A Survey of CASPER Applications

Russell Knight

Jet Propulsion Laboratory, California Institute of Technology

Abstract

We present a survey of applications of the Continuous Activity Scheduling, Planning, Execution, and Replanning (CASPER) system. CASPER is a real-time embedded planner scheduler. Its core planning and scheduling reasoning is based on the Activity Scheduling and Planning Environment (ASPEN). CASPER manages the execution and monitoring of plans and has been ported to various platforms and has been adapted to several applications. These include the Earth Observing One (EO1) spacecraft, the Three Corner Sat (3CS) mission, the rover Onboard Autonomous Science Investigation System (OASIS), aerobot, Uninhabited Aerial Vehicle Synthetic Aperture Radar (UAVSAR), unmanned sea surface and underwater vehicles, and sea gliders.

Introduction

The majority of planning and scheduling research has focused on batch-oriented models of planning. In contrast, CASPER (Chien 2000) uses iterative repair techniques to support a continuous planning process as is appropriate for autonomous control. This allows the plan to incorporate execution feedback such as early or late completion of activities, and over-use or under-use of resources. In this approach, iterative repair supports continuous modification and updating of a current working plan in light of changing operating context.

What follows is a description of CASPER and some of the applications of CASPER to real problems (either experimental systems or deployed systems).

CASPER

CASPER utilizes a *continuous planning* approach. Rather than considering planning a batch process in which a planner is presented with goals and an initial state, the planner has a current goal set, a plan, a current state, and a model of the expected future state. At any time an incremental update to the goals, current state, or planning horizon (at much smaller time increments than batch planning)[1] may update the current state of the plan and thereby invoke the planner process. This update may be an unexpected event or simply time progressing forward. The planner is then responsible for maintaining a consistent, satisficing plan with the most current information. This current plan and projection is the planner's estimation as to what it expects to happen in the world if things go as expected. However, since things rarely go exactly as expected, the planner stands ready to continually modify the plan. From the point of view of the planner, in each cycle the following occurs:

- changes to the goals and the initial state first posted to the plan,
- effects of these changes are propagated through the current plan projections (includes conflict identification)
- plan repair algorithms are invoked to remove conflicts and make the plan appropriate for the current state and goals.

This approach is shown in below in Figure 2. At each step, the plan is created by using iterative repair with:

- the portion of the old plan for the current planning horizon;
- the updated goals and state; and
- the new (extended)planning horizon.

The overall architecture for the continuous planning approach is shown in Figure 4. We now describe how each of the four basic components operates.

The planner process maintains a current plan that is used for planning (e.g. hypothesizing different courses of action). It responds to requests to replan initiated by the execution processes, activity commitments vfrom the execution module, state (and resource) updates from state estimation, and new goals (from external to the system). All of these requests are moderated by the synchronization process that queues the requests and ensures that one request is complete before another is initiated. The planners copy of the current plan is also where projection takes place and hence it is here that future conflicts are detected. However, as we will see below, requests to fix conflicts occur by a more circuitous route.

The execution process is the portion of the system concerned with a notion of "now". The execution module maintains a copy of the plan that is incrementally updated whenever the planner completes a request (e.g., a goal change, state change, or activity change). This local copy

[1] For the control domain of interest we typically see an update rate on the order of 10s of seconds real time.

includes conflict information. The execution module has three general responsibilities:

1. to commit activities in accordance with the commitment policy as they approach their execution time;
2. to actually initiate the execution of commands (e.g., processes) at the associated activity start times
3. to request re-planning when conflicts exist in the current plan

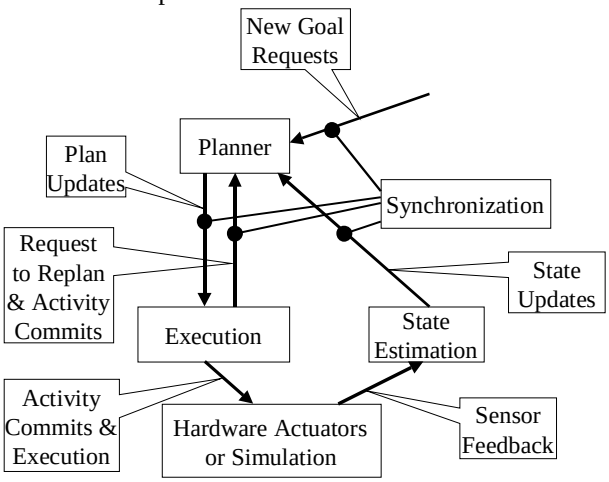

Figure 1 CASPER architecture

The execution module tracks the current time and indexing into relevant activities to commit and execute them. The execution module also tracks conflict information as computed by the projection of the planner and submits a request for replanning to the synchronization module when a conflict exists.[2]

The state estimation module is responsible for tracking sensor data and summarizing that information into state and resource updates. These updates are made to the synchronization module that passes them on to the planners plan database when coordination constraints allow.

The synchronization module ensures that the planner module(s) are correctly locked while processing. At any one time the planner can only be performing one of its four responsibilities: (re)planning, updating its goals, incorporating a state update, updating the execution module's plan for execution, or updating commitment status (otherwise we run the risk of race conditions causing undesirable results). The synchronization module serializes these requests by maintaining a FIFO task queue for the planner and forwarding the next task only when the previous task has finished.

The execution module also has a potential synchronization issue. The planner must not be allowed to modify activities (through replanning) if those activities might already have been passed on to execution. We enforce this non-interference by *commiting* all activities overlapping a temporal window extending from now to some short period of time in the future (typically on the order of several seconds). We ensure that the planner is called in a way that each replan request will always return within this time bound and we enforce that the planner never modifies a committed activity. This ensures that the planner will not complete a replan with an activity modified that is already in the past. Additionally, we use the synchronization process to ensure that the Execution module does not commit activities while the planner is replanning. This prevents the planner from modifying activities that have been committed subsequent to the planner call (but still in the future).

EO1

NASA's Earth Observing One Spacecraft (EO1) (Chien 2005) has been adapted to host an advanced suite of onboard autonomy software designed to dramatically improve the quality and timeliness of science-data returned from remote-sensing missions. The Autonomous Sciencecraft Experiment (ASE) enables the spacecraft to autonomously detect and respond to dynamic scientifically interesting events observed from EO-1's low earth orbit. ASE includes software systems that perform science data analysis, mission planning, and runtime robust execution. In this article we describe the autonomy flight software, as well as
innovative solutions to the challenges presented by autonomy, reliability, and limited computing resources.
The ASE onboard flight software includes several autonomy software components:

1) Onboard science algorithms that analyze the image data to detect trigger conditions such as science events, "interesting" features, changes relative to previous observations, and cloud detection for onboard image masking
2) Robust execution management software using the Spacecraft Command Language (SCL) [Interface & Control] package to enable event-driven processing and low-level autonomy
3) CASPER replans activities, including downlink, based on science observations in the previous orbit cycles. The onboard planner (CASPER) generates mission operations plans from goals provided by the onboard science analysis module. The model-based planning algorithms enable rapid response to a wide range of operations scenarios based on a deep model of spacecraft constraints, including faster recovery from spacecraft anomalies. The onboard planner accepts as

[2] In our implementation replanning is initiated by the execution module because this allows for the notion of urgency information (e.g. closeness of the conflict to current execution) to be incorporated in the decision to replan. If we did not wish to incorporate this information, the planner module could make this request directly to the synchronization module.

inputs the science and engineering goals and ensures high-level goal-oriented behavior.

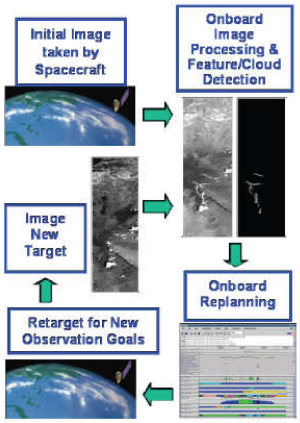

Figure 2 EO1 Autonomous Activity Flow

A typical ASE scenario involves monitoring of active volcano regions such as Mt. Etna in Italy. (See Figure 2.) Hyperion data have been used in ground-based analysis to study this phenomenon. The ASE concept is applied as follows: 1) Initially, ASE has a list of science targets to monitor that have been sent as high-level goals from the ground. 2) As part of normal operations, CASPER generates a plan to monitor the targets on this list by periodically imaging them with the Hyperion instrument. For volcanic studies, the infrared and near infrared bands are used. 3) During execution of this plan, the EO-1 spacecraft images Mt. Etna with the Hyperion instrument. 4) The onboard science algorithms analyze the image and detect a fresh lava flow, or active vent. If new activity is detected, a science goal is generated to continue monitoring the volcanic site. If no activity is observed, the image is not downlinked. 5) Assuming a new goal is generated, CASPER plans to acquire a further image of the ongoing volcanic activity. 6) The SCL software executes the CASPER generated plan to re-image the site. 7) This cycle is then repeated on subsequent observations.

3CS

The Three Corner Sat constellation (Chien 2001) consists of three satellites flying in a formation that degrades as the mission continues. The mission length is expected to be approximately three months, dependent on atmospheric drag that will eventually cause the satellites to de-orbit. Each of the three spacecraft has as its primary science instrument two fixed cameras. These cameras will be used to take images of the earth with the intention of capturing images of clouds. In order to simplify the mission and costs, the spacecraft will be tumbling (e.g., not attitude stabilized or controlled). The 3CS spacecraft are lightweight nanosatellites each weighing approximately 15 kg. The exterior envelope of the structure is a six-sided disk structure consisting of tubular supports and machined end caps to hold the bulk of the loading. Figure 3 shows the 3 spacecraft stack before deployment and a single spacecraft in the deployed state.

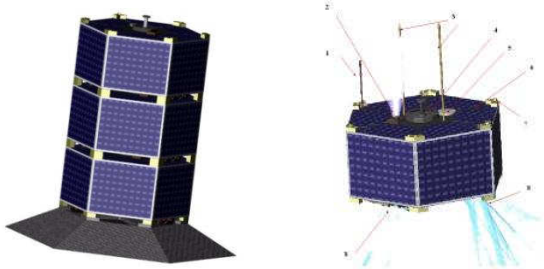

Figure 3 3CS Stack and Individual Satellite

For the 3CS mission, CASPER will manage the near-term mission plan in between daily uplink and downlink passes. This involves tracking the engineering, communications, and science activities and modifying them in response to execution feedback. The most significant of these will be replanning of science observations based on onboard data assessment. Using a science score computed for each image onboard, CASPER will prioritize images.

CASPER will then decide whether or not to discard images with little or no Earth in the frame, and plan for new images in an attempt to maximize the science return. While replanning, CASPER will need to account for available system resources: imaging opportunities, available power and energy, and available memory. CASPER will also need to manage setup procedures for downlink, uplink, command load, and imaging activities. Within the available flight processor resources, CASPER is expected to be able to respond to activity and state updates on the 10 to 60 second timescale. This will enable more recent information regarding the execution status of activities as well as monitored state and resource values to influence planning. CASPER will be integrated with the SCL execution system allowing for tight feedback from SCL rules and scripts to be reflected and acted upon within the CASPER plans.

OASIS

Surface rovers offer scientists the ability to move around a planetary surface and explore different areas of interest. The farther the rover can travel, the greater the opportunity exists for increased scientific discovery. Most mobile robot efforts at JPL and NASA have concentrated on navigation, manipulation, and control. For the Mars Exploration Rovers (MER) mission, process automation has already proven valuable in engineering areas. Due to advances in rover navigation, traverse ranges are increasing at a rate much faster than communications bandwidth. While the Sojourner rover traveled around 100m in the entire mission, the drive record for the most distance covered in a single sol (Martian day) is over 220 meters set by the MER Opportunity rover. As this trend in increased mobility

continues, the quantity of data that can be returned to Earth per meter traversed is reduced. Thus, much of the terrain the rover observes on a long traverse may never be observed or examined by scientists. We present a system developed to maximize the quality of the science data transmitted to Earth through the use of onboard science. This system expands onboard automation beyond the engineering domain to the science domain. The Onboard Autonomous Science Investigation System (OASIS) system has been developed to evaluate, and autonomously act upon, science data gathered by in-situ spacecraft such as 3 planetary landers and rovers (Castano 2007). OASIS analyzes the geologic data gathered by the rover onboard. This analysis is used to identify terrain features of interest and additional science gathering opportunities. A planning and scheduling component of the system enables the rover to take advantage of the identified science opportunity by updating the command sequence to include the opportunistic measurements. OASIS currently works in a closed loop fashion with onboard control software (e.g., navigation and vision) and has the ability to autonomously perform the following sequence of steps: analyze gray scale images to find rocks, extract the properties of the rocks, identify rocks of interest, retask the rover to take additional imagery of the identified target and then allow the rover to continue on its original mission. We have conducted a number of tests of the combined system and individual components. We describe results for the system in detecting and reacting to a science alert (identified science opportunity.

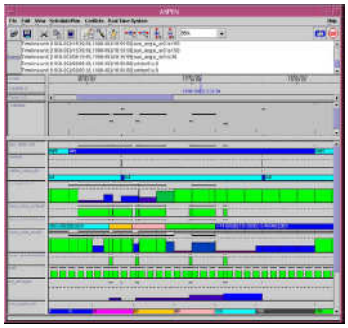

Figure 4 CASPER GUI for OASIS

Planning and scheduling capabilities for OASIS are provided by the CASPER continuous planning system. Based on an input set of science goals and the rover's current state, CASPER generates a sequence of activities that satisfies the goals while obeying relevant resource, state and temporal constraints, as well as operation (or flight) rules. 1Plans are produced using an iterative repair algorithm that classifies plan conflicts and resolves them individually by performing one or more plan modifications. CASPER also monitors current rover state and the execution status of plan activities. As this information is acquired, CASPER updates future-plan projections. Based on this new information, new conflicts and/or opportunities may arise, requiring the planner to re-plan in order to accommodate the unexpected events. An example of a plan in the CASPER GUI that was executed during a rover demonstration is shown in Figure 4.

Aerobot

JPL is developing onboard planning and execution technologies to provide robust and opportunistic mission operations for a future Titan aerobot (Huntsberger 2009). Aerobot have the potential for collecting a vast amount of high priority science data. However, to be effective, an aerobot must address several challenges including communication constraints, extended periods without contact with Earth, uncertain and changing environmental conditions, maneuvarability constraints and potentially short-lived science opportunities. We are developing the AerOASIS system (Figure 5) to develop and test technology to support autonomous science operations for a future Titan Aerobot. The planning and execution component of AerOASIS is able to generate mission operations plans that achieve science and engineering objectives while respecting mission and resource constraints as well as adapting the plan to respond to new science opportunities. This technology leverages prior work on the OASIS system for autonomous rover exploration.

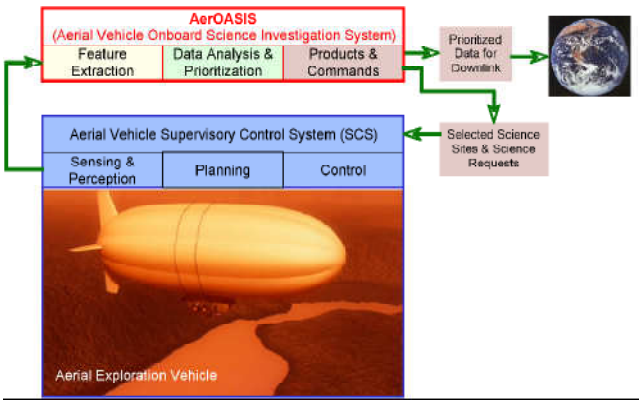

Figure 5 AerOASIS

Figure 6 JPL Aerobot in Flight Demonstration

Our objective is to enable onboard planning software to generate correct and high quality operations plans to achieve mission objectives issued from ground operations as well as respond to science opportunities detected onboard the vehicle. In particular, the system considers prioritized observation requests. This will enable the ground team to uplink a larger set of observations and let the aerobot dynamically select among them based on the scientific and engineering merit of the resulting plan and the aerobot's assessment of available resources. During execution, the aerobot will modify the plan based on the current estimate of its resources.

CASPER is used for the planning and scheduling for AerOASIS. CASPER employs its continuous planning where the planner continually evaluates the current plan and modifies it when necessary based on new state and resource information. Rather than consider planning a batch process, where planning is performed once for a certain time period and set of goals, the planner has a current goal set, a current aerobot state, and state projections into the future for that plan.

At any time an incremental update to the goals or current state may update the current plan. This update may be an unexpected event (such as a new science target) or a current reading for a particular resource level (such as battery charge). The planner is then responsible for maintaining a plan consistent with the most current information.

A plan consists of a set of grounded (i.e., time-tagged) activities that represent different aerobot actions and behaviors. Aerobot state in CASPER is modeled by a set of plan timelines, which contain information on states, such as aerobot position, and resources, such as energy. Timelines are calculated by reasoning about activity effects and represent the past, current and expected state of the aerobot over time. As time progresses, the actual state of the aerobot drifts from the state expected by the timelines, reflecting changes in the world. If an update results in a problem, such as an activity consuming more memory than expected and thereby over-subscribing RAM, CASPER re-plans, using iterative repair, to address conflicts.

CASPER includes an optimization framework for reasoning about soft constraints such as reducing the distance traversed by the aerobot and increasing the value of science data collected. User-defined preferences are used to compute plan quality based on how well the plan satisfies these constraints. Optimization proceeds similar to iterative repair. For each preference, an optimization heuristic generates modifications that could potentially improve the plan score.

CASPER's optimization framework supports a wide-range of user-defined preferences. The main loop of the algorithm interleaves iterative repair and iterative optimization to search for a conflict-free plan of high quality. The loop begins by processing any updates on state and resource timelines or on activity status. It then enters a loop in which it attempts to improve the plan by repairing conflicts or performing optimization steps.

UAVSAR

JPL is developing an onboard autonomy software package to integrate a Synthetic Aperture Radar (SAR) payload on an Uninhabited Aerial Vehicle (UAV) into an Earth observing sensorweb (Lou 2008). In the near term, tests are being conducted with a SAR instrument flying on a Gulfstream Jet. The end goal is to develop and demonstrate autonomy software that would enable a UAVSAR to (a) acquire data as directed by other nodes of a sensor network (e.g be tasked as a node in the sensorweb) and (b) based on its own data acquisition, the UAVSAR would make requests of other sensorweb assets, and (c) the UAVSAR also represents an autonomous sensorweb node that may autonomously respond to changing events, goals, and conditions.

Specifically, we are working towards demonstration scenarios where:

1. The UAVSAR acquires SAR imagery of an ongoing forest fire.
2. This imagery is processed to develop an updated fuel map and integrated with wind and fuel estimation to derive new areas for observation (e.g. locations to which the fire is likely to spread).
3. The UAVSAR and other assets (space, ground, and air) are automatically tasked to gather additional data of these areas.

In such scenarios the UAVSAR is an autonomous node interacting with other nodes in the sensorweb to achieve the overall sensorweb goals of tracking the forest fire.

The CASPER contribution to the UAVSAR automation follows a similar pattern to that of the EO1 automation.

Sea Gliders

CASPER uses a formal model of glider state and resources to validate the activity sequence and optimize schedules. The model tracks depletable resources such as battery energy, availability of the vehicle helm and communications equipment, and vehicle state variables such as the health of the system and instruments. A timeline-based user interface allows operators to alter any aspect of the plan and immediately see its impact on the mission. (Figure 7). Any resource conflicts that appear (for example, a depleted battery) will be flagged at this stage. The operator can respond by reparameterizing the existing paths with alternative behaviors, or returning to the cartographic interface to edit plan waypoint locations. During execution, CASPER activates, manages and concludes each of the activities in the sequence, decomposing them into parameters for the low-level behaviors.

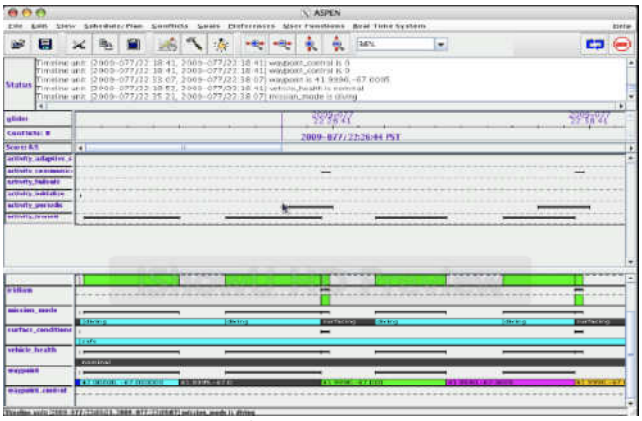

Figure 7 CASPER GUI for Sea Gliders

Acknowledgements

As with any survey paper, we stand on the shoulders of the constituents of the survey. We acknowledge the contributions of all of the authors of the papers surveys.

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

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

R. Castano, T. Estlin, R. Anderson, D. Gaines, A. Castano, B. Bornstein, C. Chouinard, M. Judd, "OASIS: Onboard Autonomous Science Investigation System for Opportunistic Rover Science." *Journal of Field Robotics*, v. 24, No. 5., May 2007, pp. 379-397, 2007.

S. Chien, R. Knight, A. Stechert, R. Sherwood, G. Rabideau , "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling." *International Conference on Artificial Intelligence Planning Systems* (AIPS 2000). Breckenridge, CO. April 2000.

S. Chien, B. Engelhardt, R. Knight, G. Rabideau, R. Sherwood, E. Hansen, A. Ortiviz, C. Wilklow, S. Wichman "Onboard Autonomy on the Three Corner Sat Mission." *International Symposium on Artificial Intelligence, Robotics and Automation for Space* (i-SAIRAS 2001). Montreal, CA. June 2001.

S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, D. Mandl, S. Frye, B. Trout, S. Shulman, D. Boyer "Using Autonomy Flight Software to Improve Science Return on Earth Observing One." *Journal of Aerospace Computing, Information, and Communication*. April 2005.

D. Gaines, T. Estlin, S. Schaffer, and C. Chouinard, "Autonomous Planning and Execution for a Future Titan Aerobot." *Proc Intl Workshop on Planning & Scheduling for Space*, Pasadena, CA, July 2009.

T. Huntsberger, H. Aghazarian, A. Castano, G. Woodward, C. Padgett, D. Gaines and C. Buzzell. ``Intelligent Autonomy for Unmanned Sea Surface and Underwater Vehicles." *AUVSI's Unmanned Systems* North America 2009. Washington DC, August 10-13, 2009.

Y. Lou, S. Chien, D. Clark, J. Doubleday, R. Muellerschoen, S. Saatchi, D. Tran, Y. Zheng, "A Forest Fire Sensor Web Concept With UAVSAR." *Proceedings IEEE International Geoscience & Remote Sensing Symposium*, Boston, MA, July 2008.

D. Thompson, S. Chien, Y. Chao, P. Li, M. Arrott, M. Meisinger, A. Balasuriya, S. Petillo, O. Schofield, "Glider Mission Planning in a Dynamic Ocean Sensorweb." *SPARK Workshop on Scheduling and Planning Applications*, Intl Conference on Automated Planning and Scheduling, Thessaloniki, Greece, September 2009.