Software Agent-based Framework Supporting Autonomous and Collaborative Sensor Utilization (AAMSRT)

Renato Levy
Wei Chen
Margaret Lyell
Intelligent Automation, Inc.
15400 Calhoun Drive, Suite 400
Rockville, MD 20855, USA
+1-301-294-5200
rlevy@i-a-i.com
wchen@i-a-i.com
mlyell@i-a-i.com

ABSTRACT

Information about the world and its local environments prevails overwhelmingly due to the rapidly growing availability and heterogeneity of sensors and sensor platforms (e.g., various advanced imagers installed on different Low Earth Orbit (LEO) satellites providing observations of an event from different views). The next logical step is to synergize these assets in order to provide a more complete and coordinated view of user interested events. Currently, users that require sensor measurements have to be able to determine which sensor platforms and sensor capabilities should be used to fulfill their observation requests. This condition incurs the dependency of an expert to intermediate and plan the observation, significantly reducing the utility of an already large and growing sensor capability.

Our approach was to develop an intermediary step that translates user's domain specific requirements into domain independent observation requests. The resultant domain-independent formulation may be viewed as coordinated service planning and execution employing heterogeneous sensor assets. In a sense, we have proposed a commoditization of sensor capabilities as well as a translation layer that determines the commodity services required to satisfy a domain specific request.

In addition, we have implemented an agent-based autonomous multiple sensor re-targeting framework (AAMSRT) that supports open, heterogeneous and dynamic environments to negotiate and track the execution of these services. This innovative approach involves the abstraction of sensor assets as software agents that encapsulate sensor services allowing for easy coordination between multiple providers, contractual support and business rules using a market approach. The tool provides a web-based interface to allow a user to formulate a service request, as well as monitoring execution tracking and replanning capabilities.

In summary, our AAMSRT framework enables coordinated employment of heterogeneous sensor assets on a service architecture for building business applications, e.g., earth modeling and observation. This paper addresses the multi-sensor retarget problem at a high level and provides a complete solution.

Categories and Subject Descriptors

I.2.11 **[Artificial Intelligence]:** Distributed Artificial Intelligence - Coherence and coordination, Multiagent Systems

Cite as: Software Agent-based Framework Supporting Autonomous and Collaborative Sensor Utilization (AAMSRT), Renato Levy, Wei Chen, Margaret Lyell, *Proc. of 8th Int. Conf. on Autonomous Agents and Multi-agent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 93 – 100

General Terms: Design, Algorithm, Experimentation

Keywords

Applications, distributed systems, multi-agent planning

1. INTRODUCTION

Information about the world and its local environments is now plentiful due to the rapidly growing availability and heterogeneity of sensors and sensor platforms. For example, NASA's "Earth Observing System" for research into the Earth's biosphere, land, atmosphere and oceans has fielded a series of satellites equipped with sophisticated sensor packages; the weather mission of NOAA and the remote sensing needs of the National Geodetic Survey component of NOAA have resulted in a series of satellites containing instruments capable of imaging. Sensor platforms extend to Aircraft, such as those offered by NOAA's airborne hyperspectral imaging sensor. As the cost and size of sensors and processing elements decline, and with fixed sensor networks becoming smarter and requiring less power, both commercial concerns and private individuals are contributing to the explosion of deployed sensors. With the rapidly growing availability of deployed sensors, the next logical step is to synergize the sensor assets in order to provide a more complete 'picture' of events that are of interest to the users. This was the motivation for the 'Sensor Web' project [6]. *But what are the characteristics of the software that will support the heterogeneous, dynamic vision of 'Sensor Web'?* A user's request for earth observations may be highly complex, e.g., evaluating the potential correlation among the vegetation change at Amazon, the weather and the human activities, and may cause a long time and a continuous commitment – consequently, high cost – for a single platform (e.g., a LEO satellite) to fulfill the aforementioned observation request. Thus, it may be favorable to employ cheaper "time shares" from multiple sensors and platforms owned by different agencies and organizations to provide a more complete and coordinated view of user interested events. *But what is a viable strategy to identify and integrate the heterogeneous assets regardless the ownership issue from the users' point of view and how to best design and develop associated underlying mechanisms to achieve this objective?*

We developed an agent-based, autonomous multiple sensor re-targeting framework, AAMSRT, as a software infrastructure that supports the open, heterogeneous and dynamic environment by flexibly leveraging various types of sensors and sensor platforms. Our multi-agent based approach eminently suits the 'Sensor Web' environment, whose dynamics and heterogeneity arise from (1) changing numbers of user requests, (2) differing characteristics of user requests, (3) various operating conditions with respect to sensor platforms, i.e. UAVs are not always in operation, (4) different organizational contributors to the overall sensor web, and (5) organizational structure of available assets and limited disclosure of their commitments. Additionally, we aim to develop a user oriented system that delivers the observation results while eliminating the need for the user to understand the underlying characteristics of sensor assets and the technical reasoning and coordination processes.

Our innovation involves the abstraction of heterogeneous sensors and sensor platforms as multiple software agents, which are organized as service providers. These agents represent various natures and capabilities of sensor assets. The agents are capable of entering into negotiations for a targeting opportunity and presenting a locally optimized bid. Furthermore, the domain problem, which involves employing multiple heterogeneous sensor assets for complex observation requests, can be decomposed into areas of capability (which sensors and sensor platforms are able to provide the observation service), feasibility (planning – whether a suitable composition of observation tasks carried out by those sensor assets can deliver exactly what the user wants), and availability (scheduling – whether the sensors and platforms are available at the right times to carry out the requested observation tasks) issues of sensor retarget opportunities. Within the planning process, we acknowledge that other factors may play a large role, such as the quality (resolution) of measurement, the likelihood of success, etc. The abstraction of sensors and platforms modeled by software agents and the dynamic planning and scheduling capabilities required to achieve a user's request clearly dictate the advantage of using multi-agent technologies as a viable solution to address the aforementioned problems.

We will specify a general research problem and conduct a brief literature review in section 2, introduce the system architecture of the AAMSRT in section 3, discuss the implementation effort in section 4, compare the strength of our approach with related effort in section 5, and finally present conclusions and future work.

2. SPECIFICATION OF RESEARCH PROBLEM

The above introduction projects two general research problems: (1) Given that a user only cares about observations he/she desires and not about the actual underlying mechanism as to how the results were generated, provided certain constraints are satisfied, how we should translate the user requests into a form that allows an optimized solution for the problem. Knowledge engineering can be performed in order to capture the users' inputs in a user-centric way. For example, a scientist wants to evaluate the potential correlation among the vegetation change at Amazon and the weather and the human activities in the past five years; she/he does not – and should not – be labored with the details that how the satellite images are acquired by which sensors on which sensor platforms that may be operated by which agencies. (2) Given that a user request has been translated into a problem specification to the AAMSRT, how multiple sensor and platform assets should be coordinated to actually carry out the observation request? This generates a multi-agent coordination problem including planning, scheduling, re-plan, etc. For example, a scan of vegetation and a scan of earth atmosphere should be carried out with different kinds of sensors on potentially distinctive satellites; different assets may provide scans on only portions of Amazon specific to the tracks and status of the satellites and how to integrate these different scans as a whole to fulfill the user's request.

The aforementioned problems have been explored in different capacity by previous and ongoing approaches. Knowledge engineering for user centric requests that will involve sensor and platform management issue has been addressed by Tang, Lyell and Colombano [13], by providing flexibility and efficiency in utilizing remote sensors on satellites for earth monitoring. They also used a market-based contract approach that involved optimized bidding by agents that represented a satellite and its host of sensors.

In [10], Mullen *et. al.* shared our view that a user is only interested in the final results and not on the means in which it was obtained (saved specific constraints). Despite the view, the authors focused on using a market based approach to select resources to perform a domain specific task rather than formulating a domain independent decomposition that could leverage heterogeneous resources across multiple organizations.

Biancho *et. al.* [1] presented the Multi-Agent Ground-Operation Automation (MAGOA), an architecture that aggregated agents responsible for automating space operation planning and execution and identifies multi-satellite conflicting tracking periods and generated a control plan for each satellite.

For multi-agent coordination issue (especially on multi-agent planning and scheduling), many different approaches have been explored. One of the most successful task planning and scheduling effort is NASA's ASPEN (Automated Scheduling and Planning ENvironment) [5] and CASPER (Continuous Activity Scheduling Planning Execution and Replanning) [7], which have been in actual use for NASA missions. However, a user needs to be an expert on recognizing and managing the assets and their capabilities to make plans and schedules, which is not suitable to the problem we address.

Lesser *et. al.* [11] led an effort to develop combinatorial auctions for resource allocation in a distributed sensor network. In their work, a radar dish has a number of parameters that can be adjusted, such as what area it sweeps, the pulse rate, etc. The resource allocation problem in this domain is to decide at each time step (allocation cycle) what setting the sensor must have, taking into account the needs of various monitoring tasks. Farinelli et. al. [8] explored the problem of performing decentralized coordination of low-power embedded devices within many environmental sensing and surveillance applications; particularly, they represented the problem as a cyclic bipartite factor graph, composed of variable and function nodes (representing the agents' states and utilities respectively) and utilized a max-sum algorithm to maximize the social welfare within a group of interacting agents.

Collaborative planning [9] was proposed to coordinate multiple agents' activities by exploiting a revised and expanded version of *SharedPlans* to handle cases in which a single agent had only partial knowledge; but the task representation was still not expressive enough to analyze arbitrary coordination mechanisms especially in response to sensor asset management.

The modern approach to general reasoning about worth-oriented goals, contingencies, and uncertainties that have prevailed in distributed problem solving is the POMDP approach [2]. But POMDP does not apply to the situation of heterogeneous resources owned by different agencies.

Chen *et. al.* [3] believed that agent planning and scheduling behaviors are inextricably linked to coordination behaviors and proposed some extensions and restrictions to the expressiveness of traditional plan and schedule representations that allowed the formal definition of the multi-agent coordination problem to present a set of general rules relating task environment characteristics and implemented a set of GPGP coordination mechanisms [4], which provided the base ground to our approach; however, this approach did not consider the impact of individual resources' own schedules that may post significant constraints to the generated task plans.

Considering the pros and cons of the related approaches, we present a distributed integrated planning and scheduling effort with enhanced negotiation process to coordinate the heterogeneous sensor assets based on their own plans and schedules to fulfill a user's observation request without the need of his/hers intervention on any technical detail.

3. SYSTEM ARCHITECTURE

In the vision behind the AAMSRT framework, the set of resources available to perform an observation request is broken into several organizational entities known as agencies. This "breaking" of the resource set serves two purposes: First, it allows the construction of restrictions and policy controls available between real life entities (e.g., NASA, US Air Force, NATO). Second, it helps to avoid the exponential explosion of alternatives in building a plan to fulfill a request. Each of these agencies has at least one coordinator agent that is able to receive and interpret requests over a pre-defined service oriented architecture interface.

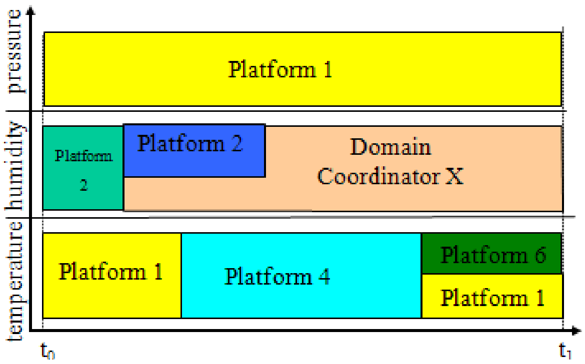

Figure 1. Composing a solution for a request.

When system users make a request through a web interface, the request is translated from its domain specific language into a generic resource allocation task. The advantage of this domain independency concept is in two-fold: It allows the coordinators to explore the easiest way to produce the results requested by the user (often there is more than one solution available). It also allows users to abstract details about sensors and underlying mechanisms to obtain those results, relying on the expertise contained in the coordinators. This "freedom" allows users that are not knowledgeable in sensor allocation and coordination to become new customers for sensor networks and the virtual pool of information that they can collect, effectively reducing the entry barrier for the technology.

A coordinator agent will then construct a plan (at least one, or maybe multiple plans) that can fulfill the request and access its own resources to determine which resources are available. The coordinator may also interact with other agencies to determine collaboration towards fulfilling the request. Based on the responses, the coordinator selects a plan and schedules the tasks using the available resources possibly across those agencies. Several levels of collaboration can be supported as indicated in Figure 1. In this example, an observation requested by the customer requires the measurement of 3 types of sensors: pressure, humidity, and temperature. Figure 1 shows that Platform 1 can support the readings of pressure measurements for the whole area of interest during the time of measurement. For temperature, Platform 1 will handle the measurements for the whole area during the initial period. After this time, Platform 4 will provide the measurements until finally Platform 6 helps Platform 1 in sharing the area of interest. In humidity, we introduce the concept of contracting another agency in support of the measurement by Platform 2 via Domain Coordinator X.

It is important to differentiate the proposed coordinator agent from a centralized scheduling system. The coordinator chooses the resource types that can fulfill the observation and queries all resources of the observation types under its knowledge to provide a bid. The actual resources in AMMSRT are self-reliant, in adjusting their schedules and providing a bid towards the service opportunity. Once all bids are in, the coordinator will pick the best options according to the agency's policies.

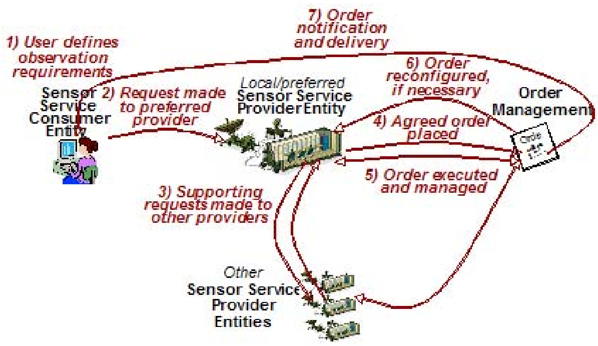

Figure 2. AAMSRT general interaction

Information flow and control flow are key to constructing complicated systems like AAMSRT. Figure 2 illustrates the high-level overview of the AAMSRT entities and their interactions. Here, the sensor-service consumer defines a user's observation requirements and submits them to their preferred sensor-service providers. The sensor service provider plays the role of agency coordinator. It determines the availability of its resources and requests additional/supplementary support from other sensor-service providers (1, and 2). The preferred provider then plans and schedules the services required for the consumer's request – resulting in an order being placed (3 and 4). The order is then executed, managed, and monitored by an order manager agent (5). Any problem that arises during the order will be detected – with the order possibly being reconfigured to complete the consumer's request (6). When the order is completed, the consumer is notified and the requested observation results are delivered to the user (7).

3.1 User Request

One of the critical aspects of the AAMSRT framework is to provide sensor services in a manner that frees the potential user from the necessity of understanding sensors and their platforms as well as limitations of their scheduling and operation. The user has access to AAMSRT, by a web-server that interacts with the user and represents him/her as a delegate for the rest of the system. This web-server support is responsible for providing a user-friendly interface. An observation request can be made from that interface; the user can also monitor the status of previous requests.

Besides describing the type of observation in which the customer is interested (forest fires, crop yield, etc…), a user must indicate the particular area in which this observation is requested and other relevant factors for the request, such as:

- Observation window: from date/time to date/time
- Maximum interval between initial and final reading
- Method of observation (if choice is presented)

In order to fulfill the user request, the service provider must embed the knowledge on how to perform the observation requested by the user, working as an automated sensor specialist (domain characterization rules).

Using the domain characterization rules provided by the service provider, a dedicated agent hosted at the web-server named sensor service consumer agent identifies and translates the request from the domain specific language from the user into a domain independent language that can then be farmed to service providers, determining the types of resources that are required to fulfill the user requests. This may include not only the sensor readings, but also recorded values of previous dates, processing and storage capabilities, and algorithms. This translation service is performed transparently to the user.

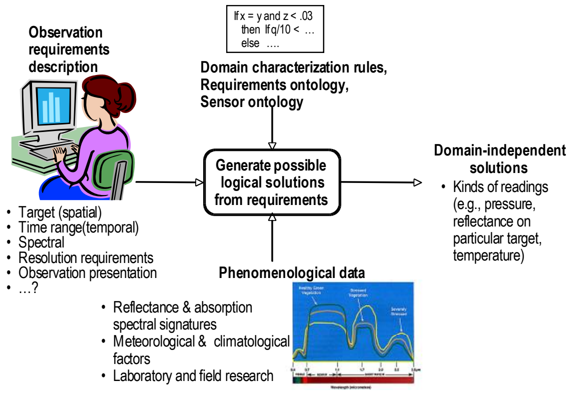

Figure 3. Conversion from domain dependent requests to domain independent solutions

Other domain dependent parameters required for a successful request are also part of the domain characterization. The parameters currently handled in AAMSRT are:

- Type of measurement (i.e. temperature, NIR, etc…)
- Accuracy (i.e. +- 1°C)
- Spatial resolution (i.e. 1-4m)

After determining which measurements are required to fulfill the request, the service consumer agent breaks the total area of observation in a set of pre-determined area geometries already known throughout the system (further specified below). This is a key step in the commoditization of the service, since from this step forward the area of interest is no longer a custom area, but rather a set of pre-determined regions, which allows service to be divided amongst multiple service providers. Note that this step is also transparent for the system user.

In our implementation, the area of interest is specified with the coordinates of two corner points (assuming to observe a rectangular area) on the earth surface. Although one may argue that the area of interest seldom is rectangular, it is very common that the sensor sweep and/or area of coverage are of rectangular shape, so even if some post-processing is needed to calculate the final value, the sensor allocation can still be done in regular rectangular areas without lost of generality.

AAMSRT uses NATO grids as a specification of pre-determined geometry. However, the NATO grid has a roughly 100 KM range per unit, which does not provide adequate granularity for our observation task planning purpose. We have further decomposed each NATO grid into 10*10 pieces (this decomposition is parameterized and can be easily modified). Thus, each sub-grid has a range of 10 KM, which is generally consistent with sensor observation ranges (based on satellite sensors which were used in our prototype).

We use the concept of sub-grid to specify an area that can be covered by a single scan of a sensor. The retrieval of type of measurement with certain minimal accuracy and resolution over a determined sub-grid is defined as an atomic observation task.

Hence, the domain independent request can be seen as a list of atomic observation tasks, associated processes, general resources, and temporal constraints.

Each atomic observation task is the commodity negotiated between service providers and sensor platforms and also between different service providers.

3.2 Sensor Service Provider

The Sensor Service Provider (SSP) is the central part of the AAMSRT system. The SSP is an agent based system within a given agency. It encapsulates the knowledge on how to perform the domain specific observations. The SSP hosts, a least one, coordination agent for planning, along with the agents that represent the resources under this SSP control. Each resource agent is aware of the rules and constraints on its resource, being it a generic resource, such as storage, or a sensor platform. The SSP also hosts the order management agents that track execution in the system.

3.2.1 Planning

The first step in this process is to identify from the domain independent request to the resources that may support this observation. Although the sensor service provider has authority over its own platforms, each platform is self-managed. This vision is shared with OGC's sensor planning approach (Open Geospatial Consortium) [12]. In order to obtain a list of the platforms that contain sensors that could fulfill the requirements, the SSP uses a publish/subscribe paradigm addressing the request for each type of sensor needed for the observation. Resources will "listen" to requests that require the kind of sensors they support, and will reply with their availability and cost for each service. Our work extended from the effort previously done in [13], by allowing multiple sensors per platform and supporting temporal and physical constraints in between multiple sensors of the same platform, as depicted in Figure 4.

Upon receiving the replies from the platforms, the SSP will build a plan to cover all the target NATO sub-grids with every measurement within the deadline and with the minimum possible cost (other objective functions can be used besides cost). Depending on the SSP policies and its current resource utilization, the SSP may forward requests to other SSPs and sub-contract the observation tasks.

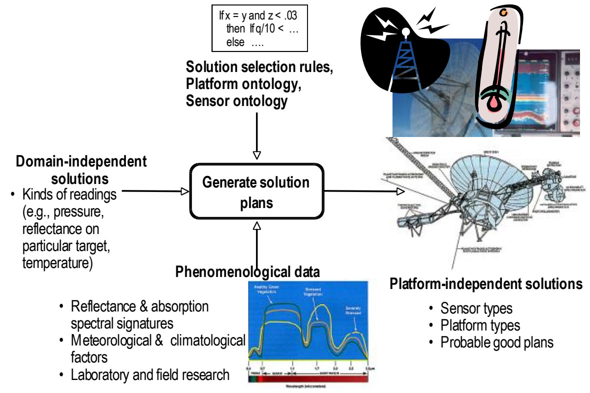

Figure 4. Map domain-independent solutions to platform-independent solutions

3.2.2 Sensor Service Provider and Resource negotiation

Once the SSP has a candidate plan that lists each resource and sensor combination to be used to fulfill the request, it sends to each resource a list of timings and grids allocation to be reserved. This list is a subset from the original list of availabilities sent by the resources. A resource will proceed to reserve these times in its own internal schedule. It is the responsibility of the resource to confirm with the SSP that the sensor and time combinations have been successfully reserved for the targeted request.

After all required resources confirm their reserved status, the SSP has achieved a plan that, according to the resources, can be performed (a viable plan). If any of the resources, which may be negotiating with multiple planners at any time, has already committed one or more of the time slots for another observation, the response will be negative and the planner will re-plan attempting to develop another viable plan based on the availabilities of the resources.

Users may request that the plan and the associated cost must be authorized. In this case, the Sensor Service Consumer (SSC) is notified to request user authorization, once a viable plan has been achieved. The user has a limited amount of time in order to confirm the plan. Failure to comply will result in the automatic denial and the cancellation of the request.

Once a viable plan is authorized, the SSP confirms it with every resource, which then accepts the order and assumes the responsibility of performing the actual observation. An acknowledgement of the commitment by each resource issues an order to fulfill the request and end the SSP planning section.

3.2.3 Negotiation between SSPs

As we have mentioned before, the AAMSRT framework includes the notion of agencies, which manage resources/sensors and hold information on how they are organized. It is possible that an SSP (which has authority limited to its own resources) will not have the resources required to fulfill a requirement, and may require help from other SSPs belonging to different agencies, which hold the capability. The use of platform independent solutions based on atomic observation tasks enables this process. In Figure 5, the service providers are identified and might be able to fulfill the requestor's observations specifications (along with the SSP's own resources). A request is sent to the provider for a quote (RFQ) that conforms to the request.

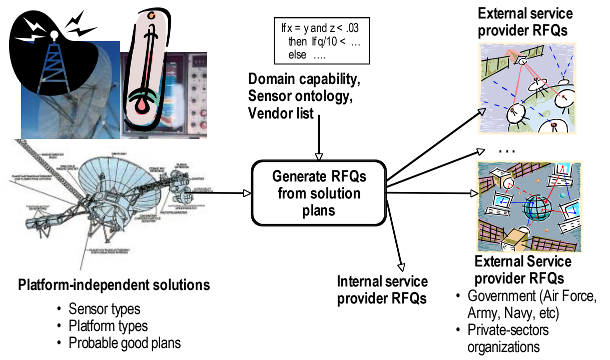

Figure 5. Request solutions from Service Providers

Negotiations between SSP's require a more complex process than the SSP's negotiation with its own resources. The SSPs exchange services with an encapsulation paradigm; they offer the observation services without disclosing which and how the resources will fulfill it. In order to make this approach viable, it is required to have the determination of a more complex ontology that describes these services with terms such as, precision, probability of success, and cost of cancellation (based on how far from the observation deadline). As a matter of fact, full contract rules and constraints are applicable here. An extensive amount of work has been done to address contractual commitments between organization entities and this discussion is outside the scope of this paper.

3.2.4 Scheduling

Our architecture assumes that each platform agent is aware of important details on determining its functional characteristics. This not only includes the capacity and current schedule for each of its sensors, but also the details on possible combinations of sensors (which combination of sensors can be used at the same time), and also which areas on earth surface can be covered by the sensors at each time, along with power issues and other constraints.

This self-awareness is critical because when queried by the Sensor Service Provider (SSP), each sensor is supposed to evaluate weather, when, and with what cost, each measurement can be obtained. And each platform is supposed to coordinate the actions between the multiple sensors that it contains and reply to the SSP.

Writing a code base that can be flexible enough to support the platform/sensors tasks is not challenging. The challenge emerges on describing the platform/sensors features and constraints with flexibility and with adequate details to assure the desired functionality. Our architecture utilizes a generic resource agent that employs SensorML [12] as the basis for its customization.

In our work, it became clear that not all information required by our system could be provided by the current version of SensorML. One example of this limitation is the relative physical constraints between multiple sensors in the same platform, i.e. two sensors which are mounted in opposite locations in the satellite body cannot be operated at the same time. In order o fulfill the gap, we had to supplement the SensorML description with configuration files. We hope that this shortcoming will be overcome with future versions of SensorML.

3.2.5 Execution Tracking

The final components of the AAMSRT framework are the order management agents. These agents are created once a plan is completed and confirmed and the resources set the engagement in their schedule. The role of these agents is to "listen", using the publish/subscribe paradigm, to any resource that has committed to support the order and has "concerns" that may not be able perform within the agreement. An example could be a satellite that cannot perform the reading as schedule due to cloud cover. When notified, the order agent will contact the customer and the SSPs, re-establishing the necessary negotiation to re-plan the support.

Once all the observations and services are rendered properly, the order agent notifies the customer and presents the instructions on how to collect the requested data package.

4. IMPLEMENTATION

Figure 6 shows the implemented architecture of AAMSRT. A user requests an observation through a web thin client via a common web server. The servlet created by the user's session contacts the SSC (potentially in the same computer as the web server), which conducts the negotiation with the SSP.

The implementation includes the simulation of the actual satellite-based sensor platforms, with their location and attitude simulations as described by their SensorML description. For the purpose of this project we have augmented the SensorML description with platform attitude and relative location of sensors in the platform.

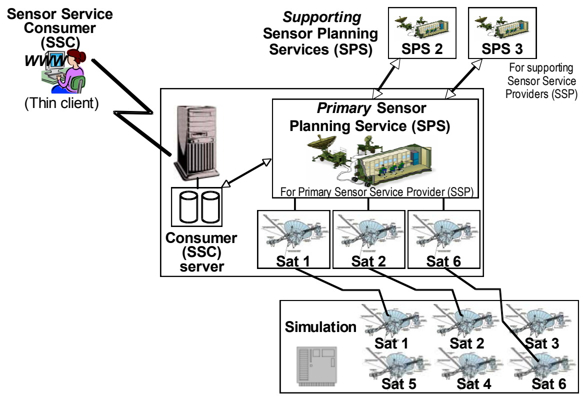

Figure 6. Implemented architecture

4.1 Web Service

The web service support for AAMSRT was created using Apache TOMCAT. Upon user's login, a servlet built in the web server creates a session that connects with a CybelePro agent (CybelePro is the Intelligent Automation's COTS product used as agent infrastructure in the project [14]) using standard socket techniques. The CybelePro agent is the Sensor Service Consumer (SSC). The SSC will respond to the new session "spinning" a dedicated activity that will be responsible to respond and keep the context of the conversation with the user.

The SSCResponder activity also communicates with its assigned preferred Sensor Service Provider (PSP) using CybelePro's publish/subscription paradigm. In order to work in between the two "worlds"; sockets which are synchronous, and agent message passing naturally asynchronous, the SSCResponder has two threads of control. The first thread connects back to the servlet passed by the SSC and receives the requests sent by the user. Once the object is retrieved through the socket connection it is transmitted as a message for the other thread, which is a pure CybelePro thread, and will respond to the request in an asynchronous manner. Some requests, such as refresh status updates, can be accomplished uniquely by the SSCResponder, other threads, such as a new request, are acknowledged by the SSCResponder, processed, and forwarded to be executed by the PSP. Figure 7 shows a diagram of the process.

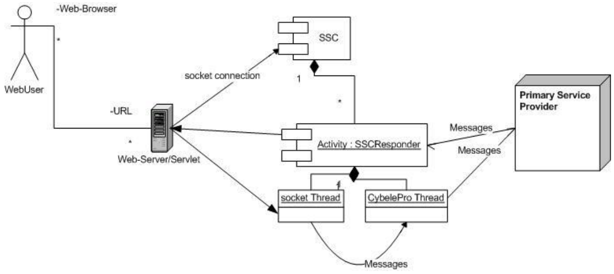

Figure 7. Web-support diagram

Before the SSCResponder can forward a new request to the SSP it performs the domain dependent to independent conversation, as described above. Table 1 shows some of domain characterizations embedded in our prototype.

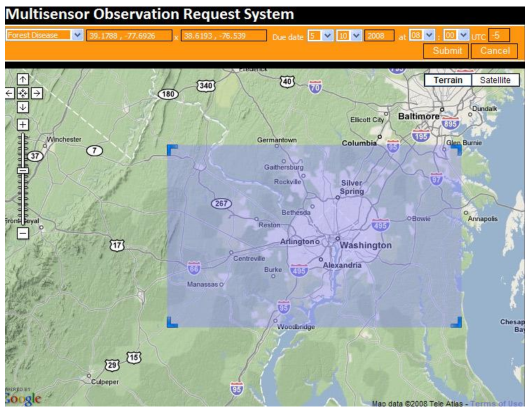

Figure 8. Web server new request screen presented in a browser

The SSCResponder will convert the type of observations into the list of measurements required to calculate the supporting indexes needed to perform the calculations. The knowledge required to perform this conversation was retrieved from the Sensor Service Provider for which this SSC can connect. The domain independent conversion includes the translation from the bounding box to the NATO sub-grids used to describe the world. Figure 8 shows the web page to insert a new request using AAMSRT.

The user can also connect to the web to receive updates on its requests or on pending orders. The system can notify the user through email if a confirmed order needs to be re-planned due to some impediment on one of its resources.

Table 1. Domain characterization in AAMSRT

Domain-based requirements	CASE 1: Forest disease	CASE 2: Crop vigour	Case 3: Fire /thermal anomalies
a.Spatial resolution			
< 1m			
1-4m	x		
4-10m			
10-50m			
50-1,000m			x
>1km			
Other		<100m	
b.Spectral resolution			
Bands	Red, NIR		MIR
Specific spectra			
Specific indices	NDVI, NVI		Fire (several possible)
c.Temporal resolution			
Repeatable	Yearly	Bi-weekly	Weekly
Season	May-June	Feb – June	Year-round
Anniversary dates?	Yes		

4.2 Service Provider

The application that hosts the Sensor Service Provider is the core of AAMSRT. The SSP agent fills the role of the coordinator in our architecture. The SSP will wait until a request is sent by the SSC. When a new request comes, the SSP will start a new activity responsible to plan the observation and coordinate with the resources. At first, the planner activity will solicit from the resources, under direct control of this provider, the time slots in which the resources are able to perform the measurements. The request for capability is broadcasted and all the resources that have sensors capable of fulfilling the request. Each resource, at its own discretion, will respond. After a given time, the planner will collect the responses and try to build a plan that is able to perform all the measurements required by the observation within the given constraints. The plan may include non-sensorial resources such as storage, previous maps, processing power to calculate indexes based on sensor readings etc… In this prototype we focused on sensor based resources. If the cost is too high or the SSP does not have enough resources to fulfill the measurements it can outsource part of the measurements to other SPs.

Once a plan, has been built, the planner will send a message for each selected resource to reserve the times of each required sensor. Each resource has to independently verify that it can perform the required measurement and agree with the reservation.

If any of the resources failed to confirm, the planner will step into re-planning mode and try to replace the resource. Once a reservation is achieved for all measurements needed, the planner will request authorization to proceed from the customer (if needed) by sending a request authorization message to the SSC. The user has a fixed amount of time to authorize; otherwise the request is cancelled. If the plan is authorized, the planner will confirm with the resources, and issue an order, creating an Order agent to track the plan's execution. Figure 9 shows the state diagram of the planner Role in the SSP agent.

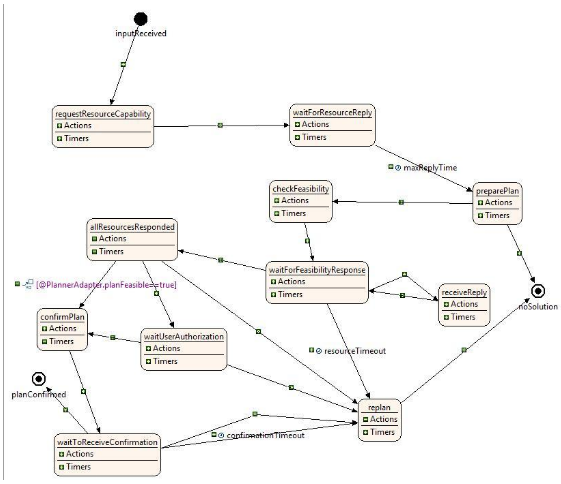

Figure 9 - Simplified state diagram of Planner Role

5. EVALUATION OF AAMSTRT

The AAMSRT framework has three key functional and qualitative achievements: embedded knowledge engineering to broaden sensor assets utilization, multi-agent based planning and scheduling in response to potentially multiple users' requests based on the capability and availability of the sensor assets (including tracking and re-planning), and ontological representation of heterogeneous assets and their organizational structure. Open Geospatial Consortium, as well as many other approaches, adopt the similar idea of ontological representation and attempted to address this issue by posing standards together with some more traditional approaches, such as hiding/ abstraction. AAMSRT specifically addresses ownership and organizational issues and provides a base for the development and implementation of complex policy and contractual interactions.

CASPER and ASPEN are not suitable as a viable solution for multi-sensor retarget problem, because: (1) the tasks for their planning/scheduling are at a low level – plan executives tied to specific hardware (e.g., Mars rover); and (2) not generally applicable to other domains, and (3) users recognize underlying assets and their capabilities very well, versus the distributed nature of managing heterogeneous assets hidden from the users for multi-sensor re-target problems.

Compared with the initial integrated planning and scheduling effort by *Chen et. al.* [2], our AAMSRT approach further considers the schedules of individual sensor assets so that the tasking assets will be available when needed (without any scheduling conflict) to the generated observation plans.

The MAGOA [1] actually shared our vision of coordinating multiple sensor assets (satellites) by leveraging their limited availability (satellite scan time visible to ground stations). But MAGOA stayed a primitive concept design and no further technical details are available for research comparison purpose.

The underlying technologies by Lesser *et.al.*[11] and Farinelli *et. al.* [8], e.g. combinatorial auction and max-sum algorithm, are viable alternatives to address only parts of our multi-sensor retarget problem, e.g., sensor and plan selection; we have explored one step further beyond these problems to not only generating detailed plans for a user requested observation, but also making sure the sensors to employ will be available for the planned tasks using multi-agent technologies of planning and scheduling and re-plan upon failures and conflicts.

POMDP is a modern approach to deal with uncertainties in DPS, but it assumes heterogeneity of resources regardless of the fact that resources may have different owners. Thus, the suitable approach should respect the assets' existing plans and schedules posted by their owners and the tasks with potentially higher priorities by reserving their observation "free time" to improve effectiveness and efficiency to fulfill a user request. Our AAMSRT framework addresses this multiple ownership issue and employs separate planning and scheduling process to resolve any potential conflict between the planned tasks to fulfill a user's request and the sensor assets' own schedules of activities.

Notably, as shown in Figure 8, AAMSRT hides the asset recognition and management processes from the end users and allows for a user specification of requirements without delving into technical details, which is something that a user does not want to, and more importantly should not, be labored.

6. CONCLUSION AND FUTURE WORK

Our work in AAMSRT has clearly addressed the two key research issues raised in our problem formulation, specifically, how one can "popularize" the use of available resources by embedding the required sensor knowledge in the system, and how to select and coordinate the required assets while keeping the organizational boundaries of the multiple providers.

The framework also includes room for other services, albeit not developed during this phase II, such as data storage, sensorial data processing, historical archives, which can also be traded in AAMSRT as commodities.

Further development of AAMSRT includes administration tools to allow easy management of policy and contractual protocols between SSP of different organizations.

7. ACKNOWLEDGMENTS

This project was partially funded by NASA through its contract# NNA06BA03C. We would like to thank Mr. James J. Odell for his contribution to the project. We would also like to thank our domain experts: Dr. Michael E. Botts, Dr. Phillip C. Dibner, and Dr. Maggi Kelly.

8. REFERENCES

[1] Biancho et. al., Software Agents for Multiple Satellite Automated Planning and Control, 24th AIAA International Communications Satellite Systems Conference (ICSSC), 11 - 14 June 2006, San Diego, California, USA.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. JAIR, 11(1):1–94, 1999.

[3] Wei Chen and Keith S. Decker, Managing Multi-Agent Coordination, Planning, and Scheduling, pp.1360-1361, 3rd International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3 (AAMAS'04), 2004

[4] Chen, W. and Decker, K. S. 2006. Analyzing characteristics of task structures to develop GPGP coordination mechanisms. T*he 5th intl. Joint Conf. on Autonomous Agents and Multiagent Systems*. AAMAS '06. ACM, New York, NY, pp. 662-669.

[5] Chien et. al., ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling. Intl. Conf. on Space Operations (SpaceOps 2000). Toulouse, France. June 2000

[6] Delin K., et. al. The JPL Sensor Webs Project: Fielded Technology, Space Mission Challenges for IT Proceedings, Annual Conference Series, 2003.

[7] T. Estlin, G. Rabideau, D. Mutz, S. Chien. Using Continuous Planning Techniques to Coordinate Multiple Rovers Electronic Transactions on Artificial Intelligence. 2000.

[8] Farinelli, A., Rogers, A., Petcu, A. and Jennings, N. R. Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In: 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08), 12-16 May 2008, Estoril, Portugal.

[9] B. Grosz and S. Kraus. Collaborative plans for complex group action. Artificial Intelligence, 86(2):269–357, 1996

[10] Tracy Mullen, Viswanath Avasarala, David L. Hall, "Customer-Driven Sensor Management," IEEE Intelligent Systems, vol. 21, no. 2, pp. 41-49, Mar/Apr, 2006.

[11] Ostwald, John; Lesser, Victor; Abdallah, Sherief. Combinatorial Auction for Resource Allocation in a Distributed Sensor Network. The 26th IEEE International Real-Time Systems Symposium, (RTSS'05), IEEE Computer Society Press, pp. 266-274. December 2005.

[12] Sensor Model languages, SensorML (http://www.opengeospatial.org/standards/sensorml), Sensor Planning Services, SensorPS (http://www.opengeospatial.org/standards/sps)

[13] Tang, K., Lyell, M., Satapathy, G., Colombano, S., Agent based Multi-criteria Optimization for Onboard Multi-sensor Retargeting, IERC, May, 2006.

[14] http://www.CybelePro.com