Context-Mediated Behavior for AI Applications*

Roy M. Turner

Department of Computer Science, University of Maine, Orono ME 04469, USA
rmt@umcs.maine.edu

Abstract. Artificial intelligence (AI) applications are often faulted for their brittleness and slowness. In this paper, we argue that both of these problems can be ameliorated if the AI program is *context-sensitive*, making use of knowledge about the context it is in to guide its perception, understanding, and action. We describe an approach to this problem, *context-mediated behavior* (CMB). CMB uses *contextual schemas* (c-schemas) to explicitly represent contexts. Features of the context are used to find the appropriate c-schemas, whose knowledge then guides all aspects of behavior.

1 Introduction

Artificial intelligence (AI) applications are often faulted for being slow and brittle. Part of their slowness stems from the inherent difficulty of their tasks, but part also is due to their need to determine which knowledge is applicable. Brittleness, or non-graceful degradation of performance, arises in part from applying knowledge that is inappropriate for the context. These problems are exacerbated if the program is required to operate in many different contexts, performing a variety of tasks.

Humans are able to cope with real-time constraints while carrying out complex tasks in a wide range of contexts. We do this in part by being *sensitive* to the context. It is well-accepted, for example, that context plays an important role in human perception, decision-making, and social interaction.[1] Upon recognizing his or her context, a person immediately has available a great deal of information about what to expect, how to interpret what is sensed, and how to behave. For example, upon entering a theater, a person knows what to expect to find there (seats, refreshment counters, ticket offices, etc.), what actions are appropriate to achieve goals (e.g., buy refreshments rather than serving oneself), and general characteristics of appropriate behavior for the context—one behaves qualitatively different in a theater than, say, at a soccer match. This contextual

* This work was funded in part by the United States National Science Foundation under grant BES–9696044.

[1] See [5] for a discussion.

In *Lecture Notes in Artificial Intelligence 1415: Methodology and Tools in Knowledge-Based Systems*, J. Mira, A.P. del Pobil, and M. Ali (eds.), Springer, New York, 1998. (Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA–98–AIE), Benicàssim, Spain, June, 1998.) Vol. I, pp. 538–547.

knowledge is available immediately, automatically, and effortlessly. As the context changes, new contextual knowledge is always at hand. The result is that a person's behavior is usually appropriate for the context, and behavioral decisions can be made quickly based on readily-available contextual knowledge.

In this paper, we discuss what is needed to endow an AI application with context-sensitive reasoning abilities. We then describe our approach to this problem, called *context-mediated behavior* (CMB), which uses *contextual schemas* [6, 7] to represent an agent's contextual knowledge. Contextual schemas (c-schemas) are stored in a conceptual memory from which they can be retrieved based on features of the current context. They are merged to form a coherent picture of that context, then knowledge from the resulting *context structure* is used to control all aspects of the AI program's behavior. CMB is being implemented in ECHO (Embedded Context-Handling Object), the context manager for the Orca program [7, 8], a mission-level controller for autonomous underwater vehicles (AUVs).

2 Requirements for Context-Sensitive Behavior

Explicit representation of contextual knowledge. An AI agent[2] needs to be able to represent knowledge about contexts in which it may find itself. All AI systems encode some contextual knowledge: antecedent clauses in rule-based systems, preconditions in planners, etc. However, with a few notable exceptions (e.g., [1, 3, 6]), contexts are not treated as objects in their own right. This leads to redundant representation of contextual knowledge, inability to make inferences about the context itself, and unnecessary effort to find appropriate contextual knowledge to apply to the context.

By representing contexts explicitly, the agent can use features of contexts it knows about as the basis to identify the context it is in. It can make predictions about unseen features of the current context based on its *a priori* contextual knowledge. This information can also help it disambiguate or otherwise interpret sensory data and fine-tune the meaning of its conceptual knowledge to fit the context.

An agent should also associate *prescriptive* knowledge with its context representations so that it knows how to behave in the corresponding contexts. When the context is recognized, the agent will then instantly and effortlessly have available the information necessary to behave appropriately in that context. For example, the contextual knowledge about being "in a harbor" should keep the agent from going too near the surface (there is likely to be surface traffic) or too deep (the water is likely to be shallow). Should there be an emergency, the contextual knowledge should immediately suggest landing on the bottom rather than surfacing (since that would risk a collision). Prescriptive knowledge an agent needs about a context includes: event-handling information; goal-achievement information; attention-focusing information; and parameters and goals that should

[2] In the remainder of the paper, "agent" will be used to refer to any type of AI application program.

become effective automatically upon entering or leaving a context ("standing orders").

Contextual knowledge should be clustered. When a context is recognized, all relevant knowledge should be automatically "brought to mind". This avoids inferences or other work needed to collect the knowledge at run-time. Moreover, the knowledge remains available until the context changes, thus allowing rapid, context-appropriate decision-making. As discussed below, CMB uses frame-like knowledge structures called contextual schemas, or c-schemas, to do this.

A problem is choosing which contexts to represent. We define a context to be any identifiable configuration of features (environmental, mission-related, or agent-related) that has predictive power for the agent's behavior [5]. However, it would be unwise to try to represent all such contexts. The number would be immense, and the agent would be unlikely ever to encounter the vast majority of them. A guideline to reduce this number is: represent a context as a c-schema only if: (1) it cannot be represented by merging the knowledge contained in existing c-schemas or (2) such a merger fails to prescribe the correct behavior for the context [5].

Still, the number of possible contextual schemas is too large. One possible solution is to abstract the features of the context represented, e.g., by using variables or constraints rather than actual values. This allows a single c-schema to represent many different contexts. Another possibility is to abstract entire contexts. For example, the context "in Bar Harbor during outgoing tide with low power" could be considered an instance of more abstract contexts, such as "in Bar Harbor" or "in a harbor", "outgoing tide", and "low power". Such abstractions can be combined in different ways to represent many different contexts.

This also allows the agent to handle novel contexts. A new context may be adequately represented by a single abstract context, or the agent may need to piece together several existing contexts. For example, if an AUV has never had a sonar failure in a harbor before, it might still merge the context representations for "in a harbor" and "sonar failure" to decide how to behave.

Context recognition. To make use of contextual knowledge, an agent must have a mechanism to recognize the current context. This task is essentially diagnosis: given a set of known contexts (c-schemas), use the features of the current context to determine which it is an instance of. Thus, diagnostic knowledge must be associated with context representations.

Mechanism for tracking context change. An agent must be able to determine when its current representation of the context no longer matches the actual context. Context representations can help by describing situations in which they no longer fit the current context. For example, the context "in a harbor" might state that if the water is very deep, then an AUV is no longer in the context. It is unlikely that all such context transitions can be handled this way, however. An agent will need periodically to check its context representation against the world.

Mechanism for changing contextual knowledge. As the agent carries out its tasks, it will learn about new contexts and new things about old contexts.

Consequently, an agent needs a way to update its contextual knowledge, e.g., by learning.

3 Context-Mediated Behavior

Context-mediated behavior is part of the AI technique called *schema-based reasoning* [7], in which packets of knowledge representing patterns in problem solving are retrieved from a conceptual memory and interpreted by a reasoner to guide its behavior. For CMB, these packets are contextual schemas representing contexts. CMB is being implemented in ECHO, the Embedded Context Handling Object. ECHO is a module in Orca, a schema-based mission controller for autonomous underwater vehicles [7, 8]. An overview of ECHO's context-management process is shown in Fig. 1.

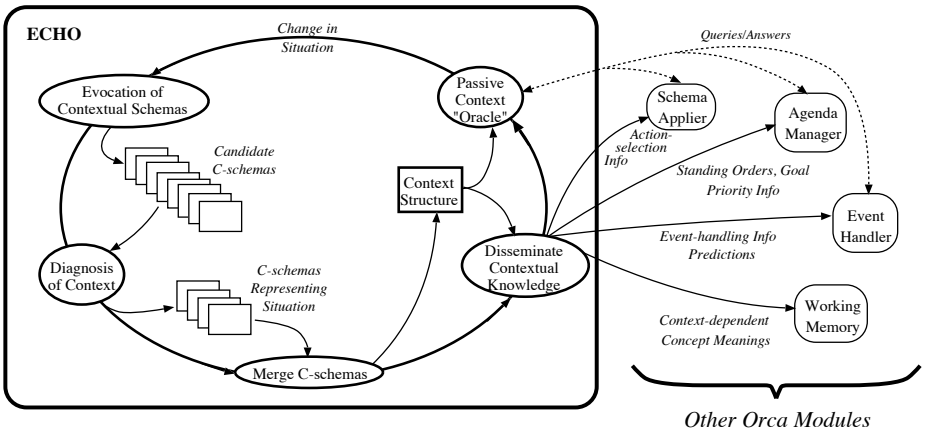

Fig. 1. ECHO.

3.1 Context Recognition

CMB's first task is recognizing the agent's current context. This is a diagnostic task. Features of the agent's current context, as reflected in sensory and other information, are compared to features predicted by known contexts, which are represented as contextual schemas. For example, `c-harbor`, a c-schema representing being in a harbor, would predict that the water column is shallow, there is surface traffic present, and so forth (see Fig. 2).

The overall diagnostic process is based on an abductive reasoning mechanism [4]. ECHO uses a conceptual memory to *evoke* candidate c-schemas based on the features of the context. In ECHO's memory, each c-schema is a memory structure that indexes related c-schemas (generally specializations of itself) based

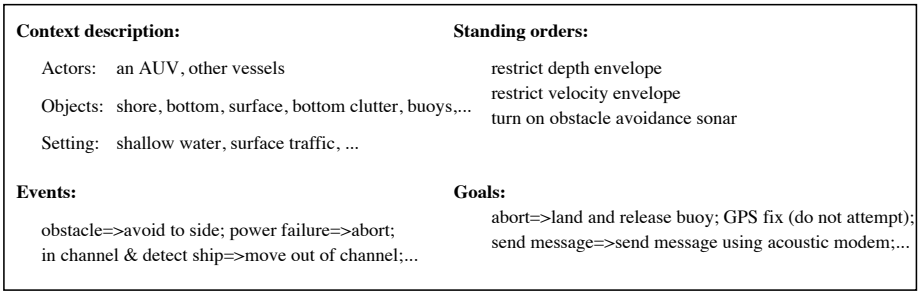

Fig. 2. Some of the information in `c-harbor`, a c-schema representing "in a harbor".

on features distinguishing them from itself [7]. Features that are different become indices that allow discrimination between c-schemas. Features are abstracted as much as possible so that similar differences point to the same specialized c-schema. When entering a new context, its features are presented to the memory, which traverses the indexing structure and returns a set of c-schemas similar in some way to the context.

Once evoked, the candidate c-schemas are combined into *logical competitor sets* (LCSs; see [2]) based on the observed features they explain. Each LCS is then solved by differential diagnosis using its c-schemas' predictions and how strongly they are evoked by the context. The c-schemas solving each LCS are merged to create the context structure.

3.2 Perception and Understanding

The context structure contains knowledge to aid perception and understanding. Predictions about unseen features of the context can bias interpretation of sensory information toward those features and can help prepare the agent for the occurrence of an unanticipated event.

In addition, the context structure contains information about context-dependent meaning of concepts [9]. For example, Orca's c-schemas contain context-dependent meanings of fuzzy linguistic variables. Orca uses a simple version of fuzzy logic in which *linguistic variables*, such as "depth", take on one of a set of *linguistic values*, such as "shallow" (e.g., [10]). The meaning of each linguistic value is specified by a membership function that maps an actual ("crisp") numeric value to a value in the range [0,1], indicating its degree of membership in the fuzzy set referred to by the linguistic value.

By storing the mapping in c-schemas, the context-dependent meaning of linguistic values can be represented. For example, a c-schema representing "in a harbor" would suggest a different meaning for the "nominal" depth than would a c-schema representing "in the open ocean". In the former, the default nominal depth should be the middle of the water column (to avoid surface traffic and bottom clutter), while in the latter, remaining near the surface would likely be

best. When c-schemas are merged to form the context structure, the fuzzy values' meanings are merged as well (see [9]).

3.3 Handling Events

One of the most important things that an AI application must do is handle unanticipated events rapidly and appropriately for the current context. Some events may be predicted quite accurately. For example, the catastrophic failure of an AUV can be predicted if it is in the context of "leaking". Other events are in a sense unanticipated: e.g., a catastrophic failure *could* occur in the context of "in a harbor", but it is not expected. However, it is critical to know the appropriate (context-specific) response should the event occur.

In CMB, c-schemas contain knowledge about both kinds of events. Knowledge about a predicted event can be used to plan for the event's occurrence as well as to detect when a predicted event has *not* occurred. Knowledge about unanticipated events allows them to be quickly recognized and handled appropriately for the context. A c-schema contains knowledge about an event only when the way that event is handled is affected by the context.

Three kinds of information are recorded for each event: how to detect it, how to estimate its importance, and how to respond to it. In Orca, event detection information is represented as fuzzy rules that are given to its Event Handler. Event importance information is represented as importance estimates as well as fuzzy rules that tailor the importance to the particulars of the context. Event response information suggests a goal to activate when the event occurs. The event is handled when Orca selects the goal as its focus of attention. This cleanly integrates event handling with normal goal achievement.

3.4 Achieving Goals

Contextual schemas contain information about how important particular goals are in the context and about context-dependent ways to achieve them. The former is used to help focus attention. For example, in most contexts, an AUV's periodic goal of identifying its location (e.g., via surfacing for a global positioning system fix) is fairly important. However, in the context of rescuing a diver who is using a sonar transponder to signal for help, the goal of determining location would be much less important. In Orca, importance information is represented by priority estimates and fuzzy rules for fine-tuning the estimate based on the particular context. These are used by its Agenda Manager to focus attention. Information about how to achieve goals is in the form of suggestions about which procedural schemas (p-schemas) to use [7].

3.5 Standing Orders

Each c-schema contains a set of things that should be done when entering the corresponding context, that should be in effect when in that context, or that

should be done when exiting the context. These "standing orders"[3] are either goals to activate/deactivate or parameters to set. CMB modifies behavior automatically by activating or deactivating standing orders when entering or leaving a context. For example, a c-schema for "in a harbor" would inform an AUV not to go too near the surface nor too deep. Similarly, a c-schema for docking would turn off obstacle avoidance behavior.

This mechanism allows an agent to modulate most aspects of its behavior based on its context, not just those associated with particular activities such as goal achievement or handling unanticipated events. Such automatic "background" behavior modulation can be very important in ensuring behavior that tracks the agent's evolving context appropriately.

4 Conclusion and Future Work

Context-sensitive behavior is necessary if an AI program is to rapidly modify its behavior to fit its context. Context-mediated behavior is an approach to this problem that explicitly represents contextual knowledge. Its contextual schemas are retrieved based on the context's features, then information from them is merged to guide all facets of the agent's behavior while in the context. This knowledge remains available until the context changes, thus allowing rapid, context-appropriate decision-making.

At the current time, implementation is in progress for the version of CMB described above. This implementation is embodied in Orca's context-management object, ECHO. We anticipate a complete, initial version of ECHO at or near the time of publication. Future versions will refine ECHO's knowledge representation and reasoning methods based on experience using the initial version. Initially, CMB will be evaluated via simulation experiments using Orca; ultimately, we plan to conduct in-water experiments.

Beyond that, future work will focus primarily on those requirements discussed above that are not currently part of ECHO's design. Contextual schema merger and detecting context change, for example, are difficult problems that will not be completely solved in the initial version of ECHO. Learning contextual knowledge is an important long-term goal of the work, since this will let an agent tailor its contextual knowledge over time to the particular set of contexts in which it operates.

We expect CMB to be useful for a wide range of AI applications. This includes not only other planning systems, but also other kinds of AI systems, such as rule-based systems and neural networks. For a rule-based system, c-schemas would contain rules. This has been suggested before [1], but CMB would add sophisticated c-schema retrieval mechanisms, c-schema merger, and the standing orders mechanism to modulate the application's behavior. A c-schema in a neural network-based system would contain weights appropriate for the network when operating in the corresponding context. When the application encounters an

[3] Term due to D.R. Blidberg.

instance of the context, the weights, and hence, the network's behavior, would change automatically to fit the context. We plan to explore these kinds of uses of CMB in future work.

References

1. Aikins, J.S.: A representation scheme using both frames and rules. In: Buchanan, B.G., and Shortliffe, E.H. (eds.): *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison–Wesley Publishing Company, Reading, Massachusetts (1984) 424–440
2. Feltovich, P.J., Johnson, P.E., Moller, J.A., and Swanson D.B.: 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*. Addison–Wesley Publishing Company, Reading, Massachusetts (1984) 275–319
3. Guha, R., and Lenat, D.B.: Cyc: A midterm report. *AI Magazine* **11** (3) (1990) 32–59
4. Miller, R.A., Pople, H.E., Jr., and Myers, J.D.: INTERNIST–1, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine* **307** (1982) 468–476
5. Turner, R.M.: Context-mediated behavior for intelligent agents. *International Journal of Human–Computer Studies* (in press)
6. Turner, R.M.: 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*, Detroit, MI (1989) 940–947
7. Turner, R.M.: *Adaptive Reasoning for Real-World Problems: A Schema-Based Approach*. Lawrence Erlbaum Associates, Hillsdale, NJ (1994)
8. Turner, R.M.: Intelligent control of autonomous underwater vehicles: The Orca project. In: *Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics*. Vancouver, Canada (1995)
9. 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)
10. Zadeh, L.A.: Fuzzy logic, neural networks, and soft computing. *Communications of the ACM* **37** (3) (1994) 77–84