A Knowledge Engineering Framework for Mission Operations of Increasingly Autonomous Spacecraft

Tiago S. Vaquero, Federico Rossi, Rebecca Castano, Ashkan Jasour, Ellen van Wyk, Nihal Dhamani, Anthony Barrett, Bennett Huffman, Marijke Jorritsma

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive,
Pasadena, California 91109

Abstract

As planning and autonomy in general become increasingly deployed onboard spacecraft, missions will face a paradigm shift in how ground operations teams command and interact with the spacecraft: moving from specifying timed sequences of commands to high level goals that on-board autonomy will elaborate based on the spacecraft's state and sensed environment. In this paper we present an ongoing effort to develop an integrated framework for supporting ground operations through modeling science and engineering intent/goals, predicting outcomes, assessing spacecraft state and performance, and maintaining models used for on-board decision-making and ground-based monitoring. Specifically, we describe the specific knowledge engineering aspects that are key in the operations of autonomous spacecraft, and how we propose to addressed the challenges posed by operations of on-board autonomy.

Introduction

Future space exploration missions will have increasingly advanced onboard autonomy capabilities to increase science return, improve spacecraft reliability, reduce operations costs, or even achieve goals that cannot be attained through a regular ground-in-the-loop operations cycle due to communication constraints or limited lifetime. Examples of autonomy capabilities being developed for future mission include autonomous planning, scheduling and execution (e.g., (Chi et al. 2021; Troesch et al. 2020)), autonomous selection of scientific targets (e.g., (Francis et al. 2017)), autonomous fault management (e.g., (Hwang et al. 2009; Kolcio, Fesq, and Mackey 2017)) and onboard data summarization and compression (e.g., (Doran et al. 2020)). Autonomy has already significantly increased the capabilities of Mars rover missions, enabling them to perform tasks such as autonomous long-distance navigation and autonomous data collection of new science targets (Estlin et al. 2012). Automated ground-based planning and scheduling, in particular, has been deployed on daily ground operations for the Perseverance rover (Yelamanchili et al. 2021a) and is projected to be deployed onboard in the near future (Rabideau et al. 2020).

As planning and autonomy are increasingly deployed onboard spacecraft, missions will face a paradigm shift in how ground operations teams command and interact with the spacecraft: specifically, operations will move from uploading a series of timed command sequences to specifying intent in the form of high level goals (e.g. a set of high level activities whose expansion in a constrained fashion achieves the humans' intent). This brings to the fore the well recognized challenges related to knowledge engineering (KE) for planning and scheduling systems, that is, the process of eliciting, representing and maintaining not only goals, requirements, and activities and task models that will serve as the main input for onboard planners, but also spacecraft models for state estimation and performance evaluation. As described in many efforts in the KE area (see survey in (McCluskey, Vaquero, and Vallati 2017)), the process of capturing and representing goals and models is quite challenging and requires a careful iterative design process, along with considerable tooling to support operators, scientists and engineers in i) expressing and refining their intents, ii) developing a shared understanding of algorithm behavior between humans and the onboard planning systems, and iii) increasing trust in autonomy.

The KE challenges are amplified for spacecraft that need to be operated further into the Solar System, such as missions to the Ice Giants. Challenges of operating faraway spacecraft with long light-time distance from Earth are exacerbated when the mission has limited communications bandwidth, short-duration science opportunities, and large uncertainty related to the environment and target science observations. As such, if mission goals, activity/behaviors, or spacecraft models are not well specified, critical science opportunities and observations may be lost, potentially jeopardizing the achievement of primary mission objectives.

Knowledge engineering research efforts in space exploration applications can been found in the literature (Barreiro et al. 2012; Bernardi et al. 2013; Verfaillie and Pralet 2020; Ai-Chang et al. 2004). However, most of the existing work has focused on the traditional paradigm of generating plans on the ground and uploading conservatively timed sequences of commands (with significant margins to cope with uncertainty) to the spacecraft. 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,

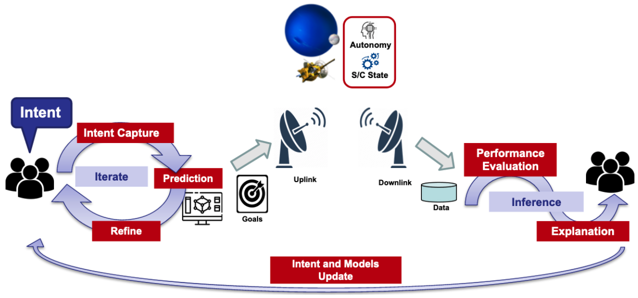

Figure 1: Overview of the mission operation planning workflow (uplink processes to the left and downlink processes to the right) highlighting key knowledge engineering processes.

i.e. spanning operations from capturing and specifying goals and plans to be uploaded to the spacecraft (uplink) to receiving and analyzing telemetry data in order to infer the state of the spacecraft and understand what executed, and what was observed (downlink).

In this work, we present an ongoing effort to develop a novel integrated framework for mission operations planning for highly autonomous spacecraft designed to facilitate the complex process of specifying and refining goals and evaluating spacecraft performance, while increasing trust in the onboard autonomy. Herein we describe the current design of the framework, focusing on key processes (illustrated in Figure 1) and tools for knowledge engineering including: *intent capture*, where goals and key performance indicators are elicited from scientists, engineers and operators; *outcome prediction*, where the team explores different scenarios and uncertainty conditions to understand possible execution paths, risks, and science trade-offs; *performance evaluation*, where the scientists and engineers analyze and understand the data from the spacecraft to estimate the state (through inference) and evaluate the autonomy performance, comparing it with prediction data; and *intent and model update* where goals and models (e.g. science phenomena, activities, spacecraft components) are modified or refined according to data from the spacecraft and performance analysis.

Our development is driven by a simulated Ice Giant tour mission to the Neptune system with an autonomous spacecraft, which helps shape the framework's requirements. Although our framework has been designed to accommodate different onboard planning and scheduling technologies, we have targeted a flight-proven planning and execution system, namely MEXEC (Troesch et al. 2020) - a system that has been demonstrated on the ASTERIA CubeSat and was used on the JPL's Europa Lander Surface Mission Autonomy project (Wang et al. 2022). Also, MEXEC shares core reasoning components (e.g. the timeline library) with the planning system used in the Perseverance rover's operations.

In what follows, we describe the vision for the framework with respect to the aforementioned key KE processes and provide an overview of the tools we designed.

Intent Capture

The effort to formulate science goals for a mission begins early in development. Strategic planning is used to determine different science investigations and how time should be allocated to them throughout the mission, based on anticipated science observation opportunities. Tactical planning, on the other hand, is conducted on a shorter time horizon. During operations, both strategic and tactical planning processes constantly capture and update goals in response to new downlinked data, each at a different cadence.

In this paper, goals are captured and organized in the form of science campaigns. Capturing and organizing goals as campaigns have been done, for example, in the work on the ASPEN-RSS scheduler for the Rosetta Orbiter (Chien et al. 2021). In this work, we use the term "campaign" to refer to *a coordinated set of observations that address a particular set of science objectives*. Herein, a *campaign* is defined by a set of goals (a desired set of high level activity, e.g. "survey the magnetosphere", or "monitor for plume activity"), key performance indicators (KPIs) and their valid range for assessment of execution (e.g. resource usage ranges, frequency of a command cycling due to delays), and relationships between goals (e.g. in the form of priorities). Relationships between goals are a critical element to be captured - they are not typically explicitly captured on missions, but rather come to light through the process of team discussions and negotiations.

In both strategic and tactical planning, campaigns and goals need to be considered from different perspectives, including those of scientists, instrument experts, engineers, and operators. Teams have to communicate and negotiate priorities, since resources (e.g., time, power, and downlink capacity) are generally insufficient to support all desired goals, and instrument utilization can be conflicting among different goals. The human factor inherent in the capture and prioritization process introduces challenges, in the sense that humans may not necessarily be able to formally articulate what they want achieved, or fully understand what trade-offs are involved. Hence, defining processes and tools to support campaigns/goals capture, representation, prioritization and refinement is critical.

In our framework, we focus on these different perspectives by providing different integrated tools for users to input goals, each targeting a specific group of users to better connect with their vocabulary, terms, focus, and needs. Specifically, we provide i) a tool for scientists and instruments teams, namely the Science Intent/Planning tool, and ii) tools designed for mission planners, engineers and autonomy experts (e.g. automated planning expert), namely the Task/Goal Network tool and the Prediction Outcome tool.

Science Planning

Science and instrument teams' intent is usually focused on science observation opportunities and related constraints (e.g., "monitor for plumes in Triton with a wide angle camera and, if detected, take follow-on observations of the limb with a narrow angle camera"). Capturing such observation- and opportunity-centric goals is considered in our tool design, illustrated in Figure 2. Scientists can specify targeted

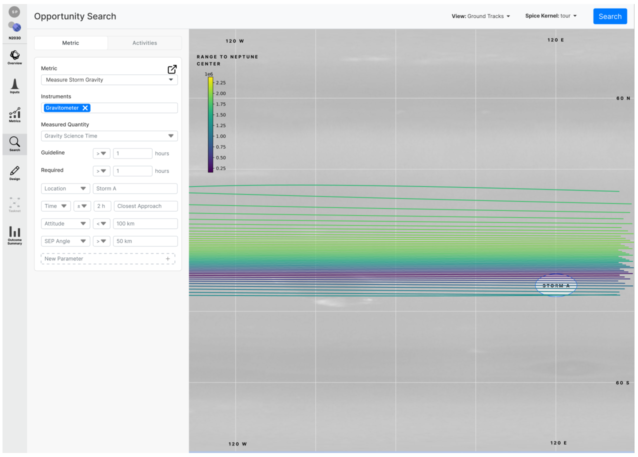

Figure 2: Science Intent/Planning tool for capturing intent from scientist and instruments teams while allowing them to explore and query observation opportunities.

observations with specific parameters, and explore the domain of possible opportunities with respect to specified constraints such as geometric, pointing, and resource constraints. One of the key elements in this opportunity exploration is to visualize the options not only with respect to geometric constraints, but also with respect to the impact against mission requirements and other key performance indicators (described later in this paper). Figure 2 shows the option of searching for Neptune gravity measurement opportunities based on observation activities or performance indicators (also called metrics). The nadir pointing options are shown on the right side of the figure on a single flyby. As users explore the opportunities, the tool helps estimate the impact of that opportunity on the remaining of the mission, analyzing whether it would support the mission goals, or whether it would impede desired progress. Once satisfactory opportunities are identified (noting that the specification and search for opportunities is already an iterative process), the goal is added to the mission goals (e.g. for the next space flyby/orbit).

The exploration of opportunities provides a good foundation for negotiation and prioritization when conflicts exist. In particular, it supports the analysis of how unique an opportunity might be, impacting the relative priority of the observation. This concept draws from existing JPL work on the Science Opportunity Analyzer (SOA) tool (Streiffert and Polanskey 2004). Our framework incorporates similar functionality, while also extending the design to allow data to be shared across the other tools in the framework, such as the Task/Goal Network tool described in the next section.

Task/Goal Network

In this work, campaign and goals are ultimately represented in a *Task Network* (which is also called *Goal Network*). This particular representation is the foundation of timeline-based temporal planning and Hierarchical Task Network (HTN) planning. While the framework is general, our current implementation uses MEXEC (Troesch et al. 2020) as the core planning and execution system onboard the spacecraft. Therefore, capturing goals follows a task network (TN) formulation, meaning that goals are expressed in the form of tasks, including their pre-, post- and maintenance conditions, impact/effect constraints, temporal and resource constraints, priority, as well as ordering constraints and how the tasks decompose into sub-tasks hierarchically, as described in (Troesch et al. 2020). Figure 3 shows a screenshot of the Task/Goal Network visual editing tool. The vision for this tool is to embrace a more broad representation of goals, including compact and expressive goal representation constructs from the work on ASPEN-RSSC scheduler for the Rosetta Orbiter (Chien et al. 2021), as well as goal representation in the form of state constraints, e.g. towards the goal network concept defined in (Shivashankar et al. 2013).

In the uplink process, the Task Network tool is meant to be used by operators, mission planners, engineers and autonomy experts to represent their intent as goals. The goals provided through the Science Planning tool are added as tasks in the TN representation managed by the Task Network tool, 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. We leverage principles and lessons learned from existing KE tools in the planning community such as ASPEN-RSS (Chien et al. 2021) for capturing intent/campaigns and constraints with a compact representation language, MapGen (Ai-Chang et al. 2004) for representing constraints graphically in space applications, and itSIMPLE (Vaquero et al. 2007) and GIPO (Simpson, Kitchin, and McCluskey 2007) for providing a workflow for inputting goals, validating the model during modeling, representing action constraints and domain variables intuitively, and integrating with planners to validate the goals.

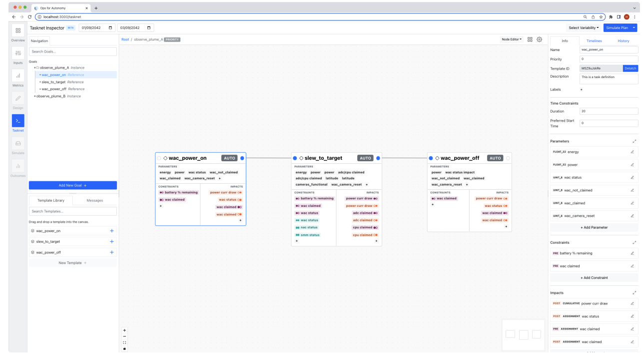

Figure 3: The Task/Goal Network tool supports the modeling of goals in the form of high level (hierarchical) tasks.

Key Performance Indicators

An important process in mission conceptualization and design is the specification of science objectives, and how their achievement can be measured, i.e., which methods and measurements can be used to answer the science questions. From an engineering perspective, we also want to be able to identify and measure performance requirements of the spacecraft and the autonomy throughout the mission. In this work we capture this information by defining *key performance indicators* (KPI), also called here *metrics*, from both science and engineering perspectives.

Studying the capture and specification process of metrics has not historically been part of mainstream of KE research for planning and scheduling applications. Traditionally, science objectives and requirements are captured in a Science Traceability Matrix (e.g. in the form of tables and spreadsheets) and tracked manually, semi-automatically or implicitly through the mission. Some existing efforts make the capture and tracking of metrics more explicit. For example, a mapping coverage requirement has been extensively applied, measured and embedded into the scheduling process in several applications of the CLASP scheduler such as in Mars Odyssey (Rabideau et al. 2010), Deformation, Ecosystem Structure, and Dynamics of Ice (DESDynI) (Knight, McLaren, and Hu 2012), NASA ISRO Synthetic Aperture Radar (NISAR) (Doubleday 2016), Intelligent Payload EXperiment (IPEX) CubeSat (Chien et al. 2016), and ECOSTRESS (Yelamanchili et al. 2021b). The work on ASPEN-RSS (Chien et al. 2021) also explicitly captured and tracked KPIs, as well as displayed them to users. The VERITaS tool (Buffington et al. 2017; McCoy et al. 2018) is another example of measuring progress towards the metrics captured in the Measurement-domain Science Traceability and Alignment Framework (M-STAF) (Susca, Jones-Wilson, and Oaida 2017), for the Europa Clipper mission. For example, a mapping coverage requirement of the Europa moon can be tracked by computing the percentage of the surface that is mapped during the mission.

In our work, capturing metrics is key to the uplink and downlink processes, in the sense that it supports the evaluation and quantification of the autonomy performance not only with respect to what has occurred, but to also drive projections and trade-off analyses towards what the spacecraft should and will do (goals) in the future. As we capture goals, we estimate the impact of those goals against the metrics to support goal refinement and prioritization. This can enable us to estimate the current progress of a mission at any point in time (past or future), assuming we have reliable models for projection.

The basic principle we propose here is to map specific goals defined in the campaigns (through the Task Networks and Science Planning tools) to a measurement method. This mapping can be done from a science perspective or an engineering perspective. In the Metric Specification tool, the measurement method is captured by specifying a mechanism to quantify progress against the goal, along with a target/required success criteria. For example, one can specify the measuring quantity as the the number of hours of a particular observation, and a desired/required number of hours as the success criteria, or the number of instances of a particular activity as the measured quantity. We provide a number of predefined templates for metric measurement to facilitate the definition process. For those measurements that no template is available (e.g. those that require analyzing images and specific science data), we provide an interface to a user-

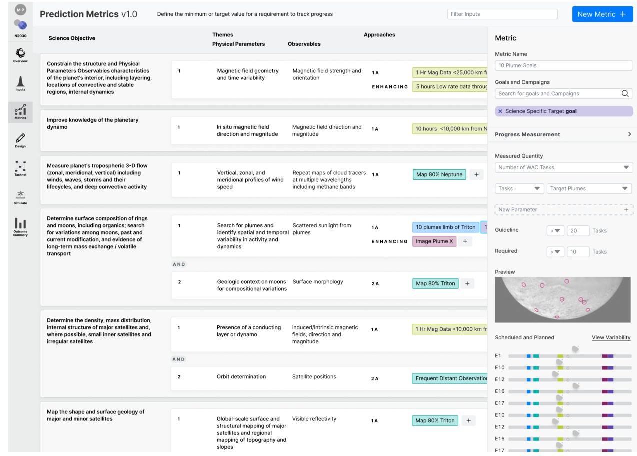

Figure 4: The Metric Specification tool captures key performance indicator and methods to measure them during mission with respect to science objectives and campaigns.

provided script to run the measurement and progress analysis. The interface for these scripts requires them to return the current measured value and progress against the target metric.

Figure 4 (right-hand side) shows the current implementation of the Metric Specification tool for capturing KPIs from a scientist and instrument expert perspective. For each goal, the user can specify the measurement method as described above, along with a representative picture that is optionally generated to visualize progress (for example a mapping campaign would track observation coverage). Figure 4 also shows that goal measurements are actually mapped to science objectives/questions and the different ways we might be able to answer these questions (left-hand side), i.e., a science question might be answered by different groups of goals represented in the task network. By capturing how metrics relate to science questions and objectives, we are able to track and estimate not only progress against campaigns, but also progress in addressing the science questions/objectives. This approach creates a foundation enabling operators, scientist, and engineers to assess mission progress and perform trade-off analysis from different perspectives at any point during a mission. The KPI/metric capture process is key in supporting the analysis about whether a set of specified goals will achieve the expected outcome.

Outcome Prediction

As we deploy more autonomous systems in environments with large uncertainties, we also need to build user trust in the decision making of the onboard planner. While it is not possible to anticipate all potential scenarios that the spacecraft will encounter, the uncertainty related to the environment (e.g. likelihood that a plume will be active a certain latitude and longitude) and the spacecraft itself (e.g. the likelihood of components failures, variations in the duration of on-board activities, etc) can be modelled on the ground.

In this work, we define a process for outcome prediction that can be performed both as goals are captured, and once

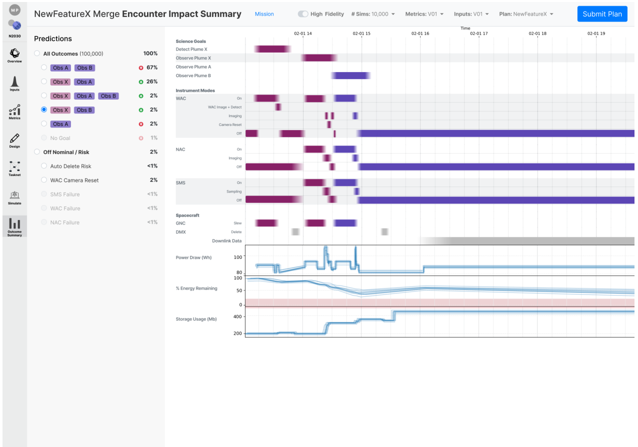

Figure 5: The Outcome Prediction tool supports the analysis of Monte Carlo simulation results by showing the distribution of outcomes with respect to goals achievement and off-nominal scenarios.

the set of goals that will be uploaded has been determined. In this framework, we first capture variability models, i.e., users specify science and engineering parameters that can vary, and a model for that variability. For example, this can entail modeling activity duration uncertainty as a Pert or Gaussian distribution (existing work with the Perseverance rover has looked into this specific aspect (Chi et al. 2021)), or modeling activity effects probabilistically (probabilistic activity models are not explicitly represented in the Task Network yet, but rather captured outside the task network representation), or modeling off-nominal behavior of instruments and components, or modeling uncertainty of science phenomena models.

Once variability information is captured, we use a Monte Carlo (MC) simulation approach for exploring different scenarios and conditions and investigating i) possible outcomes within the domain of user-specified variability and ii) the respective impact on mission progress as reflected in the KPIs/metrics. The MC simulation requires a model of both the spacecraft and environment - this is what the performance of the automated planning and scheduling systems and the goals are tested against. The prediction approach integrates both the MC system and the simulator, and stores data from each run into a database for compiling the outcome prediction results.

Figure 5 shows the design of the outcome prediction tool in our framework. A team can evaluate how likely the goals will be achieved, what is the impact on the mission, and how that translates to progress towards the campaigns and science objectives (by applying the KPIs/metrics). The left panel illustrates the distribution of outcomes with respect to goals and failures, while the right panel (inspired by the work from (Alper Ramaswamy et al. 2019)) layers the possible schedules and resources profiles into plots. Users can inspect each subset of outcomes to analyze possible performance. This automated explanation has great potential on making the inspection and analysis efficient. Navigating through the possible outcomes individually is not trivial. This method of presenting summary outcomes also provides users with insight into trends and classes of behavior. Future work will include a supporting an explanation process that helps users to understand the reasons why the onboard planner made certain decisions, which is especially helpful for unexpected or undesirable scenarios. The *Crosscheck* system developed for the Perseverance rover (Yelamanchili et al. 2021a) is an example of such an explanation tool. Moreover, the graphical user interfaces developed for the ASPEN-RSS scheduler (Chien et al. 2021) on the Rosetta Orbiter mission also has explanation features, in this case providing feedback on which constraints are preventing an observation from being scheduled.

As teams inspect and analyze the outcomes of the MC simulations, they are able to go back to the goals in the task network and refine them as needed. The prediction process can restart at any point, facilitating an iterative process of goal specification and prediction. This process is inspired by work being done for the Perseverance rover and Europa Lander simulations, which focus largely on nominal case simulations. Here we expand that concept to incorporate not just nominal cases, but also off-nominal scenarios, as well as a larger set of variability elements (both from science and engineering).

Performance Evaluation and Model Updates

Once a plan is formed, uplinked, and executed onboard the spacecraft, ground operators use downlinked information and ground tools to assess the spacecraft's state and the autonomy decisions in order to address three key questions:

- What decisions were made by autonomy?
- Why did autonomy take these decisions?
- What is the state of the spacecraft and of its environment?

Addressing these questions requires operators to compare the spacecraft's behavior with the models used by autonomy (and potentially identify areas for improvements in the models); in addition, supplemental models can be instrumental in supporting state estimation, and help form an understanding of the spacecraft's decisions by reconciling measurements taken on board the spacecraft and providing insight into states that cannot be directly measured.

State Estimation Estimating the state of the spacecraft and of its environment is critical both to assess the vehicle's health and resource availability, and to validate the models used by onboard autonomy. We have developed a family of techniques, made available to users as part the proposed framework, to estimate and track the spacecraft state and explain individual autonomy decisions.

Factor graph-based modeling A first set of tools relies on representing the spacecraft and its environment as a factor graph (Dellaert, Kaess et al. 2017; Dellaert 2012). Intuitively, factor graph models capture probabilistic relationships among state variables and between states and observations; nonlinear optimization tools are used to provide the maximum-a-posteriori (MAP) estimate of the most likely system states, e.g., the set of state variables that best explain the spacecraft's observations. While factor graphs are typically used to estimate continuous variables, state-of-the-art techniques can also capture discrete variables (Hsiao and Kaess 2019; Fourie, Teixeira, and Leonard 2019) by adding multi-modal factors. In such multi-modal models, discrete variables can be represented as separate hypotheses, e.g., one of the sensors is functioning normally or has experienced a fault.

Factor graph-based modeling is a highly effective tool for capturing nonlinear but *sparse* relations between states, where each state only affects a subset of the other states and observations. The key advantage of such modeling techniques is the ability to capture arbitrary relationships between state variables, which makes them highly suited to represent complex spacecraft models; however, the price of such flexibility is significant computational complexity.

Hidden Markov Model-based modeling A second family of tools relies on representing specific subsystems of the spacecraft (including, crucially, onboard autonomy functions such as event detection) as a hidden Markov model (HMM) (Rabiner 1989). The HMM formulation allows enumeration of the states that the spacecraft and its environment could be in, and captures the likelihood of the spacecraft transitioning between states. The Viterbi algorithm (Viterbi 1967; Forney 1973) can be used to identify the MAP estimate of the hidden states very efficiently, and the forward-backward algorithm (Dempster, Laird, and Rubin 1977) can be used to assess the marginal probabilities of individual state variables in polynomial time.

HMM-based modeling is well-suited to represent discrete spacecraft and environment states - for example, the presence or absence of a phenomenon of interest, and the likelihood of observing it based on the spacecraft's position and the sensors' capabilities.

Maintaining models consistency A key challenge in using multiple model classes (e.g., factor graphs and HMMs) for state estimation is to maintain consistency across the models, despite the models' different expressive capabilities and different degrees of discretization. This is an especially relevant concern as models evolve with the spacecraft and its environment, since the spacecraft's behavior can change with time, and the environment is often poorly-understood at first. Automated techniques to ensure model consistency, such as auto-generation of models from a main model repository, often run into insurmountable challenges due to the large differences in the models' expressive capabilities; for example, simply translating from the continuous states used in factor graphs to the discrete states employed by HMMs is nontrivial, and converting a factor in a set of transition probabilities can also prove challenging to do in an automated manner. To overcome this, our approach is to enforce mandated *processes* for the update of models used in state estimation. Specifically, we have established a "modeling consistency lead" engineering role; whenever a change to one of the models is desired, the change is vetted and approved by the modeling consistency lead, who, in turn, ensures that the owners of other models describing the same subsystem update their models accordingly.

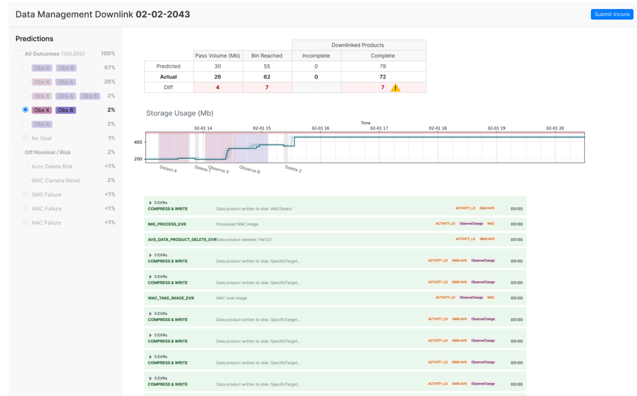

Figure 6: The "Predicts vs. Actuals" tool allows operators to filter predictions and compare them with estimates and raw measurements of the onboard state, supporting the identification of modeling inconsistencies.

Comparing estimates with on-board and ground-based prediction models The ability to provide high-quality estimates of the spacecraft's state, with quantified uncertainty, is also critical to identifying inaccuracies in the models used for onboard planning and for outcome prediction on the ground. In order to allow operators to rapidly spot inconsistencies in the models (especially the task network), we are developing software tools (shown in Figure 6) to compare "predicts vs. actuals", i.e., to juxtapose model prediction with estimates of the onboard state and with raw measurements. Outcome predictions are widely used in the uplink process, as discussed in previous sections; however, the goal of these predictions is to characterize the autonomy's decisions, and their impact on the spacecraft state, across a broad range of events that the spacecraft could encounter. In contrast, when comparing predictions with a posteriori estimates, only predictions that match the events actually encountered by the spacecraft are relevant (for example, if a scientific event of interest is detected on board, MC simulations performed in the uplink process that resulted in no event being detected can be ignored as uninformative; if the on-board autonomy decided to perform follow-on observations, simulations that featured no such observations can be similarly filtered out). Accordingly, this user interface provides the operators with tools to *filter* predictions based on the presence of events of interest and on the autonomy's decisions, ensuring that only relevant predictions are compared with the spacecraft's estimated state.

Conclusion and Future Work

We presented a software framework, currently under development, for mission operations planning of highly autonomous spacecraft. Our framework addresses several knowledge engineering processes in the uplink-downlink cycle, including the interaction between uplink and downlink phases. We focus on capturing and refining intent and spacecraft modeling from different perspectives in an integrated fashion, acknowledging that this is an iterative process at multiple levels. We are in the process of implementing and refining the full set of tools presented in this paper. We than plan to conduct a series of user studies to refine their design and gather recommendations for future mission operations with increasingly onboard autonomy.

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

Ai-Chang, M.; Bresina, J.; Charest, L.; Chase, A.; Hsu, J.; Jonsson, A.; Kanefsky, B.; Morris, P.; Rajan, K.; Yglesias, J.; Chafin, B.; Dias, W.; and Maldague, P. 2004. MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission. *IEEE Intelligent Systems*, 19: 8–12.

Alper Ramaswamy, B. B.; Agrawal, J.; Chi, W.; Kim Castet, S. Y.; Davidoff, S.; and Chien, S. 2019. Supporting Automation in Spacecraft Activity Planning with Simulation and Visualization. In *AIAA Scitech Forum 2019*, SciTech 2019. American Institute of Aeronautics and Astronautics.

Barreiro, J.; Boyce, M. E.; Do, M. B.; Frank, J. D.; Iatauro, M.; Kichkaylo, T.; Morris, P. H.; Ong, J. C.; Remolina, E.; Smith, T. B.; and Smith, D. E. 2012. 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*.

Bernardi, G.; Cesta, A.; Finzi, A.; and Orlandini, A. 2013. 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)*.

Buffington, B.; Lam, T.; Campagnola, S.; Ludwinski, J.; Ferguson, E.; Bradley, B.; Scott, C.; Ozimek, M.; Haapala, A.; and Siddique, F. 2017. Evolution of trajectory design requirement of NASA's planned Europa Clipper mission. In *International Astronautical Congress (IAC)*.

Chi, W.; Agrawal, J.; Chien, S.; Fosse, E.; and Guduri, U. 2021. Optimizing Parameters for Uncertain Execution and Rescheduling Robustness. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29(1): 501–509.

Chien, S.; Doubleday, J.; Thompson, D. R.; Wagstaff, K.; Bellardo, J.; Francis, C.; Baumgarten, E.; Williams, A.; Yee, E.; Stanton, E.; and Piug-Suari, J. 2016. Onboard Autonomy on the Intelligent Payload EXperiment (IPEX) CubeSat Mission. *Journal of Aerospace Information Systems (JAIS)*.

Chien, S. A.; Rabideau, G. R.; Tran, D.; Troesch, M.; Nespoli, F.; Ayucar, M. P.; Sitja, M. C.; Vallat, C.; Geiger, B.; Vallejo, F.; Andres, R.; Altobelli, N.; and Kueppers, M. 2021. Activity-Based Scheduling of Science Campaigns for the Rosetta Orbiter. *Journal of Aerospace Information Systems (JAIS)*, 18(10).

Dellaert, F. 2012. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology.

Dellaert, F.; Kaess, M.; et al. 2017. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2): 1–139.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22.

Doran, G.; Lu, S.; Liukis, M.; Mandrake, L.; Rebbapragada, U.; Wagstaff, K. L.; Young, J.; Langert, E.; Braunegg, A.; Horton, P.; Jeong, D.; and Trockman, A. 2020. COSMIC: Content-based Onboard Summarization to Monitor Infrequent Change. In *2020 IEEE Aerospace Conference*, 1–12.

Doubleday, J. R. 2016. Three petabytes or bust: planning science observations for NISAR. In *SPIE, Earth Observing Missions and Sensors: Development, Implementation, and Characterization IV*, volume 9881.

Estlin, T. A.; Bornstein, B. J.; Gaines, D. M.; Anderson, R. C.; Thompson, D. R.; Burl, M.; Castaño, R.; and Judd, M. 2012. AEGIS Automated Science Targeting for the MER Opportunity Rover. *ACM Trans. Intell. Syst. Technol.*, 3(3).

Forney, G. D. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3): 268–278.

Fourie, D.; Teixeira, P. V.; and Leonard, J. 2019. Non-parametric mixed-manifold products using multiscale kernel densities. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6656–6662.

Francis, R.; Estlin, T.; Doran, G.; Johnstone, S.; Gaines, D.; Verma, V.; Burl, M.; Frydenvang, J.; Montaño, S.; Wiens, R.; et al. 2017. AEGIS autonomous targeting for ChemCam on Mars Science Laboratory: Deployment and results of initial science team use. *Science Robotics*, 2(7).

Hsiao, M.; and Kaess, M. 2019. Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree. In *2019 International Conference on Robotics and Automation (ICRA)*, 1274–1280. IEEE.

Hwang, I.; Kim, S.; Kim, Y.; and Seah, C. E. 2009. A survey of fault detection, isolation, and reconfiguration methods. *IEEE transactions on control systems technology*, 18(3): 636–653.

Knight, R.; McLaren, D.; and Hu, S. 2012. Planning Coverage Campaigns for Mission Design and Analysis: CLASP for the proposed DESDynI Mission. In *International Symposium on Artificial Intelligence, Robotics, and Automation for Space (ISAIRAS 2012)*. Turin, Italy.

Kolcio, K.; Fesq, L.; and Mackey, R. 2017. Model-based approach to rover health assessment for increased productivity. In *2017 IEEE Aerospace Conference*, 1–13. IEEE.

McCluskey, T. L.; Vaquero, T. S.; and Vallati, M. 2017. Engineering Knowledge for Automated Planning: Towards a Notion of Quality. In *Proceedings of the Knowledge Capture Conference*, K-CAP 2017. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355537.

McCoy, K.; Nairouz, B.; Bradley, B.; Jones-Wilson, L.; and Susca, S. 2018. Assessing the science robustness of the Europa clipper mission: Science sensitivity model. In *IEEE Aerospace Conference*, 1–13.

Rabideau, G.; Chien, S.; Mclaren, D.; Knight, R.; Anwar, S.; and Mehall, G. 2010. A Tool for Scheduling THEMIS Observations. In *International Symposium on Space Artificial Intelligence, Robotics, and Automation for Space (ISAIRAS 2010)*. Sapporo, Japan.

Rabideau, G.; Wong, V.; Gaines, D.; Agrawal, J.; Chien, S.; Kuhn, S.; Fosse, E.; and Biehl, J. 2020. Onboard Automated Scheduling for the Mars 2020 Rover. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation for Space*, i-SAIRAS'2020. Noordwijk, NL: European Space Agency.

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2): 257–286.

Shivashankar, V.; Alford, R.; Kuter, U.; and Nau, D. S. 2013. Hierarchical Goal Networks and Goal-Driven Autonomy: Going where AI Planning Meets Goal Reasoning. In *Goal Reasoning: Papers from the ACS Workshop*, 95–110. Baltimore, MD, USA.

Simpson, R. M.; Kitchin, D. E.; and McCluskey, T. L. 2007. Planning domain definition using GIPO. *The Knowledge Engineering Review*, 22(2): 117–134.

Streiffert, B. A.; and Polanskey, C. A. 2004. Observation Planning Made Simple with Science Opportunity Analyzer (SOA). In *Core Technologies for Space Systems Conference*.

Susca, S.; Jones-Wilson, L. L.; and Oaida, B. V. 2017. A Framework for Writing Science Measurement Requirements and its Application to the Europa Multiple Flyby Mission. In *IEEE Aerospace Conference*. Jet Propulsion Laboratory, National Aeronautics and Space Agency, Pasadena, CA.

Troesch, M.; Mirza, F.; Hughes, K.; Rothstein-Dowden, A.; Bocchino, R.; Donner, A.; Feather, M.; Smith, B.; Fesq, L.; Barker, B.; et al. 2020. 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)*.

Vaquero, T.; Romero, V.; Tonidandel, F.; and Silva, J. 2007. itSIMPLE 2.0: An Integrated Tool for Designing Planning Domains. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 336–343.

Verfaillie, G.; and Pralet, C. 2020. 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)*.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2): 260–269.

Wang, D.; Russino, J. A.; Basich, C.; and Chien, S. 2022. Analyzing the Efficacy of Flexible Execution, Replanning, and Plan Optimization for a Planetary Lander. In *International Conference on Automated Planning and Scheduling*.

Yelamanchili, A.; Agrawal, J.; Chien, S.; Biehl, J.; Connell, A.; Guduri, U.; Hazelrig, J.; Ip, I.; Maxwell, K.; Steadman, K.; and Towey, S. 2021a. Ground-based Automated Scheduling for Operations of the Mars 2020 Rover Mission. In *Proceedings Space Operations 2021*.

Yelamanchili, A.; Chien, S.; Cawse-Nicholson, K.; and Freeborn, D. 2021b. Scheduling and Operations of the ECOSTRESS Mission. In *Proceedings Space Operations 2021*.