Context-Mediated Behavior: An Approach to Explicitly Representing Contexts and Contextual Knowledge for AI Applications

Roy M. Turner
Department of Computer Science
University of Maine
Orono, ME 04469 USA
(207)581-3909
rmt@umcs.mine.edu

Abstract

Explicit representation of context and contextual knowledge is critical to AI applications. In this paper, we discuss conclusions drawn from several years of work on representing and using contextual knowledge. We describe our approach to context-sensitive reasoning, called *context-mediated behavior* (CMB), and discuss our experience related to reasoning in context in AI applications and our ongoing and future work in the area.

The context in which an intelligent agent operates profoundly affects how it behaves. Not only is this intuitive, it has been shown to be the case by psychological and sociological studies (see discussion in (Turner 1998)). This is true for artificial agents as well, such as artificial intelligence (AI) applications.

AI programs, except for the most trivial ones, have always taken context into account. Usually, this has occurred without explicitly recognizing that it is being done, and without the application itself having any explicit representation of what context it is in or any clear sense of its own contextual knowledge. For example, AI planners (Fikes & Nilsson 1971; Wilkins 1984) create plans that must work in a particular task context, yet they do not represent that context as an object in its own right, nor do they explicitly identify the contextual aspects of the planning knowledge. The context is taken to be the observable features of the current situation, and contextual knowledge is distributed in the operator preconditions, inference rules the planner may have, and implicitly in the assumptions encoded in the program itself.[1] Similar implicit context representation occurs in rule-based systems, neural networks, and other AI applications.

The result is that these AI applications cannot capitalize on knowing what context they are in and how to behave in that context. To the extent they do so at all, they are forced to do situation assessment without any clear notion of what the space of possible situations (contexts) might be. Without explicit representations of contexts, they cannot take advantage of *a priori* knowledge about them to do differential diagnosis to guide information gathering during situation assessment. Without explicit context representation, the application is left with no clear idea of what the implications might be of being in any particular situation. Behavior is conditioned by aspects of the context, but not by the context as a whole. The application is without the ability to truly reason about the context it is in and how that should affect behavior. It cannot conclude, "I am in context X", then behave appropriately until that context changes. Instead, it must waste effort constantly deciding if behavioral knowledge is appropriate for the situation (e.g., by checking antecedent clauses of potential rules, goals/preconditions of potential operators, etc.). An application cannot easily learn important information about how to behave in a context, since it doesn't have any clear notion about what it means to *be* in the context. Finally, it is difficult to acquire and maintain knowledge about context, since it may be distributed across many pieces of knowledge (e.g., rules); this may also lead to problems maintaining the consistency of the knowledge base.

For over ten years, we have investigated explicit representation of contexts and contextual knowledge for AI applications, first in the domain of medical diagnosis (Turner 1989; 1994) and now in autonomous underwater vehicle control (Turner & Stevenson 1991; Turner 1994; 1995; 1998), multi-agent systems (Turner & Turner 1998), and multi-modal interfaces to geographic information systems (Turner *et al.* submitted). In this paper, we discuss the conclusions we have drawn from this work about how knowledge about context should be represented and used. We then describe our approach to doing this, called called *context-mediated behavior* (CMB) (Turner 1998). Finally, we relate CMB back to application domains by briefly describing our work in several AI application areas.

Representing and Using Context

In this section, we present conclusions we have arrived at from our past and current work on context. This

[1]Some planners do include meta-rules (Davis & Buchanan 1984), but even here, only rudimentary attention is paid to representing context.

section can be viewed as our "position" on reasoning in context for AI applications. We present these conclusions as a set of assertions, with explanations, below.

A note on terminology is in order. Throughout the paper, we use the term "contextual knowledge" to mean *a priori* knowledge about contexts or about how to behave in contexts. We do not explicitly consider here "contextualized knowledge", general knowledge that must be taken into account explicitly in the current context, as do some authors (Brézillon *et al.* 1997), although some of our contextual knowledge as well as other kinds of knowledge the application uses could also fall under that category. "Situation" is, to us, the constellation of features of the world, the agent, and the task existing at any particular time, devoid of most subjective assessment. A "context" is a recognized situation type, with the subjective assessment that the context is important for behavior. Many different situations might then be instances of a given context. For example, situation one, characterized by the facts "at Disney World", "hungry", and "day is Monday", and situation two, "at Disney World", "hungry", and "day is Tuesday", are both instances of the context "hungry while at Disney World". Finally, throughout the paper we use the term "agent" interchangeably with "AI application".

Contexts should be explicitly represented. An agent should explicitly represent the contexts it knows about. This allows it to *diagnose* the current situation as being an instance of a context it knows about, much as a medical reasoner diagnoses a particular pattern of signs and symptoms as having particular meaning from the standpoint of diagnosis—in that case, as being a known disease. In the general case, diagnosing a situation as a context categorizes the situation and allows the agent to select its behavior based on the context. Instead of predicating behavior on features of a potentially infinite set of situations, the agent can decide how to behave based on recognizing that it is in one of a relatively small set of known situation types (contexts) with particular, known implications for how it should behave.

Contextual knowledge is structured and should be represented in a structured manner. An agent could represent the contexts it knows about as symbols naming the contexts. Knowledge about a particular context could then be indexed by the context name. Behavior could similarly be conditioned on the context name, for example, "If context $C1$ then turn on obstacle avoidance sonar". However, it makes more sense to cluster together the knowledge an agent has about a context. This allows the agent to have immediate access to all context-appropriate knowledge when it recognizes the context it is in without searching its knowledge base. The sum of all the agent's knowledge about the context can be examined and reasoned about as a unit, which has benefits such as preventing, detecting, and eliminating inconsistencies. Bringing together all the knowledge about a context facilitates the update and acquisition of contextual knowledge, both by humans and machine learning techniques.

Knowledge about how to behave in a context should be associated with other information about the context. For most AI applications, identifying the context is not the end of the story: the goal is for the application to behave appropriately for the context it is in. Consequently, the agent needs knowledge linking the context to how to behave in that context. This knowledge should can most fruitfully be associated with the agent's other contextual knowledge. This allows the agent, when it recognizes the context it is in, to immediately have access to knowledge about how to behave. If done properly, the agent can use this knowledge to automatically condition its behavior to fit the context it is in. This includes not only "behavior" in the sense of behavior-based agents, but also knowledge it can use for goal-directed actions such as planning.

Knowledge about contexts should be composable. An immediate and reasonable objection to the idea of explicit context representation is that if not done carefully, it can lead to an unmanageably large number of knowledge structures corresponding to all the contexts that have implications for the agent's behavior. A solution to this problem is to represent a relatively few important kinds of contexts, then merge those as needed to represent others. For example, an autonomous underwater vehicle (AUV) might find itself in a situation in which it has low power, it is under sea ice, it is searching for a sunken ship, and its obstacle avoidance sonar has malfunctioned. It is unreasonable to expect the AUV's designers to have given it knowledge about such a context, and it would not make much sense for the AUV to store information about it for later use, since it is unlikely to occur again. Instead, the AUV could be given explicit representations of more commonly-occurring or at least predictably-important contexts, such as "operating on low power", "operating under ice", "search mission", and "OAS malfunction". It could then merge these to come up with a reasonable description of its context and how to behave in it.

Contextual knowledge should be updateable from experience. Some AI applications do not need to learn from their experiences. Ideally, however, most should. For those agents that will operate in a variety of different contexts, it is important that their contextual knowledge be represented in such a manner as to facilitate its modification and learning by the agent. This allows the agent to not only tailor its behavior to the immediate situation using its contextual knowledge (short-term adaptation), but also to change its behavioral repertoire over time (long-term adaptation) (Turner 1994).

Context-Mediated Behavior

In the past few years, we have developed an approach, called context-mediated behavior (CMB), to explicitly representing and using contextual knowledge in AI applications. This approach relies on representing contexts as knowledge structures called *contextual schemas*, or c-schemas, that are retrieved from a schema memory based on features observed in the current situation. All of the agent's knowledge about a context and how to behave in it is stored in the corresponding c-schema. C-schemas are merged to create a complete picture of the context. Information from this is then used to affect all aspects of the agent's behavior. A *context manager* is responsible for all of this. The context manager can be thought of as an agent in its own right whose expertise lies in managing the application's view of its context and its context-sensitive behavior. In its current incarnation in the Orca AUV mission controller, the context manager is being implemented as ECHO, the embedded context handling object (Turner 1998).

Contextual Schemas

Contextual schemas are knowledge structures that are descendents of Schank's *memory organization packets* (MOPs) (Schank 1982). They are stored in, and organize, a content-addressable conceptual memory similar to the CYRUS (Kolodner 1984) program.

C-schemas contain both *descriptive* and *prescriptive* knowledge about the contexts they represent. The descriptive knowledge consists of:

- Features of the situation that must be present (or not present) in order for it to be considered an instance of the context. This allows the agent to "diagnose" the current situation as an instance of a context it knows about.
- Features of the situation, perhaps yet unseen, that are expected in this context. This allows the agent to make predictions about things it is likely to see that may impact problem solving (e.g., to allow it to recognize unanticipated events). It also allows the agent to disambiguate sensory input based on the contextual, top-down predictions.
- Context-specific ontology/meaning of particular concepts. Some concepts have different meanings in different contexts; this contextual knowledge provides this information to the agent. For example, changes in the meaning of fuzzy linguistic values can be handled by storing context-specific membership functions for the values in c-schemas (Turner 1997). Similarly, neural networks could be made to recognize different things in different contexts by storing context-specific weights in c-schemas.

Prescriptive knowledge, that is, information about how to behave in the context, is also stored in c-schemas. This includes information about:

- handling unanticipated events: how to detect them, how to diagnose their meaning in the context, their context-specific importance, and how to appropriately handle them;
- focusing attention: which goals should/should not be worked on and how important particular goals are in the context;
- goal-directed behavior: knowledge about how to achieve goals appropriately in the situation;
- non-goal-directed behavior: knowledge governing the expression of behavior that is not directly related to goals, such as turning off obstacle avoidance behavior when in the context of docking an AUV, etc; and
- new goals that should be pursued because the agent is in the context.

Using C-Schemas

A context manager such as ECHO must identify the agent's context at each point in problem solving. This can be done by a diagnostic process. Features of the current situation cause c-schemas representing similar contexts to be retrieved, or *evoked*, from memory. Descriptive knowledge in these c-schemas can then be used to determine which of the c-schemas best fits the situation. Differential diagnosis in the manner of INTERNIST-I/CADUCEUS (Miller, Pople, Jr., & Myers 1982) can be done. The c-schemas retrieved represent hypotheses about the classification of the situation as a context. They can be grouped into *logical competitor sets* (Feltovich *et al.* 1984), each of which contains c-schemas that are competing to explain a certain set of the features of the situation.

The top competitors from all the sets, after differential diagnosis, then constitute the set of c-schemas that best fit the situation. These can be merged to create a complete picture of the context. In ECHO, this is called the *context object*.

The context manager parcels out information from the context object to the rest of the agent. In ECHO, the current plan is to implement this as follows. The agent's other modules register their interests with the context manager, much as agents register with facilitators in multiagent systems based on KQML (knowledge query and manipulation language) (Patil *et al.* 1992). When the a new context has been diagnosed, ECHO will either tell the interested modules that there is a new context or send them the information they requested, depending on how they registered.

The context manager constantly monitors the situation to determine if the context has changed. If so, then a new context object is created.

CMB in AI Applications

The context-mediated behavior approach was partially implemented in a medical diagnostic reasoner, and a full version is currently being implemented in an AUV controller. In addition, we have plans to test the approach in other applications, as discussed here.

MEDIC. MEDIC (Turner 1989; 1994) was a schema-based diagnostic reasoner whose domain was pulmonary infections. It grew out of both work in case-based reasoning and reactive planning. It was an *adaptive, schema-based reasoner*, using generalized cases to control its reasoning and capable of changing the way it behaved based on techniques from reactive planning research (Georgeff & Lansky 1987).

Contextual information is very important in medical diagnosis. The meaning of a sign (objective finding) or symptom (subjective) depends on context; for example, a persistent cough in a young, generally healthy person should make the diagnostician think of something different than when observed in a chronic smoker or an inner city dweller with HIV (e.g., respiratory infection, cancer, and tuberculosis, respectively).

In MEDIC, the contexts we were interested in had to do with patient presentation. These were early on called *diagnostic MOPs*, then later the name was changed to "contextual schemas" as it became clear that they were an instance of a larger, more generally-useful class of knowledge structures. Each c-schema represented a picture of the current diagnostic session centered around the patient presentation. For example, MEDIC had c-schemas for "consultation", "cardiopulmonary consultation", "cardiopulmonary consultation in which the patient is an alcoholic". Contextual schemas in MEDIC were monolithic structures representing the entire problem solving context; the best c-schema returned by memory was used as the context object (though not referred to by that name). In a superficial way, MEDIC's c-schemas were similar to earlier work on prototypes in diagnosis by Aikins (1980).

Other contexts are important in medicine that MEDIC did not examine. For example, diseases themselves, or rather their presence, define contexts; indeed, the ultimate goal of a purely diagnostic program is to determine the current context to the level of what disease (or set of diseases) is present in the patient. Disease contexts can provide additional information that is very important, such as the prognosis and suggestions for treatment.

MEDIC ignored an important feature of contexts in general, and in medicine in particular: the evolution of contexts over time. For instance, a context defined by patient P having disease D has many "sub-contexts" corresponding to the evolution of the disease, possibly in response to treatment. This fluid nature of contexts is very difficult to capture in AI knowledge structures. MEDIC was concerned with diagnosing "snapshots" of a patient, similar to the clinicopathological conference (CPC) exercises that doctors engage in, or to diagnosis on an outpatient basis. Consequently, tracking the patient through time was not necessary.

There was, however, some evolution of contexts during a session with MEDIC. As the program's understanding of the case grew as findings were presented and questions answered, different c-schemas would match the situation. Usually, the new c-schemas were specializations of the old, allowing fine-tuning of MEDIC's behavior, but sometimes the c-schema would correspond to a different context altogether, which would change the hypotheses MEDIC was considering.

Orca. The CMB process as described above is being implemented in Orca, an intelligent mission controller for oceanographic AUVs (Turner & Stevenson 1991; Turner 1994; 1995; 1998). In particular, Orca's ECHO context manager will overcome some of the limitations of MEDIC's approach. It will diagnose c-schemas and merge them into a coherent picture of the overall context. A variety of context types is being considered, for example having to do with the vehicle, the environment, and the mission. Some work has begun on handling the changing character of the situation while within a context; for example, the physical properties of the environment change as an AUV transits a harbor, but throughout it makes sense to consider the AUV in the context of "in a harbor". We will not say more about Orca here, since it is covered adequately elsewhere (Turner 1998, e.g.), and the description of CMB above is from ongoing work on Orca.

CoDA. We are beginning to look at the role and use of contextual knowledge in multiagent systems. Our testbed system is an autonomous oceanographic sampling network (AOSN) (Curtin *et al.* 1993), and our project is CoDA (Cooperative Distributed AOSN controller) (Turner & Turner 1998). We are just beginning to look at context in this setting. Some of the issues that come to mind are how to use context to help select organizational structures, to select communication modes and channels, and to recognize and respond to opportunities to reorganize the system. Other interesting issues to be explored include the notion of shared context between the agents and how the agents can agree on what the context is. We believe CMB can be extended to the multiagent case, and we will explore this in the near future.

Sketch-and-Talk. We have also begun to examine the role of context in multi-modal (natural language and graphics) interfaces to geographical information systems. That work is the subject of another submission to this workshop (Turner *et al.* submitted). Briefly, we have identified several kinds of contexts (or components of the context) active in this application: the natural language discourse context; the graphics context, including a graphical equivalent to discourse context; the task context; the context defined by the kind of user and the particular user; and the temporal context of what is being discussed (e.g., "a building used to be here" versus "there is a building here now"). We are investigating the use of contextual knowledge for understanding ellipsis and other phenomena important to the application. We intend to investigate the applicability of CMB to the Sketch-and-Talk application in the near future.

Conclusion

In this paper, we have presented a brief discussion of our position on context-sensitive reasoning for AI applications. We intend to continue work on the approach developed so far, context-mediated behavior, in future single and multiagent systems.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. BES-9696044 and IIS-9613646 and under contracts N0001-14-96-1-5009 and N0001-14-98-1-0648 from the U.S. Office of Naval Research. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The author would also like to thank Elise Turner and other members of the University of Maine Cooperative Distributed Problem Solving Research Group as well as Patrick Brézillon for many helpful discussions.

References

Aikins, J. S. 1980. *Prototypes and Production Rules: A Knowledge Representation for Computer Consultations.* Ph.D. Dissertation, Stanford University.

Brézillon, P.; Gentile, C.; Saker, I.; and Secron, M. 1997. SART: A system for supporting operators with contextual knowledge. In *Proceedings of the 1997 International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*, 209–222.

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

Davis, R., and Buchanan, B. G. 1984. Meta-level knowledge. In Buchanan, B. G., and Shortliffe, E. H., eds., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, Massachusetts: Addison–Wesley Publishing Company.

Feltovich, P. J.; Johnson, P. E.; Moller, J. A.; and Swanson, D. B. 1984. LCS: The role and development of medical knowledge and diagnostic expertise. In Clancey, W. J., and Shortliffe, E. H., eds., *Readings in Medical Artificial Intelligence.* Reading, Massachusetts: Addison–Wesley Publishing Company. 275–319.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Georgeff, M. P., and Lansky, A. L. 1987. Reactive reasoning and planning: An experiment with a mobile robot. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 677–682.

Kolodner, J. L. 1984. *Retrieval and Organizational Strategies in Conceptual Memory.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Miller, R. A.; Pople, Jr., 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.

Patil, R.; Fikes, R.; Patel-Schneider, P.; McKay, D.; Finin, T.; Gruber, T.; and Neches, R. 1992. The DARPA Knowledge Sharing Effort: Progress report. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning.*

Schank, R. C. 1982. *Dynamic Memory.* New York: Cambridge University Press.

Turner, R. M., and Stevenson, R. A. G. 1991. ORCA: An adaptive, context-sensitive reasoner for controlling AUVs. In *Proceedings of the 7th International Symposium on Unmanned Untethered Submersible Technology (UUST '91)*, 423–432.

Turner, R. M., and Turner, E. H. 1998. Organization and reorganization of autonomous oceanographic sampling networks. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA'98)*, 2060–2067.

Turner, E. H.; Turner, R. M.; Grunden, C.; Mailman, J.; Neale, M.; and Phelps, J. submitted. The need for context in multi-modal interfaces. In *Proceedings of the 1999 AAAI Workshop on Reasoning in Context for AI Applications.*

Turner, R. M. 1989. When reactive planning is not enough: Using contextual schemas to react appropriately to environmental change. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, 940–947.

Turner, R. M. 1994. *Adaptive Reasoning for Real-World Problems: A Schema-Based Approach.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Turner, R. M. 1995. Intelligent control of autonomous underwater vehicles: The Orca project. In *Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics.* Vancouver, Canada.

Turner, R. M. 1997. Determining the context-dependent meaning of fuzzy subsets. In *Proceedings of the 1997 International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97).*

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

Wilkins, D. E. 1984. Domain-independent planning: Representation and plan generation. *Artificial Intelligence* 22(3).