Teamwork Reasoning
and
Multi-Satellite Missions

NASA Agreement
Investigating Teamwork Among Spacecraft
Number
NAG 2-1489
Period of Performance: 7/1/01- 4/30/02

Stacy C. Marsella
Information Sciences Institute
University of Southern California

Abstract

NASA is rapidly moving towards the use of spatially distributed multiple satellites operating in near Earth orbit and Deep Space. Effective operation of such multi-satellite constellations raises many key research issues. In particular, the satellites will be required to cooperate with each other as a team that must achieve common objectives with a high degree of autonomy from ground based operations. The multi-agent research community has made considerable progress in investigating the challenges of realizing such teamwork. In this report, we discuss some of the teamwork issues that will be faced by multi-satellite operations. The basis of the discussion is a particular proposed mission, the Magnetospheric MultiScale mission to explore Earth's magnetosphere. We describe this mission and then consider how multi-agent technologies might be applied in the design and operation of these missions. We consider the potential benefits of these technologies as well as the research challenges that will be raised in applying them to NASA multi-satellite missions. We conclude with some recommendations for future work.

Contents

1 Introduction

NASA is rapidly moving towards the use of spatially distributed multiple satellites operating in near Earth orbit and Deep Space. The satellites will be required to cooperate with each other as a *team* that must achieve common objectives with a high degree of autonomy from ground based operations. Such satellite teams will be able to perform spatially separated, synchronized observations that are currently not feasible in single satellite missions. This will enable or improve multi-point observations of large scale phenomenon, co-observation of single phenomenon and interferometry. Autonomous operations will reduce the need for ground based support that would otherwise be prohibitively expensive in such missions. However, the underlying control systems necessary to enable such missions will raise many new challenges in autonomous, multi-platform operations.

In particular, a critical requirement for these satellite constellations is that they must act coherently as a coordinated, often autonomous team, and to do so even in the face of unanticipated events. This ability to operate as an autonomous team will need to be satisfied in many of the multi-satellite missions being planned. Therefore, it is important to understand this requirement, elucidate the research challenges it presents and consider approaches to satisfying it.

For example, consider the Magnetospheric Multiscale (MMS) mission. The mission involves 5 satellites flying in various formation configurations while making coordinated, simultaneous observations of the three dimensional structure of the magnetosphere. MMS's observation plan has a projected 2 year life span involving multiple phases with different orbits and formation scales. The satellite "constellation" will face and need to respond in a timely fashion to hard to predict and unexpected events such as solar flare observation opportunities or equipment failures. The constellation will likely have to address most of these events without human operator intervention; there will be limited and delayed communication with earth based human operators.

If an observation event occurs, the constellation may need to make a coordinated decision concerning the onset of observations and which sensor to use, decisions which in turn may be impacted by the status of each craft's sensor equipment. To realize this coordination, the satellites will need to communicate with each other. An effective policy for that communication is clearly a key requirement for the success of the mission. MMS also raises key issues about coordination between the constellation and ground-based operations. For example, in the face of unexpected events, the satellites must balance the need to react coherently in a timely fashion against the need for human oversight at critical junctures. At times, it may be best for the constellation to make an autonomous decision as how to proceed. At other times it may be best to seek human operator intervention. If that intervention does not come in a timely fashion, the constellation may still need to make an autonomous decision. An effective policy for such adjustable autonomy will be critical to the long-term survivability and success of the mission.

Of course, the question of how to achieve the necessary coordination between these craft and the adjustable autonomy with ground operations are requirements that are not unique to MMS or even multi-satellite operations in general. The multi-agent research community has been investigating these issues and has made considerable progress in addressing them. Various general approaches to coordinated teamwork and adjustable autonomy have been proposed, have been implemented in a variety of domains and have demonstrated considerable robustness. These approaches, for example, lay out prescriptions for when teammates should communicate and what they should communicate in order to achieve effective coordination on a team task such as a multi-point observation of an event. The design of multi-satellite missions will likely benefit greatly from this research. At the same time, a multi-satellite constellation will face difficult challenges that raise research questions which are not only at the frontiers of multi-agent research but will likely push that frontier forward.

One of these research challenges concerns the complexity of the process by which the satellites come to some coordinated decision and the quality of the resulting decision. For example, we might consider how MMS decides to make a joint observation with some sensor and whether it is the best decision they could make. Any approach will require certain communications, which consume power and time, and result in a specific decision that is, more or less, the optimal decision given the situation, the time it took to make the decision, etc. Furthermore, the MMS craft must make decisions in the context of a mission that has a 2 year lifespan, therefore the optimal decision for a specific observation event, for example, may be far from optimal in the context of subsequent tasks that must be performed. Indeed the very concept of optimal must take into account that the tasks the mission faces cannot be a priori specified with certainty, given the opportunistic nature of the observations, unexpected equipment failures, etc.

To address this challenge, it is useful to know certain baselines, such as what is the *optimal* decision for the team to make in any given situation and what is the *complexity* of finding that decision. However, to date insufficient progress has been made in precisely characterizing what constitutes an optimal decision and understanding the complexity of finding such optimal decisions. Given the lack of such baselines, it is not surprising that the various practical approaches to making teamwork decisions that have been proposed by the research community have also not been comparatively analyzed in terms of their optimality or complexity. Thus the optimality/complexity tradeoffs of proposed approaches cannot be determined, making it difficult to evaluate alternative approaches. Indeed, the optimal policy for a particular domain or application is typically unknown. This lack of progress in evaluating alternative approaches to central problems in teamwork is particularly worrisome in high cost, critical applications such as satellite constellations.

A second, closely related, research question concerns the *limited resources* any multi-satellite mission faces. Only limited progress has been made by the multi-agent research in explicitly modeling the real world constraints that are fundamental to the success of a satellite mission. For example, communication is in general a cornerstone of effective teamwork and will likely be key to maintaining MMS satellite coordination. However, communication has a cost. It can consume considerable power, can impact certain kinds of data collection and can delay other actions if for example one member of team communicates and waits for a response from other teammates. Similar real world issues arise in the case of adjustable autonomy. Traditionally, the adjustable autonomy issue has been framed as a one-shot decision to either make an autonomous decision or pass control to a human (e.g., ground controllers). However, the decision to pass control may lead to costly delays which ideally should be factored into the decision to transfer control. But in the real world the length of the delay is typically indeterminate, drawing into question the advisability of making such a one-shot decision.

However, recent advances in *formal models* of teamwork and adjustable autonomy have begun to address these challenges. For example, work in casting teamwork into a formal framework, what we call an MTDP (*multi-agent team decision problem)*, provides a tool to address a range of analyses critical to fielding teams in real world applications. Using the MTDP framework, the complexity of deriving optimal teamwork policies across various classes of problem domains can be determined. The framework also provides a means of contrasting the optimality of alternative approaches to key teamwork issues like role replacement. Finally, the framework also allows us to empirically analyze a specific problem domain or application of interest. To that end, a suite of domain independent algorithms has been developed that allow a problem domain to be cast into the MTDP framework. This allows the empirical comparison of alternative teamwork approaches in that domain. Derivation of the optimal policy for the problem domain serves not only as the basis of comparison but also can inform the design of more practical policies. Most recently, progress is being made in addressing how real world operating constraints like power consumption can be modeled in this framework.

Another critical research question concerns *integration*. Clearly, these teamwork and autonomy decisions cannot be made independently from the rest of the operational decisions being made on the craft. But the question of how they integrate is yet another research question.

In this report, our goal is to illuminate several basic issues in the application of multi-agent research to multi-satellite missions. We discuss the need to develop robust and effective coordination prescriptions for multi-satellite teamwork. Rather than mission-by-mission ad hoc approaches to coordination, we focus on a general approach to teamwork that will be both more robust in a particular mission while also building across mission teamwork infrastructure. We also stress the need for analysis and suggest an approach to assessing the quality of alternative prescriptions, based on MTDPs, that allows both formal and empirical evaluation. We illustrate how the approach could be applied to MMS and discuss how it could be extended to provide a faithful rendering of difficult resource limits that such missions will operate under. In addition, we discuss alternatives to realizing the teamwork reasoning and how teamwork and autonomy is integrated into a craft's overall software architecture.

The discussion of these issues begins in Section 2, by describing the MMS mission and pointing out some of the technical challenges it raises for teamwork and adjustable autonomy. But of course, teamwork and autonomy reasoning are just one part of the constellation's operation, which must include various flying, observation, communication and maintenance tasks over the duration of the mission. So we briefly introduce the supervisory control software that manages and schedules these tasks. In particular, we discuss one approach to the design of this supervisory software in order to facilitate later discussions. We then discuss in Section 4 the issue of realizing robust teamwork, as the problem is approached by the STEAM architecture. Section 5, *Analysis and Synthesis of Teamwork*, presents one of the central proposals of this report, the use of formal models for analyzing teamwork. Section 6 presents some prior work in teamwork analysis. Sections 7 and 8 discuss in turn adjustable autonomy and the integration of teamwork reasoning with the supervisory control software. Finally, Section 9, *Recommendations*, suggests several directions for the research and also potential collaborations with NASA.

2 Magnetospheric Multiscale

The Magnetospheric Multiscale (MMS) mission is being designed to investigate the processes of magnetic reconnection, charged particle acceleration and turbulence in the Earth's magnetosphere. The study is concerned with the dynamic and spatial structure of these processes and thus it can not feasibly be undertaken by a single craft. A multi-satellite mission design is being developed that uses identical spacecraft capable of flying in formation and making the simultaneous, coordinated observations required. The 5 satellites of MMS will fly in a hexahedral formation near apogee, comprising two tetrahedral with three of the satellites in a plane with the fourth satellite above and a fifth below that plane. See Figure 1. An alternative design will have four craft defining a tetradron with the fifth craft (potentially) placed within that tetrahedron. See Figure 2. The formation will at times elongate into a string of pearls, depending on where it is in the orbit and the temporal/spatial goals of the observations. Whereas observations that could separate spatial and temporal characteristics of observed phenomenon could be done by two craft, the ability to resolve these characteristics are significantly improved by 5 craft. In order to capture data from different regions of the magnetosphere, there are multiple phases to the mission with different orbits and different inter-satellite distances. Specifically, Phase 3 and Phase 4 will involve more distant observations, including magnetotail studies at up to 120 RE. Depending on the phase of the mission, the spacing between satellites will range for from tens of kilometers to tens of thousands of kilometers, with separations sometimes increasing or decreasing over orbital phase. The MMS mission has an operational duration of 2 years.

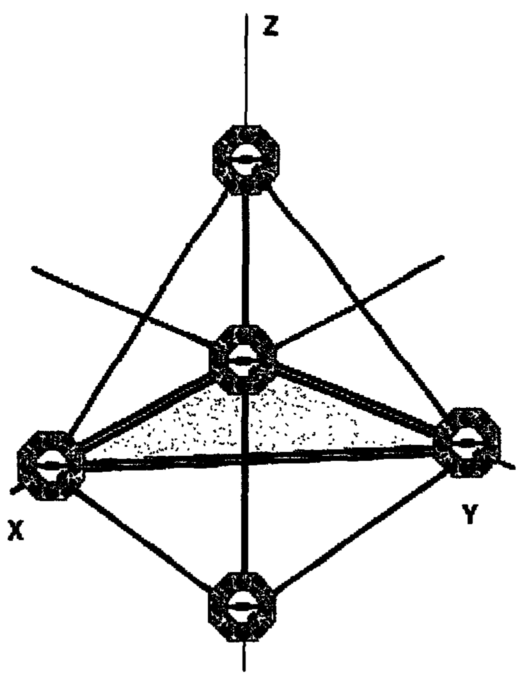

Figure 1. MMS Spacecraft in hexahedral configuration.

Each craft has memory on board to record data which must be transferred to ground stations at appropriate times. As of early 2002, the craft design proposed a sensor system that has two data rates, high and low, which give them different resolution observations. At interesting events, such as a solar flare, the craft should go into high data rate to get the greatest amount/resolution of data. However, some highly desired events happen quickly enough that they can be missed, at least partially. The memory also fills quickly at high data rates. Plus the buffers on the craft may have different amounts of free memory at any time, making it more or less feasible for them to go into high data rate. Not all craft need to be at the same rate during an observation, surprisingly, but the more the better. Also, Phase 3 and 4 of the mission will require the DSN 34-meter dish. Since the downlink of data is sensitive to distance and ground station cost can be prohibitive, the craft will be required to store weeks of data until their orbit brings them close enough for high speed downlinks.

Additionally, the MMS satellites will carry a range of instruments, including plasma instrumentation, energetic particle detector, electric field/plasma wave instruments and magnetometer. For various reasons, a craft's instruments may not be operable simultaneously. For example, they may share electronics. The operation of the sensors will also need to be coordinated between craft.

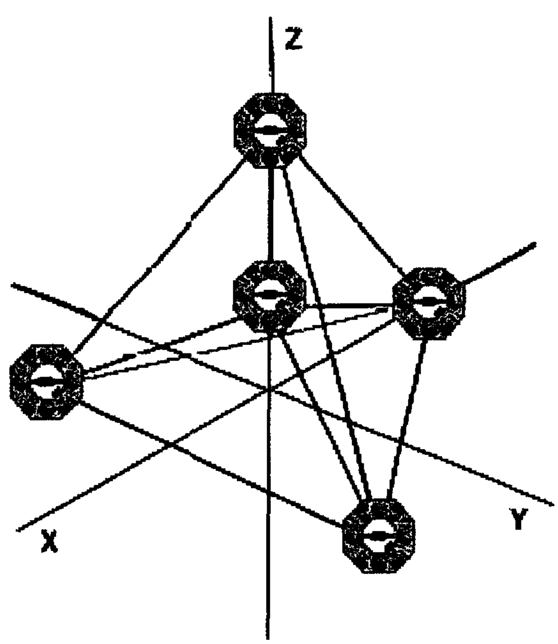

Figure 2. MMS Spacecraft, five tetrahedral configuration.

Finally, the formation is not designed to dynamically reconfigure - the reconfiguration is preplanned depending on where they are in orbit and which phase of the mission it is. Apparently, it is too expensive to consider dynamic reconfiguration - it costs too much in fuel (and consequently liftoff weight). This in principle limits the kinds of coordination tasks that need to be addressed, but does not eliminate the need for coordination. Because of this limitation, however, this document will not address in great detail possible relations between teamwork reasoning or adjustable autonomy and low-level control algorithms that will be used to maintain the crafts' formation. Rather the focus will in large measure be on the science operations.

2.1 MMS and Teamwork.

A decision to make an observation potentially faces various tradeoffs with respect to the interestingness of the observation, the feasibility of any particular satellite going into high rate given its free memory, the quality of the observations that results or when the next downlink is feasible. Additional factors may arise that affect the high/low data rate decision. It is also not clear when to turn back to low data rate - presumably because it is not clear when the observation of an event should end. Closely related is the possibility of foregone future observations due to too full memory prior to any downlink. Or for that matter some satellite could run out of memory mid-observation. The state/precision of the constellation's formation will also impact observation quality and arguably should be factored into the high and low data rate decision.

Related to this observation decision, there are also interesting coordination issues and tradeoffs to be considered. The current proposed coordination approach is an alarm system. A satellite individually detects interesting events and signals others that it spot the event and is going into high data rate, other satellites should in turn signal that they are going into high data rate, assuming they have the buffer space. This is "what's my state" coordination technique that appears topreclude the possibility of coordinating the high/low data rate decision as a team decision which arguably might be a better approach --- since the individual decision may need to take into account the state of the team such as the other satellites state of memory, value of only part of the formation going into high rate, the teams current formation, the possibility of false alerts, whether all craft's sensors are working, power levels in the various craft, etc.

Moreover, the high/low data rate decision is clearly just one decision to coordinate. For example, which instruments will be activated to perform an observation clearly must be coordinated between craft. Again, there may be many factors that could impact this decision and might argue that a coordinated, team decision is preferable. For example, if one or more craft has an instrument failure, then this might argue for changing the observation to other instruments. Since useful, but degraded, observations of spatial/temporal characteristics can possibly be made by even two craft, the appropriate decision may not be obvious.

There also is another planned mission, solar sentinel, that will be closer to the sun that could coordinate with MMS. Specifically, it could be used as early warning sensors for interesting events.

Finally autonomy is critical here. Ground links in general are expensive, especially when, in Phase 3 and 4, DSN is required. Communication would thus drive up costs astronomically and will be relatively infrequent. The uplink of data is designed to be quite contained, one design for the mission specifies that commanding for the instruments will be 100 bytes per day per craft.

2.2 MMS, Formation Flying Testbed and Distributed Satellite Simulation.

The MMS mission has been chosen as the first mission design to be explored within the Formation Flying TestBed (FFTB) being developed at Goddard. FFTB is specifically designed to evaluate the low-level distributed control algorithm (DCA) and hardware involved in realizing the low-level formation maintenance and station-keeping necessary for a mission like MMS. However, the FFTB is also becoming the kernel of a distributed satellite simulation system (DSS) that will bring software and hardware together within a distributed system that will allow the simulation of an entire mission. Since the FFTB and DSS presents special opportunities for evaluating the constellations control software, we briefly describe these components here and raise certain implications of their design for the teamwork research.

The Formation Flying Testbed (FFTB) at NASA GSFC is a modular, hybrid dynamic simulation facility being developed as a platform for the evaluation of guidance, navigation, and control of formation flying clusters and constellations of satellites. The FFTB is being developed to support both hardware and software development for a wide range of missions involving distributed spacecraft operations.

The FFTB has several features of special note here. It is being designed to realize very high fidelity simulations of a constellations formation flying that will provide a strong test for software design. It is a hybrid simulation system that can employ a blend of hardware of software components. The use of hardware within the simulation system can constrain the simulation to run in real time. However, the FFTB design is modular. Software modules can be swapped in for the hardware modules, which would allow faster than real time simulation but at the cost of some loss in the fidelity of the simulation.

Most critically, FFTB is the core of an evolving distributed simulation system/environment (DSS) for satellite constellations. DSS could potentially support the simulation of all aspects of a mission, including the multiple sensors, absolute and relative position determination and control, in all (attitude and orbit) degrees of freedom, information management, high-level supervisory control as well as the underlying physical phenomenon the constellation is designed to observe.

This implementation is therefore an ideal framework for exploring and evaluating alternative approaches to the high-level supervisory control of the craft and its coordination with other craft in the constellation and ground control. The supervisory control has the general functions of validating the data in the navigation system, switching the modes of operation of the vehicle based on either events or schedules, and interfacing the on-board functions with ground functions. Thus teamwork reasoning will need to play some integrated role in supervisory control.

3 Supervisory Control

The main focus of this paper is the adjustable autonomy and teamwork decision-making that arise in missions like MMS. However, these capabilities are realized within the context of each crafts supervisory control software that overall decides what tasks are performed and when they are performed. Thus the relation between the teamwork reasoning and supervisory control is a central issue. For example, one issue that will arise in later discussions concerns the tradeoffs between realizing teamwork reasoning as a separate module versus a tighter integration.

In order to help set the context for that subsequent discussion, we introduce here an example of a general purpose architecture for supervisory control of remote craft that has been proposed by NASA Ames. We leave out many architectural specifics such as the relation of supervisory control to the distributed control algorithms used for formation maintenance.

3.1 Planning and Scheduling

NASA Ames's Intelligent Deployable Execution Agents (IDEA) [14] framework for planning and scheduling, which is a continuation of the work begun for the Remote Agent. IDEA has four main components; (1) a plan database which represents all possible plans that are consistent with the current set of instantiated constraints, (2) a domain model which defines the operational constraints for the craft, (3) a set of planners that generate plans in the plan database and (4) the plan runner which performs execution. Figure 1, borrowed from a NASA report, depicts IDEA.

The IDEA has many interesting capabilities but for our subsequent discussions, two features are most relevant. IDEA allows for multiple planners with different planning time responses, some of which may be more deliberative while others may be more reactive or scripted. The plan database provides a uniform representation for these planners and the execution of the resulting (partial) plan. Within the plan database, it is possible to represent not only partial plans for execution tasks but also flexibly represent planning tasks and reason about the scheduling constraints on those planning tasks. For example, IDEA could schedule a planning task, based on other operational constraints (e.g., whether the cpu is available). IDEA could also choose between alternative planning strategies based on scheduling constraints or modify other mission tasks to ensure time for planning (e.g., go into a wait loop).

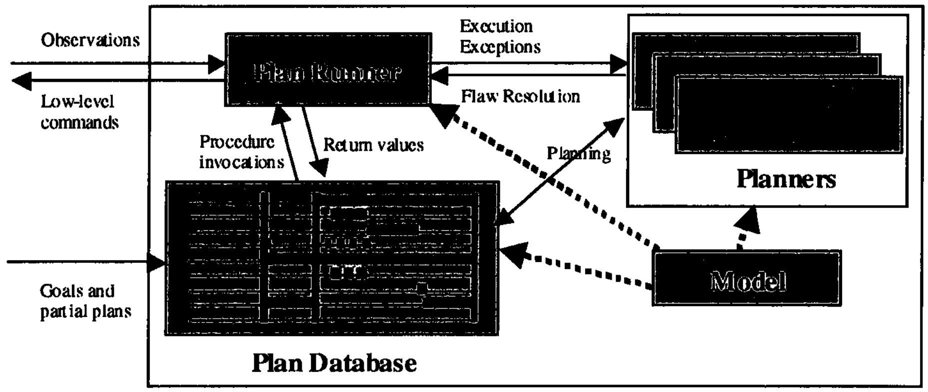

Figure 3. The IDEA Framework (from NASA report)

4 Teamwork

Although there is an increasing demand for multi-agent systems that enable a team of agents to work together, getting the team to perform well in a dynamic environment remains a difficult challenge. It is particularly difficult to ensure robust and flexible performance in the face of unexpected events. Individual agents may fail and there may also be coordination breakdowns, due to agent's not having a shared mental model. Building a system may require a potentially large number of special purpose coordination plans to cover all the low-level coordination details. If the underlying system tasks change or new agents are added, new coordination plans will be needed.

Considerable progress has been made over the years in developing and implementing practical models of teamwork that address these design challenges. Theoretical work on teamwork [Cohen & Levesque, Grosz, etc] laid the solid basis for implemented systems, such as STEAM [Jair], a general model of teamwork that explicitly reasons about commitments in teamwork. STEAM demonstrated the real-world utility of explicit reasoning about teamwork commitments for designing robust organizations of agents that coordinate amongst themselves. In a STEAM system, each team member has general purpose teamwork reasoning skills as well as an explict model of the team plan and its commitments to teammates. Thus each teammate knows that it is in a team and it has commitments to achieve team goals. Plus they possess rules for achieving the coordination required by those commitments in the face of unforeseen events. So, for example, if a teammate sees another teammate fail in a key task, it will reason about whether to warn teammates.

In particular, STEAM contains maintenance-and-repair rules that enable team members to monitor the impact of failing teammates and suggest recovery for such failures (e.g., by substitution of a failing team member with another). It also contains coherence-preserving rules which enable team members to supply each other key information to maintain coherence within a team, and communication-selectivity rules that help agents limit their communication using decision-theoretic reasoning. For example, one coherency preserving rule is that teammates need to know when a team task is achievable. Therefore, if an agent observes that a team task is achievable, this rule comes into play and the agent will decide to communicate the information, based on the communication-selectivity rules.

These rules realize general teamwork reasoning and therefore apply across any team task. They are as well practical in the sense that they take into account tradeoffs. In particular, the communication-selectivity rules take into account the criticality of the task, the cost of the communication and the likelihood that teammates already know. Our experience in a host of difficult domains is that this combination of general teamwork reasoning skills, explicit team plans and decision-theoretic reasoning about tradeoffs is robust. It's robustness follows from the emphasis on giving general teamwork reasoning skills to each teammate. The underlying assumption is that the world is "open", that the unexpected event can happen in the world. The designer of the team cannot pre-plan for every such event but rather must design general methods for teamwork reasoning about failures that allow the teamwork to maintain a coordinated, effective response.

STEAM's successful applications of teamwork to multi-agent systems lead to the Teamcore architecture. The key hypothesis behind Teamcore is that teamwork among agents can enhance robust execution even among heterogeneous agents in an open environment. The Teamcore architecture enables teamwork among agents with no coordination capabilities, and it establishes and automates consistent teamwork among agents with some coordination capabilities, by providing each agent with a proxy capable of general teamwork reasoning. At the heart of each Teamcore proxy is the STEAM teamwork model, which provides the set of rules that enable heterogeneous agents to act as responsible team members. The power of the resulting architecture stems from these built-in teamwork capabilities that provide the required robustness and flexibility in agent integration, without requiring modification of the agents themselves.

The Teamcore/STEAM framework has been successfully applied in several different domains. STEAM's original application was in the battlefield simulation environment where it was successfully used to build a team of synthetic helicopter pilots that participated in DARPA's synthetic theater of war (STOW'97) exercise, a large scale exercise involving virtual and real entities, including human pilots. STEAM was later reused in RoboCup Soccer, where it led to top performing teams in International RoboCup Soccer tournaments. STEAM is at the heart of the Teamcore proxies, which now enable distributed heterogeneous agents to be integrated in teams. Teamcore has been applied to bring together agents developed by different developers in DARPA's COABS program; these agents had no teamwork capabilities to begin with, but Teamcore allowed their smooth integration. Finally, Teamcore is also being used in the "Electric Elves" project, a deployed agent system at USC/ISI, which has been running 24/7 since June, 2000. This system provides Teamcore proxies for individual researchers and students at USC/ISI, as well as proxies for a variety of schedulers, matchmakers, information agents. The resulting team of 15-20 agents helps to reschedule meetings, decide presenters for our research meetings, track people and even order our meals.

4.1 TEAMCORE and MMS

It is useful to consider how we might apply Teamcore to MMS. Consider the previously discussed observation coordination example. To realize coordinated observations, observations would be defined as a team task, which would be achievable, for instance, when a solar flare happened. If a satellite now observed a solar flare, the coherency-preserving and communication rules would lead it to communicate to its teammate satellites that observation was now achievable (i.e., should be jointly executed). This would lead them to turn on high data rate as a team.

We can also consider somewhat more ambitious, speculative scenarios based on general teamwork reasoning and **team reformation**. For instance, assume MMS is in operation when some other mission is launched, enabling in some way better or earlier sensing of interesting events. For example, Solar Sentinel would be such a mission. To exploit this new capability, the already in-flight MMS craft, in principle, would not have to be modified (which would be risky and costly to do via command uplink). The new craft would just be added as a member of the MMS observation team, using the same Teamcore reasoning and observation team task. When it sensed an event, it would inform its teammates, the original MMS team. In practice, of course, this flexibility presumes that the new craft has some communication channel with the MMS craft, a network in some sense. Although such a network may not be currently feasible, if it were one can envision such plug and play teams of heterogenous satellites helping each other on their missions by dynamically taking on new roles in each other's tasks.

Let's also consider the case of failures. Assume some planned action by the supervisory control software, such as an attitude adjustment, suffers a failure of some kind. If the failure impacts a team task such as an observation, then the agent will signal its Teamcore proxy teamwork layer that it cannot perform its role in the team task as planned. The proxy will communicate with the other satellites in the team which will attempt to adjust their plans. If they cannot, they will in turn communicate failure on the team task that will in turn lead to a coordinated response to the initial failure.

5 Analysis and Synthesis of Teamwork

Based on systems like Teamcore/STEAM, multi-agent systems have moved out of the research lab into a wide range of applications areas. But of course, multi-satellite control is a highly critical application, where seemingly minor control decisions can have drastic consequences when made incorrectly. To meet the challenge of such a bold application, multi-agent research will need to provide high-performing, robust designs that performs such control as optimally as feasible given the inherent uncertainty of the domain. Unfortunately, in practice, research on implemented systems has often fallen short in assessing the optimality of their proposed approaches with respect to mission-level performance criteria.

To address this shortcoming, researchers have increasingly resorted to decision-theoretic models as a framework in which to formulate and evaluate multi-agent designs. Given some group of agents, the problem of deriving separate policies for them that maximize some joint reward (i.e., performance metric) can be modeled as a decentralized partially observable Markov decision process (DEC-POMDP). In particular, the DEC-POMDP model is a generalization of a POMDP to the case where there are multiple, distributed agents basing their actions on their separate observations. POMDP is in turn a generalization of a single agent Markov decision process, or MDP, whereby the agent makes decisions based on only partial observations of the state.

The Com-MTDP model is a closely related framework that extends DEC-POMDP by explicitly modeling communication. R-COM-MTDP in turn extends Com-MTDP to enable explicit reasoning about Team Formation and Re-Formation.

These MTDP frameworks allow a variety of key issues to be posed and answered. Of particular interest here, these frameworks allow us to formulate what constitutes an optimal policy for a multi-agent system and in principle to derive that policy.

For example, the COMmunicative Multiagent Team Decision Problem (COM-MTDP) provides a general-purpose language for representing the interactions among intelligent agents sharing a complex environment. The COM-MTDP can capture the different capabilities of the various agents in the world to perform actions and send messages. The model can represent the uncertainty in the occurrence of events, in the ability of the agents to observe such events, and in the effects of those events on the state of the world. The model also uses a reward function to quantify fine-grained preferences over various states of the world. The overall model provides a decision-theoretic basis for examining and evaluating possible courses of action and communication for the agents so as to maximize the expected reward in the face of their environment's ubiquitous uncertainty.

In a COM-MTDP, the behavior of the team is modeled as a joint policy that determines each agent's action based on its observations. There is also a reward function that assigns a value for an agent performing that action in the current state of the world. This framework allows us to determine what is the expected utility of any policy and in principle derive the optimal policy for a team of agents. It may also be a robust policy but only if the probabilistic models have done a faithful rendering of what could happen in the world. Note that in this analysis framework the individual agent knows nothing about being in a team. Knowledge about being in a team is not explicitly being modeled. Rather, a central planner derives a joint policy and each agent only has its part of the policy which tells it what to do next based on its current beliefs. This is quite different from the STEAM teamwork reasoning where each teammate knows it is in a team and can reason individually and as a team about how to best maintain team coordination in pursuit of team goals.

5.1 Technical Details

The COMmunicative Multiagent Team Decision Problem (COM-MTDP)} model subsumes previous distributed models in control theory, decision-theoretic planning, multiagent systems, and game theory. An instantiated COM-MTDP model represents a team of selfless agents who intend to perform some joint task. This COM-MTDP is specified as a tuple, <S,A,O,B,R>.

S is a set of world states which describes the state of the overall system at a particular point in time. For example, the state of a typicai COM_MTDP system would capture the status of the agents (e.g., satellites) themselves, including their positions, their available power, their communication queue, etc. The state would also represent the current environment, external to the agents themselves (e.g., position of other satellites or observation targets).

A_i, is the set of control decisions that each agent i can make to change itself or its environment, implicitly defining a set of combined system actions, A. The actions of an individual agent/satellite, for example, may include choice of sensor, choice of orientation, choice of power consumption (perhaps selecting between high- and low-quality sensing), and potentially even a choice to do no sensing at all (e.g., to maximize power conservation).

The state of the world evolves in stages that represents the progression of the system over time. For nontrivial domains, the state transitions are non-deterministic and depend on the actions selected by the agents in the interval. The non-determinism inherent in these transitions is quantified by specifying transitions as a probabilistic distribution. The transition probability function can represent the non-deterministic effects of each agent's choice of action.

O_i is a set of observations that each agent, i, can experience of its world, implicitly defining a combined observation. O_i may include elements corresponding to indirect evidence of the state (e.g., sensor readings) and actions of other agents (e.g., movement of other satellites or robots). The observations that a particular agent receives are non-deterministic (e.g., due to sensor noise), and this non-determinism is quantified with a set of observation functions. Each such observation function defines a distribution over possible observations that an agent can make. Each observation function represent the noise model of a node's sensors, so that we can determine the relative likelihood of the various possible sensor readings for that node, conditioned on the real state of the system and its environment.

C_i is a set of possible messages for each agent, i, implicitly defining a set of combined communications. An agent may communicate messages to its teammates.

Each agent forms a belief state based on its observations seen and messages received through time, where B_i circumscribes the set of possible belief states for the agent. The agents update their belief states at two distinct points within each decision epoch: once upon receiving observation (producing the *pre-communication* belief state) and again upon receiving the other agents' messages (producing the *post-communication* belief state). The distinction allows us to differentiate between the belief state used by the agents in selecting their communication actions and the more ``up-to-date" belief state used in selecting their domain-level actions.

An agent's belief state forms the basis of its decision-making in selecting both domain-level actions and communication. This decision-making is summarized by mappings from belief states into actions and messages, using a domain-level *policy* that maps an agent's belief state to an action and a communication-level policy.

A common reward function R is central to the notion of teamwork in this model. This function represents the performance metric by which the system's overall performance is evaluated. The reward function represents the team's joint preferences over states, the cost of domain-level actions and the cost of communicative acts (e.g., communication channels may have associated cost).

5.2 An MTDP - MMS analysis example

The COM-MTDP work was originally envisioned as a framework for analyzing teamwork strategies but increasingly we have begun to explore its use in synthesizing teamwork strategies. However, let's first exemplify its use in analysis.

MTDP can be applied to represent the MMS spacecraft's data acquisition discussed earlier. To do this, we would represent each of the spacecraft as an agent, with state features representing the status of each spacecraft. We could also potentially represent a spacecraft's power limitations and consumption within the MTDP model's state space and transition probability. In particular, for each spacecraft, there would be a corresponding state feature representing its available power. The transition probability function would model the dynamics of this available power as a stochastic process, with the change in available power as a function of the spacecraft's choice of action (e.g., data transmission accelerates the rate of power consumption). We can use similar state features to represent the position, orientation, amount of data recorded for each spacecraft, as well as a similar transition probability function to represent the dynamics of each. Such state-based representations have proven successful in modeling distributed systems, and we have had similar success ourselves in applying them to multiagent systems.

There would be additional state features to represent the state of the magnetosphere around them. These features would capture the presence/absence of the various phenomena of interest to the mission. The transition probability function would capture the stochastic evolution of the magnetosphere state, perhaps by incorporating existing models (e.g., MHD models). An agent's observation function would provide a probabilistic model of its corresponding spacecraft's sensors in relation to the state of the surrounding magnetosphere.

Each agent would have a choice of recording or not recording data. The MTDP reward function represents the relative value of its choice after taking into consideration the magnetosphere state. In other words, recording data will have a high value when phenomena of interest are present in the current state. The magnitude of the value will correspond to the relative value of the present phenomena. When an agent decides to record data, the transition probability function will represent the change in the spacecraft's state (i.e., it will have less memory left for recording data).

Given such an MTDP model, we can evaluate data acquisition procedures by encoding them as agent policies. In other words, each agent's policy would represent its corresponding spacecraft's decision process in deciding when to record data, based on its sensor readings. We can then use MTDP algorithms to simulate the behavior of these policies over the possible magnetosphere events. By evaluating the reward earned by the agents over these possible events, weighed against their likelihood, we can derive an expected reward of the policies selected, which in turn allows us to characterize the various performance tradeoffs. We can manipulate the MTDP reward function to isolate the dimensions of interest for each such tradeoff. For instance, if we wish to quantify the ability of an acquisition procedure to avoid running out of power, we can define a reward function that has value 1 in a state where a spacecraft has no available power and 0 in all other states. We can then use our evaluation algorithm to compute the expected reward earned by the nodes, which, with this reward function, will exactly measure the probability that a spacecraft runs out of power. We can make similar reward function definitions that allow our evaluation algorithm to compute expected amount of data recorded, amount of data transmitted, expected number of interesting phenomena *missed*, etc. We can combine reward functions over different dimensions into a single reward function to consider the two dimensions simultaneously and thus quantify the tradeoffs between them. Furthermore, by replacing the expectation in these algorithms with minimization and maximization, we can compute best- and worst-case statistics as well.

This provides a potential basis for selecting between various candidate data acquisition policies. The MTDP model can also potentially provide feedback into the design process underlying data acquisition. A system designer can consider the output of our evaluation algorithms (i.e., the separate predictions and the tradeoffs between them) when choosing among various candidate data-acquisition procedures. These performance predictions will provide the algorithm designers with concrete performance profiles of their algorithms' performance under realistic conditions. The designers can then take these profiles (e.g., too many messages, low probability of success) and use them to make informed improvements to the means by which they achieve data acquisition. This will help our research but in addition provide useful, practical information and software tools (the MTDP analysis framework in particular) for developing these missions.

5.3 Modeling Real World Constraints

Formal models of distributed systems have typically neglected to model real world resource limits. In contrast, one of the features of the previous example use of MTDP was the proposal to model power consumption dynamics. This represents current research in which we are investigating how various real world resource limits such as power consumption can be modeled as first class entities. Since one of the difficult challenges faced by many NASA missions and MMS in particular is the tight resource constraints they operate under, this added capability will clearly have special relevance for using the MTDP framework for NASA mission analyses.

5.4 Synthesis and Re-Synthesis potential.

As commented earlier, the MTDP work was originally envisioned as a framework for analyzing teamwork algorithms. Our experiences to date have also revealed an extremely interesting potential for synthesis. For example, we can use the MTDP work to derive an optimal policy for some team mission by simply simulating all possible policies out to some bounded point in the simulation and picking the best one. This optimal policy is of course only optimal under the assumptions about the world built into the probabilistic models used in the simulation. And it is not a tractable simulation to perform in general. Nevertheless, it does provide a benchmark against which to measure the optimality of alternative teamwork reasoning approaches such as the TEAMCORE work mentioned above. When we have done this kind of benchmarking, we found that the MTDP may generate optimal policies that were entirely unexpected. For example, the optimal policy might replace "failed" teammates before they fail - in essence employing a redundancy approach in high-risk situations. The optimal policy might flexibly decide to replace or not replace based on the expected utility. Finally in some cases it might choose to abandon the mission. None of these capabilities were built into the experiments by the designers - they were discovered by deriving and then inspecting the optimal policy.

This discovery suggests a third approach to building agent teams – the iterative combined approach. Here the domain is modeled probabilistically, the optimal policy is derived and this policy is analyzed to suggest possible improvements to the more general-purpose teamwork reasoning strategies such as employed in TEAMCORE. In other words, by examining the optimal policy (which may be infeasible with real-world resource constraints), we could identify deviations made by our more practical TEAMCORE architecture. We can then modify the architecture to be more in line with the ideal behavior specified by the optimal policy, and thus minimize the suboptimality that we achieve in practice.

5.5 Effective policy derivation algorithms

COM-MTDP and decentralized POMDPs clearly show considerable promise for multi-agent research as well as the application of that research. One key step to using these formalisms is the derivation of the policies. However effective algorithms for deriving policies for decentralized POMDPS is ongoing research. Significant progress has been achieved in efficient single-agent POMDP policy generation algorithms (refs, Monahan, etc). However, it is unlikely such research can be directly carried over to the decentralized case. Finding an optimal policies for decentralized POMDPs is NEXP-complete and therefore provably does not admit a polynomial time algorithm (Bernstein, Zilberstein and Immerman). In contrast, solving a POMDP is PSPACE-complete (Papadimitriou and Tsitsiklis). As Bernstein et al. note (ref), this suggests a fundamental difference in the nature of the problems. Since the reward function is a joint one, the decentralized problem can not be treated as one of separate POMDPs in which individual policies can be generated for individual agents. (For any one action of one agent, there may be many different rewards possible, based on the actions that other agents may take.)

In our own work, we have developed several policy derivation algorithms. Among these is an exact algorithm that generates optimal policies via a full search of the space of policies. This exact algorithm is of course expensive to compute which limits its applicability to problems for which there is sufficient time to offline pre-compute such an exact solution or some way of decomposing the problem a priori. Therefore, we have also developed approximate algorithms. For example, one approach is to search the space of policies incrementally. This algorithm iterates through the agents, finding an optimal policy for each agent assuming the policies of the other agents are fixed. The algorithm terminates when no improvements to the joint reward is achieved, thus achieving a local optimum similar to a Nash Equilibrium.

This question of effective algorithms will likely be of special relevance to the application of these formalisms to MMS. Given its projected mission duration of two years, a brute force search for the optimal policy would not be feasible. However, although the resource constraints of such missions will complicate our representation, they may actually *simplify* such algorithms by restricting the search space of implementable policies. For example, the optimal policy for many COM-MTDP problems requires that the agents remember *all* of their observations throughout their lifetime and then choose different actions based on all possible such observation sequences. Spacecraft with the limited memory resources cannot store such a policy, let alone execute it. The number of possible policies that are executable is much smaller than the number of unrestricted policies, which suggests that finding optimal policies *subject to the mission resource constraints* may be feasible through novel COM-MTDP synthesis algorithms.

6 An aside: Data-driven analysis

The COM-MTDP work provides an approach to analyzing team performance. A key requirement for the analysis is the probabilistic models of the domain and task, for example the state transition probabilities and the observation function. This begs the question of where these models come from.

In the case of MMS, these models could be derived directly or indirectly from the models of the magnetosphere, of the low-level flight control, etc. that are part of the Formation Flying Test Bed (FFTB) and Distributed Satellite Simulation (DSS) mentioned earlier which are being developed at Goddard. For example, an indirect derivation would rely on the simulation of these models within the DSS that could be sampled to derive estimates of the probabilistic models needed for COM-MTDP. By combining the COM-MTDP framework and the DSS simulation, the overall approach to the analysis would be more driven by the data in the simulations. More generally, we envision such combinations of analytical analysis and simulation to be a particularly fruitful research path.

This optimism stems from our prior experiences in researching data-driven approaches to analysis that used simulation data to derive models that were subsequently used for teamwork analysis. In particular, such an approach was used by the ISAAC teamwork analysis tool (ref). ISAAC performs post-hoc, off-line analysis of teams using agent-behavior traces derived from the team's performance in the domain or simulation of the domain. This analysis is performed using data mining and inductive learning techniques to derive models of the team's performance in the domain.. Using data from the agents' external behavior traces, ISAAC is able to analyze a team with very little in the way of pre-existing models of the domain or the team's internals.

In fact, ISAAC develops multiple models of a team. To fully understand team performance, multiple levels of analysis are criticial. One must understand individual agent behavior at critical junctures, how agents interact with each other at critical junctures as well as the overall trends and consequences of team behavior throughout the life of a mission. Thus ISAAC is similarly capable of analyzing from multiple perspectives and multiple levels of granularity. To support such analyses, ISAAC derives multiple models of team behavior, each covering a different level of granularity. More specifically, ISAAC relies on three heterogeneous models that analyze events at three separate levels of granularity: an individual agent action, agent interactions, and overall team behavior. These models are automatically acquired using different methods (inductive learning and pattern matching) -- indeed, with multiple models, the method of acquisition can be tailored to the model being acquired.

Yet, team analysts such as ISAAC must not only be experts in team analysis, they must also be experts in conveying this information to humans. The constraint of multiple models has strong implications for the type of presentation as well. Analysis of an agent action can show the action and highlight features of that action that played a prominent role in its success or failure, but a similar presentation would be incongruous for a global analysis, since no single action would suffice. Global analysis requires a more comprehensive explanation that ties together seemingly unconnected aspects and trends of team behavior. ISAAC uses a natural language summary to explain the team's overall performance, using its multimedia viewer to show examples where appropriate. The content for the summary is chosen based on ISAAC's analysis of key factors determining the outcome of the engagement.

Additionally, ISAAC presents alternative courses of action to improve a team using a technique called 'perturbation analysis'. A key feature of perturbation analysis is that it finds actions within the agents' skill set, such that recommendations are plausible. In particular, this analysis mines data from actions that the team has already performed.

ISAAC has been applied to all of the teams from several RoboCup tournaments in a fully automated fashion. This analysis has revealed many interesting results including surprising weaknesses of the leading teams in both the RoboCup '97 and RoboCup '98 tournaments and provided natural language summaries at RoboCup '99. ISAAC was also awarded the 'Scientific Challenge Award' at the RoboCup '99 international tournament. ISAAC is available on the web at http://coach.isi.edu and has been used remotely by teams preparing for these competitions.

While ISAAC is currently applied in RoboCup, ISAAC's techniques are intended to apply in other team domains such as agent-teams in satellite constellations. For example, ISAAC could produce a similar analysis for the DSS simulation system and use similar presentation techniques as well. Indeed, we believe that the COM-MTDP analysis work could be incorporated into a ISAAC-like tool for the DSS system.

6.1 OVERVIEW OF ISAAC

(Perhaps delete this section)

ISAAC uses a two-tiered approach to the team analysis problem. The first step is acquiring models that will compactly describe team behavior, providing a basis for analyzing the behavior of the team. As mentioned earlier, this involves using multiple models at different levels of granularity to capture various aspects of team performance. The second step is to make efficient use of these models in analyzing the team and presenting this analysis to the user An overview of the entire process is shown in Figure 4.

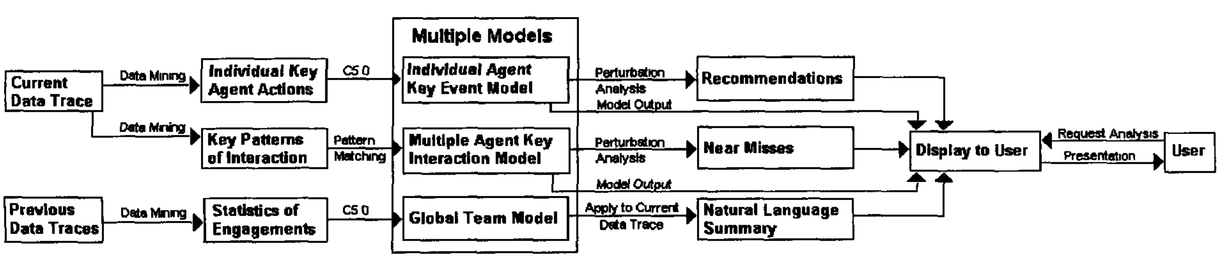

Figure 3. ISAAC analysis process.

Input to all models comes in the form of data traces of agent behaviors. In the current implementation of ISAAC, these traces have been uploaded from users around the world through the Internet.

As shown in figure 4, acquiring the models involves a mix of data mining and inductive learning but is specific to the granularity of analysis being modeled. Analysis of an individual agent action (*individual agent key event model*) uses the C5.0 decision tree inductive learning algorithm, an extension to C4.5, to create rules of success or failure [ref]. For analysis of agent interactions (*multiple agent key interaction model*), pre-defined patterns are matched to find prevalent patterns of success. To develop rules of team successes or failures (*global team model*), game level statistics are mined from all available previous games and again inductive learning is used to determine reasons for success and failure.

Utilizing the models involves catering the presentation to the granularity of analysis to maximize human understandability. ISAAC uses different presentation techniques in each situation. For the individual agent key event model, the rules and the cases they govern are displayed to the user who is free to make the final determination about the validity of the analysis. By themselves, the features that compose a rule provide implicit advice for improving the team. To further elucidate, a multimedia viewer is used to show cases matching the rule, allowing the user to better understand the situation and to validate the rules (See figure 5). A *perturbation analysis* is then performed to recommend changes to the team by changing the rule condition by condition and mining cases of success and failure for this perturbed rule. The cases of this analysis are also displayed in the multimedia viewer, enabling the user to verify or refute the analysis.

For the multiple agent key interaction model, patterns of agent actions are analyzed similar to the individual agent actions. A perturbation analysis is also performed here, to find patterns that are similar to successful patterns but were unsuccessful. Both successful patterns and these 'near misses' are displayed to the user as implicit advice. This model makes no recommendations, but does allow the user to scrutinize these cases.

The global team model requires a different method of presentation. For the analysis of overall team performance, the current engagement is matched against previous rules, and if there are any matches, ISAAC concludes that the reasons given by the rule were the determining factors in the result of the engagement. A natural language summary of the engagement is generated using this rule for content selection and sentence planning. ISAAC makes use of the multimedia display here as well, linking text in the summary to corresponding selected highlights.

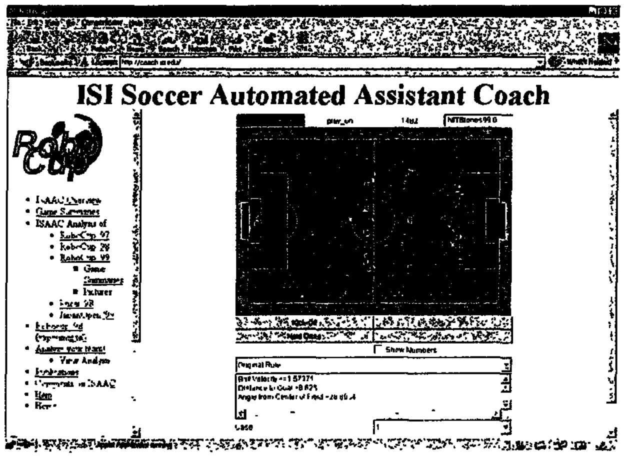

Figure 5: Multimedia viewer highlighting key features

7 Adjustable Autonomy

One of the interesting issues raised by MMS is how the team of satellites interact with ground control. Recall that at various times, the constellation is quite distant from Earth and requires the DSN to communicate (and then only when the orbit takes them close enough high speed data links). This makes communication more costly and harder to schedule. The planned daily uplink of command data has been estimated in one report (1999) to be 100 bytes. Clearly, the MMS satellite will need to exhibit considerable autonomy but nevertheless it is not hard to imagine that system anomalies may occur that require human intervention.

Increasing interest in applications where humans must act as part of agent teams, has led to a burgeoning of research in adjustable autonomy, i.e., in agents that dynamically adjust their own level of autonomy. Essentially, for effective task performance, an agent may act with full autonomy or with reduced autonomy --- harnessing human knowledge or skills when needed, but without overly burdening the humans. The results of this research are both practically important and theoretically significant.

The need for agent teamwork and coordination in a multi-satellite mission leads to critical and novel challenges in adjustable autonomy --- challenges not addressed in previous work, given that it has mostly focused on individual agents' interactions with individual humans. For instance, consider one of the central problems in adjustable autonomy: when should an agent transfer decision-making control to a human (or vice versa). The presence of agent teams adds a novel challenge of avoiding team miscoordination during such transfer.

To get a more concrete sense of the Adjustable Autonomy Issues here, consider a simple example. If the MMS constellation's formation deteriorates beyond some safe bound, the side-effects of making the adjustment may make it undesirable to leave it to the low-level distributed control algorithm (DCA) to make adjustments. There may be more than one way for the individual satellites to adjust with different fuel requirements across satellites, while the satellites may differ in amount of fuel they have. One of the satellites may have a persistent but not detected/diagnosed anomaly in its attitude control that is leading to the formation degradation, which should be factored into the decision-making. The necessary adjustments may also subsequently impact the transformations of the orbits over time, which are part of the planned mission phase transitions. Finally, these factors are happening in some part of the orbit/mission that makes communication with ground more or less feasible in some amount of time.

Clearly, if a single agent were to transfer control for this decision to the human user involved, and the human fails to respond, the agent may end up mis-coordinating with its teammates who may need to act urgently. Yet, given the risks in the decision, acting autonomously may be problematic as well. Clearly, the adjustable autonomy in this context applies to the entire team of agents rather than any individual spacecraft. Further, if the decision is to transfer control, the team could not expect to wait indefinitely for a response from a human operator.

Clearly, the need for real-time response, the serious potential costs of errors, and the inability of the human to directly monitor the state of the different spacecraft add to the complexity of the adjustable autonomy problem. In addressing such challenges, on-going work in adjustable autonomy will play a critical role.

For example, one approach to avoid team miscoordination due to transfer of control decisions is for an agent to take into account the cost of potential mis-coordination with teammates before transferring decision-making control. For example, if a satellite is having persistent difficulty maintaining formation, one response might be to ask ground control what to do and go into a wait loop waiting for a response. But such a response needs to take into account the miscoordination consequences before it decided to transfer the control decision to ground. This would avoid rigidly committing to a transfer of control decision and allow the craft to continual reevaluating the situation, reversing control and taking autonomous action when needed. This suggests that transfer of control must be more strategic.

7.1 Transfer of Control Strategies

Previous approaches to transfer-of-control were quite too rigid, employing one-shot transfers-of-control that can result in unacceptable coordination failures. Furthermore, the previous approaches ignore potential costs (e.g., from delays) to an agent's team due to such transfers of control.

To remedy such problems, more recent work (ref to Scerri et al) emphasizes the notion of a transfer-of-control strategy. A transfer-of-control strategy consists of a conditional sequence of two types of actions: (i) actions to transfer decision-making control (e.g., from the agent to the user or vice versa) and (ii) actions to change an agent's pre-specified coordination constraints with team members, aimed at minimizing mis-coordination costs. An agent executes such a strategy by performing the actions in sequence, transferring control to the specified entity and changing coordination as required, until some point in time when the entity currently in control exercises that control and makes the decision. When the agent transfers decision-making control to an entity, it may stipulate a limit on the time that it will wait for a response from that entity.

Since the outcome of a transfer-of-control action is uncertain and some potential outcomes are undesirable, an agent needs to carefully consider the potential consequences of its actions and plan for the various contingencies that might arise. Moreover, the agent needs to consider sequences of transfer-of-control actions to properly deal with a single decision. Considering multi-step strategies can allow an agent to attempt to exploit decision making sources that might be too risky to exploit without the possibility of retaking control. For example, control could be transferred to a very capable but not always available decision maker then taken back if the decision was not made before serious miscoordination occurred. More complex strategies, possibly including several changes in coordination constraints, can provide even more opportunity for obtaining high quality input.

7.2 Implications of Strategies

The goal for a transfer of control strategy is for high quality individual decisions to be made with minimal disruption to the coordination of the team. Clearly however there are dependencies. Transfer of control actions, whether they are one-shot or strategies, take time. Further the decision to use a particular transfer of control strategies may not be independent from the other task facing the team and individual craft. This clearly factors in to the question of how adjustable autonomy is realized within the overall software architecture and in particular its relation to supervisory control – a question we return to later.

Of course, one approach to deriving good transfer of control strategies is to conjoin decision-making about adjustable autonomy with the other planning and scheduling decisions. For example, one can operationalize transfer of control strategies via Markov decision processes (MDPs) which select the optimal strategy given an uncertain environment and costs to individuals and teams. Scerri et al. have also developed a general reward function and state representation for such an MDP, to facilitate application of the approach to different domains.

7.3 MMS and AA

Currently, it is not clear to what extent adjustable autonomy will play a major role in MMS. MMS is being planned with an apparent high degree of autonomy. However, it is interesting to note that the costs in time and money of any transfer of control to human operators on the ground will vary over the course of an orbit as well as the phase of the mission. For example, in phases 3 and 4 of the mission, as noted earlier, MMS will be quite distant at times and require scheduling time on the DSN for communication. This would make any interaction with ground more costly and more time consuming. The implication of this is that if Adjustable Autonomy becomes part of the mission design, the transfer of control strategies will be quite different over the course of the mission.

8 Integration

Until now, we have only briefly touched on how the teamwork reasoning and adjustable autonomy reasoning could be folded into each craft's supervisory control procedures. However, the discussions of the underlying decision-making and communication involved in teamwork and adjustable autonomy made it clear that these processes take time. For that reason, they may interact with the scheduling of other tasks. For example, the decision to turn on a sensor could be made autonomously by a craft, negotiated with other craft, transferred to ground or decided by executing some transfer of control strategy. Each of these strategies will have some kind of temporal footprint with potential tradefoffs on whether the conjoined sensing acting succeeds, whether other mission critical tasks are delayed, which tasks need to be performed, how the power levels are impacted and how much the data buffer is filled. The tradeoffs in principle might work both ways. Thus, the teamwork and autonomy decision-making processes may impact the scheduling decisions made by the supervisory control and conversely the scheduling decisions may impact which teamwork strategy is preferred. And overall solution quality may, in fact likely will, depend on the teamwork, autonomy and supervisory control decisions.

This argues for a tight integration of these teamwork and supervisory procedures, for an integration that makes teamwork decisions part of the supervisor's planning, scheduling and execution. Of course, this need for tight, uniform integration is precisely the kind of need that architectures like IDEA, specifically its plan database, are supposed to address. IDEA gives planning decisions first class status in its plan database and it could likewise incorporate teamwork and adjustable autonomy decisions. Thus, one model of a general software architecture for multi-satellite missions like MMS is to integrate the teamwork and adjustable autonomy reasoning into the rest of the decision-making. The constraints that the alternative decision choices impose on each other can then be explicitly reasoned about. For example, in such a system, the planning/scheduling decides which transfer of control strategy to use in concert with decisions being made about other tasks.

An alternative is to treat these decision-making processes as separate modules. Indeed this is often the norm in the design of multi-agent teams. Teamcore in particular is an architecture designed around the assumption that teamwork reasoning can be a distinct module or wrapper around the rest of the agent's individual task reasoning. This approach has many benefits. The separation has no doubt played a key role in the advances made in multi-agent teamwork theory. More pragmatically, the separation provides a strong decomposition that greatly simplifies the software engineering task. It also allows existing agent designs to be wrapped. It would likely work well in many multi-satellite missions. But it does, by design, enforce a separation between individual task reasoning and team task reasoning. If the tradeoffs between these tasks are inconsequential, then there needs to be someway to make those tradeoffs explicit in the interactions between decision modules. For example, supervisory control might communicate to the teamwork reasoning various time windows available to make a decision along with its estimate of their impact on solution quality. The teamwork reasoning module would make a decision on the appropriate coordination strategy based on this information and its own estimates. One might also imagine some form of iterative communication between a single craft's modules, or even negotiation, to come to a joint decision.

A third alternative is arguably more radical. We mention it here only since we earlier discussed decentralized POMDPs as a framework of analysis. This naturally raises the question of why not consider them for synthesis. In this approach, there is no flexible supervisory control and no teamwork reasoning module. Rather a decentralized policy is derived for all the craft. Each craft's software simply implements that policy that drives their behavior based on the history of their observations. The individual craft sense, communicate, make attitude adjustments, uplink and downlink because their individual policies informed them to perform these actions. We do not envision this approach being feasible for anything but perhaps the shorter, simpler missions. Given the complexity of generating Dec-Pomdp algorithms, it may not be feasible to derive the policy for longer, more complex missions in the first place. Further, the probabilistic models for state transitions, observations, etc. are not known with sufficient accuracy to entrust mission success to them. The policies themselves may be too large to store on board. Arguably most important is the fact that there are alternative approaches with well-demonstrated track records. IDEA is the follow-on to Remote Agent which has mission experience. STEAM has been used in many applications where it is has demonstrated its robustness and has even evaluated in several domains within the COM-MTDP framework where it has demonstrated that it can provide a cheaper-to-compute good approximation to optimal performance.

9 Recommendations

It is a difficult challenge to design a team of agents that can coherently and efficiently pursue common goals in dynamic, uncertain environments. Indeed, the magnitude of the challenge is often underestimated. However considerable progress has been made by the multi-agent research community in understanding this challenge, designing teamwork algorithms and implementing agent teams. Clearly, this research could play an important role in facilitating the development of NASA multi-satellite missions. As has been noted throughout this paper, the application of this research to NASA missions like MMS raises several issues and opportunities. In this conclusion, we summarize these issues and make suggestions for future directions.

NASA is embarking on a wide range of ambitious multi-satellite mission designs. By establishing FFTB and DSS, NASA has already recognized and acted on the pressing need for systematic evaluation and experimentation of any distributed spacecraft system. This presents a clear opportunity for NASA and the multi-agent research community to collaborate. In particular, models of teamwork reasoning could be part of this experimentation. Without such models, key questions about satellite coordination and performance will remain unanswered. Incorporating a teamwork module would be relatively straightforward. Indeed, there are no technological barriers to incorporating Teamcore into DSS since Teamcore, like FFTB and the DSS system, is designed to be a modular component.

In particular, the formal MTDP work can and should play a key role in analyzing designs for distributed satellite missions. These formal frameworks will likely have a major impact on multi-agent research. For example, the best-case, worst-case and average case analyses they support will be a critical part of any real-world, high-cost application of multi-agent systems. In terms of NASA missions, the formal analyses could be performed, rapidly, outside of DSS, resulting in tested and improved teamwork prescriptions that would then be tested inside of DSS. Alternatively, a hybrid approach might be feasible where some of the probabilistic functions of the MTDP framework are realized by software modules that are part of the DSS.

Note, as discussed earlier, we envision that the main role for MTDP to be in the analysis of algorithms or informing the design of new algorithms, as opposed to synthesis of MTDP policies as a replacement for existing algorithms.

As a first step towards applying this MTDP framework to the problem of designing better satellite teams, we would propose to cast an example NASA satellite constellation problem, specifically MMS, into the MTDP framework. This will allow us to evaluate alternative approaches to role replacement and adjustable autonomy eventually and contrast them with optimal policies. We also envision that an ISAAC-like tool that incorporates the MTDP framework could be readily incorporated into the DSS environment. To fully exploit the potential of the MTDP work, research is needed to develop efficient algorithms for finding approximately optimal policies.

As NASA embarks on developing multi-satellite missions, we believe it is important to explore general approaches to teamwork reasoning and analysis from the start. We believe this is true even in early multi-satellite missions that may seemingly require minimal teamwork coordination. For example, it may seem that a mission like MMS is simple enough that it does not require general architectures for teamwork or extensive analysis of alternative coordination schemes.. However, ad hoc coordination schemes that address specific coordination tasks as special cases are too brittle. This conclusion has come to the multi-agent community through hard-earned experience. Quite simply, human designers cannot think of every way coordination can break down, so there is always another special case rule to add. Further, it ends up being more time consuming and costly to come up with the host of ad hoc rules. Finally, by incorporating general teamwork reasoning and analysis early on, these initial missions could lay critical groundwork that could be exploited in later more ambitious missions.

10 Acknowledgements

I would like to thank NASA Ames researcher, Dr. Kanna Rajan, for his key help and support of this project. I would also like to thank NASA Goddard researchers Dr. Jesse Leitner, Dr. Russell Carpenter and Dr. John Bristow for their time and input. Dr. David Pynadath, provided critical input to this report. In addition, my colleagues at ISI, Dr. Milind Tambe, Dr. Paul Scerri and Ranjit Nair for their suggestions and critiques.

11 Bibliography

Curtis, S. The Magnetospheric Multiscale Mission… Resolving Fundamental Processes in Space Plasmas. Report of the NASA Science and Technology Definition Team for the Magnetospheric Multiscale (MMS) Mission. NASA/TM-2000-209883, December 1999.

Nicola Muscettola, Gregory A. Dorais, Chuck Fry, Richard Levinson, and Christian Plaunt, *QSS* A Unified Approach to Model-Based Planning and Execution

J. Russell Carpenter, David C. Folta, and David A. Quinn. Integration of Decentralized Linear-Quadratic-Gaussian Control into GSFC'S Universal 3-D AUutonomous Formation Flying Alogorithm, AIAA-99-4269.

J. Russell Carpenter. A preliminary investigation of decentralized control for Satellite Formations.

Jesse Leitner. A Hardware-in-the-Loop Testbed for Spacecraft Formation Flying Applications.

C. Schrijver, K. Carpenter. A dream of a mission: Stellar Imager and Seismic Probe.

Steve Chien, Rob Sherwood, Michael Burl, Russell Knight, Gregg Rabideau, Barbara Engelhardt, Ashley Davies, Paul Zetocha, Ross Wainwright, Pete Klupar, Pat Cappelaere, Derek Surka, Brian Williams, Ronald Greeley,Victor Baker, James Doan. The Techsat-21 Autonomous Sciencecraft Constellation.

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

Ranjit Nair, Milind Tambe, Stacy Marsella and David Pyndath: "Model for Team Formation for Reformation in Multiagent System", *AAAI Workshop on Coalition Formation*, 2002.

Ranjit Nair, Milind Tambe, Stacy Marsella and David Pyndath: "R-COM-MTDP: Forming and Comparing Plans for Team Formation in Multiagent Domains", *AAAI Workshop on Planning for and with Multiagent Systems* , 2002.

Ranjit Nair, David Pyndath, Makoto Yokoo, Milind Tambe and Stacy Marsella: "Towards Computing Optimal Policies for Decentralized POMDPs", *AAAI Workshop on Game Theoretic and Decision Theoretic Agents* , 2002.

Milind Tambe, Ranjit Nair, Stacy Marsella and David Pynadath: "Practical Teamwork: Architectures and Analysis", *Proceedings of European Agent Systems Summer School, 2002.*

Ranjit Nair, Milind Tambe, Stacy Marsella and David Pynadath: "Formalizing Team Formation and Reformation in Multiagent Systems", *AAMAS Workshop on Teamwork and Coalition Formation* , 2002.

Raines, T., Tambe, M., and Marsella, S. Automated agents that help humans understand agent team behaviors, *Proceedings of the 4th International conference on Autonomous Agents*, 2000.

Marsella, S., Adibi, J., Alonaizan, Y., Kaminka, G., Muslea, I., Tambe, M..Experiences acquired in the design of robocup teams: A comparison of two fielded teams. *Journal of Autonomous Agents and Multi-agent Systems*, special issue on "Best of Agents'99", Vol. 4, 115-129.

D.V. Pynadath and M. Tambe, 2002. The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models. *Journal of Artificial Intelligence Research*, to appear.

Pynadath, D.V. and Tambe, M. **Multiagent Teamwork: Analyzing the Optimality and Complexity of Key Theories and Models**. In the Proceedings of the First Autonomous Agents and Multiagent Systems Conference (AAMAS), 2002.

Nair, R., Tambe, M., Marsella, S., and Raines, R. Automated assistants for analyzing team behaviors. Journal of Autonomous Agents and Multiagent Systems (JAAMAS), 2002. (to appear)

Scerri, P., Pynadath, D., and Tambe, M. Why the elf acted autonomously: Towards a theory of adjustable autonomy. In the First Autonomous Agents and Multiagent Systems Conference (AAMAS), 2002.

Tambe, M. 1997 Towards Flexible Teamwork *Journal of Artificial Intelligence Research, Volume 7, Pages 83-124*

Summary of the Report:
Teamwork Reasoning
and
Multi-Satellite Missions

NASA Agreement
Investigating Teamwork Among Spacecraft
Number
NAG 2-1489
Period of Performance: 7/1/01- 4/30/02

Stacy C. Marsella
Information Sciences Institute
University of Southern California

NASA is rapidly moving towards the use of spatially distributed multiple satellites operating in near Earth orbit and Deep Space. The satellites will be required to cooperate with each other as a *team* that must achieve common objectives with a high degree of autonomy from ground based operations. Such satellite teams will be able to perform spatially separated, synchronized observations that are currently not feasible in single satellite missions. Autonomous operations will reduce the need for ground-based support that would otherwise be prohibitively expensive in such missions. However, the underlying control systems necessary to enable such missions will raise many new challenges in autonomous, multi-platform operations.

In particular, a critical requirement for these satellite constellations is that they must act coherently as a coordinated, at times autonomous team, even in the face of unanticipated events such as observation opportunities or equipment failures. Further, the satellites will need to take actions that will not only impact the constellation's current tasks but may also impact subsequent tasks, an issue that is particularly relevant given the often long duration of some missions and the limited power and fuel resources available to each satellite. Overall, the ability to operate as a team will need to be satisfied in many of the multi-satellite missions being planned. Therefore, it is important to understand this requirement, elucidate the research challenges it presents and consider approaches to satisfying it.

The multi-agent research community has made considerable progress in investigating the challenges of realizing such teamwork. In the full report, we discuss some of the teamwork issues that will be faced by multi-satellite operations. In particular, we discuss the Magnetospheric Multiscale mission (MMS) to explore Earth's magnetosphere. We describe this mission and then consider how multi-agent technologies might be applied to improve the design and operation of such missions.

Specifically, the report illuminates several basic issues. It discusses the need to develop robust and effective coordination techniques for multi-satellite teamwork. Rather than mission-by-mission ad hoc approaches to coordination, we focus on a general approach to teamwork that will be both more robust in a particular mission while also building across mission, teamwork-technology infrastructure. We also stress the need for analysis and suggest a formal approach to assessing the quality of alternative coordination techniques, based on the MTDP (*Multi-agent Team Decision Problem*) framework that allows both formal and empirical evaluation. We illustrate how this approach could be applied to MMS's science operations and discuss how it could be extended to provide a faithful rendering of difficult resource limits that such missions will operate under. In addition, we discuss alternatives to realizing the teamwork reasoning and how teamwork and autonomy is integrated into a craft's overall software architecture.

MTDP provides a tool to address a range of analyses critical to fielding teams in real world applications. Using the MTDP framework, the complexity of deriving optimal teamwork policies across various classes of problem domains can be determined. The framework also provides a means of contrasting the optimality of alternative approaches to key teamwork issues like role replacement and communication. Finally, the framework allows us to empirically analyze a specific problem domain or application of interest. To that end, a suite of domain independent algorithms has been developed in prior work that allows a problem domain to be cast into the MTDP framework. This allows the empirical comparison of alternative teamwork approaches in that domain. Derivation of the optimal policy for the problem domain serves not only as the basis of comparison but also can inform the design of more practical policies. Most recently, progress is being made in addressing how real world operating constraints like power consumption can be modeled in this framework.

But of course, teamwork and autonomy reasoning are just one part of the multi-satellite team's operation, which must include various flying, observation, communication and maintenance tasks over the duration of the mission. So, the report also discusses the supervisory control software that manages and schedules these tasks. In particular, we discuss one approach to the design of this supervisory software and the integration of teamwork reasoning within this supervisory control software.

The report makes several recommendations for the future of the research and also potential collaborations with NASA. In particular, it is suggests that the formal MTDP work could play a key role in analyzing designs for distributed satellite missions. MTDP formalisms could be used in the analysis of algorithms or informing the design of new algorithms. For example, the best-case, worst-case and average case analyses that the MTDP models support could be of critical assistance in the design and development of any real-world, high-cost application of multi-agent systems. In terms of NASA missions, the formal analyses could be performed entirely within the MTDP framework, resulting in tested and improved teamwork prescriptions. Alternatively, the MTDP framework could be realized by software modules that are incorporated into ongoing NASA Goddard work in distributed satellite simulation.

As a first step towards applying this MTDP framework to the problem of designing better satellite teams, we propose to cast an example NASA satellite constellation problem, specifically MMS, into the MTDP framework. This will allow us to evaluate alternative approaches to teamwork and adjustable autonomy as well as contrast them with optimal policies.

As NASA embarks on developing multi-satellite missions, we believe it is important to explore general approaches to teamwork reasoning and analysis from the start. We believe this is true even in early multi-satellite missions that may seemingly require minimal teamwork coordination. For example, it may seem that early missions will be simple enough that they will not require general architectures for teamwork or extensive analysis of alternative coordination schemes. However, ad hoc coordination schemes that address specific coordination tasks as special cases are too brittle. This conclusion has come to the multi-agent community through hard-earned experience. Quite simply, human designers cannot think of every way coordination can break down, so there is always another special case rule to add. Further, it ends up being more time consuming and costly to come up with the host of ad hoc rules. Finally, by incorporating general teamwork reasoning and analysis early on, these early multi-satellite missions could lay critical groundwork that could be exploited in later even more ambitious missions.