanical Engineering from McGill University in Montreal, Canada.

Vandi Verma is the Principal Investigator for the Autonomy for Operations, Spacecraft State Estimation, Research and Technology Development task. She also serves as the Chief Engineer of Robotic Operations for M2020 Perseverance and as the Deputy Section Manager for Mobility and Robotics Systems at JPL. She has additional roles on Perseverance and works on robotics and autonomy flight software development, Rover Planning, and the Ingenuity helicopter technology demonstration. She holds a Ph.D. in Robotics from Carnegie Mellon University and specializes in space robotics, autonomous robots and robotic operations. She has worked on the Mars Exploration Rovers, Curiosity rover, Europa Clipper Autonomy Prototype, Europa Lander, and autonomous research robots in the Arctic, Antarctica and Atacama.

Rebecca Castano is the Directorate Technologist for the Interplanetary Network Directorate at JPL. In this role, she works on ensuring that relevant new technologies are developed, matured, validated, and infused. She is also the JPL Program Manager for the Center Innovation Fund (CIF). Prior to her directorate position, Dr. Castano was the Division Technologist for Division 39, the Mission Systems and Operations Division. She earned a Ph.D. in Electrical Engineering from the University of Illinois with her dissertation in the area of computer vision. She has contributed to autonomy software flying on Earth orbiters and Mars rovers.