Tracking Logistical Constraints Across Missions and Organizations: A Multipurpose Information Infrastructure

Michael Orosz[*]
Tatiana Kichkaylo[*]
Edward Balaban[**]
Andre Goforth[***]
Adam Sweet[**]
Robert Neches[*]

[*] *University Of Southern California Information Sciences Institute*
4676 Admiralty Way, Marina Del Rey, CA 90292

[**] *QSS Group Inc./NASA Ames Research Center, Moffett Field, CA 94035*

[***] *NASA Ames Research Center, Moffett Field, CA 94035*

Abstract

The goal of many enterprises is to develop and execute plans and schedules that achieve stated goals while simultaneously minimizing the cost of logistics management and maximizing resource productivity. This goal is a challenge in space flight environments where just-in-time logistics management can't be supported and large scale planning and scheduling requires collaborations and negotiations that cross many divisions and departments.

Although there have been many systems proposed and/or developed that address one or more of these concerns, a key element missing in these systems is the tight coupling that is necessary between maintenance, logistics, and operations. This close relationship is particularly important in space operations where changes to scheduled missions and/or the logistics chain can greatly impact overall operations.

Based on previous work on combined maintenance and operations planning/scheduling for Marine Aviation, we propose a software-based infrastructure that coordinates planning, scheduling, and execution across multiple departments and disciplines.

1. Introduction

Space-based missions require considerable resources. In most cases, these missions occur far from sources of service and re-supply, severely limiting what and how much can be transported or stored. In addition, this limitation on available resources greatly reduces the types and numbers of missions that can be planned or updated and reduces planning/scheduling flexibility. For example, unplanned maintenance typically requires a maintenance-specific mission and resources. The ability to quickly respond to such a request will be limited by a lack of resources (e.g., spare parts, tools, skills, etc.). The ability to handle these types of unplanned missions will become critical as space exploration moves beyond low-earth orbit to the moon, Mars, and beyond where planning and scheduling will have to occur on-board and in real-time.

Space exploration planning and scheduling involves both operations and maintenance missions. Ensuring that proper resources are available at the right place and at the right time is critical to efficient mission execution. As the number of orbiter launches, crew and cargo vehicles, and habitats increase, the need to coordinate multiple source and multiple location supply-chain logistics increases as well. When plans and schedules change, resource inventories at multiple locations and cargo manifests on multiple vehicles must adjust to ensure efficient execution and successful mission completion. One of the challenges which differentiates this from many other technical initiatives is the need to flow effects all the way back through a lengthy supply chain. This requires coordination across very different temporal ranges.

The ability to generate and maintain plans and schedules in a multi-discipline and highly distributed environment is a subject of much research **[1]**. Examples of such environments include military and civilian aviation, commercial trucking and freight operations. Typically, stakeholders in these environments collaborate and generate plans and schedules using a collection of disparate tools and manual techniques. Although these methods are sufficient in many environments, they do not scale well as the size and complexity of the problem increases.

Coordinated Multi-source Maintenance on Demand (CMMD) is an integrated multi-agent planning and scheduling decision support system targeted for both current and future NASA manned and robotic space flight operations. CMMD is not an autonomous planning/scheduling tool. Rather, CMMD is an open system that provides planning, scheduling, and execution decision-support tools that *aid* NASA planners in developing, maintaining and executing plans and schedules.

A key feature of CMMD is the integration of maintenance and supply-chain logistics planning into the overall planning and scheduling process. This avoids the scheduling and execution problems that often result when maintenance and supply-chain logistics are planned and executed separately – often after the operations plan and/or schedule has been generated. This is particularly a problem during real-time schedule execution when changes made by various disciplines are difficult to coordinate because there are multiple plans to consider. In CMMD, planners and schedulers from all areas (disciplines) collaborate through negotiation technologies to achieve a single coordinated plan/schedule (and associated logistics artifacts, such as cargo manifests and inventory lists) that meets the needs of all stakeholders and that of the overall mission.

The CMMD infrastructure allows multiple agents to work on a virtually centralized constraint-based representation of the plan called the Living Schedule. The plan is modified by executing rules, which correspond to procedures, and the Living Schedule explicitly encodes produced dependencies. Publish-subscribe mechanisms regulate propagation of updates through the system.

In an initial implementation of the CMMD software, the management of cargo manifests for a single increment will be supported. The software determines what resources are required for specific tasks and then determines if the resources are in the current habitat and/or orbiter inventory. If the resources are not in the current inventory, the CMMD software is designed to interface to an external cargo manifest generating system to ensure that the proper resources are made available.

2. The Problem

Mission durations, already in months with the International Space Station (ISS), will only increase as space exploration moves to the moon, Mars, and beyond. The planning and scheduling of both manned and unmanned missions involves determining what resources make it on cargo manifests and into station/habitat inventories. These decisions are influenced by concerns such as safety, volume, lift capacity, cost, as well as current and future mission requirements.

Long-running operations, such as the ISS and future lunar habitats, require tracking equipment health for service life estimation, repair, and refurbishment. Emergency repairs, re-supply, and unscheduled events such as launch delays and unscheduled maintenance, illustrate needs for minimally disruptive plan repair. The large number of people and organizations involved in space operations further complicate the situation -- which will be compounded as multiple ISS/habitat, manned CEV/Shuttle, and robotic operations overlap and need to support each other.

It is not feasible to maintain large teams of human experts to develop mission plans that ensure all logistical constraints are satisfied, manage the execution of the plans, and then re-plan/repair if problems arise. This is particularly true as manned and unmanned operations extend further from Earth where planning/scheduling decisions must be made locally, with little input from ground control. As NASA Administrator Michael D. Griffin recently announced, *"We need a system that needs a smaller support base .... The only way to do that is to use fewer people on any given thing… [and]… do more things - other things than have everybody crowd around one space launch vehicle."*

We address these issues by providing an infrastructure for *coordination* among multiple, discipline-oriented planning/scheduling systems. The infrastructure provides ties to the information environment that allow it to inform each participant of changing situations. Managing and enforcing logistical constraints across organizational and mission boundaries, entails handling several challenges:

Encoding constraints. Currently, knowledge about operations and logistic chain management is recorded in text describing ground rules and constraints. To automate any decision support, these rules and constraints must be formally represented. Since cost reductions depend on tying logistics to operations planning, the resource needs of scientific and maintenance missions as well as the requirements necessary to maintain high safety standards must also be known to the system.

Supporting collaboration. Logistical chains usually span multiple organizations. Automatic tracking and propagation of dependencies along such chains requires simultaneous shared access to the global plan with means to resolve conflicts and race conditions caused by such parallel access.

Scaling. Thousands of people in hundreds of disciplines are involved in planning and executing space missions. An IT infrastructure for logistic chain management should be scalable to accommodate such chains.

3. Current Planning Approach to Space-Based Operations

3.1. Domain Background

Currently, NASA's manned space operations consist of the Space Shuttle and the International Space Station (ISS). These programs provide a manned presence in Low Earth Orbit (LEO) where space operations such as satellite servicing, ISS maintenance, and scientific research are undertaken. These programs will eventually be replaced by Exploration Systems program. This new program has the mission of returning humans to the moon and ultimately Mars. The Space Shuttle will be replaced with the CEV and the ISS will serve as a test-bed for validating systems and technologies for crewed lunar and planetary outposts, including the technologies developed in this effort.

Unlike the Space Shuttle program that has long-established operational processes, procedures and planning/scheduling tools, the ISS is still in the process of construction and so its planning processes have to accommodate both normal and unique one-time construction activities for completion of station assembly.

This manned space program is divided into the following branches: flight operations, planning, training, and facilities. The major functions within the flight operations area are real-time operations (involving flight directors and controllers), integration of operations with international partners, and real-time planning and scheduling. The planning activities consist of mission concept definition, mission requirements integration, long-term and short-term planning of Systems, EVA and Intra-Vehicular Activity (IVA) tasks, robotics activities, payload and launch integration, and several others.

CMMD is initially focused on ISS operations and logistics planning, with a particular focus on EVA tasks; however, support for the execution area and its real-time specifics will be incorporated in later phases. The section below describes in detail how operations planning is done currently in order to provide the reader with the understanding of motivation guiding CMMD work.

3.2. ISS Operations Planning

In planning ISS operations, the largest unit of time ordinarily dealt with is an *increment*. An increment is the period of time covering activities of a specific crew onboard a space station or at a habitat. The currently used duration of ISS increments is roughly six to eight months.

3.2.1. Initial Increment Definition Phase. Planning for an increment starts with the assignment of an Increment Manager (IM). The IM assembles the Increment Management Team (IMT) consisting of representatives from various disciplines involved in the mission, such as engineering, science, EVA, and others. This happens a year to a year and a half prior to actual increment starts. The group's first objective is to come up with the Increment Definition Requirements Document (IDRD). This document defines, in broad strokes, the major stages of the increment, such as manned and cargo spacecraft arrivals and departures, crew time allocation, supply-chain logistics, as well as primary science, assembly, and maintenance tasks. Some event dates are defined but assumed to be approximate, unless related to a time-sensitive event such as a launch.

3.2.2. On-Orbit Summary (OOS). About six months prior to the start of the increment, and often overlapping the final stages of IDRD planning, the OOS is started. The OOS is developed by the Mission Operations Directorate (MOD) and is derived from the IDRD and is a more detailed plan of the increment. The OOS covers the duration of the entire increment and describes it on the day-by-day basis. Tasks for each day are listed, although the order and the duration of tasks are kept flexible. There is also a practice of inserting "padding" between tasks to help deal with variations in crew working styles as well as with the situations where a task takes much longer than expected.

3.2.3. Weekly Look-ahead Plan (WLP). At the start of an increment and throughout its execution, detailed weekly plans, called Weekly Look-ahead Plans (WLP) are created. Two weeks prior to the week being planned, the WLP is created. Tasks within each day are ordered and are assigned time slots, crew, and resources. During this planning process, various teams and international partner representatives negotiate with each other in an attempt to get key tasks and sub-tasks scheduled. This process can become quite tense towards the end of an increment where time and resources are limited.

3.2.4. Short Term Plan. The final level of detail is provided in the Short Term Plan (STP). The STP is the daily schedule that is executed on station. Planners produce it by extracting the next days plan out of the WLP a week in advance and solidifying the times/ordering for tasks, crew assignments, and other details. The plan is then circulated among the international partners and stakeholders for approval. The approval process takes roughly a day.

Next, the plan is recorded in an electronic form as the On-Orbit Short Term Plan (OSTP) and uploaded onto the on-board computers. Once execution of an OSTP begins, the crew can mark the completion status of each task and provide comments. Status information is also collected by the ground personnel at the end of each day via a videoconference and is used to adjust the next day's OSTP.

3.2.1. Tools Utilized. For the most part, ISS planning and scheduling is primarily a manual process. This is particularly true when handling changes and unexpected events. Members of the planning team typically rely on commercially available tools such as Microsoft Excel and Project, as well as on verbal and electronic communications (e.g., electronic mail, etc.). Attempts made to automate portions of this process have met with limited success.

4. Challenges

Over the next 5-8 years, NASA will simultaneously undertake three major programs; completion of the ISS, continued Space Shuttle operations, and development and testing of the Crew Exploration Vehicle (CEV). Below is an overview of the challenges that will arise in the near and longer terms for space exploration and their implications for planning/scheduling work.

4.1. Increased level of activities

Over the next several years, ISS build-out activities will put a severe strain on the current planning/scheduling infrastructure [2]. For instance, it is anticipated that the number of planned EVAs to handle the build-out will increase from the typical 4-5 per year to around 25-30 per year. In addition, existing and new equipment (e.g., instrumentation, robotics, etc.) on-board will need to be maintained and replaced which will require additional maintenance EVAs and IVAs to be scheduled.

As new modules are added during the ISS build-out, such as Columbus science module from the European Space Agency (ESA), there will be increased scheduling of scientific tasks. In addition, crew size will also likely increase to accommodate the new science missions and to support the increased maintenance activities. .

4.2. New types of spacecraft

Several new spacecraft types are nearing deployment and will have to be integrated into the planning and scheduling process. These new spacecraft include ESA's Automated Transfer Vehicle (ATV), the aforementioned CEV (intended to replace the Space Shuttle in the US space program), and possibly Russia's new manned Kliper. In each case, the planning and scheduling process must take into account a variety of new information, such as payload capacity, size of crew, length of flight, and other specialized needs and capabilities.

4.3. Logistical difficulties

With the availability of the US Space Shuttle fleet to undertake frequent flights in question, sustaining an effective logistical pipeline to and from the station will be difficult. Currently, the other available cargo spacecraft, the Russian Progress, has a limited capacity (2.7 metric tons vs. Shuttle's 25 tons) and cannot return cargo from orbit. The ESA's ATV, while increasing the amount of cargo that can be launched to approximately 9 metric tons, will, like the Russian Progress, not be able to return cargo from orbit. The CEV and the Kliper will be capable of returning back some cargo, however, it will most likely be on the order of a few hundred kilograms and limited in volume. All of these factors will need to be taken into account when planning logistical support for larger crews, an aggressive ISS build-out schedule, and an increased science agenda.

The Moon-Mars Exploration Program, in addition to dealing with many of the present challenges, will also present some new ones:

4.4. Planetary base operation

While NASA and its international partners have by now accumulated significant experience operating space stations in LEO, no manned planetary bases have ever been established. NASA plans to operate such bases on the lunar surface and use this experience as the foundation for eventual expeditions to the Martian surface. It is assumed that operating a lunar base will be similar in many respects to operating a space station, but there will also be significant differences, such as those related to in-situ resource utilization and lengthy surface excursions, possibly with the help of robotic assets.

4.5. Greater operational distances

While multi-million mile expeditions to Mars are at least a couple of decades away, even lunar operations, conducted at a distance of "mere" 240000 miles, will present significant obstacles in terms of logistical support, communications, and situational awareness of the ground support personnel. To accommodate these challenges, more of the planning and scheduling authority (particularly in the day-to-day operations) will need to be given to astronaut crews. This will increase the complexity of the overall planning and scheduling process since both local and remote planning/scheduling needs and requirements will have to be integrated – a particularly challenging prospect given the distance and anticipated communication breakdowns between ground and lunar operations.

4.6. Operating multiple simultaneous missions

Another new challenge will be the planning of simultaneous, possibly interdependent, multiple missions. One example would be a crew operating in lunar orbit and another crew in a lunar habitat. This will raise the issues of task coordination and resource sharing, and will require rethinking of the current planning/scheduling approach as well.

5. CMMD

CMMD is a distributed planning/scheduling and decision-support system that addresses many of the challenges previously discussed. The overall goal of CMMD is to not only support current Space Shuttle and ISS operations, but also scale and support future lunar and habitat operations. CMMD is an open system that *aids* the user in decision-support. It provides suggested plans and schedules, alerts users to changing conditions, and offers advice. Ultimately, the user has the final say and relies on CMMD as a decision-support system.

CMMD consists of multiple independent agents that communicate with a virtual Backbone. In the Backbone, a representation of the current state of the increment plan/schedule, called the Living Schedule (LS) is maintained as well as the states of various cargo manifests and habitat inventories. The Backbone performs access control and notification propagation allowing agents with different areas of expertise to collaborate on the LS, manifests, and inventories.

Domain information, such as rules, constraints, preferences, and variables is collected through the PRISM (Planning & Real-time Information for Space Missions) component that connects to already existing databases (such as JSC's Orbital Data Reduction Complex) and supplements it with user-entered data. The information is then processed, organized, and indexed in an internal domain data repository. As needed, it is transmitted to the agents (and LS) via Backbone and translated into the appropriate representation.

Information is contained in the LS in a declarative form. The CMMD data model separates representation of the plan/schedule from rules and procedures used to modify the plan and also from user preferences. In addition, the LS stores dependencies between elements of the plan, so that every planning decision made by CMMD is traceable to the rule or preference this decision is based on. Dependencies between elements of the plan and the contents of cargo manifests and habitat inventories are also maintained.

Such cause-effect dependencies can be individually inspected and disabled by users. CMMD propagates the effects of any changes made by the user. This way the user has complete information and control over planning decisions, while the CMMD system handles the tedious and error-prone dependency propagation.

Note that dependency propagation performed by CMMD takes into account logistics rules and constraints. This way, when a user adds a new task to the schedule, the system automatically computes logistics implications of this task and assesses its feasibility, cost, and impact on other activities.

The size of the plan stored in the LS and the total number of agents working on it may be quite large. Finding an optimal solution would require a complete search, which is infeasible in such a large problem. Moreover, the problem with undertaking a complete search of the solution space is compounded by constantly changing conditions and user preferences. CMMD addresses this issue through algorithms that focus on finding "good enough" solutions, plan repair, and evaluation of alternatives.

Finally, CMMD is a multi-agent system. The virtual Backbone that connects the agents is also responsible for storing the Living Schedule and related cargo manifests and habitat inventories, as well as for notifying agents about relevant changes in the LS.

Various services provided by the Backbone allow agents to cope with temporary disconnections and facilitate negotiations by keeping track of dependencies while minimizing traffic.

In the rest of this section we discuss various elements of the CMMD architecture in more detail.

5.1. Domain and plan representation

CMMD relies on a data model that combines features of constraint-based systems (**[3], [4], [5]**) with those of hierarchical task network-based (HTN) planners **([6], [7], [8])**. In CMMD, physical and virtual resources are described in terms of timelines. Activities that use these resources are represented by tokens placed on the timelines. Relationships between the activities are captured by procedural constraints connecting variables of the tokens. For example, one may restrict the end time of one activity to be less than the start time of another activity (the temporal *Before* constraint).

In addition to resource timelines, the CMMD data model supports real-valued capability timelines. Capabilities are tied to resources. For example, resource Battery may have capability Charge, whose value is affected by Use and Recharge tokens placed on the Battery timeline.

At any given moment, the state of the CMMD plan is described by the tokens residing on the various resource timelines. A user can change the plan by explicitly adding and removing tokens. Often such a change brings the plan to an invalid state, which can be fixed by adding, removing, and/or modifying other tokens. For example, scheduling a particular crewmember to perform an EVA at some point in the future requires scheduling the same crewmember for training before the proposed mission.

One of the goals of our system is to help the user keep track of such dependencies between activities. CMMD captures such dependencies using two kinds of rules: safety and achievement rules. Safety rules describe legal states of the world, while achievement rules represent standard expansions of high-level activities into procedures.

As with most constraint-based planning systems, CMMD allows non-singleton values for variables at different stages of the problem solving process. A distinguishing feature of the CMMD representation is that *all* choice points are captured using variables. This includes, among other things, the choice of a resource to be used in a particular activity (*timeline* variable of a token) and the choice of expansion rule for a high-level activity (*support* variable of a token).

5.2. Multi-agent architecture

A CMMD system consists of multiple agents connected to the Backbone. Backbone is a virtually centralized entity. However, it can be implemented using either a centralized solution or peer-to-peer technology.

The Backbone provides the following functionality:

- Persistence and 24x7 access to the Living Schedule, cargo manifests, and habitat inventories
- Session management and access control for agents
- Query and subscription service

For the most part, agents represent the various planning disciplines in the current NASA planning process. However, neither the set, nor the types of agents connected to the Backbone are fixed. In a typical mission-planning environment, each human user would have a dedicated agent. Legacy systems and external data sources would also have their own agents. Various physical resources might also be connected to the Backbone using agents.

All CMMD agents implement the same interface to connect to the Backbone. Agents can read and write different portions of the Living Schedule as determined by their access rights. For example, an agent corresponding to a physical resource submits real-time data about the state of this resource to the Living Schedule. Agents interested in the state of this resource post a subscription on that resource to the Living Schedule. Upon receipt of an update from the physical resource agent, the Backbone checks the list of current subscriptions and sends notifications to all interested agents.

In addition to querying and changing the state of the Living Schedule, agents can negotiate with each other by exchanging proposed “patches” to the current state of the Living Schedule before committing their changes. This process is discussed in more detail below.

5.3. CMMD agent

CMMD agents can operate independently or as part of a multi-agent community by communicating and negotiating with each other via the Backbone.

Figure 1 shows a simplified structure of a CMMD agent. Agent modules can be divided into two classes: domain-independent and domain-specific.

At the core of the domain-independent portion of the agent sits the Knowledge Base (KB). The KB stores the local view of the data from the Living Schedule, the agent’s internal data, and intermediate results of computations. The KB is also responsible for communicating with the Backbone and performs other functions, such as detecting inconsistencies and keeping track of dependencies among alternative solutions.

The Rule Engine (RE) relies on safety and achievement rules to help ensure the validity of the plan. When modifications are made to the plan, the RE checks the state of the plan stored in the KB and fires applicable safety and achievement rules. In many cases, these rule firings may cause the plan to be updated to ensure the validity and legality of the plan.

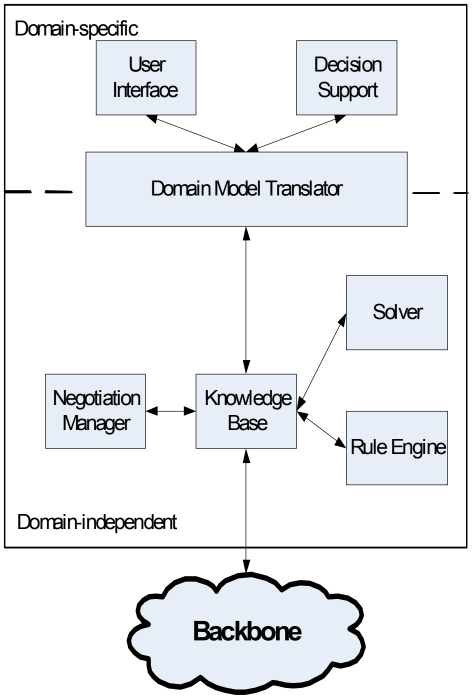

Figure 1. CMMD agent architecture

The Rule Engine is a purely production system. It does not perform search or any expensive computations. The Solver, on the other hand, is responsible for finding a consistent assignment of values to all variables in the plan – this requires search. CMMD architecture does not prescribe which algorithm to use for this purpose. Currently we are investigating several implementations of the Solver module, including algorithms that accept external guidance in the form of variable and value ordering heuristics.

As mentioned earlier, CMMD agents can evaluate multiple alternatives. Moreover, agents can "discuss" alternatives with each other before submitting them to the Living Schedule. For example, during the planning of an EVA, the EVA agent may consult with the Maintenance agent and the Flight Surgeon to confirm that both the equipment (e.g., space suit) and the crewmembers are ready for the space walk. The CMMD system allows for "discussing" multiple alternatives with different agents without involving human users. These localized negotiations (undertaken by the Negotiation Manager) eventually generate and present different options – some more feasible than others – to the user. The benefit of this negotiation process is that the agents do most of the work, leaving the human decision-maker to make the final choices.

All modules described so far are domain-independent. The same code can be used in other domains in addition to space exploration. The only difference will be the information stored in the Living Schedule and the sets of rules used by the agents. Some modules, however, are necessarily domain-dependent. The most obvious example is the user interface. A CMMD agent can also include a Decision Support module responsible for generating suggestions and detecting opportunities based on the current state of the schedule.

The domain-specific modules of the architecture read and modify information stored in the Knowledge Base via the Domain Model Translator (DMT). The DMT permits more natural access to domain objects without jeopardizing reusability of the domain-independent modules.

5.4. Current Status

A preliminary CMMD prototype capable of generating a plan and schedule that supports multiple EVAs over a one-week planning horizon has been developed. The prototype relies on approximately 100 achievement and safety rules, the skills and availability of crewmembers, and the availability of inventory to generate a detailed schedule that includes dates, start times, and expected durations of each task.

The next phase of development will focus on extending the prototype to support multi-month increment plans and schedules

6. Conclusions

CMMD is a multi-agent planning and scheduling decision support tool being developed for future NASA space flight operations. Although targeted for the Space Exploration Program, the initial prototypes will, as a proof-of-concept, be used to support planning and scheduling of the current International Space Station operations.

CMMD is not being designed as an autonomous planning/scheduling tool. Rather, it is a planning, scheduling, and execution decision-support system that aids planners in developing and, if needed, quickly changing their plans and schedules.

A key feature of CMMD is the integration of maintenance into the overall planning and scheduling process. Through negotiation technologies, both scheduled and unscheduled maintenance, along with its impact on the overall logistic supply-chain are considered every time a mission plan, weekly, or daily schedule is created, updated, or executed. This close integration of maintenance and logistics into every planning/scheduling operation is key to preventing unexpected mission delays and/or aborts due to maintenance or supply-chain shortcomings.

The CMMD architecture is centered on the concept of the "Living Schedule" in which multiple agents – each representing a unique NASA planning discipline – interact. The Living Schedule is persistent and available to all agents on a 24x7 basis. The Living Schedule contains both the current and future increment plans/schedules, allowing for both mission planning and current plan execution.

Other key features of the system include generating so-called "good enough, soon enough" plans, legal schedules, localized planning/scheduling, and the ability to undertake multiple-path "what-if" analysis.

7. Acknowledgements

This research was supported by NASA grant NNA05CS29A. The authors would like to acknowledge the contributions of their CMMD teammates from USC ISI, Vanderbilt University, JSC, and Hamilton Sundstrand who devoted their considerable talents and energy to making this project possible (in alphabetical order): Chris Van Buskirk, Steven Gonzalez, Shashi Gowda, Gabor Karsai, Keith Massei, Himanshu Neema, Sandy Peterson, Carolina Quinteros, and Suzie Shimamoto. The authors would also like to express their gratitude to following individuals affiliated with JSC, Hamilton Sundstrand, and USA for being exceptionally generous with their time and expertise in helping the CMMD team: Melissa Arnold, Frank Birkenseher, Heath Borders, Susan Brandt, Stephen Broussard, Anthony Butina, Wade Frost, Michele Gonzales, David Hughes, Joseph Kitchen, Robert Knight, David Korth, Kenneth Kruse, Troy LeBlanc, Kathleen Leary, Chris Looper, Robert McCormick, Patricia Meyer, Blair Nader, Carlos Noriega, William Robbins, Jim Ruszkowski, Sarah Shull, Ernest Smith, Kenneth Todd, Debbie Trainor, Mark Trevino, Kevin Watson, and Dennis Webb.

8. References

[1] Durfee, E. and Musliner, D. (eds), *Distributed Plan and Schedule Management: Technical Report SS-06-04,* AAAI Press, Menlo Park, CA, 2006.

[2] Ney, Z. and Looper, C., "Integrated Planning and Logistics of Exploration Initiatives for Long Term Maintenance EVA/EVR Operations," AIAA 1st Space Exploration Conference, 2004.

[3] Ghallab, M. and Laruelle, H. "Representation and Control in IxTeT, a Temporal Planner," AIPS, 1994.

[4] Smith, D., Frank, J., and Jonsson, A. "Bridging the Gap Between Planning and Scheduling," Knowledge Engineering Review 15(1), 2000.

[5] Jonsson, A. and Frank, J. "A framework for dynamic constraint reasoning using procedural constraints," ECAI, 2000.

[6] Erol, K., Hendler, J., and Nau, D., "UMCP: A Sound and Complete Procedure for Hierarchical Task-network Planning," AIPS, 1994.

[7] Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, W., Wu, D., and Yaman, F. "SHOP2: An HTN Planning System," JAIR 20:379-404, 2003.

[8] Currie, K. and Tate, A. "O-Plan: the open planning architecture," AI 52:49-86, 1991.