UNSYS digital

Journal of Unmanned System Technology

CoDA: Decentralized, Context-Based Organization and Reorganization for Multi-AUV Systems

Roy M Turner[†], Sonia Rode[†], and David Gagne[§]

[†]School of Computing and Information Science, University of Maine Orono, United States

[§]BlueTarp Financial, Inc., United States

***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.

***Keywords*—**Autonomous underwater vehicles, multiagent systems, distributed organization, context-sensitive reasoning.

I. 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. An example of such a task is long-term monitoring and data gathering by an autonomous oceanographic sampling network (AOSN [1]) consisting of AUVs and other instrument platforms. The goals to be carried out may be complex and dynamic, changing as the scientists learn more about the area of interest or require data for different purposes. The environment itself will be dynamic and often little-known. The system may be *open*, meaning that AUVs can come and go, either due to failure or to new AUVs becoming available for the AOSN. The AUVs present may be heterogeneous and designed by different organizations; consequently, their capabilities and processing styles may not be known before they arrive at the work site.

All of this means that an organization for such a multi-AUV system cannot be created ahead of time, since the total capabilities of the system—indeed, which AUVs are even present—may not be known until the AUVs are on-site. The dynamic nature of the environment and mission as well as the system's composition will also prevent this, and it will cause the system to need a new organization as the situation changes.

Instead, the AUVs themselves will have to self-organize and reorganize cooperatively and as needed. The system will need to take into account the kind of situation it is in—its *context*—to determine the best kind of organization (hierarchy, consensus-based, contracting, etc.) to use, based on what it knows of the AUVs present, their capabilities and resources, the mission, and the environment. It will need to recognize when the situation has changed enough that it needs to reorganize. And, ideally, all the work of context recognition, organization, and reorganization will be done in a distributed manner so that there is no single point of failure.

The CoDA (Cooperative Distributed AOSN control) project is developing a distributed, context-sensitive approach to self-organization/reorganization of multi-AUV and other multiagent systems [2]. Initial work focused on developing protocols and mechanisms for self-organization and reorganization. Later work focused on using contextual knowledge for organization design, and current work is on distributing the processes of context assessment and organization to capitalize on all AUVs' knowledge and viewpoints and to avoid any single point of failure.

In this article, we first present a running example that we will use to motivate and ground our discussion of CoDA. Second, we describe CoDA's overall approach to multi-AUV system control. Third, we discuss context-based organization design, including context representation and assessment. Fourth, we describe current work aimed at decentralizing CoDA's organization design across multiple AUVs. Finally, we conclude and discuss future directions.

Corresponding author: Roy M. Turner (e-mail: rturner@maine.edu).
This paper was submitted on November 30, 2013; revised on February 14, 2014; and accepted on May 14, 2014.

I. An Example Multi-AUV System

There are many examples of tasks that can benefit or even require a multi-AUV system: autonomous oceanographic sampling, mine hunting, underwater construction or inspection, and so forth. In this section, we describe one task that is both important and difficult, yet that is beyond the capabilities of current multi-AUV approaches. This will provide a motivating example of the kind of multi-AUV systems and missions we hope to control using techniques developed in CoDA.

For our example, we consider the problem of using a multi-AUV system when a plane goes down in a remote, hostile, or inaccessible ocean region, for example, the North Atlantic. 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. Depending on the sea state and weather, using surface ships or airplanes for this will be infeasible.

Instead, we imagine a future scenario in which AUVs are used. Since AUVs are expensive and crashes thankfully few and far between, it would make sense to have AUVs that can be quickly allocated to the task, but that also have other duties. Perhaps some AUVs across the world operated by universities, research labs, and governments could be called upon when there is a crash.

Given the distances involved and the likelihood that some of the AUVs would be too busy to be freed up, the composition of the multi-AUV system would not be known until the AUVs arrive on site. Indeed, since they may arrive via several different means—e.g., transiting under their own power, being deployed from ships or submarines, or even dropped by air—with some means being more problematic and dangerous than others, there may be no way to know *a priori* which AUVs will be participating.

The goals the multi-AUV system focuses on will also change during the mission. For example, initially the goal might be to find the crash site, then it would switch to look for survivors. At some point, the goal would be to characterize the debris field, while looking for black boxes. Once a black box is located, a new goal to retrieve and return it would become active.

Since we cannot count on airplanes crashing conveniently close to populated areas in calm seas, the multi-AUV system will likely have to operate autonomously, at least most of the time. Communication might be curtailed by lack of safe access to the surface (high sea state, e.g., and/or high winds), or all the AUVs may be needed underwater, with none to spare for a relay between the underwater vehicles and shore.

Over time, vehicles that were delayed or that have been freed up from other tasks will arrive, and some vehicles that were part of the system will need to leave, either because they are needed elsewhere or because of equipment failure or power issues.

In this scenario, it is obvious that the system’s organization cannot be designed ahead of time, and it is likely that it will have to be organized and reorganized autonomously as the situation changes.

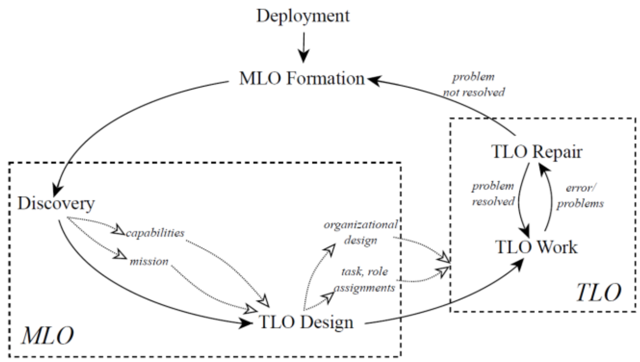

Figure 1 Overview of the CoDA two-level organization scheme (This and all figures from [15] used by permission.)

II. CoDA

The primary problem facing a group of AUVs arriving at the site of work is how to organize themselves into a useful multi-AUV system. There are some approaches to doing this in the multiagent systems literature, for example, using the Contract Net Protocol [3] to create a dynamic hierarchy or partial global planning [4] to create a partial plan for carrying out the mission. However, there are many more kinds of organizations possible (see, e.g., [5]), although some of them are not amenable to self-organization by themselves. The question is, which organization to use for the situation?

As in many other areas of engineering, there is in organization design a tradeoff between flexibility and efficiency. Very flexible organizations, such as committees or consensus-based organizations, are far from efficient, while efficient organizations, such as hierarchies, are not very flexible. Unfortunately, initially a nascent multi-AUV system will need a very flexible organization, one that makes few assumptions about knowledge of the agents present or their capabilities, but during the actual mission, it will need one that is highly efficient and tailored to the task at hand and the environment.

In CoDA, we take a two-level approach to organization, as shown in **Figure 1**. Initially, a very flexible organization is formed, called the *meta-level organization* (MLO). This is formed without needing to know ahead of time much about which AUVs are present or their capabilities. The first goal of this organization is to discover the capabilities of the system as a whole as well as to come to a joint understanding of what the mission is. This then serves to advance the organization’s primary goal: to create an efficient *task-level organization* (TLO) that will actually carry out the system’s mission.

The MLO is created by cooperative activity of a subset of the AUVs, namely the ones capable of following the requisite set of *cooperation protocols.* These “MLO agents” communicate to discover each other, their location, and their capabilities using a set of protocols that begins with the one shown in **Figure 2**.

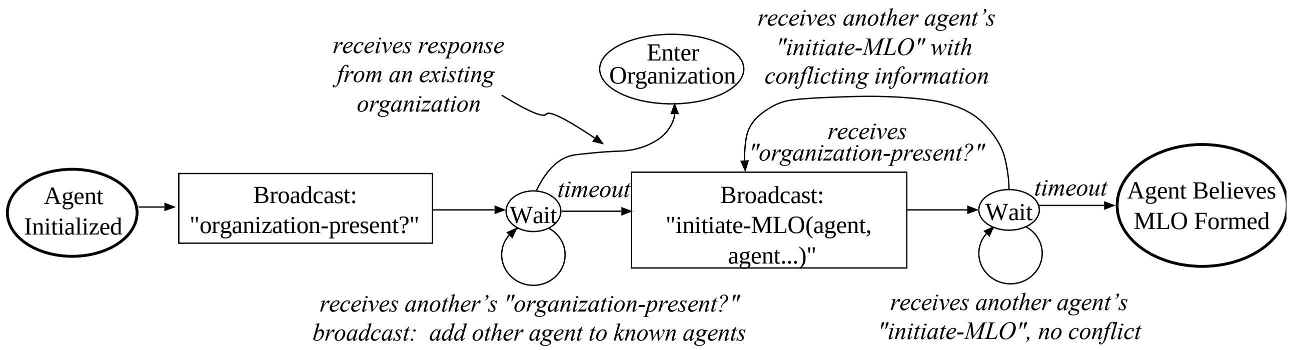

Figure 2 The MLO formation protocol (© [2001] IEEE. Reprinted, with permission, from [2].)

Consider our running example, and assume that several AUVs have arrived at the crash site at roughly the same time. Two of them, EAVE-Arista and EAVE-Ariel, are capable of forming an MLO, and others, along with various instrument platforms, are not. No agent knows about any other, nor about any possibly-existing organization (either MLO or TLO). Each agent attempts to find its peers by broadcasting (using an acoustic modem, in this example) an "organization-present?" message, with different variants for MLO agents and non-MLO agents. If there were an existing organization present, then that organization would reply, and the MLO agent would transition to a different protocol to enter the organization. However, no organization is present, so the AUV waits, listening for others' "organization-present?" messages, which serve also to announce an AUV's presence. Once the AUV has judged adequate time has passed (as specified in the protocol), then it assumes it knows all other agents present, and so sends an "initiate-MLO" message listing them to begin negotiation to create an MLO. All agents are doing this nearly simultaneously (when all agents arrive or are initialized nearly simultaneously), and so they hear each others' messages. They use this information to update their own knowledge about the agents present that could form an MLO or to correct others' when they detect a conflict. Finally, after adequate time has passed with no more corrections, all agents independently recognize that the MLO has effectively been formed and move into the next phase, discovering the system's total resources, which is governed by a different protocol. **Figure 3** shows sample output from the CoDA simulator during MLO formation.

Once the MLO has discovered the total resources available to it—all the AUVs, not just the MLO agents, and their capabilities—then it progresses to designing the TLO, which is described in more detail below.

In the current version of CoDA, once the TLO is designed and roles assigned to AUVs, then the MLO hands off control to it and disbands. The TLO then conducts the mission. Depending on the type of organization designed, the TLO may have the ability to handle some problems as they arise, as shown in **Figure 4**. In the example, a hierarchy has been formed, and it has some slack resources. Thus, the manager may be able to reassign an AUV to take over for another that must leave due to power failure. When another AUV arrives, it also must be integrated into the TLO, if possible. It could have needed resources, and so take a non-filled role or replace a suboptimal AUV in a role, or it could be held on a slack resources list for later use.

```
00:00:00.0 (MLO) new agent EAVE-Ariel broadcasting organization-present?.
00:00:00.0 (MLO) new agent EAVE-Ariel setting timer 1 to wait for replies.
00:00:00.0 (MLO) new agent EAVE-Arista broadcasting organization-present?.
00:00:00.0 (MLO) new agent EAVE-Arista setting timer 1 to wait for replies.
00:00:00.0 (MLO) new agent mooring-Able broadcasting (non-CDPS) organization-present?.
[...]
00:00:01.01 (MLO) EAVE-Arista: received organization-present? message from EAVE-Ariel
00:00:01.01 (MLO) EAVE-Ariel: received organization-present? message from EAVE-Arista
[...]
00:00:30.0 (MLO) EAVE-Ariel: waited long enough for organization-present? replies.
00:00:30.0 (MLO) EAVE-Ariel: initiating MLO formation with agents = (EAVE-Arista
EAVE-Ariel)
00:00:30.0 (MLO) EAVE-Ariel: broadcasting first initiate-MLO message.
00:00:30.0 (MLO) EAVE-Ariel: setting timer 2 to wait for replies.
00:00:30.0 (MLO) EAVE-Arista: waited long enough for organization-present? replies.
00:00:30.0 (MLO) EAVE-Arista: initiating MLO formation with agents = (EAVE-Ariel
EAVE-Arista)
00:00:30.0 (MLO) EAVE-Arista: broadcasting first initiate-MLO message.
00:00:30.0 (MLO) EAVE-Arista: setting timer 2 to wait for replies.
00:00:31.01 (MLO) EAVE-Arista: In wait 2: initiate MLO received; no conflict.
00:00:31.01 (MLO) EAVE-Ariel: In wait 2: initiate MLO received; no conflict.
00:01:00.0 (MLO) EAVE-Ariel: completed MLO formation.
```

Figure 3 Simulator output during MLO formation

```
00:03:01.64 (MLO)new agent Phoenix broadcasting (non-CDPS) organization-present?.
00:03:03.64 (TLO) Phoenix: received notification that TLO exists, managed by EAVE-Arista;
00:03:03.64 (TLO) sending identity, location, capabilities as requested.
00:03:04.66 (TLO) EAVE-Arista: received ID message from Phoenix; at (0,0,0),
00:03:04.66 (TLO) caps=(survey-magnetometer manage manage manage manage manage manage
manage).
00:03:04.66 Adding Phoenix to EAVE-Arista's knowledge about TLO. [...]
00:03:15.67 (TLO) Top-level manager (EAVE-Arista) received message from EAVE-Ariel that
mooring-Charlie has exited; attempting to repair TLO.
00:03:15.67 Attempting to repair TLO after exit of mooring-Charlie.
00:03:15.67 Affected roles:
00:03:15.67 Labor role LABOR-ROLE510 (mooring-Charlie using LBL (1 unit) for LBL3)
00:03:15.67 Building repair problem REPAIR35.
00:03:15.67 Agent assignment successful.
00:03:15.67 Assigning mooring-Able to LABOR-ROLE510, using capability LBL for task LBL3.
00:03:15.67 Repair successful; TLO updated with fix.
00:03:15.67 (TLO) EAVE-Arista -> mooring-Able: you now fill role LABOR-ROLE510.
00:03:15.67 (TLO) EAVE-Arista -> EAVE-Ariel: mooring-Able now fills role LABOR-ROLE510,
which you manage.
00:03:15.67 (TLO) EAVE-Arista: status of repair of exit of mooring-Charlie is SUCCESS.
00:03:01.64 (MLO) new agent Phoenix broadcasting (non-CDPS) organization-present?.
00:03:03.64 (TLO) Phoenix: received notification that TLO exists, managed by EAVE-Arista;
00:03:03.64 (TLO) sending identity, location, capabilities as requested.
00:03:04.66 (TLO) EAVE-Arista: received ID message from Phoenix; at (0,0,0),
00:03:04.66 (TLO) caps=(survey-magnetometer manage manage manage manage manage manage
manage).
00:03:04.66 Adding Phoenix to EAVE-Arista's knowledge about TLO. [...]
```

Figure 4 Output of TLO handling agent entry and repair

```
00:03:17.68 (TLO) ** Top-level manager EAVE-Arista exiting system **
00:03:27.68 (TLO) ** TLO has noticed that TL manager is not responding **
00:03:27.68 (TLO) ** Initiating reorganization **
00:03:57.68 (MLO) AUV is initiating MLO formation
00:03:57.68 (TLO) AUV->all: re-form-MLO
00:03:57.68 (MLO) new agent AUV broadcasting (non-CDPS) organization-present?.
00:03:58.68 (MLO) new agent AUV2 broadcasting (non-CDPS) organization-present?.
00:03:58.68 (MLO) new agent AUV3 broadcasting (non-CDPS) organization-present?.
00:03:58.68 (MLO) new agent AUV4 broadcasting (non-CDPS) organization-present?.
00:03:58.69 (MLO) new agent mooring-Delta broadcasting (non-CDPS)
          organization-present?.
00:03:58.73 (MLO) new agent EAVE-Ariel broadcasting organization-present?.
00:03:58.73 (MLO) new agent EAVE-Ariel setting timer 1 to wait for replies.
[...]
00:04:28.73 (MLO) EAVE-Ariel: waited long enough for organization-present? replies.
00:04:28.73 (MLO) EAVE-Ariel: initiating MLO formation with agents = (EAVE-Ariel)
00:04:28.73 (MLO) EAVE-Ariel: broadcasting first initiate-MLO message.
00:04:28.73 (MLO) EAVE-Ariel: setting timer 2 to wait for replies.
00:04:58.73 (MLO) EAVE-Ariel: completed MLO formation.
```

Figure 5 TLO failure and MLO re-formation

If problems occur beyond what the TLO can handle, or if the situation changes so that the TLO is no longer a good fit for the situation (noticed, e.g., by suboptimal performance or failure), then the MLO is reformed and repairs or redesigns the TLO based on the changed situation, as shown in **Figure 5**. In this case, while the TLO is conducting the mission, the top-level manager of the hierarchy, EAVE-Arista, fails. Another (non-MLO-capable, in this case) AUV notices a lack of response from it and initiates the reformation of the MLO, which will then re-design or repair the TLO.

III. CONTEXT-BASED ORGANIZATION DESIGN

In the initial version of CoDA, work was focused on developing cooperation protocols and task-assignment mechanisms [6]. Consequently, the only organization considered was a hierarchy, constructed based on the available resources to match the needs of the mission.

However, from the start it was realized that different situations call for different organizations. There are many different possibilities for organizing a group of agents, including: static hierarchies of various kinds (e.g., [7], [8]); dynamic hierarchies, such as created by the Contract Net Protocol [3]; teams [9]; committees; coordination structures created by partial global planning [4]; consensus-based organizations; various organizations created by collaborative planning [10]; and various auction schemes (e.g., [11]). 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.

Returning to our example of dealing with an airplane crash, if there are few AUVs available with substantial intelligence, but all agents have good communication abilities, then it may be that the best organization is a hierarchy that has the intelligent agents as high-level managers and the rest doing as they are told. On the other hand, if there are many intelligent AUVs, but communication is poor, the area is very large, or the environment is highly dynamic, then it may make more sense to design an organization more akin to a team of the intelligent agents, each of which has its own non-intelligent agents assigned to it to use in a simple hierarchy.

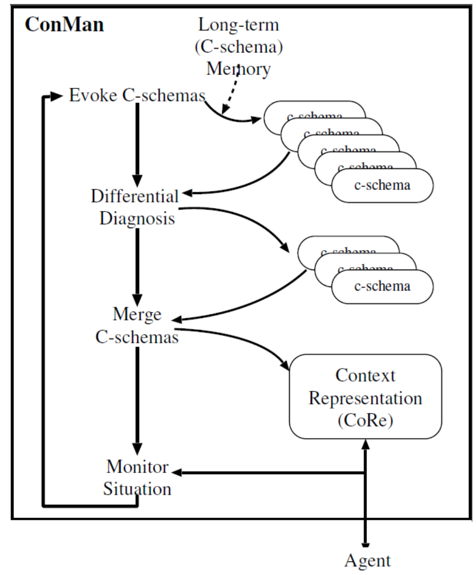

Figure 6 Context-mediated behavior process

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 to be good in the past by trial and error.

CoDA instead takes the approach of using contextual knowledge to direct the organization design process. CoDA extends a type of reasoning known as *context-mediated behavior* (CMB) [12] to work in the area of organization design.

In CMB, classes of situations that have implications for how an agent (e.g., an AUV) behaves are considered to be *contexts*. Contexts are explicitly represented by knowledge structures called *contextual schemas* (c-schemas). The CMB process for a single agent is shown in **Figure 6**.

The first step in CMB is context assessment, in which the current situation is identified as an instance of one or more known contexts. This is a differential diagnosis process, with the c-schemas that seem to superficially match the situation ("in the North Atlantic", "on a search and rescue mission", "looking for aircraft black boxes", "working with EAVE-Arista") first being *evoked* from schema memory, then grouped into competitor sets based on what is explained about the situation. Then strategies are used to gather information or make inferences with the goal of determining which c-schema in each set is the best explanation for the features explained. Thus, the diagnosed collection of c-schemas represents the context of which the situation is an instance.

The next step is to merge the information from these c-schemas to create a coherent picture of the current context called the context representation (CoRe). This allows a system to know how to behave in a situation even if the exact context has never been seen before. For example, an AUV participating in the example mission may have never been in the North Atlantic before on a similar mission. However, it may have c-schemas representing: being in cold, deep water; searching for objects; rescuing humans; and working with other AUVs in multi-AUV systems. These c-schemas can be merged into a CoRe to represent the current situation.

Context assessment is just the beginning, however. The goal in CMB is to give an agent (or in this case, a multi-AUV system) the knowledge it needs to behave appropriately in the context. Thus, c-schemas contain not only information characterizing contexts (*descriptive knowledge*), but also knowledge about how to behave in the context (*prescriptive knowledge*). This includes, in the single-AUV case, knowledge about: how to interpret new information; how to focus the AUV's attention on the appropriate goal(s) in the context; how best to achieve goals in the context; behavioral parameters ("standing orders"[1]), such as sensors to activate, depth envelope, etc.; and how to handle unanticipated events in the context (e.g., how to detect/diagnose, evaluate the importance of, and respond to them).

It is relatively straightforward to extend CMB to also guide CoDA to choose context-appropriate organizational structures during organization design. Contextual schemas, in addition to their usual prescriptive knowledge, can also contain knowledge about which organizational structures (hierarchies, committees, etc.) work well in the context. These can then be instantiated and modified by CoDA to create an organization that is appropriate for the current situation.

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, such as in our downed-aircraft example, and some self-interest among the agents, then something like the contract net or other contracting schemes might be recommended.

CMB 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., [13]) 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., [14]). And, if the agents happen to themselves be controlled by context-aware reasoners, such as Orca [12], then context assessment is already being done, and so the organizational knowledge is found essentially for free. 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 most often addressed by merging several different known contexts the situation resembles to arrive at suggestions for the organizational structures to use. In the worst case, the system falls back on from-scratch reasoning.

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 choose 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.

IV. Distributed Context-Based Organization Design

So far in CoDa, the work of organization design has not been distributed among the MLO agents except in the most rudimentary way. One of the agents is selected based on a convention (e.g., alphabetical order of name, first to initiate MLO, etc.) to create the task-level organization. That agent then designs the organization with minimal input from others and tells the rest of the system what the organization will be.

There are several problems with this approach. First, the selected agent is a single point of failure in a distributed system, which is rarely desirable. Second, all MLO agents are considered equally capable of being the TLO designer, which may not be the case—some may have specialized organizational design knowledge, for example, or others may have limited computational resources available for the problem. Third, all the knowledge necessary to design the organization has to be gotten to the agent selected, which may needlessly increase demands on or even exceed the capacity of the communication channel; consider that in our example problem, the agents will have to communicate acoustically, and that mode of communication can have extremely low bandwidth. And, finally, the process of organization design might be complex enough that it takes the single agent a long time to complete or even exceeds its computational capacity.

What is needed, then, is a way to distribute the work of organization design across multiple MLO agents. Given that we

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

take a context-based approach to organization design, this means that context-mediated behavior will itself need to be distributed.

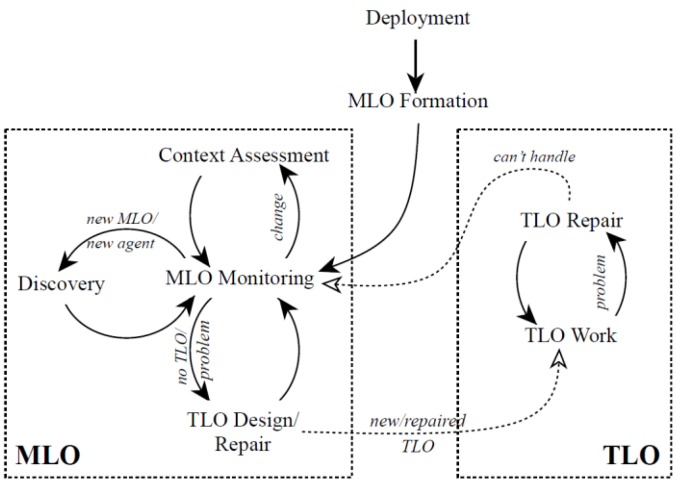

Figure 7 Distributed context-based organization design [15]

The overall process of distributed organization design is shown in **Figure 7**. It requires some changes in CoDA's prior approach. For example, 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

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. Since the MLO agents are distinct from the other agents only in that they are sophisticated enough to handle the MLO protocols, 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.

Context assessment 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, 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 [16].

A first problem faced by the MLO is how to distribute the assessment process itself. Recall from Section IV above 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, such as our example mission, it may make sense for some agents to take on entire tasks, such as context merger, to avoid message traffic; in other situations, for example when AUVs can surface and communicate by radio, there may be enough 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 elsewhere [16], it is somewhat more complicated.

Creating competitor sets of candidate c-schemas and "solving" each set by finding a clear favorite among its elements [17] can both 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 [4]). Solving competitor sets can be fully distributed, or competitor sets can be assigned to different agents for solution.

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

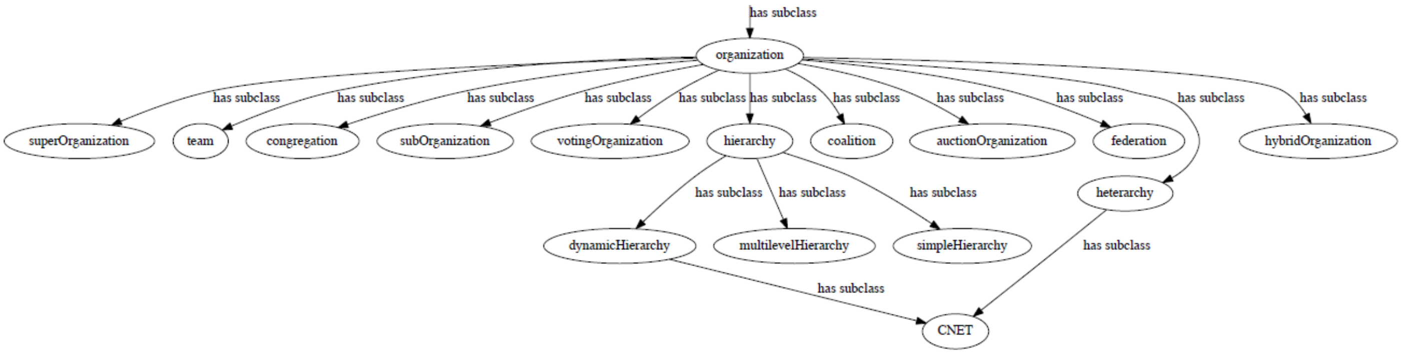

Figure 8 Part of an ontology of organizational structures [15]

Distributed organization design The CoRe will provide suggested types of organizational structures appropriate 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 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 8**). Part of this is shared 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.

Once agreement has been reached about the overall organizational structure to use, then the MLO needs to instantiate it 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 9 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 each subgoal (cf. [6]). This can be done recursively, involving more MLO agents, until the entire hierarchy structure is determined and AUVs are 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.

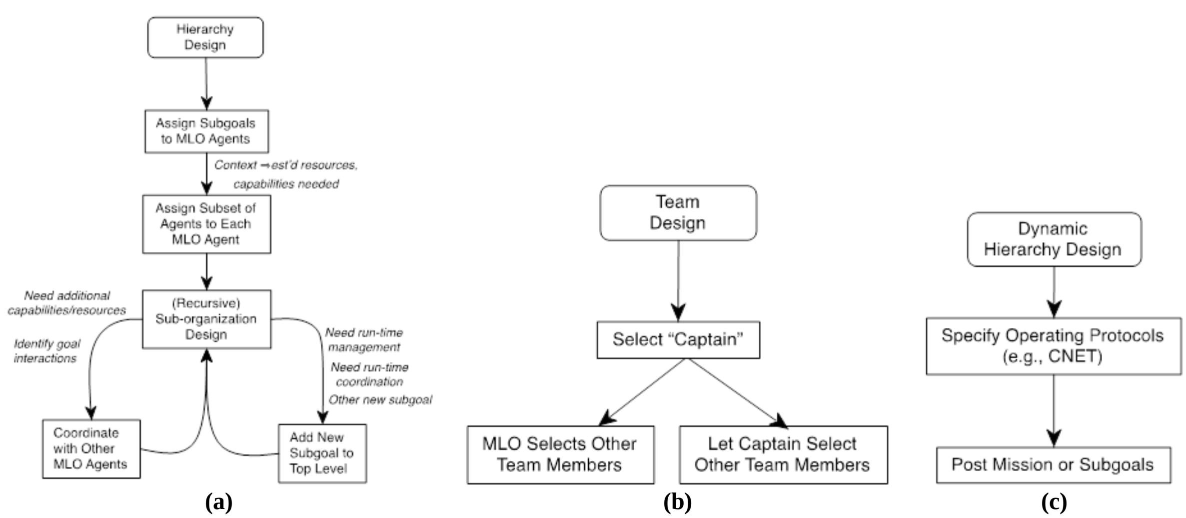

Figure 9 Instantiating organizational structures: (a) a hierarchy; (b) a team; and (c) a dynamic hierarchy [15]

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 Protocol (CNP) [3]. 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., [6]). The MLO will need different protocols and mechanisms for each. So far, we are concentrating on the three organizational structures mentioned above.

V. Conclusion and Future Work

Many missions for which a multi-AUV system would be desirable require that the system be able to operate autonomously, including being able to initially self-organize and to reorganize as necessary. We have made the case that this is true for our example downed-aircraft mission, but it is equally true for many long-term data-gathering missions, AOSNs operating under sea ice, and mine hunting in hostile locations.

As important as it is for a multi-AUV system to be able to organize/reorganize itself, it is just as important that its organization be appropriate the context. This means that the system will need the ability to: determine what the (joint) context is; what appropriate organizational structures are for the context; and how to create an organization based on them.

CoDA is an ongoing, long-term project whose aim is to develop distributed, context-based organization/reorganization mechanisms for sophisticated multi-AUV and other multiagent systems. At the present, a two-level organization scheme has been developed, initial protocols for many phases of operation have been defined, a simple organization design mechanism has been implemented, and task (role) assignment mechanisms [6] have been developed and tested. We have recently begun both applying context-mediated behavior to the problem of context-based organization design [18] and, even more recently, distributing CMB across multiple agents [16]. These latter areas constitute the bulk of the future work to be done, along with changes they will necessitate in CoDA's overall design (as discussed briefly above).

CoDA has already shown itself to be capable of controlling simulated multi-AUV systems in the face of a dynamic situation. We believe that with the addition of context-based organization design, it will be able to quickly and accurately choose from among the organizational structures it knows the ones appropriate for the situation. We further are confident that by distributing the CMB and organization design processes across multiple agents, the processes will be more robust and better able to take into account the different viewpoints of all the agents in the system. The result will be a mechanism for multi-AUV control that is fast, reliable, and highly context-specific.

References

[1] T. Curtin, J. Bellingham, J. Catipovic, and D. Webb, "Autonomous oceanographic sampling networks," Oceanography, vol. 6, no. 3, 1993.

[2] R.M. Turner and E.H. Turner, "A two-level, protocol-based approach to controlling autonomous oceanographic sampling networks," *IEEE Journal of Ocean Engineering*, vol. 26, no. 4.

[3] R. Davis and R.G. Smith, "Negotiation as a metaphor for distributed problem solving," *Artificial Intelligence*, vol. 20, pp. 63-109, 1982. CrossRef

[4] E.H. Durfee and V.R. Lesser, "Using partial global plans to coordinate distributed problem solvers," in *Proceedings of the 1987 International Joint Conference on Artificial Intelligence*, 1987, pp. 875-883.

[5] B. Horling and V. Lesser, "A survey of multi-agent organizational paradigms," *The Knowledge Engineering Review*, vol. 19, no. 4, pp. 281-316, 2004. CrossRef

[6] E.H. Turner and Turner R.M., "A constraint-based approach to assigning system components to tasks," in *Lecture Notes in Artificial Intelligence 1415: Methodology and Tools in Knowledge-Based Systems*, J. Mira, A.P. del Pobil, and M. Ali, Eds. New York, USA: Springer, 1998, vol. I, pp. 312-320, Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-98-AIE).

[7] M.S. Fox, "An organizational view of distrubuted systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, pp. 70-80, 1981. CrossRef

[8] T.W. Malone, "Modeling coordination in organizations and markets," *Management Science*, vol. 33, no. 10, pp. 1317-1332, 1987. CrossRef

[9] M. Tambe, "Agent architectures for flexible, practical teamwork," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1997, pp. 22-28.

[10] B.J. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artificial Intelligence*, vol. 86, no. 2, pp. 269-357, 1996. CrossRef

[11] T. Sandholm and Q. Huai, "Nomad: Mobile agent system for an Internet-based auction house," *Internet Computing*, vol. 4, no. 2, pp. 80-86, 2000. CrossRef

[12] R.M. Turner, "Context-mediated behavior for intelligent agents," *International Journal of Human-Computer Studies*, vol. 48, no. 3, pp. 307-330, 1998. CrossRef

[13] S.I Sagatun, "A situation assessment system for the MSEL EAVE-III AUVs," in *Proceedings of the Sixth International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, 1989. CrossRef

[14] J.H. Lawton, R.M. Turner, and E.H. Turner, "A unified long-term memory system," in *Proceedings of the International*

Conference on Case-Based Reasoning (ICCBR'99), Munich, Germany, 1999.

[15] David Gagne, Sonia Rode, and Roy M. Turner, "Distributed context-based organization and reorganization of multi-AUV systems," in *Proceedings of the 18th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Portsmouth, NH, 2013.

[16] R.M. Turner, S. Rode, and D. Gagne, "Toward distributed context-mediated behavior for multiagent systems," in *Proceedings of the 2013 International and Interdisciplinary Conference on Modeling and Using Context*, Annecy, France, 2013.

[17] R.A. Miller, H.E. Pople, and J.D. Myers, "INTERNIST-1, an experimental computer-based diagnostic consultant for general internal medicine," *New England Journal of Medicine*, vol. 307, pp. 468-476, 1982. CrossRef

[18] R.M. Turner and E.H. Turner, "Selecting organizational structures for advanced multi-AUV systems," in *Proceedings of the Fourteenth International Symposium on Unmanned Untethered Systems (UUST)*, Durham, NH, 2005.

[19] N. Carlesi, F. Michel, B. Jouvencel, and J. Ferber, "Generic architecture for multi-AUV cooperation based on a multi-agent reactive organizational approach," in *IEEE/RSJ Internaional Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 5041-5047.

[20] M. Kemp et al., "MASA: A multi-AUV underwater search and data acquisition system," in *OCEANS'02 MTS/IEEE*, vol. 1, 2002, pp. 311-315.

[21] V.R. Lesser and D.D. Corkill, "Functionally accurate, cooperative distributed systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 1, pp. 81-96, 1981. CrossRef

[22] H. Li, A. Popa, C. Thibault, M. Trentini, and M. Seto, "A software framework for multi-agent control of multiple autonomous underwater vehicles for underwater mine counter-measures," in *2010 International Conference on Autonomous and Intelligent Systems*, 2010, pp. 1-6.

[23] C.C. Sotzing, J. Evans, and D.M. Lane, "A multi-agent architecture to increase coordination efficiency in multi-AUV operations," in *OCEANS'02-Europe*, 2007, pp. 1-6.

[24] R.M. Turner and E.H. Turner, "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, 2000.