Toward Distributed Context-Mediated Behavior for Multiagent Systems

Roy M. Turner[1], Sonia Rode[1], and David Gagne[2]

[1] School of Computing, University of Maine, Orono, ME 04444 USA
{rturner,sonia.rode}@maine.edu
[2] Department of Computer Science, University of Southern Maine,
Portland, ME 04104 USA
david.gagne1@maine.edu

Abstract. Although much attention has been devoted to modeling and using context in intelligent agents, relatively little has been given to the problem for multiagent systems (MASs). Yet, just as with an individual agent, context affects how a MAS should behave. In this paper, we discuss an approach to distributed context management for multiagent systems. The approach is based on earlier work on context-mediated behavior (CMB) for single agents, which explicitly represents contexts as c-schemas that contain knowledge about how to behave in the contexts represented. We are distributing CMB for use in advanced multiagent systems. This work is just beginning, and so the paper discusses issues and potential approaches to distributing CMB.

Keywords: Context-mediated behavior, multiagent systems, context assessment

1 Introduction

Modeling context and the use of contextual knowledge has been the subject of intense interest in recent years, not only in the interdisciplinary context community (as represented, e.g., in the CONTEXT conference series), but also in natural language understanding, ubiquitous computing, and context-aware applications. With the exception of work in natural language understanding, most work has focused on understanding the role of context and contextual knowledge in the decision processes of single agents. The literature is far too broad to synopsize here, but our own past work (e.g., [1, 2]) is somewhat representative, focusing on explicitly representing contexts as first-class objects, having agents assess their current situation in terms of known contexts, and then using the resulting contextual knowledge to guide the agent to behave appropriately.

Context is also important for multiagent systems (MAS), however. In the simplest case, a context-aware agent will know how best to behave within the structure and environment of a MAS. But the role of context in a MAS goes beyond this. One can also think of the context of the MAS as a whole. If the MAS' agents can, together, recognize this *global context* (joint context, shared

Appears in the *Proceedings of the 2013 International and Interdisciplinary Conference on Modeling and Using Context*, © 2013 Springer and the authors. It is posted here for your personal use: no redistribution. The definitive version appears in the conference proceedings, published by Springer.

context), then potentially they can all behave more appropriately and effectively as members of the MAS, and, consequently, the MAS as an entity will behave appropriately for its context.

As an example, consider a complex scenario: using a MAS composed of autonomous underwater vehicles (AUVs) to respond to a plane going down in the North Atlantic. The MAS will need to characterize the debris field, search for any survivors, and find the airplane's black boxes. Using a MAS for this is ideal in many respects, since the area is remote, the environment is hostile, and the task may take a long time, all things that argue against a human presence. However, such a MAS faces many practical problems. First, the AUVs must somehow arrive at the site of the crash. This means that either they must travel there under their own power, be delivered by ship or submarine, or be dropped from an airplane, so it is likely some of the AUVs may not survive, or may arrive very late. Second, since the frequency of crashes is low and the cost of AUVs is very high, it would likely not make sense to have a dedicated set of vehicles for this task. Rather, the MAS should be able to use any AUVs that can be made available by governments, industry, or academia. Thus the resulting MAS will be heterogeneous, and, due also to the delivery problem, its composition and capabilities may not be predictable ahead of time. This means that an organization for the MAS cannot be devised ahead of time, but rather must be designed by the MAS itself, on-site. Third, agents by their nature will occasionally fail, need to refuel/recharge, or be needed elsewhere; others may become available. Consequently, the composition of the MAS will change over time, which, coupled with the fact that the environment will be dynamic, the sensors uncertain, and the agents' knowledge uncertain and incomplete, means that the MAS will need to be able to reorganize itself as needed.

Attention to context comes into play in several ways for such a MAS. Individual agents that are aware of the global context can make better decisions about how to behave within the MAS by matching their local behavior to the needs and constraints of the MAS as a whole. They can interpret their sensory information better by making use of knowledge about the global context, for instance, and they can focus their attention on goals that are most supportive of the goals of the MAS, either those explicitly known or those inferred from the context. They can choose actions to take to achieve goals that are appropriate for the MAS' context.

Beyond the local behavior of individual agents, however, knowledge about the global context can directly benefit the MAS as a whole. This is most apparent for the kind of MAS just described. The problem of designing an organization for such a MAS is context-dependent. Different organizations (e.g., hierarchies, teams, etc.) have different strengths and weaknesses depending on properties of the environment (e.g., uncertainty and change), communication (e.g., bandwidth, type of communication channel, whether or not the mission is covert), and the MAS itself (e.g., how many agents are present, their intelligence level, etc.). Identifying the global context that is implied by such properties of the current situation can help the MAS decide which organization or organizations (if it can merge several) are best.

In past work, we have concentrated on single-agent context assessment and use, and we have considered the problem of extending this to the multiagent case by having a single agent design the organization based on its view of the global context [**?**]. However, a much better approach, and the one we consider in this paper, is decentralizing the context assessment process. This removes a potential single point of failure, offloads from a single agent some of the burden of context assessment, and makes use of different agents' viewpoints and contextual knowledge.

The work presented in this paper is preliminary. We first discuss our overall approach, called *context-mediated behavior* (CMB) [1]. We then discuss issues relating to distributing this process across a subset of agents of a MAS and some directions we are exploring to address these issues.

2 Context-Mediated Behavior

In context-mediated behavior, an agent's contextual knowledge is stored in knowledge structures called *contextual schemas* (c-schemas).[3] Each is a frame-like structure representing a *context*, which in our approach is a class of similar situations, each of which has similar or the same implications for the agent's behavior. C-schemas are usually stored in a content-addressable memory (e.g., [4]) to allow features of the situation to be used to retrieve c-schemas that are similar to the current situation.

A given situation can be a member of more than one context. For example, if an AUV is taking data samples under sea ice while its batteries are low, then this situation can be viewed as an instance of each of the contexts "data collection mission", "under ice", and "low power", depending on which contexts the agent knows about before hand. If this situation turned out to have different implications about behavior than could be derived from combining information in the c-schemas, then a new c-schema would be learned for this context and stored appropriately by relating it to the other contexts.

The process of context-mediated behavior for an individual agent is shown in Figure 1. We call the part of the agent that does context assessment the *context manager* (ConMan). ConMan contains functionality to assess the context as well as interface with the rest of the agent to distribute the contextual knowledge as needed.

The overall process of assessing the context is a diagnostic process analogous to medical diagnosis, where features of the situation (cf. "signs and symptoms" of medical diagnosis) are used to diagnose the context (i.e., select a context that can explain the features). We use a differential diagnosis process based on work in the artificial intelligence in medicine program INTERNIST-I [5] that allows

[3] The name was chosen to differentiate these schemas from others used in the original work for procedural and strategic knowledge, p-schemas and s-schemas, respectively [3].

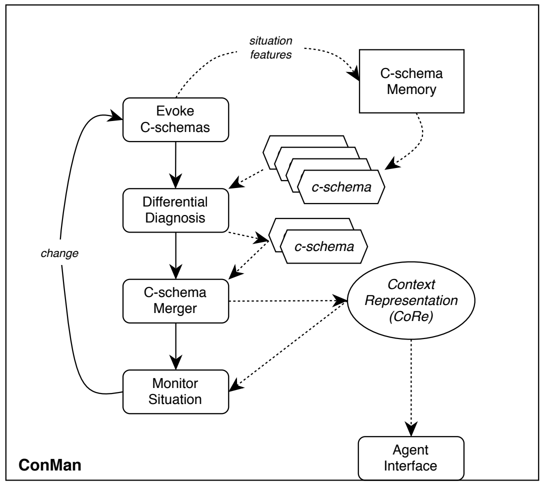

Fig. 1. The context-mediated behavior process. For clarity, lines representing information flow about situation features into the context manager (ConMan) are not shown.

multiple context hypotheses to be played off against one another to find the best one(s) that fit the situation.

The process starts when ConMan uses features of the situation to probe its c-schema memory. This will elicit, or *evoke*, one or more c-schemas, each of which is a candidate to represent some facet of the current situation; we can think of this as ConMan being "reminded" of these c-schemas based on the situation (cf. [6]).

The next step is to more closely examine and compare the c-schemas to find those that truly represent aspects of the current context. This is done by comparing the c-schemas with respect to the features they predict that are present and those that are not, and those that are present they do not predict. To do this, c-schemas are grouped into *logical competitor sets* [7], each element of which is a c-schema that can basically explain the same set of situational features. The hypotheses are scored, and then ConMan attempts to *solve* the set by increasing confidence in one hypothesis relative to the others by some given amount. This is done using various strategies based on those described by Miller *et al.* [5]. This is done for each competitor set until ConMan is left one or more c-schemas, each of which represents part of the context.[4]

The c-schemas remaining are then merged to create an overall picture of the context called the *context representation*, or (CoRe). This is not a simple problem, since the elements of each c-schema can have various relationships with each other, such as compatible, overlapping, superseding, conflicting, and so forth. Note that our approach differs from, e.g., that of [8], who use a simple

4 The process is not quite this simple, since the act of trying to solve a competitor set can cause the sets to need to be recomputed.

algebra for this purpose (and no differential diagnosis). This aspect of CMB is an area of active research.

The CoRe serves as the repository for knowledge about the current context. This knowledge is given to other parts of the agent via ConMan's agent interface.

After the context is assessed, ConMan monitors the situation, comparing it to predictions from the CoRe. When it detects a significant change (which depends partly on the context), the process repeats so that at all times, the agent attempts to maintain a coherent, current view of the context.

3 Communicating About the Group Context

Given that the kind of MAS in which we are interested is open, meaning agents can come and go, and that we do not wish to restrict the kinds of agents that can participate, the first thing we must consider is how the different, likely heterogeneous, agents can communicate about the group context.

In order for this to happen, the agents obviously must share a common communication language. There are many existing agent communication languages, and our approach is agnostic as to which to use, as long as all agents have access to it and the language is sufficiently expressive to carry the knowledge needed. Second, along with the language, the agents also need to be able to express their own knowledge, regardless of their internal knowledge representation, in a common representation that can be transmitted via the language. In our work, we have used a frame-based representation, and we are now considering augmenting or replacing this with a description logic. Mastrogiovanni *et al.* [9] has made a start toward a situation description language. However, our approach is agnostic as to this shared representation language as well.

The third thing that is needed is a common ontology for context and contextual knowledge. There has been some work on ontologies for context (e.g., [10–12]). However, many of these approaches take a simplistic view of context (e.g., context is location or user task), have a shallow ontology, or both. What is needed is not only an ontology for contexts per se, but also one that includes the kinds of things that comprise contextual knowledge for open MASs.

Unfortunately, an ontology of contexts is somewhat difficult to specify a priori given our approach to context representation. Contextual schemas grew out of work in case-based reasoning: they are essentially generalized cases. Our approach relies on an agent being able to update its contextual knowledge based on its own experience, including modifying existing c-schemas, learning new relationships between them, and learning new ones. This is supported by the kind of schema memory we use (e.g., [4]). One can view the c-schema memory as an evolving, changing ontology.

We can, however, provide agents with a basic ontology for contexts to serve as the basis for their (ultimately) idiosyncratic ontologies. A start toward such an ontology is shown in Figure 2. To the extent that the agents do not modify this "upper ontology", they will have at least some basis for communication. Idiosyncratic contexts derived from the agents' own experiences will need to be

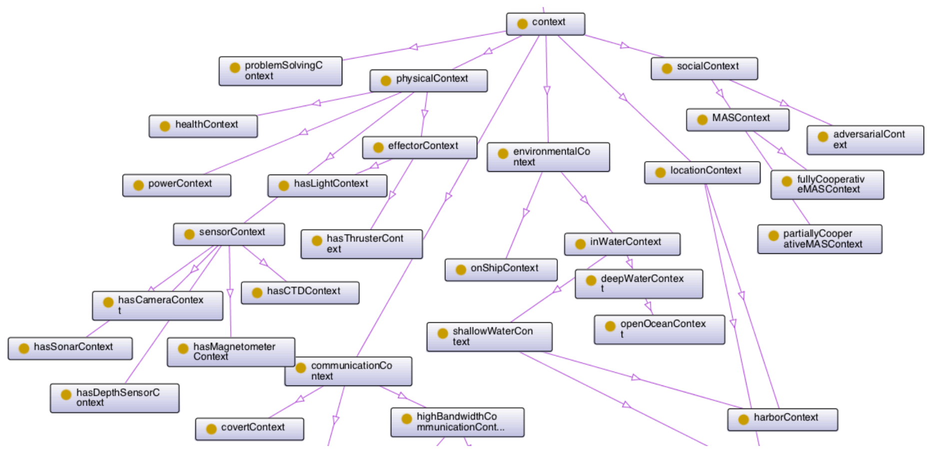

Fig. 2. A starting point for an ontology of contexts. (Figure produced by OntoGraf.)

discussed in relation to a shared upper ontology. Work on how this is to be done is ongoing.

With respect to the contents of contextual schemas, a shared ontology is more feasible and straightforward. The classes in the ontology reflect the kinds of knowledge useful in our c-schemas (and, we believe, for contextual reasoning in general), as shown in Figure 3. This includes knowledge about predicted features of the situation, context-dependent meaning of concepts (e.g., [13]), event-handling knowledge, knowledge about goal priority (attention focusing knowledge), knowledge of how to achieve goals, and various behavioral settings ("standing orders") that should automatically come into effect in the context.

A key problem for an agent during context assessment is deciding if others are referring to the same context it is. This is a variant of the reference problem from natural language processing [e.g., E. Turner and Matthias, 1998]. There are three possibilities here, if agent A believes the context is represented by c-schema C_A and agent B is believes it is represented by its c-schema C_B. First, C_A and C_B could actually refer to the same context. Determining this seems at first glance straightforward, but it is not. The context may be labeled differently by A and B, for example, if the c-schemas have been learned from their own experience (and hence, were not part of the common context ontology). Even if they are labeled the same, the knowledge contained in each may differ, even about the same context, again due to the differences in the agent's experiences. However, if the agents can recognize that their c-schemas represent the same context, they may be able to synchronize their knowledge.

A second case is when C_A and C_B are not identical, but each represent variants of the same context. For example, C_A may refer to "in Boston Harbor on a weekend" while C_B is "in Boston Harbor on a holiday". Here, the agents may be able to use their ontologies to identify a common ancestor of the c-schemas (e.g., "in Boston Harbor") as a basis to begin reasoning about the context.

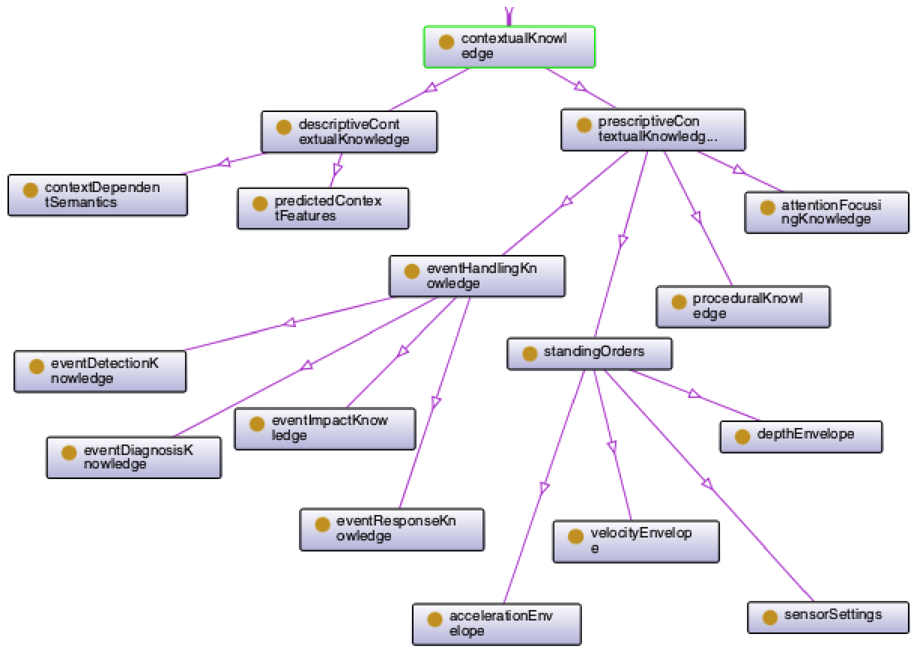

Fig. 3. A portion of the ontology for contextual knowledge. (Figure produced by OntoGraf.)

Finally, C_A and C_B may represent entirely different, possibly incommensurate, contexts. In this case, the agents will need to negotiate to attempt to resolve the conflict.

This problem, as well as the related problem of ensuring that contextual knowledge stored in c-schemas is mutually commensurate, is an active area of research.

4 Deciding How to Distribute the Process

The problem of distributed context assessment is, itself, context-dependent. The appropriate way to distribute the task is determined by such things as the number of agents capable of participating, the communication characteristics (bandwidth, speed of channel, broadcast versus point-to-point channel, etc.), and the degree of time pressure. For example, if there are many agents, reasonable bandwidth, and no significant time pressure, then distributing the process over all agents may make sense; if there are only a very few agents capable of participating, very low bandwidth, or very high time pressure, then it may make sense to allow one agent to assess the context for everyone.

The first step, then, in distributing context assessment is for each agent to "pre-assess" the context.[5] Depending on the assessment, the agents may have to seek agreement from others, or the outcome may be so clear that no further communication is needed. This will depend on the cooperation protocols in use by the MAS.[6]

The distribution mechanism may vary as well, depending on the context. For example, there are four basic tasks for context assessment in CMB: evoke hypotheses, form competitor sets, solve competitor sets, and merge the results. Any or all of these could be distributed, depending on the context pre-assessment. For example, to reduce communication, the process could be distributed as follows: agents all evoke hypotheses based on their local context and communicate the hypotheses to everyone; competitor sets are formed by each agent, with the (possibly fallacious) assumption that agents will all create the same sets; agents select which set(s) they will attempt to solve based on some a priori convention (e.g., an agent might select a set if it was the first to evoke its top hypothesis); and then the final set of hypotheses would be used for distributed context merger to create a context representation. In a different context, it might be better to distribute each of the parts.

At present, we are concentrating on the case in which all parts of the CMB process will be distributed. Future work will look more closely at this issue of pre-assessment and context-dependent selection of distribution strategy.

5 Distributed Context Hypothesis Evocation

The first step of context assessment is finding candidate context hypotheses by determining which c-schemas are *evoked* from memory based on the situation. This could be done by a single agent if necessary, but the different viewpoints, agent knowledge, and c-schema repertoires all argue for having each agent perform this task.

Each agent's evocation of some candidate hypotheses for the global context is a natural consequence of its own context assessment. (Here, we assume that distributed context assessment is restricted to context-aware agents.) The problem is determining which c-schemas have global rather than purely local scope. This is somewhat harder than it seems. For example, if a local hypothesis is that the agent is in the context of operating on low power, this would seem to be a purely local context; however, it may be the case that the global context is affected by this, as well, since the MAS may need to take into account that some of its assets (e.g., this agent) may have to leave the system before the mission is done.

A question also arises of which of the locally-evoked c-schemas should be shared. While the most general solution would be to share all of them, this may

[5] The c-schemas representing this "meta-context" likely will be similar to our earlier strategic schemas that determined the style of problem solving [3].

[6] See [15] for an example of protocols where individual decisions can be followed with little need for communication.

not be the most efficient, both from the standpoint of communication bandwidth and computational load on the overall system. It may be best to share only those that have gone through the local agent's differential diagnosis process to become part of its own CoRE; however, this may cause the MAS to miss some reasonable candidates that were ruled out by the local agent because it lacked global knowledge that would have included it.

Although the set of agents' c-schemas evoked this way will be a good source of global context hypotheses, it may not be sufficient. Some c-schemas might have been evoked locally by an agent if only it had access to information another agent has about the environment or other situational features. For example, suppose agent A has knowledge about operating in a context in which a thermocline (a temperature/density discontinuity which affects acoustic communication) is present, but does not observe one from its location, and agent B observes a thermocline, but does not have any knowledge about such a context. In this case, the information about the environmental feature should be communicated from B to A. In general, though, it is a difficult to determine what should be communicated: too much, and the communication channel will possibly be saturated; too little, and some c-schemas will not be evoked that should be.

It may be that some kinds of information can be identified as generally evocative, for example, particular environmental features, or the properties of an agent's schema memory may predict the value of asking others for particular information. For example, in a dynamic conceptual memory as we have used in the past [4, 3], an agent could during memory search identify salient features that, if it knew their value, would allow it to retrieve important c-schemas. Addressing this problem in general will be an active area of future research.

6 Competitor Set Formation

The next step is to create logical competitor sets from the evoked hypotheses by grouping them according to what they explain. As part of this process, the hypotheses are scored and ranked according to what they do and do not explain. The issues involved in distributing this process are determining who makes the decision about which hypothesis belongs in which set and determining which situational features each hypothesis does/does not explain.

The entire MAS (or rather, the context-aware members) could decide on the composition of the sets. This could be done by all agents reaching common knowledge (by communication and possibly negotiation) about the set of evoked c-schemas, then negotiating about set membership. Alternatively, this process could progress in a general sense like the process of partial global plan formation [16]. Agents could each decide on the set of competitor sets, then share this with their neighbors (by location, e.g., to reduce communication lag time), which then critique the set based on their own sets and knowledge of what each hypothesis explains. Over time, a (partial) global set of competitor sets could evolve via negotiation. Or, finally, the problem of competitor set creation could be divided amongst the agents by negotiation or convention, as mentioned above. The best way to do this has yet to be determined.

7 Solving Competitor Sets

Differential diagnosis is used to "solve" the competitor sets to arrive at a final set of c-schemas. It involves comparing hypotheses within each competitor set based on what they each explain or fail to explain about the current situation and gathering new information until one hypothesis exceeds some threshold value beyond the nearest competitor.

Similarly to the above discussion about distributing context evocation, this process can be fully distributed or done largely by individual agents. In a fully-distributed version, the agents would all have common knowledge of the competitor sets and their composition, and they would exchange information about situational features and negotiate to come to agreement about the scores of the hypotheses. The agents would also need to gather additional information to solve the sets, either by eliciting information already known by some agent(s) in the system or by taking actions (e.g., using a sensor) to gather new information. With agents having common knowledge about the set being worked on, some communication might be avoided: an agent that had the requisite information could just supply it rather than having to be asked.

Another possibility is for the competitor sets to be parceled out to individual agents for them to solve. This could be done by convention, for example, based on which agent suggested the topmost hypothesis in a set (with ties also being broken by convention). Or a more sophisticated distribution could be done, with the kinds of information needed to solve the set being matched to what knowledge particular agents have. Responsibility for solving a set could even be shifted among the agents over time based on what information is currently needed to make progress on the solution.

Regardless of the distribution, the agents will likely disagree on some aspects of the process, in particular, which situational features are or are not explained by a given c-schema. Consequently, there will need to be negotiation mechanisms in place to allow the agents to arrive at some consensus on such issues.

8 Merging Contextual Knowledge

Once all the competitor sets have been solved, the MAS will be left with a set of c-schemas, each of which represents some part of the current context. The next step is to merge the knowledge from these to form the overall context representation (CoRe). Context merger could be handed off to a single agent, but to make use of all agents' different knowledge and viewpoints, it should be distributed. Merger can be done proactively, with all knowledge merged immediately, or more lazily, with knowledge merged only when needed, e.g., to make a decision about an aspect of organization design.

Merging contextual knowledge is difficult, especially in the distributed case. Not only can different c-schemas provide conflicting knowledge (e.g., the predicted impact of an unanticipated event), but different agents can have different beliefs about it as well.

We have looked at the former problem to some extent and have some idea about how to merge knowledge from different c-schemas. For example, if the knowledge is numerical, depending on the context, it may be reasonable to abstract the information to a range of values or a set of possible values; ranges can be intersected or unioned, as can fuzzy sets; and symbolic values can sometimes be merged by appeal to the ontology (e.g., by abstracting to a common ancestor). Others, for example, Bikakis and Antoniou [17], have looked at this problem of conflict resolution in the multiagent case, but the strategies for merger tend to be much simpler than what we feel is needed. In addition, getting different agents to agree on which features a c-schema does/does not explain will also be difficult and will likely involve negotiation.

All agents could participate in all aspects of the merger process. Alternatively, an agent or a small set of agents could be identified for different elements of the c-schemas, for example, for event-handling knowledge. It would then be responsible for merging that portion of the CoRe. The CoRe might itself be distributed this way, with no agent having knowledge of the whole thing; instead, the agents that merged portions of the CoRe could be responsible for that portion. Although this is attractive from the standpoint of reducing any particular agent's need to store the CoRE, drawbacks include having possible single points of failure for some parts of the CoRe as well increasing message traffic to access parts of the CoRe that an agent does not have.

9 Using the Contextual Knowledge

Once the MAS has assessed the context and has a CoRe available, the contextual knowledge it contains needs to be made available to the agents as they require it. If the CoRe is disseminated in its entirety to all agents, then this problem is trivial. However, if not, then an agent needing, say, organizational design knowledge, would first have to determine where such knowledge resides, then obtain it. Finding the knowledge could be done easily by giving all agents common knowledge of which agents are responsible for which parts of the CoRe. However, since we are interested in an open MAS, that may change over time. A better approach might be either to have a broker (e.g., [18]) for the information or to have agents broadcast requests for contextual knowledge, depending on the communication constraints (which are, of course, context-dependent).

10 Continuous Context Assessment

Creating the CoRe is only one phase of the overall process of context management. As the situation changes, the MAS will have to assess the context in response. Thus, in addition to carrying out the tasks assigned to the MLO, the

MLO will also have to devote effort and communication bandwidth to monitoring and assessing the context. For example, in our work on multiagent systems, a *meta-level organization* (MLO) first self-organizes in order to design an efficient *task-level organization* (TLO) to carry out the mission [15]. In past work, the MLO disappeared as the system transitioned to the TLO. To add decentralized context assessment to this approach, the MLO will need to continue in some capacity as an entity that can continuously assess the context.

11 Conclusions and Future Work

In this paper, we have discussed some issues related to distributed context assessment for multiagent systems, in particular for distributing our context-mediated behavior approach. As should be apparent, although we have identified important issues and some mechanisms to address them, this work is still in an early stage.

We are currently working to integrate a distributed version of CMB into our CoDA (Cooperative Distributed AOSN[7] control) approach to multiagent organization/reorganization [15]. Work is currently focusing on developing an ontology for context and a representation language to allow communication between the agents and developing the distributed CMB approach described above.

We anticipate that adding contextual reasoning abilities to multiagent systems will dramatically improve the performance of individual agents as well as that of the MAS as a whole, in particular by improving the speed and quality of organization design. Whether or not this improvement is outweighed by the overhead of distributed context assessment, which may entail adding ConMan modules to non-context-aware agents, is an open question, although we believe that it will be worth it. As our work matures, we intend to test this hypothesis via simulation experiments and experiments using our autonomous land robots.

References

1. Turner, R.M.: Context-mediated behavior for intelligent agents. International Journal of Human–Computer Studies **48**(3) (March 1998) 307–330
2. Turner, R.M.: A model of explicit context representation and use for intelligent agents. In: Lecture Notes in Artificial Intelligence 1688: Modeling and Using Context (Proceedings of the 1999 International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-99), Trento, Italy), Springer-Verlag (September 1999)
3. Turner, R.M.: Adaptive Reasoning for Real-World Problems: A Schema-Based Approach. Lawrence Erlbaum Associates, Hillsdale, NJ (1994)
4. Lawton, J.H., Turner, R.M., Turner, E.H.: A unified long-term memory system. In: Proceedings of the International Conference on Case-Based Reasoning (ICCBR'99), Monastery Seeon, Munich, Germany (July 1999)

[7] Autonomous oceanographic sampling networks [19].

5. Miller, R.A., Pople, H.E., Myers, J.D.: INTERNIST–1, an experimental computer-based diagnostic consultant for general internal medicine. New England Journal of Medicine **307** (1982) 468–476
6. Kolodner, J.L.: Retrieval and Organizational Strategies in Conceptual Memory. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1984)
7. Feltovich, P.J., Johnson, P.E., Moller, J.A., Swanson, D.B.: LCS: The role and development of medical knowledge and diagnostic expertise. In Clancey, W.J., Shortliffe, E.H., eds.: Readings in Medical Artificial Intelligence. Addison–Wesley Publishing Company, Reading, Massachusetts (1984) 275–319
8. Padovitz, A., Loke, S.W., Zaslavsky, A.: Multiple-agent perspectives in reasoning about situations for context-aware pervasive computing systems. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and HumansIEEE Transactions on **38**(4) (2008) 729–742
9. Mastrogiovanni, F., Scalmato, A., Sgorbissa, A., Zaccaria, R.: Robots and Intelligent Environments: Knowledge Representation and Distributed Context Assessment. Automatika–Journal for Control, Measurement, Electronics, Computing and Communications **52**(3) (2011)
10. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications. (2004) 18–22
11. Bucur, O., Beaune, P., Boissier, O.: Representing context in an agent architecture for context-based decision making. In: Proceedings of the Workshop on Context Representation and Reasoning (CRR'05), Paris, France (2005) Co-located with CONTEXT'05.
12. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. The Knowledge Engineering Review **18**(03) (2003) 197–207
13. Turner, R.M.: Determining the context-dependent meaning of fuzzy subsets. In: Proceedings of the 1997 International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97), Rio de Janeiro (1997)
14. Turner, E.H., Matthias, C.: Understanding pronomial references to quantified expressions. Journal of Natural Language Engineering **4**(4) (1998) 345–361
15. Turner, R.M., Turner, E.H.: A two-level, protocol-based approach to controlling autonomous oceanographic sampling networks. IEEE Journal of Oceanic Engineering **26**(4) (October 2001)
16. Durfee, E.H., Lesser, V.R.: Using partial global plans to coordinate distributed problem solvers. In: Proceedings of the 1987 International Joint Conference on Artificial Intelligence. (1987) 875–883
17. Bikakis, A., Antoniou, G.: Distributed Defeasible Reasoning in Multi-Context Systems. NMR (2008)
18. Decker, K., Williamson, M., Sycara, K.: Matchmaking and brokering. In: Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96). (1996) 432
19. Curtin, T., Bellingham, J., Catipovic, J., Webb, D.: Autonomous oceanographic sampling networks. Oceanography **6**(3) (1993)