Appears in the *Proceedings of the 18th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Portsmouth, NH, August, 2013. © 2013 The Autonomous Undersea Systems Institute (AUSI) and the authors. It is posted here for your personal use: no redistribution. The definitive version appears in the conference proceedings, published by AUSI.

DISTRIBUTED, CONTEXT-BASED ORGANIZATION AND REORGANIZATION OF MULTI-AUV SYSTEMS

David Gagne
Department of Computer Science
University of Southern Maine, Portland, ME
david.gagne1@maine.edu

Sonia Rode and Roy M. Turner*
School of Computing and Information Science
University of Maine, Orono, ME
sonia.rode@maine.edu/rturner@maine.edu

Abstract

Many tasks requiring multiple autonomous underwater vehicles (AUVs) are simple, with static goals, of short duration, and require few AUVs, often of the same type. Simple coordination mechanisms that assign roles to AUVs before the mission are sufficient for these multi-AUV systems. However, for tasks that are complex and dynamic, of long duration (implying that AUVs will come and go during the mission), and that have many heterogeneous AUVs, *a priori* organization of the system will not work. In addition, due to changes in the situation, the system will likely need to be reorganized during the mission.

We are developing a distributed, context-aware self-organization/reorganization scheme for advanced multi-AUV systems. This is a two-level approach in which a meta-level organization first self-organizes, assesses the context, and uses contextual knowledge to design a task-level organization appropriate for the context that can then carry out the mission. We are extending our prior work by distributing both the context assessment process and the organization design process. The result will be a system that can self-organize efficiently and effectively for its context and that can reorganize appropriately as the context changes.

*Authors listed in alphabetical order. Corresponding author is Roy M. Turner, School of Computing and Information Science, University of Maine, Orono, ME 04469.

Introduction

Many tasks requiring multiple autonomous underwater vehicles (AUVs) are rather simple, with static goals, a relatively static environment, and known, homogeneous vehicles. For these tasks, simple coordination mechanisms are appropriate, and an organization for the vehicles can be designed *a priori*, then the vehicles fielded.

However, there are tasks that are more challenging with respect to organizing, reorganizing, and controlling the operation of a multi-AUV system. Some characteristics of this class of tasks include: complex, dynamic goals that may change throughout the course of the mission; a dynamic, possibly little-known environment; and a group of AUVs that may be heterogeneous, open (with vehicles coming and going over time), and, in fact, whose composition may not be completely known ahead of time.

For tasks of this type, it is not possible to create an organization for the vehicles ahead of time and to give them plans to carry out together. Instead, the group would itself have to be responsible for its own operation, organizing itself to use whatever resources (i.e., vehicles and their capabilities) are available, carrying out mission tasks, and reorganizing as the situation changes.

We earlier proposed (including in this symposium series) an overall approach toward accomplishing autonomous organization/reorganization of multi-AUV systems, CoDA (Cooperative Distributed AOSN[1] control) [Turner and Turner, 2005, 2001].

[1] Autonomous oceanographic sampling network [Curtin et al., 1993].

Agents follow a set of *coordination protocols*, with variants for different kinds of agents (e.g., those with more or less intelligence), that allow them to work together to self-organize a heterogeneous set of AUVs and other instrument packages into an efficient multiagent system for accomplishing a mission and to reorganize as the situation changes.

Initial work on CoDA focused on the protocols themselves and how they could be used, with relatively little attention to organization design per se [Turner and Turner, 2001]. As a first cut, it was assumed that a single agent would be identified to perform organization design. Subsequent work began to look at the context-dependent nature of organization design [Turner and Turner, 2005] and how organizations could be selected based on *a priori* contextual knowledge. However, little attention was given to distributing either of the processes of context assessment or organization design.

There are significant advantages to distributed versus centralized context assessment and organization design. As in the case of all distributed systems, distributing the work can lessen the cognitive load on any one agent, speed up the process, and ameliorate problems arising from having a single point of failure in a centralized system, with agent failure instead causing a graceful degradation of the system's performance. In addition, there are advantages specific to intelligent systems. It is often difficult if not impossible for a centralized scheme to have a global view of the entire situation [e.g., Lesser and Corkill, 1981], whereas the system as a whole can have a more global perspective on the overall system's context, since the system encompasses all agents' knowledge of the world. Indeed, given that some agents may have knowledge of contexts that others do not and that in our approach multiple known contexts can be used to characterize a novel context, distributing context assessment can allow the system to recognize contexts that no single agent could itself recognize, which allows the system to behave more appropriately for the context. Different agents will also likely have different knowledge and abilities pertaining to organization, which will allow a better organization to be designed than if any single agent were responsible. And to the extent that the task of organization is decomposable into "sub-organizations" (as can be the case in hierarchies, for example), pieces of the organization task can be parceled out to individual agents, thus minimizing both the effort required by any single agent as well as the message traffic needed (since work on the subtasks can be localized to the agents with knowledge needed to accomplish them).

This paper discusses recent work on extending our approach by introducing distributed context assessment and, especially, distributed organization self-design. The work reported is ongoing, and consequently much of what is discussed is somewhat preliminary. We first provide an overview of the problem of self-organization/reorganization for advanced multi-AUV systems and of the CoDA approach. We then look at the role of context assessment in organizational design. We next discuss how context assessment and organization design can be distributed across a multi-AUV system. Finally, we discuss the project's current status and future work.

Self-Organization/Reorganization

Simple multiagent AUV systems, although presenting some challenging problems, in general are relatively straightforward to organize. If the AUVs will all persist in the system until the conclusion of the mission, and if the mission itself is not complex or dynamic, then the behavior of the agents with respect to one another can be specified ahead of time by the system designers/controllers, including the roles the agents play (e.g., master/slave, sensor platform/data aggregator, etc.) and the communication pathways and types. If the system is in contact with humans, then even if the mission or the environment is dynamic, the humans can still modify the systems' organization without the need for on-board organization design capabilities. Many, if not most, current multi-AUV systems are of this type [e.g., Sotzing et al., 2007; Li et al., 2010; Kemp et al., 2002].[2]

However, if we look ahead a few years to more advanced multi-AUV systems, *a priori* organization and human-directed reorganization will not work. In some cases, the composition of the system will change over time (i.e., the system will be *open*) as AUVs come and go due to (e.g.) failure. In other cases, the environment in which the AUVs are to be deployed will be changing and/or largely unknown. These problems will be worse for long-term missions. And in some missions, communication with humans will be largely impossible (e.g., under sea ice) or ill-advised (a covert mission).

As an example, consider the problem of using a multi-AUV system when a plane goes down in a remote, hostile, or inaccessible ocean region, for

[2] But cf. Carlési et al. [2011].

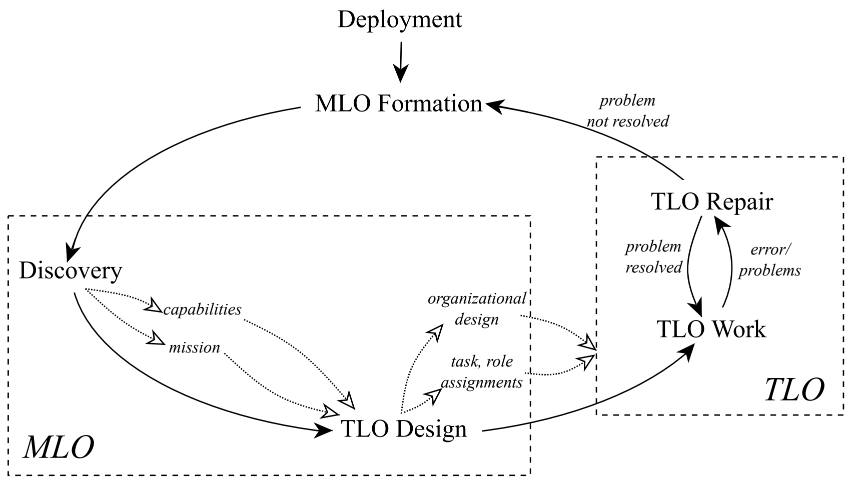

Figure 1: Overall CoDA approach.

example, the North Atlantic, where using surface ships or airplanes would be infeasible. The crash site needs to be found, any survivors identified and rescued, the debris field characterized (e.g., to help determine the cause of the crash), and the "black boxes" found. Which tasks the system would need to work on would change during the course of the mission based on what is found at the site. The environment may be unknown to any level of detail. Which AUVs are present may change as AUVs leave or enter the system. The initial composition itself may not be known ahead of time. Since dedicating AUVs to this would likely not make sense, given the infrequent occurrence of crashes and the cost of AUVs and their operation, it is likely the group of AUVs would be put together as needed, drawing on the combined resources of the community, including government, industry, and academic AUVs. Thus the group would ultimately consist of a heterogeneous group of AUVs, the membership of which cannot be predicted ahead of time—indeed, given the problems involved in delivering the AUVs to the site (e.g., via air drop, surface vessel, submarine launch, or autonomous transit), the exact composition of the group would not be known until the mission begins. And, finally, given the remote location, likely sea state, and possible presence of storms, the group would have to function autonomously.

In this scenario, organization cannot be done ahead of time, since it is unknown what the situation or even the system composition will be before the vehicles arrive at the mission site. Given the dynamic nature of an open system such as this, as well as the dynamic and unknown situation, it is likely that the initial organization will need to be changed as vehicles leave and enter the system, and yet the system may not be able to communicate with humans for help. Consequently, a control scheme for multi-AUV systems of this sort will need to be able to autonomously self-organize and to reorganize as the situation and system changes.

CoDA

The CoDA project was initially focused on intelligent control of autonomous oceanographic sampling networks [Curtin et al., 1993], a kind of multiagent system (MAS) composed of AUVs and other instrument platforms that can return long-term data from an area of interest.

We recognized early in the project that controlling advanced multiagent systems has two opposing criteria: flexibility, so that the system can self-organize easily from whichever agents happen to be present; and efficiency, so that the system accomplishes its mission goals in a timely manner using the capabilities it has while remaining within any resource constraints. While this could be seen as a tradeoff to be addressed in the design of the control mechanism, this would result in a multiagent system that was neither truly flexible nor truly efficient.

A better approach results from realizing that flexibility and efficiency are needed at different points during the operation of the system. The control mechanism needs to be most flexible when there is no organization yet present, so that the agents that are present can easily cooperate with each other

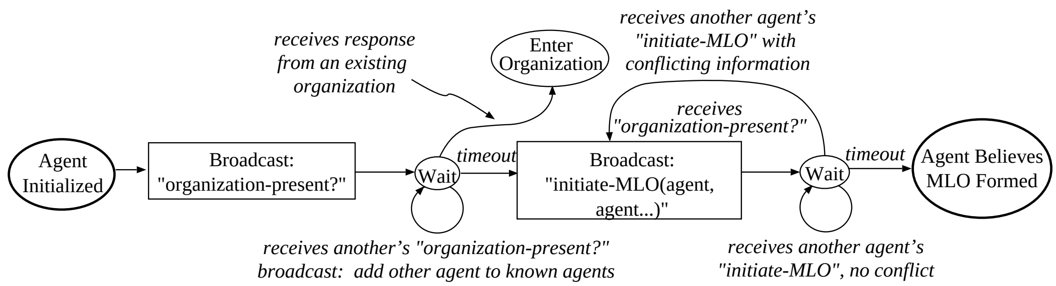

Figure 2: A protocol for MLO formation (after [Turner & Turner, 2001]).

with no need for much (if any) *a priori* knowledge, and when the situation or the system has changed enough to require reorganization. On the other hand, the control mechanism needs to be efficient when the system is actually carrying out the mission goals. This led to the design of a two-level organization mechanism in CoDA, shown in Figure 1.

Initially, the system self-organizes into a *meta-level organization* (MLO), a loosely-coupled organization created from a subset of the agents that are present (the "MLO agents") that have the intelligence needed to follow the MLO coordination protocols. The purpose of the MLO is to discover the capabilities and resources available in the entire system as well as to agree on the mission, then to design a *task-level organization* (TLO) to actually carry out the mission. Since the TLO is designed by the MLO, which has all the available knowledge about the environment, the mission, and the capabilities of all the agents present, it can be specifically designed to be very efficient for the situation. In the original approach, the MLO transitions control to the TLO and disbands.

CoDA relies on all agents sharing coordination protocols that they follow to participate in the multiagent system. Figure 2 shows one of CoDA's protocols, in this case, the protocol that an MLO agent follows during MLO formation. A simple agent, such as an instrumented buoy, would follow one set of protocols (e.g., one that would cause it to respond to simple queries), whereas a more intelligent agent, such as an AUV, would follow another (e.g., one to actually participate in the system's organization).

The task-level organization can be designed with some ability to handle small changes in the situation. However, the best-laid plans of MAS and men can go awry, and so there will often be changes that are beyond the TLO's ability to adapt—the TLO will no longer be a good design for the situation, in other words. When this happens, the MLO is reformed and can either repair the TLO or design a new one for the changed situation.

Context-Based Organization Design

There are many different possibilities for organizing a group of agents, including: static hierarchies of various kinds [e.g., Malone, 1987; Fox, 1981]; dynamic hierarchies, such as created by the Contract Net Protocol [Smith, 1980]; teams [Tambe, 1997]; committees; coordination structures created by partial global planning [Durfee and Lesser, 1987]; consensus-based organizations; various organizations created by collaborative planning [Grosz and Kraus, 1996]; and various auction schemes [e.g., Sandholm and Huai, 2000]. There is no one best organization type. Instead, each kind of organization has properties (e.g., communication overhead, requirements on agent sophistication, span of control, tolerance of uncertainty, etc.) that are advantageous for some situations and disadvantageous for others.

Consequently, an important aspect of CoDA is designing an appropriate task-level organization for the given situation and designing a new one when necessitated by changes in the situation. This can be done from first principles, for example, by analyzing elements of the situation and comparing them to properties of known organization types. However, this can be time-consuming. Possibly more important, there is a limited ability to handle special cases where a particular organization has been found to be good for a given kind of situation, but it is not known (to the system, at least) why that is so. This could arise, for example, if the system has been told by humans that the organization is good for a kind of situation, or if the organization has been found by trial and error by the system in the past to be good.

In CoDA, organization design is context-based.

That is, the situation is first identified as a known kind of situation in a process of *context assessment*. Then, knowledge about the context is used to select organizational structures (e.g., hierarchies, etc.) that can be instantiated for the current situation. This can speed organization design by shortcutting the reasoning required to match organizations with the situation, and it can compensate for missing knowledge or the need for idiosyncratic organization–situation pairings: the contextual knowledge can specify an appropriate organization type directly for the situation. And, as the system gains experience using organizations it has designed, it can update the contextual knowledge with knowledge of how they performed for the kind of situation.

There are drawbacks to this approach, of course. Context assessment, which is similar to what is often referred to as situation assessment [e.g., Sagatun, 1989] in the AUV literature, requires effort. And by prescribing a particular organization design, others that from-scratch reasoning might have selected are not considered. The first problem is addressed in part in our approach by using memory retrieval mechanisms that are fast [e.g., Lawton et al., 1999]. And, if the agents happen to themselves be controlled by context-aware reasoners, such as Orca [Turner, 1995, 1998], then context assessment is already being done. The second problem is more difficult, and is similar to functional fixedness in humans. However, truly bad pairings will ultimately be detected as failures occur, and some optimization of context-based organization selection will occur as reasoning is done to instantiate the organizational structures suggested by the context. A third problem—coming up with organizations for novel situations—is addressed by falling back on from-scratch reasoning in the worst case, but most often by merging several different known contexts the situation resembles to arrive at suggestions for the organizational structures for the current situation.

CoDA makes use of a style of context-based reasoning called *context-mediated behavior* [Turner, 1998], as shown in Figure 3. In this approach, contexts are represented explicitly by knowledge structures called *contextual schemas* (c-schemas) that both describe a class of situations as well as provide knowledge about appropriate behavior for the context. To assess the context, an agent first finds known contexts that the features of the current situation evokes, then uses differential diagnosis to compare and contrast them. The result is a set of one or more c-schemas that each match the current situation. These are then merged into a *context representation* (CoRe) for a complete view of the current context. The CoRe contains information that the agent can use to predict as yet unseen situational features, understand sensory input, and change the semantics of its knowledge representation. It also contains prescriptive knowledge to use to handle unanticipated events, focus attention, select ways to achieve goals, and to modify behavioral parameters ("standing orders"[3]).

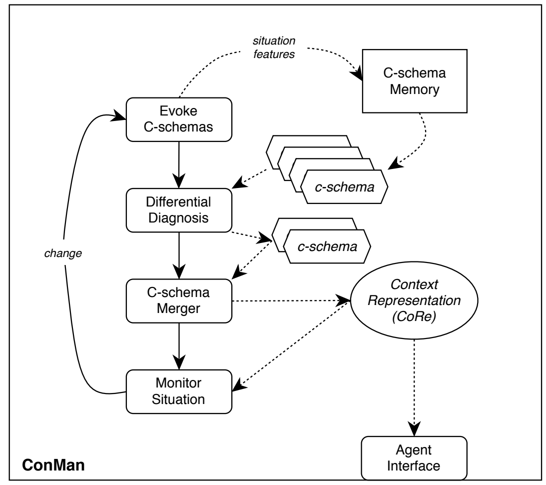

Figure 3: Context-mediated behavior. (From [Turner et al., submitted].)

In CoDA, c-schemas also contain suggested organizational structures that are predicted to be appropriate in the context. For example, in a context where there is reasonable point-to-point communication bandwidth, the need for rapid response of agents in carrying out actions, and little uncertainty, a hierarchy might be suggested, whereas if there is broadcast capability, high uncertainty and a dynamic environment, and some self-interest among the agents, then something like the contract net or other contracting schemes might be recommended.

Since the CoRe is possibly a composite of multiple c-schemas, each may suggest different organizational structures. This is not necessarily a bad thing, as the system can either pick from among them based on its domain knowledge or merge them to create a highly-tailored organizational structure, for example, one that is a hierarchy overall, yet with local groups collaborating as peers to achieve goals.

[3]Thanks to D.R. Blidberg for the term.

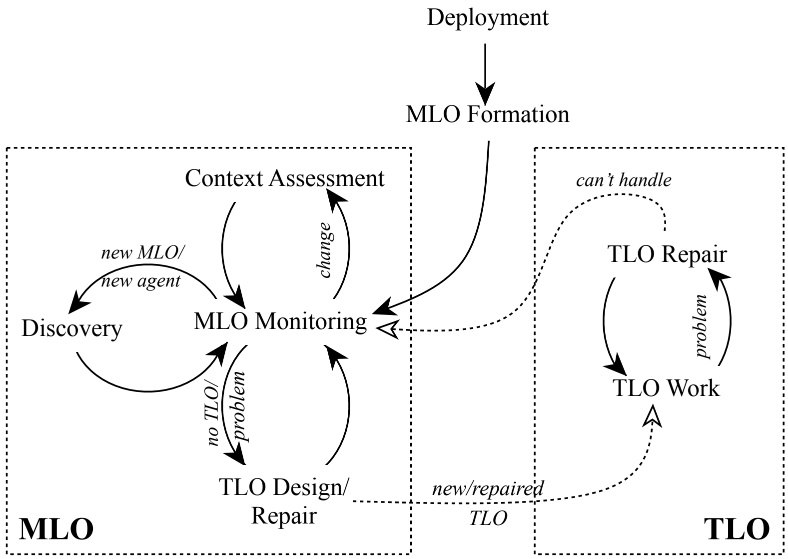

Figure 4: Distributed, context-based organization design.

Distributed Organization Process Overview

For distributed organization design, the overall process shown in Figure 1 is modified as shown in Figure 4. The meta-level organization now assumes a greater role than previously. It now not only designs and repairs the task-level organization, but is also responsible for maintaining a shared notion of what the current context is. Also, instead of disbanding while the TLO is working, the MLO now remains in existence to continuously monitor and assess the context as the situation changes. This allows it to more quickly respond to the need to repair or redesign the TLO, since it will not itself have to reorganize. It also allows the MLO the ability to critique its TLO design based on the evolving context, and thus to suggest context-appropriate changes to the design in a way the TLO, not necessarily being context-aware, cannot.

Once the AUVs are deployed, they follow protocols much the same as before to self-organize into a loosely-coupled MLO. The MLO will first note that it has not discovered all of the system's agents and capabilities, and so it will enter a discovery phase much like before.

At this point, the MLO will assess the context, based on its knowledge of the mission, the environment, and the agents and their capabilities. This process is distributed across the MLO, as discussed below. Once the context has been assessed and a common context representation created, the MLO makes use of organizational design knowledge in the CoRe in order to create a TLO that is appropriate for the situation. This process, too, is distributed across the MLO, as discussed below.

The MLO then initiates the TLO, which begins work on the mission. The MLO remains active in a "background" processing mode to assess the context as necessary and to handle the arrival of new agents by learning about them through discovery and incorporating them into the MLO and/or TLO, as appropriate. The MLO agents are distinct from the other agents only in that they are sophisticated enough to handle the MLO protocols, and so they, too, are assigned roles in the TLO. Thus a goal of the continuing MLO processing is to minimize its effect on the TLO's work, both in terms of communication and processing.

Distributed Context Assessment

In the past, we have concentrated on context assessment by a single AUV in a multiagent system. However, this is not optimal for several reasons, including introducing a single point of failure, increased demands on the agent's computing resources, and increased message traffic to get knowledge from other agents in the system to where it is needed at the central assessor. As mentioned above, there is also the very real possibility that no single agent in the system can adequately assess the context due to the distribution of contextual knowledge

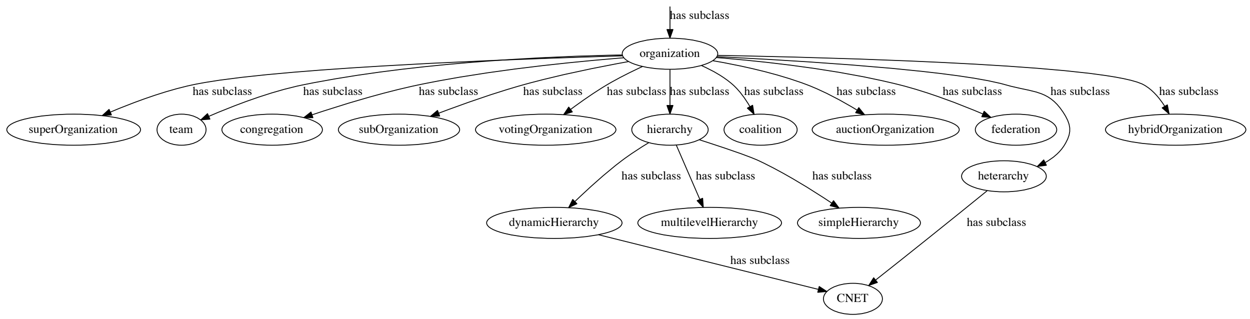

Figure 5: Part of an ontology of organizations.

among the agents, even if the system as a whole has the requisite knowledge to do so. Consequently, distributing the context assessment process across some or all of the agents in the system makes sense.

In order to distribute the process of context assessment, agents need to be able to communicate about contexts and contextual knowledge. Since CoDA is concerned with controlling open, likely heterogeneous systems, this means that all agents involved in context assessment must share a common communication language, a knowledge representation for contextual knowledge, and an ontology for contexts and contextual knowledge. There are many agent communication languages available, and we discuss the issues involved in shared representation and ontologies for context assessment elsewhere [Turner et al., submitted].

A first problem faced by the MLO is how to distribute the assessment process itself. Recall from Figure 3 that context assessment in our approach has several parts: evoking c-schemas potentially matching the current situation; differential diagnosis; and merging the resulting c-schemas to form the context representation. Each of these pieces can potentially be distributed across multiple agents, sometimes in multiple ways. For example, in a situation with limited bandwidth, it may make sense for some agents to take on entire tasks, such as context merger, to avoid message traffic; in other situations, there may be enough communication bandwidth to make use of all agents' expertise in all areas. In our approach, MLO agents that can themselves assess the context each engage in a "pre-assessment" to determine the best way, given the situation, to distribute context assessment; some communication and negotiation may be needed here, as well, to come to agreement, depending on the cooperation protocols in use.

Assuming that all parts of the process are distributed, then the MLO agents will together evoke a set of candidate c-schemas matching the current situation. This will be done by the agents each coming up with their own set, and then communicating and negotiating to arrive at the final set. A problem arises here in determining which c-schemas from different agents actually represent the same context; this can be partially resolved by recourse to the shared ontology, but as discussed by Turner et al. [submitted], it is somewhat more complicated than that.

Differential diagnosis itself involves creating *competitor sets* of c-schemas, each member of which explains/predicts roughly the same set of situational features, then "solving" each set by playing one c-schema against the others until one is a clear favorite (i.e., confidence in it exceeds some threshold amount beyond the others) [Miller et al., 1982]. Both of these processes can be distributed in multiple ways. For example, competitor sets could be formed by negotiating between all agents, or by pairs of agents exchanging competitor sets and resolving differences until a global view has crystallized (cf. partial global planning [Durfee and Lesser, 1987]). Solving competitor sets can be fully distributed, or competitor sets can be assigned to different agents for solution.

Finally, the task of merging the remaining c-schemas to form a coherent context representation (CoRe) can be distributed. This, too, can be done in multiple ways, depending on the situation.

Distributed Organization Design

In CoDA, organization design depends heavily on context assessment. Knowledge from the CoRe is used by the MLO to determine the overall kind of organizational structure(s) to use as well as how to instantiate the structure(s).

The CoRe provides suggested types of organiza-

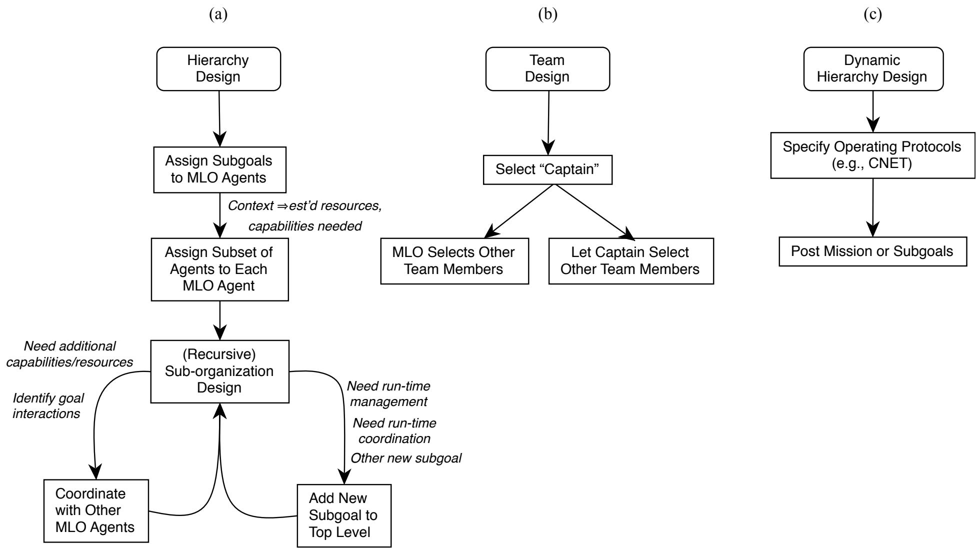

Figure 6: Instantiating organizational structures: (a) a hierarchy; (b) a team; (c) a dynamic hierarchy.

tional structures for the current situation, based on knowledge about the context. Since multiple c-schemas form the CoRe, there may be multiple suggestions, and thus the first task facing the MLO is determining which to use.

In our approach, this is done by the MLO as a whole. In the best case, all agents will agree about which to use. However, different MLO agents will have some different views of the situation, even given the shared CoRe, due to their local knowledge gained from sensing the environment. Consequently, deciding on the organizational structure will still involve some negotiation.

To facilitate negotiation, agents are assumed to share an ontology of organizations (see Figure 5). Part of this shared knowledge will be knowledge about the properties of organizations, including such things as their needs for communication bandwidth, cognitive abilities of participants, tolerance of uncertainty, and so forth. The MLO agents can make use of this knowledge in deciding which organizational structure to use based on the CoRe and their own idiosyncratic knowledge.

It may be that the final outcome of negotiation is not a unique organizational structure (e.g., a hierarchy), but multiple ones. This would be the case if two or more organizational structures fit different aspects of the situation. In this case, it may be that a hybrid organizational structure should be created. For example, some organizations, such as universities, are hybrids: there is an overall hierarchical organizational structure, but departments are run more or less as committees or teams, with only some of the authority vested in the department chair. We leave hybrid organizational structure design for future work.

Once agreement has been reached about the overall organizational structure to use, then the MLO needs to instantiate that structure given the available agents and their capabilities. The way this is done, as well as how this is distributed, will be different depending on the kind of organizational structure to be instantiated.

Figure 6 shows examples of how three different organizational structures could be instantiated in a distributed manner. Part (a) of the figure shows one way that a hierarchy could be instantiated. If the mission task naturally has subgoals (subtasks), or if it can be decomposed (e.g., via planning techniques) into subgoals, then the MLO can identify an agent to be the overall manager, then assign MLO agents to create a sub-organization for a each subgoal. This can be done recursively, involving more MLO agents, until the entire hierarchy structure is determined and AUVs assigned to roles. As shown in the diagram, if an MLO agent realizes that additional resources are needed, or that there will be interactions with its subgoal and others, then it can communicate with the other MLO agents to coordinate the sub-organization designs. If an agent realizes that a subgoal it is working on needs run-time management or coordination during the mission, then it can generate a new subgoal that can then be worked on to add management or coordination roles to the hierarchy by adding additional levels.

Part (b) of the figure shows a simple distributed design for a team organization. The MLO agents can negotiate to determine which AUV would likely be the best captain for the team, then they can decide which other agents to add to the team, or they could delegate this to the team captain, if it has sufficient sophistication to do so.

Part (c) of the figure shows how a dynamic hierarchy could be created, that is, one that can change its structure during the TLO work phase. One such hierarchy is created by contracting, for example, by the Contract Net Protocols (CNP) [Smith, 1980]. The MLO can together decide which agents and protocols to use (e.g., CNP), then, based on the protocols, make sure the goals get to the "organization". For CNP, this would entail either identifying an overall contractor and giving it the mission to achieve, or identifying several agents and giving them the subgoals, with the MLO itself monitoring the overall mission performance. Guidance for which alternative to use would come in part from the current contextual knowledge as represented in the CoRe.

There are many other organizational structures: heterarchies, federations, congregations, voting organizations, auction-based organizations, coalitions, consensus-based organizations, and hybrids of these [see, e.g., Horling and Lesser, 2004]. The MLO will need different protocols and mechanisms for each. So far, we are concentrating on the three organizational structures mentioned above.

Current Status and Future Work

The CoDA protocols and overall two-level organization scheme are well worked-out, as is single-agent context-mediated behavior. We have only recently begun work on the distributed version of each, as described above. At the current time, the CoDA simulator [Turner and Turner, 2000] is being redesigned and reimplemented to allow development and testing of the distributed organization design. In parallel with this, we will be implementing and refining the distributed context-based organization mechanism discussed in this paper.

References

Carlési, N., Michel, F., Jouvencel, B., and Ferber, J. (2011). Generic architecture for multi-auv cooperation based on a multi-agent reactive organizational approach. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 5041–5047. IEEE.

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

Durfee, E. H. and 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.

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

Horling, B. and Lesser, V. (2004). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4):281–316.

Kemp, M., Hobson, B., Meyer, J., Moody, R., Pinnix, H., and Schulz, B. (2002). Masa: a multi-auv underwater search and data acquisition system. In *OCEANS'02 MTS/IEEE*, volume 1, pages 311–315. IEEE.

Lawton, J. H., Turner, R. M., and Turner, E. H. (1999). A unified long-term memory system. In *Proceedings of the International Conference on Case-Based Reasoning (ICCBR'99)*, Monastery Seeon, Munich, Germany.

Lesser, V. R. and Corkill, D. D. (1981). Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC–11(1):81–96.

Li, H., Popa, A., Thibault, C., Trentini, M., and Seto, M. (2010). A software framework for multi-agent control of multiple autonomous underwater vehicles for underwater mine counter-measures. In *Autonomous and Intelligent Systems (AIS), 2010 International Conference on*, pages 1–6. IEEE.

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

Miller, R. A., Pople, H. E., and Myers, J. D. (1982). INTERNIST–1, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, 307:468–476.

Sagatun, S. I. (1989). A situation assessment system for the MSEL EAVE–III AUVs. In

Proceedings of the Sixth International Symposium on Unmanned Untethered Submersible Technology, Durham, New Hampshire.

Sandholm, T. and Huai, Q. (2000). Nomad: mobile agent system for an internet-based auction house. *Internet Computing, IEEE*, 4(2):80–86.

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.

Sotzing, C. C., Evans, J., and Lane, D. M. (2007). A multi-agent architecture to increase coordination efficiency in multi-auv operations. In *OCEANS 2007-Europe*, pages 1–6. IEEE.

Tambe, M. (1997). Agent architectures for flexible, practical teamwork. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 22–28.

Turner, R. M. (1995). Orca: Intelligent adaptive reasoning for autonomous underwater vehicle control. In *Proceedings of the FLAIRS–95 International Workshop on Intelligent Adaptive Systems*, pages 52–62, Melbourne, Florida.

Turner, R. M. (1998). Context-mediated behavior for intelligent agents. *International Journal of Human–Computer Studies*, 48(3):307–330.

Turner, R. M., Rode, S., and Gagne, D. (submitted). Toward distributed context-mediated behavior for multiagent systems. Submitted to the 2013 International and Interdisciplinary Conference on Modeling and Using Context. Also available as a preprint at `Mainesail.umcs.maine.edu`.

Turner, R. M. and Turner, E. H. (2000). Simulating an autonomous oceanographic sampling network: A multi-fidelity approach to simulating systems of systems. In *Proceedings of the Conference of the IEEE Oceanic Engineering Society (OCEANS'2000)*, Providence, RI.

Turner, R. M. and Turner, E. H. (2001). A two-level, protocol-based approach to controlling autonomous oceanographic sampling networks. *IEEE Journal of Oceanic Engineering*, 26(4).

Turner, R. M. and Turner, E. H. (2005). Selecting organizational structures for advanced multi-AUV systems. In *Proceedings fo the Fourteenth International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Durham, NH.