Organization and Reorganization of Autonomous Oceanographic Sampling Networks*

Roy M. Turner & Elise H. Turner
Department of Computer Science
University of Maine
Orono, ME 04469
{rmt,eht}@umcs.maine.edu

Abstract

As robots and other autonomous agents become more widespread, there will be a greater need for them to work cooperatively to accomplish complex, long-duration tasks. Such groups of agents will often be highly heterogeneous and *open*: that is, agents will enter or leave the system while the task is being performed. This presents a formidable problem for organizing and controlling the agents. In this paper, we present an approach to this problem for autonomous oceanographic sampling networks (AOSNs), which are groups of robots and sensor platforms that cooperate to sample an area over a long period of time. The approach uses two organizations to control the agents. A *task-level organization* (TLO) controls the system during the actual mission, and a *meta-level organization* (MLO) is responsible for self-organization of the system, design of the TLO to fit the situation, and reorganization as necessary. This approach allows the TLO to be highly efficient, while allowing the MLO to give the overall system a great deal of flexibility in adapting to change.

I Introduction

Autonomous oceanographic sampling networks (AOSNs) [Curtin *et al.*, 1993] are multi-robot systems being developed to collect data from the ocean over long periods of time. During deployment, the AOSN may alter its task in response to previously-collected data or new instructions from scientists. An AOSN will be composed of a wide variety of components, including autonomous underwater vehicles (AUVs), remotely-operated vehicles (ROVs), and non-mobile instrument platforms. Because underwater robots are so expensive to develop, and because so few exist, components of the AOSN will include AUVs and ROVs that have been loaned to the AOSN by undersea robotics laboratories.

Like all multi-agent systems, the AOSN requires a mechanism for organizing its components. However, AOSNs have several characteristics which make existing approaches for organization inappropriate. We believe these characteristics will be shared by an increasing number of systems as more robots become available and are used to perform more complex missions over longer periods of time. These characteristics are:

Organization and reorganization must be autonomous. Finding the appropriate organization for a system requires knowledge of organizational theory as well as knowledge of the components of the system. Because the system will interact with "users" instead of "operators," we do not want to burden the user with having to define the organization. In addition, we would like the system to be able to operate without any user intervention. Since the situation may change and reorganization may be required at any time, the system must be able to reorganize autonomously.

The system will be composed of a wide variety of agents. The system will be composed of robots with a wide range of capabilities depending on the available robots and the tasks which the system is to perform. Consequently, there will be many more differences between robots than in most previous heterogeneous systems. In particular, robots will have different levels of intelligence: some will be able to reason about the organization while others will have only the intelligence required to perform some specific task.

The composition of the system will change. Because the system will be deployed for long periods, we expect robots to come and go. Robots will leave the system due to failure or because they are needed
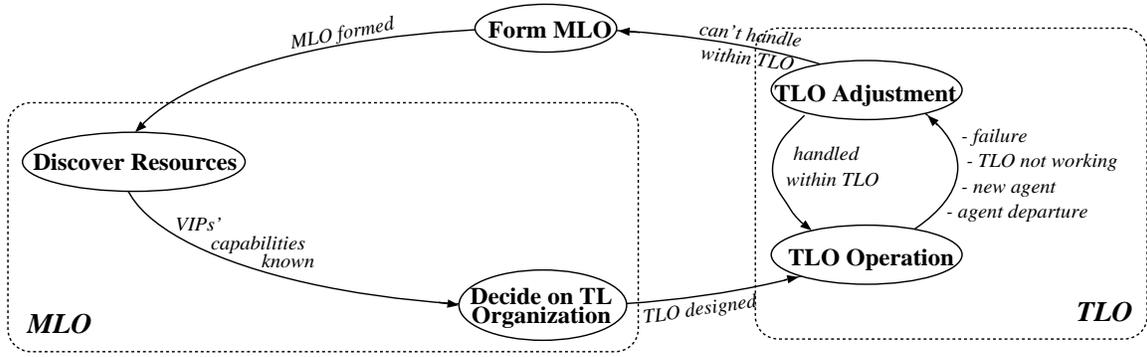
Figure 1: A two-level approach to AOSN control.

outside of the AOSN. Robots may also join the system when they are no longer needed elsewhere. In addition, new robots may be developed which are added to the system.

The tasks of the system may change. Whenever a system interacts with the real world, its task may change in response to unexpected features of the environment or in response to the system's own activities. A task may be successfully completed or the results of one task may cause another to be created. For example, exploratory robots may be able to determine when they have found something interesting and, based on that information, generate a task to explore that area in more detail. The user may also change the assigned task of the system. This is particularly likely for systems that are deployed for long periods, because the situation, the users' goals, or the users may change.

Communication is severely limited. In the ocean, acoustic communication is essentially the only option for communication that does not require breaking the surface. Current acoustic modems have baud rates significantly below that of off-the-shelf telephone modems. Many other domains and tasks also have communication constraints. For example, in covert operations, it is essential that communication be kept to a minimum. Even domains that have seemingly unlimited communication have some constraints on communication, including processing constraints which limit the amount of information that the agent can understand, as well as its ability to perform other tasks while communicating.

Excellent performance is expected. As robotics research advances, users will have elevated expectations about the performance of the system. Systems, at a minimum, will have to perform their task effectively. In addition, the systems should be efficient and reliable. For example, an AOSN will need

to guarantee its users high levels of data quality and coverage. To these ends, there must be some global control to ensure efficiency and monitor expected results.

Existing CDPS approaches fall short on addressing one or more of these characteristics. For example, Partial Global Planning (PGP) [Durfee & Lesser, 1987] requires all agents to have sophisticated problem-solving abilities, and it does not take the sort of global perspective necessary to ensure data quality. The Contract Net Protocol (CNP) [Smith, 1980] also fails to take a global approach to designing the organization it creates, and it is not clear how well it handles the failure or exit of mid-level or top-level managers. Multiagent planning approaches [e.g., Cammarata *et al.*, 1983; Georgeff, 1983], though taking a global perspective, do not cope well with the possibility of the planning agent(s) failing or leaving the system. Limited bandwidth is a problem for almost all approaches. In addition, most existing approaches suffer from a trade-off between flexibility and efficiency. Efficient approaches are not particularly adaptable when the situation changes, and flexible approaches are not as efficient as might be desired. A middle-ground approach is worse than either: the system will be neither adaptive enough nor efficient enough.

We have developed a two-level approach to controlling AOSNs to address these problems. Two organizations are used to control the agents. A *task-level organization* (TLO) controls the actual mission-related tasks, and a *meta-level organization* (MLO) is used to design as efficient a TLO as possible. Only the robots in the MLO are required to reason about the organization. The MLO self-organizes and can take a global perspective to design a TLO. The TLO then conducts the mission until there is a change beyond its capability to adapt. At that point, the MLO again
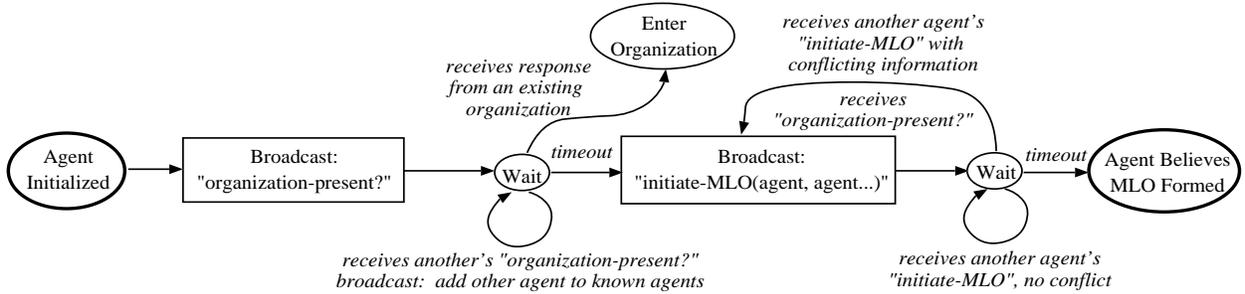
Figure 2: MLO formation protocol.

regains control to reorganize the system by repairing or designing a new TLO. This provides both efficiency and flexibility.

In the remainder of this paper, we first discuss the approach in general, then look more closely at task assignment and at how reorganization takes place in the system. We then describe the simulation testbed in which this work is being developed.

II A Two-Level Approach to Organizing CDPS Systems

Our overall approach is shown in Figure 1. The operation of the system is partitioned into two general phases, one controlled by the meta-level organization and one controlled by the task-level organization. Within these are smaller phases. All activities of the system and the agents are controlled by *protocols* that dictate acceptable actions given the situation and the phase the system is in. All vehicles and instrument platforms (*VIPs*) that participate in the system must abide by the protocols or a subset of the protocols appropriate for their level of participation. Thus those that can participate in the MLO must be able to follow most of the protocols, while those that can participate only in the TLO must follow the TLO protocols relevant to them.

A Meta-Level Organization

The meta-level organization is composed of those agents present that have the capability to reason about which organization is appropriate for the mission. This means that these agents, called *MLO agents*, must be fairly sophisticated. The MLO, then, will contain a subset of the VIPs present.

When first deployed, the task facing the AOSN's agents is to self-organize into a meta-level organization capable of designing the task-level organization. We assume that the system may not initially have much knowledge of itself. This could happen, for example, if the VIPs were deployed by air drop, in which case not all may survive, or if they were supposed to rendezvous at a location, in which case not all may have arrived. Thus, the MLO cannot be created *a priori*, nor should it rely on the presence of particular VIPs.

Consequently, our MLO formation protocol makes few assumptions about which agents will be present. The agents capable of participating in the MLO (called *MLO agents*) each carry out a simple sequence of actions designed to identify their peers. Figure 2 shows the protocol used for forming the MLO.[1] The result is a "flat" meta-level organization in which each member can communicate with all the others and no single agent is in charge. As long as there is a single MLO agent present, the MLO can form and the mission can continue.

The exchange of information specified by the MLO-formation protocol establishes common knowledge of the MLO's membership and location of the agents. The next task is to determine the total capabilities available to the AOSN to conduct its task. This is done by the MLO discovery protocol (see Figure 3), which guides the MLO agents in querying non-MLO VIPs about their abilities.

We treat each agent as a black box with advertised capabilities [see, e.g., Turner *et al.*, 1993]. This simplifies designing the TLO, as it abstracts away unnecessary detail about the specifics of the agent's implementation. It also facilitates interoperability of disparate agents, since a "vocabulary" of capabilities provides a common interface to the agents and a standard way of communicating about them. In many ways, the benefits are similar to those from object-oriented programming. This representation of agents means that, practically speaking, any VIP can participate in the system as long as it is capable of respond-

---

[1] In this and the other protocol diagrams, only the major pathways are shown for clarity.
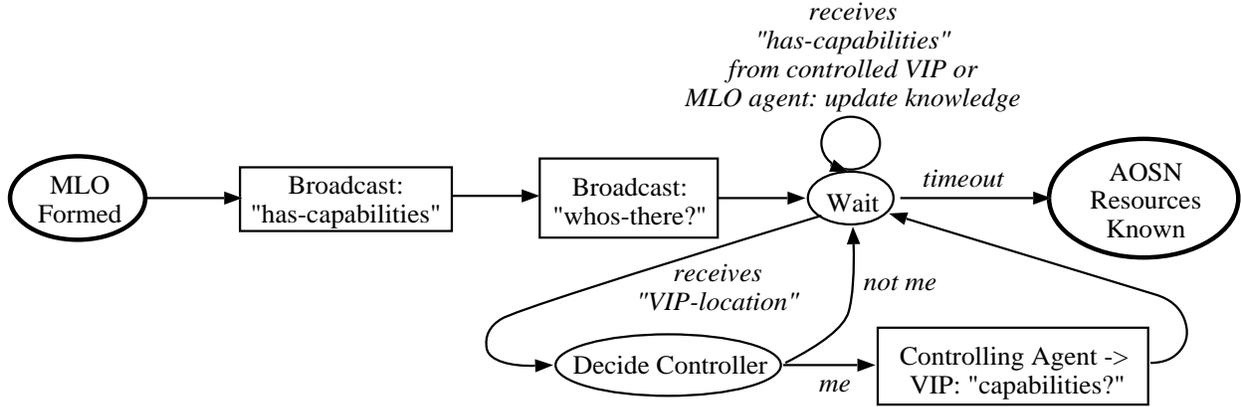
Figure 3: MLO discovery protocol.

ing appropriately to queries about its capabilities.

At the end of the protocol shown in Figure 3, the MLO knows all the capabilities of all agents in the system. To reduce bandwidth requirements, our current MLO organization distributes responsibility for this knowledge among the MLO agents. Each MLO agent is responsible for knowing about the capabilities of those VIPs closest to it.

At this point, the MLO can proceed to design the TLO. The current protocol, shown in Figure 4, calls for a single planner to be selected by convention from among the MLO agents. The planner gathers information from its peers about capabilities related to accomplishing the mission at hand, then it designs a task-level organization. In the future, we will examine how and when to distribute the planning task among multiple MLO agents.

We currently restrict the kinds of organizations considered for the TLO to be hierarchies. Hierarchical organizational structures have several benefits, including efficiency of communication [Malone, 1987], effectiveness in the face of uncertainty [Fox, 1981], and the possibility of global coherence of actions, since there is a top-level manager. Even with this restriction, however, the problem of designing a TLO is rich enough to be interesting, since hierarchies can vary along a number of axes, including the kind and timing of communication allowed, number of levels, and whether or not peers are allowed to cooperate outside the management structure. In the future, we plan to broaden our consideration to include other kinds of task-level organizational structures.

The planner designs the TLO by matching VIP capabilities to mission tasks, creating the hierarchy's management structure, and assigning VIPs to management roles, based again on their advertised ca-

pabilities. Our first implementation of task assignment used a very simple first-fit mechanism, and the assignment of management roles is done using simple heuristics (e.g., pick a manager from among the agents working on a task, if possible). The second version is based on constrained heuristic search [Fox et al., 1989], in which task assignment is treated as a constraint satisfaction problem. This is described in more detail in Section III. In the future, we intend to examine the feasibility of using an enhanced version of the distributed constrained heuristic search algorithm [Sycara et al., 1991] for task assignment to allow the entire MLO to participate. We will also extend the approach to organizational design in two ways. First, we are developing a representation for organizations that will allow an agent to analyze the situation, then quickly choose an organization type and specific parameters based on the situation's features. Second, we will ultimately examine the feasibility of incorporating management roles into the "tasks" assigned by CHS.

Once the TLO is designed, the planner informs the new managers of their roles and whom they control and tells the top-level manager to begin work. The planner then sends a message to its peers in the MLO informing them that the MLO is now dissolved.

B Task-Level Organization

Once formed, the task-level organization controls the AOSN until the mission is complete or there is a change severe enough to be beyond the TLO's ability to accommodate.

Figure 5 shows the major protocol used in hierarchical TLOs. When a TLO manager receives notification of its assignment, it analyzes itself and its subordinates to identify any slack resources (e.g., unused
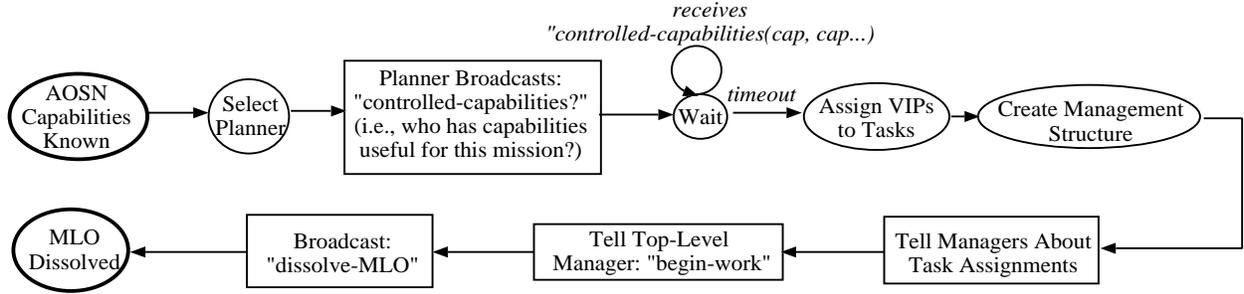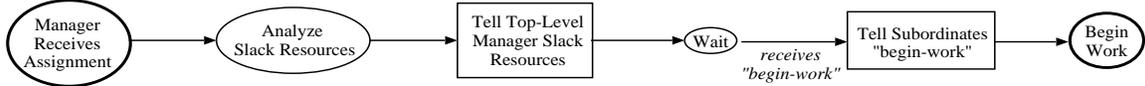
Figure 4: TLO formation protocol.

Figure 5: A TLO protocol.

capabilities). This information is passed to the TLO's top-level manager, which keeps a record of the slack capabilities and VIPs present in the system. This gives the TLO a limited ability to respond to changes in the task or its own composition, as discussed in Section IV. When the manager receives a "begin work" message, it passes it along to its subordinates (which pass it along to their subordinates, etc.) and begins work on its own task(s).

We have not yet defined protocols governing mission-related aspects of the AOSN control such as communication between manager and subordinate during work on tasks. To a large extent, this will depend on the specifics of the task, manager, and subordinate. Protocols will be needed, however, to allow the managers of the TLO to gather and maintain current information about the state of the TLO and the mission. This will be addressed in future work.

III   Task Assignment

Task assignment identifies the VIP which will perform each required task. The task assignment algorithm receives a task decomposition tree as input. A task decomposition tree is a standard representation used by problem solvers that can be produced for a given mission by the planner selected for TLO formation. The tree represents all of the alternative methods for carrying out the assigned mission and the capabilities required by each alternative. Only a VIP that has some capability can be assigned to deliver that capability, and no VIP can be assigned tasks in excess of its resources.

Task assignment is currently implemented using a variant of *constrained heuristic search (CHS)* [Fox *et al.*, 1989] that has been extended for task decomposition trees [Turner & Turner, in press]. CHS was chosen as the basis for our task assignment algorithm because it efficiently finds a solution if one exists. CHS ties constraint satisfaction to heuristic search by supplying operators which add constraints and variables to the constraint graph. The alternative methods for carrying out the task are selected using heuristics which indicate how likely it is that an assignment can be found within the resource constraints of the VIP. The selected capabilities are placed in a constraint graph as variables whose values are constrained by the resource limitations of the VIPs. Then constraint satisfaction techniques are used to find a solution.

IV   Handling Change

Changes will occur during the operation of an AOSN that may require reorganizing the system. We are currently focusing on changes in composition of the AOSN. Agents will fail during long duration missions. In systems such as AOSNs, in which robots may be on loan to the system, agents will also need to exit the system as they are needed elsewhere. In addition, as robots become available, it is desirable to allow them to enter the system to augment its resources.

If an agent has time and sufficient knowledge, it may be able to gracefully leave the system by notifying the MLO or TLO, whichever currently exists. A protocol for doing this in the context of the TLO is shown in Figure 6. This protocol is specific for hierarchical TLOs and requires the top-level manager to maintain a list of slack resources present in the sys-
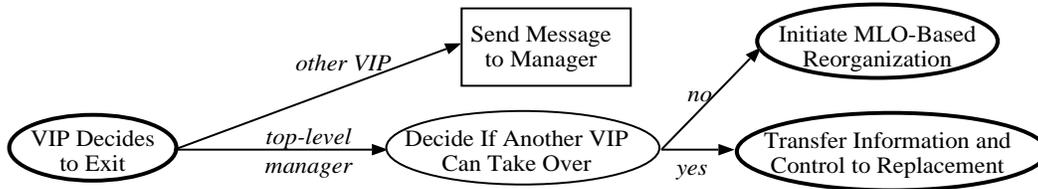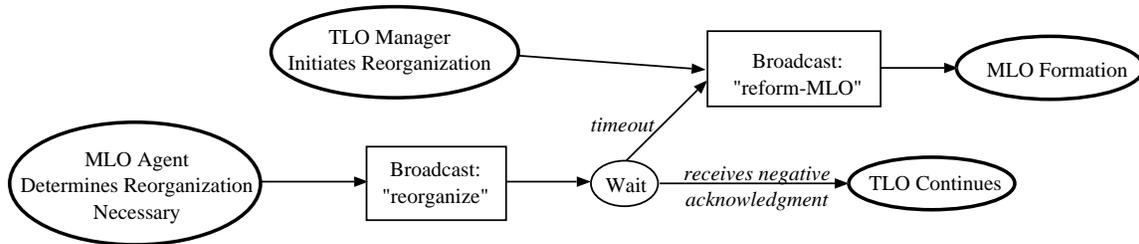
Figure 6: Exit protocol.

Figure 7: A reorganization protocol.

tem. If the top-level manager itself needs to leave, it can consult this list to determine if there is another agent present that can take over for it. If not, then the manager has exhausted the TLO's ability to adapt and initiates reorganization via re-forming an MLO, as discussed below. If the exiting agent is not the top-level manager, it signals its manager and, if that manager has knowledge about resources under its control that can be assigned to replace the agent, it replaces the agent and passes information up the hierarchy about the change. Otherwise, it notifies its own manager about the agent's intention to exit and lets that manager attempt to handle the change in the context of its more global viewpoint. Ultimately, if no other manager can handle the change, then the top-level manager will attempt to replace the exiting agent. If it cannot, it will initiate reorganization by the MLO.

It would be overly optimistic to assume that in all (or even most) cases, agents can exit gracefully. When an agent fails, it often will not have the time or the ability to signal its manager. In such cases, the agent's manager and peers will have to determine that the agent has failed. This is, in general, a very difficult problem that we will address in future work. Once failure is detected, however, it can be treated similarly to a more graceful exit.

The entry of new agents is also important. A new agent's arrival will not force the TLO or MLO to reorganize, but the availability of its capabilities will often provide an opportunity to achieve a task more effectively or to work on a task impossible to achieve

previously. For example, at the time the TLO was created, perhaps the only vehicle present that could take data samples of a particular kind was one whose navigational precision was poor enough that some data points would be missed. Later, if a vehicle with the required sensor and a much better navigational system enters the AOSN, it would make sense to assign it to take over the old agent's task.

Our agent entry protocols are sufficiently complex, due to the need to handle agent entry in any phase of AOSN operation, that space does not permit a discussion here. They are, however, discussed elsewhere [Chappell *et al.*, 1997].

The task-level organization will not always be able to accommodate to agents entering or leaving, since it is meant to be efficient, not flexible. In these cases, the MLO regains control to reorganize the system. Other sorts of changes can also lead to reorganization. For example, the environment may change significantly, the mission can be changed by a user, or a task can fail. In these cases, some agent or agents in the TLO must detect the change. The agent can then either send a message up the hierarchy or attempt to initiate a reorganization without the participation of the top-level manager. The latter may be the case if the agent detecting the change has more information or is more intelligent than the top-level manager.

Figure 7 shows a reorganization protocol. When the top-level manager requests a reorganization, the MLO is immediately re-formed. Other agents follow a different protocol. First, they broadcast a message to ensure that all agents agree to the reorganization. If

so, the MLO is re-formed. This protocol is tentative, and will be changed as work progresses.

MLO re-formation is identical to its initial formation. This is because the situation will likely have changed significantly since the last time the MLO existed. Consequently, we cannot rely on the presence of any of the MLO agents that participated previously. In the future, we will evaluate the usefulness of allowing the MLO to remain in existence while the TLO is controlling the system. This will likely decrease the time necessary to reorganize at the expense of increased bandwidth utilization during the mission.

V THE SIMULATOR

A simulation testbed was created in which to develop and evaluate the approach described above. Instead of building the usual sort of AI simulator, in which the decision-making processes of individual agents is simulated in detail, we chose to create a rule-based simulator that focuses on the aggregate properties of a group of agents following the protocols. This allows us to focus on the protocols rather than on how the agents are implemented.

This approach has several advantages. Development time is saved by not concerning ourselves with the internal decision-making of agents. Time that would normally be spent designing and programming the agents is instead available for developing and evaluating the protocols. This is also in keeping with the approach being simulated, since this does not over-commit to the kinds of agents that will be in the simulated system: agents are treated as black boxes with well-defined behavior (i.e., as governed by the protocols). The rule-based approach makes it very easy to implement new protocols or change existing ones, since protocols are readily implemented as sets of rules. This allows rapid testing of new ideas in the simulator. A very important benefit of this kind of simulator is that we can concentrate on evaluating the protocols without worrying about how the particular decision-making processes of the agents impacted the results. This type of simulator also allows higher-fidelity rules and algorithms to replace lower-fidelity versions as they become available.

The simulator is written in the CLIPS [Giarratano, 1993] forward-chaining rule-based system. There are approximately 270 rules currently in the simulator that implement the protocol simulation itself, the environment, vehicle motion, and discrete-event simulation capabilities. In addition, the simulator includes C and Lisp code. The CLIPS and C portions of the

```
;; MLO formation phase:
00:00:00 (SIM) new agent EAVE-Ariel is broadcasting organization-present?
     message.
00:00:00 (SIM) new agent EAVE-Arista is broadcasting organization-present?
     message.
;; [...]
00:00:05 (MLO) EAVE-Ariel: received organization-present? message from
     EAVE-Arista
00:00:30 (MLO) EAVE-Ariel believes it has waited long enough for replies.
00:00:30 (MLO) EAVE-Ariel is initiating MLO formation with agents =
     (EAVE-Arista EAVE-Ariel)
;; MLO discovery phase:
00:00:35 (MLO) Agents are attempting to discover other VIPs.
00:00:35 (MLO) EAVE-Arista --> EAVE-Ariel: I have capability(ies) (CDPS
     survey-side-scan-sonar transit search acoustic-link manage manage
     manage manage manage manage), and I'm at location (10 10 10).
;; [...]
00:00:40 (MLO) mooring-Able broadcasting: I am at (0 0 0).
00:00:40 (MLO) mooring-Baker broadcasting: I am at (200 0 0).
;; [...]
00:01:35 (MLO) Closest MLO agent EAVE-Ariel now controls mooring-Able
00:01:35 (MLO) EAVE-Ariel --> mooring-Able: tell me your capabilities.
;; [...]
00:01:40 (MLO) mooring-Able -> EAVE-Ariel: I have capabilities (radio
     acoustic-link LBL).
;; [...]
00:01:45 (MLO) MLO formation complete.
;; TLO design phase:
00:01:45 (MLO) Selecting EAVE-Ariel as planner (convention: first in MLO).
00:01:45 (MLO) Planner EAVE-Ariel querying others about capabilities
     they may contribute for tasks: (background-survey LBL3 LBL2 LBL1
     communication-relay convex).
;; [...]
00:02:00 (MLO) EAVE-Arista --> EAVE-Ariel: these agents may work: (AUV
     mooring-Charlie mooring-Baker mooring-Delta CONVEX-mooring
     EAVE-Arista).
;; [...]
00:02:05 (MLO) EAVE-Arista manages mooring-Able.
00:02:05 (MLO) EAVE-Arista manages EAVE-Ariel.
00:02:05 (MLO) EAVE-Arista manages mooring-Delta.
00:02:05 (MLO) EAVE-Arista manages mooring-Charlie.
00:02:05 (MLO)  EAVE-Ariel manages CONVEX-mooring.
00:02:05 (MLO)  EAVE-Ariel manages AUV.
00:02:05 (MLO)  EAVE-Ariel manages mooring-Baker.
00:02:05 (MLO)  mooring-Able is working on task LBL1.
00:02:05 (MLO)  mooring-Baker is working on task LBL3.
00:02:05 (MLO)  mooring-Charlie is working on task LBL2.
00:02:05 (MLO)  CONVEX-mooring is working on task convex.
00:02:05 (MLO)  mooring-Delta is working on task communication-relay.
00:02:05 (MLO)  AUV is working on task background-survey.
00:02:05 (MLO) EAVE-Ariel -> all: dissolve meta-level organization.
```

Figure 8: Example simulator output.

simulator are available on the Web.[2] Figure 8 shows example output from the simulator.

VI CONCLUSION AND FUTURE WORK

As robots and other autonomous agents become more widely used, there will be a greater need for them to work cooperatively to accomplish complex, long-duration tasks. Often these systems of agents will have characteristics and requirements similar to those of autonomous oceanographic sampling networks: autonomous organization/reorganization will be needed,

---

[2] http://cdps.umcs.maine.edu/MAUV

the system will be highly heterogeneous, its composition and tasks will change over time, communication bandwidth will be limited, and rigorous performance measures will have to be satisfied.

In this paper, we have discussed our approach to controlling such systems. We have developed a two-level organizational approach that addresses concerns of flexibility and efficiency without trading them off against one another. The task-level organization can be designed to be highly efficient, though inflexible, with the meta-level organization stepping in to reorganize when the situation changes.

We are currently developing and evaluating protocols that implement this approach. Future work will refine and extend the protocols based on the results of empirical simulation studies that are currently in progress. We will explore distributing the task assignment mechanisms across multiple MLO agents. Characteristics of organizational structures will be delineated, along with features of situations in which they are appropriate. This will allow the development of a mechanism for quickly selecting an appropriate organizational structure based on the current situation. In the longer term, we anticipate testing this approach in a fielded AOSN.

REFERENCES

Bond, A. & Gasser, L. (1988). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, Los Altos, California.

Cammarata, S., McArthur, D., & Steeb, R. (1983). Strategies of cooperation in distributed problem solving. In *Proceedings of the 1983 International Joint Conference on Artificial Intelligence*, pages 767–770.

Chappell, S. G., Turner, R. M., Turner, E. H., & Grunden, C. M. (1997). Cooperative behavior in an autonomous oceanographic sampling network: MAUV Project update. In *Proceedings of the 10th International Symposium on Unmanned Untethered Submersible Technology (UUST)*.

Curtin, T., Bellingham, J., Catipovic, J., & Webb, D. (1993). Autonomous oceanographic sampling networks. *Oceanography*, 6(3).

Durfee, E. H. & Lesser, V. R. (1987). Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the 1987 International Joint Conference on Artificial Intelligence*, pages 875–883.

Fox, M. S. (1981). An organizational view of distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, 11:70–80.

Fox, M. S., Sadeh, N., & Baykan, C. (1989). Constrained heuristic search. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*.

Georgeff, M. P. (1983). Communication and interaction in multiagent planning. In *Proceedings of the 1983 Conference of the American Association for Artificial Intelligence*, pages 125–129. (Reprinted in [Bond & Gasser, 1988].)

Giarratano, J. C. (1993). *CLIPS User's Guide*. NASA, Information Systems Directorate, Software Technology Branch, Lyndon B. Johnson Space Center, Houston, TX.

Malone, T. W. (1987). Modeling coordination in organizations and markets. *Management Science*, 33(10):1317–1332.

Smith, R. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C–29(12):1104–1113.

Sycara, K., Roth, S., Sadeh, N., & Fox, M. (1991). Distributed constrained heuristic search. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1446–1461.

Turner, E. H. & Turner, R. M. (in press). A constraint-based approach to assigning system components to tasks. To appear in the *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*.

Turner, R. M., Blidberg, D. R., Chappell, S. G., & Jalbert, J. C. (1993). Generic behaviors: An approach to modularity in intelligent systems control. In *Proceedings of the 8th International Symposium on Unmanned Untethered Submersible Technology (AUV'93)*, Durham, New Hampshire.