COORDINATED SCIENCE CAMPAIGN SCHEDULING FOR SENSOR WEBS

Robert Morris*, Jennifer Dungan*, Will Edgington, Jenny Williams***, Jean Carlson***, Damian Fleming***, Terri Wood****, and Neil Yorke-Smith*****

**QSS Group Inc
*NASA Ames Research Center, MS 269-1, Moffett Field, CA 94035
*** Computer Sciences Corporation
**** NASA Goddard Space Flight Center
*****Artificial Intelligence Center, SRI International, Menlo Park, CA 94025, USA

ABSTRACT

Future Earth observing missions will study different aspects and interacting pieces of the Earth's eco-system. Scientists are designing increasingly complex, interdisciplinary campaigns to exploit the diverse capabilities of multiple Earth sensing assets. In addition, spacecraft platforms are being configured into clusters, trains, or other distributed organizations in order to improve either the quality or the coverage of observations. These simultaneous advances in the design of science campaigns and in the missions that will provide the sensing resources to support them offer new challenges in the coordination of data and operations that are not addressed by current practice. For example, the scheduling of scientific observations for satellites in low Earth orbit is currently conducted independently by each mission operations center. An absence of an information infrastructure to enable the scheduling of coordinated observations involving multiple sensors makes it difficult to execute campaigns involving multiple assets. This paper proposes a software architecture and describes a prototype system called DESOPS (Distributed Earth Science Observation Planning and Scheduling) that will address this deficiency.

Key words: Observation scheduling, remote sensing.

1. INTRODUCTION

NASA's Earth Science vision emphasizes the importance of establishing a tighter link among Earth Science models, data analysis, and observational activities at all relevant spatial and temporal scales. To enable such a tight linkage, there needs to be an associated information infrastructure binding the cycle of observation, on-board data handling and computing, transmission to ground, storage, data mining and product distribution to support activities such as inverse modeling, data assimilation and model evaluation. The cyclical nature of the linkage implies that new observation goals will emerge out of the products generated from previous observations.

Future remote sensing environment will consist of large numbers of networked sensors that are frequency-agile and capable of multi-scene observations from different space vantage points. Data acquired from such platforms will be merged with those acquired by more traditional systematic missions (such as Landsat). Second, for the purpose of validation and model robustness, data acquired by other observational platforms, including suborbital measurements using ground-, airborne-, and balloon sensors, will be merged with data from remote sensing platforms to form a sensor-web. Furthermore, the focus will be on the development of complex compositional Earth Science models, wherein focused process models combine iteratively to form interactive multi-component models that simulate the coupled behavior of two or more Earth system components. A complete multi-component model of the Earth is considered the holy grail of Earth Science research.

Consequently, Earth scientists will require data from multiple sources distributed in space, over significant periods of time, with choices available to the users of the data with respect to when, where and how these data will be acquired. Planning and executing a series of observations will benefit from information technology that provides an interface and set of automated tools for accessing the sensing resources available to meet observation goals, in a way analogous to the way that web-based archive data retrieval tools such as EOSDIS provide an interface for retrieving data that has been acquired in the past.

This paper provides an overview of a set of capabilities for addressing the need for coordination of observations. The system is based on a methodology called model-based observing. By *Model-based observing* is meant here the process of allocating and scheduling sensing resources based on the goal of validating a specific hypothesis derived from an Earth science model. Model-based observing allows observation scheduling to be *campaign-driven*, where a campaign is defined as a systematic set of activities undertaken to meet a particular science objective. Campaign goals require the collection of data on several variables, on different observing resources at different times and potentially at varying locations.

In the following sections we first present the overall architecture for model-based observing that links the Earth Science community to observation resources. Part of the architecture forms the set of capabilities for coordinating observations, which is the focus of the remainder of the paper. These capabilities are organized into a set of components of a system, called DESOPS (Distributed Earth Science Operations Planning and Scheduling System). DESOPS solves a constraint optimization problem as well as a schedule execution, monitoring and replanning problem, all of which are discussed here.

2. ARCHITECTURE FOR MODEL-BASED OBSERVING

Model-based observing requires coordinating the assignments of observation tasks among a collection of remote sensors or sub-orbital platforms such as ground-, airborne-, and balloon sensors, possibly configured into an organization (e,g, a train or a sensor web) (4). It is assumed here that each sensing or satellite resource has a distinct, geographically separated, operations team for managing the daily activities of each sensor. Using an economic metaphor, the interests and objectives of these "resource owners" are potentially different from those of the consumers; in particular, the users want maximum utility of the data received associated with their specific science goals, whereas the resource owners have other, potentially conflicting goals. In this regard, the operation environment for model-based observing offer challenges similar to those potentially solved by so-called *computational grid* systems (9), namely the need to provide visibility and access to a set of resources while maintaining the security and autonomy of operations for each.

A system for coordinating observations provides an added layer between the users of sensing resources and mission operations planners. Part of this coordination layer consists of software tools that allow consumers and providers to express requirements for facilitating the successful completion of observation goals. Resource users need to specify a set of measurements as well as a utility model for the data to be acquired. They need to be able to specify constraints on cost and completion time for their campaign goals. They need a mechanism to act as a broker to identify available resources and dynamically submit requests to schedule observations on them. The tool should monitor the execution of these requests and adapt to uncertainties in the availability of resources during execution, which potentially involves rescheduling observations on the same or different resources. Resource owners need a flexible means to specify constraints on the utilization of the resource, as well as a way to continuously supply updated statistics on current load and capacity. They need a system that will facilitate improved utilization of their resource without interrupting normal mission operations.

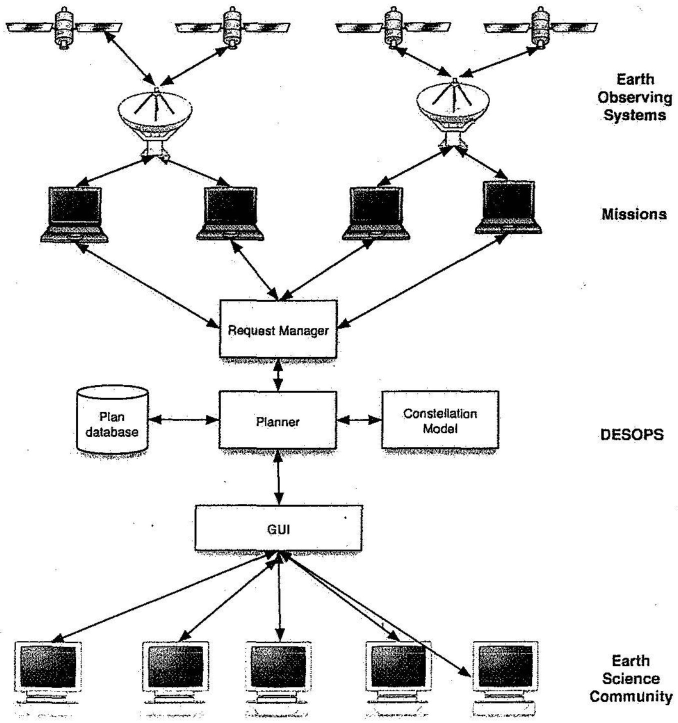

Figure 1. Architecture for Model-based Observing

The overall architecture,is displayed in Figure 1. The remainder of this paper discusses the layer marked DESOPS (Distributed Earth Science Observation Planning and Scheduling).

3. DESOPS ARCHITECTURE

The set of system components in Figure 1 labeled DESOPS consists of part of the information infrastructure for constructing and executing campaign plans involving a collection of sensors, and enables more direct contact between Earth Scientists and the mission planning process. The next sections describe these component capabilities in more detail.

3.1. Constellation Model

A **constellation model** consists of a database and set of functions for defining the capabilities and dynamics of resources available to the user for observation. There are five components to a constellation model:

1. a description of the capabilities of a collection of geophysical *sensors*;
2. a model of *time*. For the purposes of coordinated campaign, time can be viewed as a finite set of totally ordered values naturally interpreted as the set of days in which some observation can be taken or some other event of interest happens;
3. a *global notation* for data, enabling a user to inquire about satellite imagery over any portion of the world by specifying the location of the data of interest. Examples of global notation systems are the Worldwide Reference System (WRS) (6), or latitude-longitude.
4. a *satellite orbit* function for determining the set of sensor viewing times for a specified region of interest; and
5. for each resource, a *mission model* that describes constraints on the process by which tasks on the sensor are scheduled by the mission that manages it.

Collectively the constellation model provides a language for specifying the requirements for using a collection of sensing resources.

3.2. Graphical User Interface

Users define and revise campaigns over time through a **graphical user interface**. Users define a campaign to consist of a set of *measurements*; an (optional) set of *exogenous events* (such as a fire or volcano); and a set of *constraints*, restrictions the way a campaign can be carried out. Constraints are described more fully below.

The main screen of the interface is displayed in Figure 2. This screen shows a map for specifying regions of interest for a campaign, a flexible plan (defined in more detail below) and a textual representation of a campaign as a hierarchy of measurements and constraints. The view path swaths (defined below) for one of the requested satellites has also been computed automatically and is visually displayed.

3.3. Planner

The role of the **Planner** is to build and manage flexible plans. First, the planner constructs an initial *flexible plan* based on user inputs. Second, new constraints are added by propagating the effects of the initial set of constraints. In particular, the planner generates start times for each sensor in the domain of each measurement from *view paths* over specified regions of interest during specified time windows. A view path is the intersection of a specified region of interest with the path followed by a satellite over the user-specified time window. In DESOPS, view paths are generated by conducting a web search for this data from mission web sites. Alternatively, it is possible to generate this data directly through the use of simulators such as STK (Satellite Tool Kit). Converging on a flexible plan is an iterative process in which the user is allowed to view and revise the inputs to the problem.

3.4. Request Manager

An *observation request* is a specific assignment of a sensor, a time, and a location to the measurement. A *feasible observation schedule* is a sequence of observation requests that satisfy the user specified constraints. In general, a flexible plan gives rise to a number of feasible observation schedules. The **Request Manager** incrementally executes a feasible observation schedule by submitting observation requests to missions. The Request Manager also *monitors* the state of the executing plan, and initiates *rescheduling* activities where necessary. To carry out these functions the Request Manager implements an execution strategy (described in more detail below) for dealing with uncertainty in the execution environment and applies a state- transition model to monitor the progress of the plan.

3.5. Plan Database

The **Plan Database** stores all the current information about every campaign being managed in DESOPS. The database is used in all phases of campaign planning and execution, and contains

1. the definitions of all the measurements in the campaign;
2. constraint information
3. a description of the observation requests generated and submitted to missions; and
4. other information used by the DESOPS components associated with the state of a campaign.

4. CAMPAIGN PLANNING AS A CONSTRAINT OPTIMIZATION PROBLEM

DESOPS' scheduling problem can be mapped into a *constraint optimization problem* (COP). A COP consists of a set of *variables*, each associated with a *domain* of possible assignments, and a set of *constraints* defined over the subset of the variables. Associated with each solution (complete set of assignments) is an *objective function* that evaluates the quality of each solution. In a *Partial Order COP* (PCOP) the objective function induces a partial ordering of the set of solutions based on quality. In a *multi-attribute* COP, the objective function can be decomposed into a collection of criteria that are combined in some

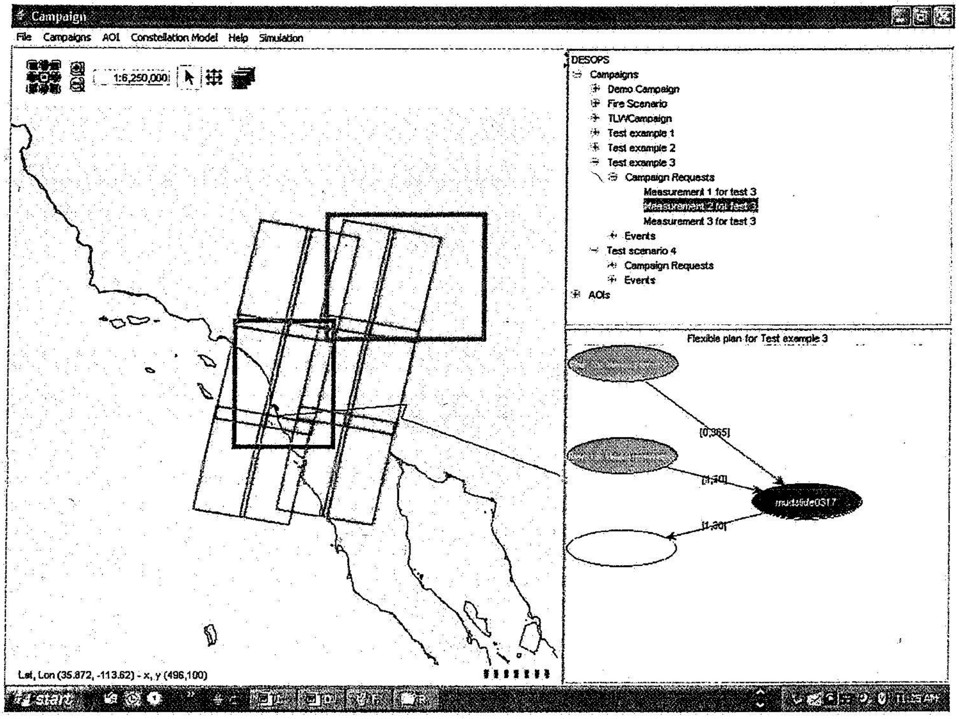

Figure 2. Graphical user interface for defining coordinated campaigns. The blue box represents the location constraint, green boxes represent view paths that satisfy that constraint. Also shown are an executing flexible campaign plan depicted as a network, and a tree representation of the plan objects and constraints.

manner to obtain the objective function value. In DESOPS, variables are associated with measurements and the domains will be triples of times, locations and sensors. In this section we formulate the problem and describe the approach to generating plans.

4.1. Campaign Constraint Definition

The constellation model provides domains of sensors S, locations L , and times T, and defines *location constraints* on $S \times L \times T$ that restrict possible values for measurement variables. Formally, we define a function $orb_{S_k} : GN \rightarrow T$, where GN is an area on the Earth described using some global notation, and $orb_{S_k}(r) \subseteq T$ is a set of viewing times for a specified location r on a sensor $S_k \in S$. Each sensor $S_i \in S$ is associated with a cost $cost(S_i)$ for using it to take an image.

Each campaign consists of a set of **measurements** $\{M_i\}$, a set $\{E_k\}$ of **exogenous events**, and a set of **constraints.** Each measurement M_i is associated with a tuple of **request variables** $s_{M_i}, t_{M_i}, l_{M_i}$, where the domain of s_{M_i} $(dom(s_{M_i}))$ is a subset of S, the domain of t_{M_i} $(dom(t_{M_i}))$ is a subset of T, and $dom(l_{M_i})$ is a set of locations. Let $s_{var} = \{s_{M_i} : 1 \leq i \leq n\}$; define $t_{var} and l_{var}$ similarly, and let $m_{var} = t_{var} \cup s_{var} \cup l_{var}$. The user may designate a subset of M to be "optional", meaning informally that if no observation is taken of the measurement variable, the campaign will still have value. A measurement that is not optional is *required.*

An exogenous event E_K can be specified as a pair of **parameters** $t^s_{E_k}, t^f_{E_k}$, of start and end times, with domain T. These variables can be viewed as discrete random variables with a probability distribution over T. For a given campaign M, let $v_M = m_{var} \cup e_{par}$.

The set C_M of user-specified constraints on a campaign M are defined over, and restrict the permitted values of, variables in m_{var}. There are six kinds of constraints:

1. For each $M_j \in M$, there are **sensor domain** constraints of the form $dom(s_{M_j}) = S_{M_j}, <_{s_{M_j}}$, where $S_{M_j} \subseteq S$, and $<_{s_{M_j}}$ specifies a preference ordering for elements in S_{M_j}; $S_k <_{s_{M_j}} S_p$ means "prefer acquiring data with S_k over S_p for measurement M_j".
2. A **temporal constraint** has one of two flavors. The first provides a means to coordinate measurements, or to constrain start times for measurements, and has the form $t_{x_j} - t_{x_i} \in \{T_i, \ldots, T_{i+m}\}, pref$, where $pref \in \{min, max, mid\}$ is an (optional) preference function for times in $\{T_i, \ldots, T_{i+m}\}$, and where $x_i, x_j \in m_{var} \cup e_{par}$. The other flavor of temporal constraint restricts the times of measurements with respect to exogenous events, and has the form $t_{x_j} - t_{x_i} \in \{T_i, \ldots, T_{i+m}\}, p(\mu, \sigma)$, where $p(\mu, \sigma)$ specifies a probability distribution over the

set of times, and where $x_i, x_j \in m_{var} \cup e_{par}$. Such a constraint expresses a prediction about when an exogenous event will occur.

3. A **region-of-interest** l_{M_i}, and an associated **time window** $[T_s, T_e], T_s, T_e, \in T$, are specified for each measurement M_i. A **location constraint** induced by the specified region-of-interest is of the form $dom(t_{M_i}) = \bigcup_{S \in S_{M_i}} orbs_S(l_{M_i})$, i.e. the domain of t_{M_i} is restricted to the set of times for which there is a sensor $S \in S_{M_i}$ available to take the observation of the region-of-interest l_{M_i}

This core set of constraints can be expanded to include a *budget constraint*, a restriction on the overall cost $\Sigma cost(M_i)$ of the campaign, and a *cloud cover constraint* which restricts the amount of acceptable cloud cover associated with the image.

A multi-attribute objective function for evaluating solutions to the DESOPS COP is constructed from a set of user-specified criteria for good solutions. These criteria include

1. World feasible: intuitively, a solution is world feasible if it satisfies the constraints and is consistent with the expected behavior of the exogenous events.
2. Minimal cost: The cost of a solution is the sum of the cost of the sensors used on each measurement.
3. Temporally preferred: preference for solutions that maximize the overall preferences for times.
4. Resource preferred: preference for solutions that maximize the overall preferences for sensors.

Given a complete assignment to the variables in m_{var}, the value of this assignment with respect to each of these criteria can be rigorously derived from techniques discussed elsewhere, e.g. (3), and (2).

4.2. Campaign Generation

The process of converging on an high quality campaign is an iterative process performed jointly by the human user, the DESOPS planner and request manager. There are three main steps in the process. First, a minimal flexible plan is constructed. A flexible plan for this problem can be represented as an augmented Simple Temporal Network (STN) (8) consisting of nodes representing events or measurements, and directed arcs labeled by constraint information. The STN is augmented by a representation of temporal preference, uncontrollable events and associated probabilities associated with the time intervals, as defined in (3), (12), and (2) (the augmented STN represents a *Simple Temporal Problem with Preferences and Probabilities*). An example is found in Figure 3. The plan consists of three measurements and one event. The constraint $[40, 100]$ represents the belief that event $E1$ is expected to happen sometime beteen day 40 and day 100 of the campaign. The other constraints represent temporal ordering constraints; for example, the label between $M1$ and $E1$ expresses the constraint that $M1$ should occur between 1 and 30 days before $E1$, with a preference for times as close to $E1$ as possible. The sensor constraints are also attached to each measurement in the plan. As illustrated in Figure 2, a flexible plan network is part of the visual displays available to the user during campaign definition. The same display is used during campaign execution to provide the user with status information about the plan. For example, in the figure, the blue nodes indicate measurements that have been acquired, and the yellow node represents an exogenous event that has yet to occur.

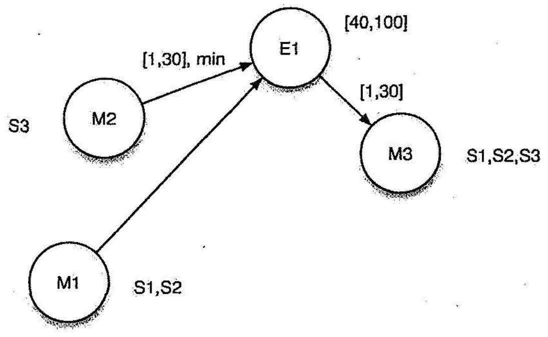

Figure 3. A simple Flexible Plan. Each node of the network represents a measurement or event. Directed arcs depict temporal orderings, labeled with the duration between event(s) and measurements(s) The sensors are listed with each measurement.

Second, from the initial network, new temporal constraints are added by propagating the effects of initial set of temporal orderings. Shortest path algorithms (10) can be applied to generate the *minimal network*, a set of constraints that precisely describe the set of solutions to the initial problem. In addition, the set of temporally optimal solutions can be derived by techniques discussed in (3).

Third, *request sequences* of observations are created to submit to the missions. To build a single request sequence, it must be decided

1. Which M_i to select next (ordering of measurements);
2. Which admissible sensor S_j to associate with M_i;
3. Which admissible time T_k to assign to the observation request, based on the selection of (2).

Specified user preferences guide these decisions, and the user is allowed to interactively revise the set of observation requests to consider. Specifically, the system generates the set of candidate observation requests for each measurement; the user is allowed to delete from this candidate list.

4.3. Example

Here, we present a hypothetical campaign based on a science goal to test an emissions model predicting the aerosols released by wildfires. Data on several measurement variables must be gathered in order to accomplish the analysis. In particular, vegetation type or biomass, atmospheric aerosol concentration and burned area are needed for the region. Fuel moisture content is a variable that also would be useful for the objectives of the science, though not a necessity. Sensors that provide products at various spatial resolutions relevant to these variables include Landsat Enhanced Thematic Mapper+ (ETM+) or Thematic Mapper (TM) can be used for mapping vegetation type. Optimal timing for acquiring Landsat data for this purpose would be June or July in the same year that the fires burned, when forested land can most easily be spectrally distinguished from grassland. For mapping aerosol concentration, images coincident to burning must be obtained. Moderate Resolution Imaging Spectroradiometer (MODIS) on the Terra and/or the Aqua satellites would provide data for this variable. MODIS data from either platform could also be used to provide coarse spatial resolution burned area after (though not too long after) the fires were out. For mapping vegetation moisture content, hyperspectral data from EO-1 Hyperion instrument are relevant. The most useful data for this purpose would need to be acquired just preceeding the fire. We assume the day the campaign is constructed is Day 0 and the campaign window is until Day 122.

- *Constellation Model*
  - Locations and times are as defined above
  - Sensors

    $S = \{ETM^+, TM,\ Hyperion,\ MODIS\}$

    with associated orbit functions $orb_{s \in S}$ and the following cost functions: $cost(ETM^+) = 1500$,$cost(TM) = 500$, $cost(Hyperion) = 2000$,$cost(MODIS) = 1000$.
- *Problem Inputs*
  - Variables: A set M of measurement variables $aero$, $moist$, veg representing aerosol concentration, moisture content, and amount of vegetation in burned area. $moist$ is designated as optional. A single exogenous event, $fire$.
  - Sensor constraints and preferences:
    $s_{veg} \in \{ETM^+, TM\}, ETM^+ <_{aero} TM,$
    $s_{moist} \in \{Hyperion\},$
    $s_{aero} \in \{MODIS\}.$
  - Temporal Ordering Constraints:
    $t_{veg} - T_0 \in \{31, 32, \ldots, 92\},$
    $t^s_{fire} - t_{veg} \in \{0, 1, \ldots, 14\}, min,$

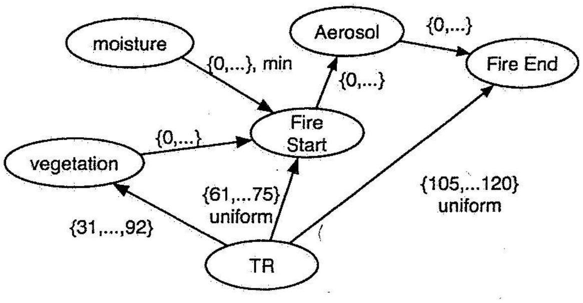

Figure 4. **Network representation of inputs to the fire example**

    $t_{aero} - t^s_{fire} \geq 0,$
    $t^e_{fire} - t_{aero} \geq 0$
    $t^s_{fire} - t_{moist} \geq 0, min,$
    $t^e_{fire} - T_0 \in \{105, 106, \ldots, 120\}, uniform,$
    $t^s_{fire} - T_0 \in \{61, 62, \ldots, 75\}, uniform$ [1].
  - Regions-of-interest $l_{aero}, l_{moist}, l_{veg}$, each with time window $[0, 122]$.
  - Budget constraint: $cost(M) \leq 2000$.
- *Flexible Plan for Campaign.* The flexible plan for the original constraints (before propagation) is found in Figure 4. There are nodes for each measurement type and for the start and end of the fire. There is a reference node (representing time 0) and directed labeled arcs for each constraint. As is customary, the interpretation of an arc between A and B is that the temporal gap $B - A$ is constrained to have a value in the set of values displayed on the edge label.

  Assume the constellation model generates the following set of values:
  $orb_{ETM^+}(l_{veg}) = \{3, 19, 35, 51, 67, 83, 99, 115\};$
  $orb_{TM}(l_{veg}) = \{5, 21, 37, 53, 69, 85, 101, 117\};$
  $orb_{MODIS}(l_{aero}) = \{36, 52, 68, 84, 100, 116\};$
  $orb_{Hyperion}(l_{moist}) = \{0, 30, 60, 90, 120\}.$
  Then the possible observation times can be represented by the following location constraints:
  $dom(t_{veg}) = orb_{ETM^+}(r_{aero}) \cup orb_{TM}(r_{aero}) = \{3, 19, 35, 51, 67, 83, 99, 37, 53, 69, 85\}$
  $dom(t_{moist}) = orb_{Hyperion}(r_{moist}) = \{0, 30, 60, 90, 120\}$
  $dom(t_{aero}) = orb_{MODIS}(r_{aero}) = \{36, 52, 68, 84, 100, 116\}.$

  The domain constraints, coupled with the user-specified ordering constraints, allow for further domain refinements or new ordering constraints. For example, from $t_{veg} - T_0 \in \{31, 32, \ldots, 92\}$ and the domain constraint above, we can derive the new domain constraint by intersection of the two sets, resulting in $t_{veg} \in \{35, 51, 67, 83, 99, 5, 21, 37, 53, 69, 85\}$.

[1] For simplicity, we represent the fire start and fire end as a range of expected values in a uniform distribution.

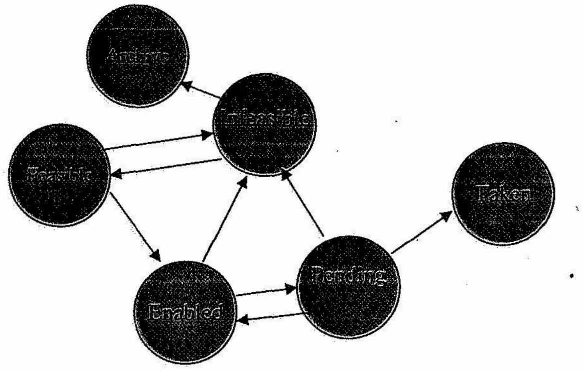

Figure 5. A state transition model for measurements. States and possible transitions between them are depicted.

5. EXECUTING A CAMPAIGN REQUEST SEQUENCE

Executing a campaign requires formulating requests and submitting them to missions, monitoring the progress of a campaign over time, and initiating rescheduling actions as needed in response to unexpected events. The Request Manager's behavior is constrained by an *execution strategy* on when and how requests for measurements are submitted to missions. An execution strategy is part of a mission model. The mission model informs the Request Manager on matters related to which mission is most likely to be able to fulfill a request, as well as how and when to submit the request. For example, the mission model may contain a *load profile* for each sensor, which indicates the percentage of time the sensor has been idle during a specified period. The Request Manager may apply this information by preferring sensors with a smaller load. Second, a mission model contains formatting rules for request submssion. Third, a mission model contains deadlines for submitting requests based on the mission-scheduling process.

The Request Manager monitors a campaign by implementing a *state transition model*, which identifies possible states of the overall campaign, the component measurements defined for the campaign, and, for each measurement, the state of each associated observation request and legal transitions between states. The Request Manager observes whether enabling conditions for a transition are met, and, if they are, records the change in state. The state transition model also allows the Request Manager to detect when a campaign has failed during execution, which triggers a suspension of the campaign and notification to the user for rescheduling purposes.

Figure 5 shows a state transitions for a measurement. A measurement starts in a feasible state. It becomes enabled when the temporal preconditions for taking the measurement are met (for example, an exogenous event happens or a dependent measurement has been acquired). It becomes infeasible if the constraints make it impossible for it to be taken; this can happen, for example, if all submissions of requests for the measurement are rejected. Otherwise, a measurement is pending if at least one request for the measurement has been submitted. If a mission accepts the request and the image is acquired, the measurement enters the terminal node *Taken*. The user may decide during execution to use data in an archive to acquire the needed data. If so, the Request Manager no longer submits requests for the observation to the missions.

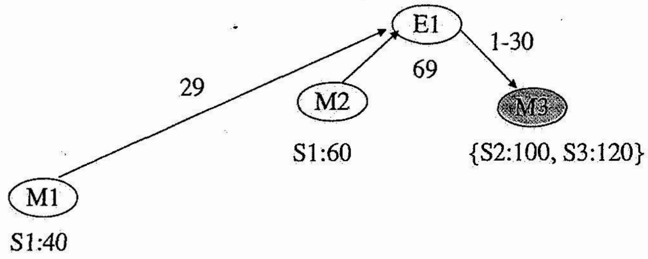

Figure 6. A replanning scenario. The occurrence of $E1$ at time 69 has made it impossible to schedule an observation of $M3$ that satisfies the constraint between $E1$ and $M3$. The user must decide whether to relax the constraints on the plan to restore its feasibility.

5.1. Replanning

As the campaign plan executes, missions may fail to accept requests or exogenous events may happen, or fail to happen, at unexpected times. As a result, a campaign may become infeasible. If a mission rejects a request and there are alternative viewing opportunities, the Request Manager will automatically resubmit requests. If for any reason, the set of viewing opportunities for a measurement is empty during campaign execution, the human user decides whether to restore feasibility to the plan or to abort it.

Plans are restored to feasibility by relaxing constraints. Figure 6 shows a simple plan that was made infeasible during execution. Exogenous event $E1$ happened at time 69. A constraint requires measurement $M3$, which has yet to occur, happen between 1 and 30 days after $E1$. $M3$ has two observation opportunities: with sensor $S2$ at time 100, or with sensor $S3$ at time 120. Clearly, both exceed the upper bound on the temporal ordering constraint, and so this constraint is violated. The user may relax the upper bound of the temporal constraint to make the observation opportunities consistent with the plan. Alternatively, the user may add additional sensors for $M3$ that include opportunities consistent with the ordering constraint, or may decide to acquire $M3$ data through an archive.

DESOPS provides the user continuous plan execution status. It also provides notification of the need for plan repair when the plan becomes infeasible during execution. Visual and textual information will be provided by

DESOPS' explanation facility, using a model to map plan state information into useful textual or visual advice.

DESOPS is implemented in C++ and Java. The implementation is built upon previous work on the AMPS/MOPSS system and the EUROPA constraint-based planning system (7). An end-to-end prototype with the capabilities described in this paper is currently being tested and evaluated.

6. DISCUSSION

We view observation scheduling, the core capability of DESOPS, is one part of a broader vision for managing sensing resources for Earth Science. We view the ultimate goal of developing automated tools for campaign management to be the efficient and effective acquisition of data products, not simply the scheduling of sensing resources. Many sensing resources are not "controllable" in the sense that they are designed to continuously acquire images. For such a mixed environment of controllable and uncontrollable sensors, the overall planning problem requires the joint management and scheduling of remote sensing as well as data archive services. We are currently extending the DESOPS architecture to integrate observation scheduling with planning for data analysis as discussed in (5), which would lead to an end-to-end planning system for generating data products. Such systems could also plan for "virtual sensing" activities, as discussed in (11).

Second, we are exploring the integration of Earth Science domain models into DESOPS planning. This would enable the system to provide more robust advice to a user in formulating campaigns. For example, such models could advise users on the detailed selection of promising regions-of-interest for developing a fire campaign. Finally, we are working on devising a means for providing a "feedback loop" from the results of data analysis to the formulation of new campaign goals for DESOPS. Each of these enhancements contribute to the overall vision, as described in (1), to forge a tighter link between modeling, observing, and data analysis.

7. CONCLUSION

This paper has described a set of capabilities for building and executing sequences of observations for accomplishing complex campaign goals. Observation requests generated from user inputs describing campaign goals and constraints are submitted electronically to mission operations planners, who then decide whether and how to incorporate the request into future mission schedules. The system also supports dynamic replanning in response to request rejection or unexpected changes in the observing environment. The overall approach to distributed planning has the advantage of allowing missions to maintain ultimate control over their instruments while at the same time allowing Earth scientists more visibility into the resources available for accomplishing their science objectives.

ACKNOWLEDGMENT

Thanks to Jeremy Frank, Paul Morris and Joseph Coughlan for comments on previous drafts. Also to Lina temp-Pref, Christopher Bruno and Andrew Bachman for past contributions of ideas and for their development efforts.

REFERENCES

[1] *NASA Earth Science Enterprise Strategy*, 2003.

[2] Morris, R., et. al., Temporal Planning with Preferences and Probabilities, Proceedings of the ICAPS Workshop on Constraint Programming for Planning and Scheduling, 2005.

[3] Morris, P., et. al, Strategies for Global Optimization of Temporal Preferences, Proceedings of CP 2004: 408-422.

[4] Clement, B. and A. Barrett, Coordination Challenges for Autonomous Spacecraft, AAMAS-02 Workshop Notes on Toward an Application Science: MAS Problem Spaces and Their Implications to Achieving Globally Coherent Behavior, 2002.

[5] Golden, K. et. al., Automating the processing of Earth observation data, Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation for Space (i-SAIRAS), 2003.

[6] The Worldwide Reference System. http://landsat.gsfc.nasa.gov/documentation/wrs.html

[7] Smith, D., et. al, Bridging the Gap Between Planning and Scheduling, Knowledge Engineering Review, 15:1, 2000.

[8] Dechter, R. et. al., Temporal Constraint Networks, Artificial Intelligence, 49:61–95,1991.

[9] Buyya, R., et. al., The Grid Economy, Special Issue on Grid Computing, Proceedings of the IEEE, Manish Parashar and Craig Lee (editors), Volume 93, Issue 3, 698-714pp, IEEE Press, New York, USA, March 2005.

[10] Cormen. T. H. et. al., Introduction to Algorithms, MIT Press,1990.

[11] Srivastava, A. et. al., Mixture Density Mercer Kernels: A Method to Learn Kernels Directly from Data, SIAM International Conference on Data Mining, 2004.

[12] Morris, P. and N. Muscettola, Execution of Temporal Plans with Uncertainty, Proceedings of AAAI-2000, 491–496, 2000.