

Placing Newly-Arising Goals in the Proper Context

Elise H. Turner, Roy M. Turner & Erik Albert

Computer Science Department

University of Maine

Abstract

When agents collaborate in the real world, they must be able to handle new goals that arise as they are executing a plan. If we can predict how goals will be satisfied, we can predict the context in which actions will be executed. When new goals arise, we can place them in the predicted context where they will be most effectively achieved. In this paper, we first discuss JUDIS, an implemented system which places newly-arising goals into predicted discourse contexts. Next, we discuss how the approach used in JUDIS is being extended to more general problem solving.

1 Introduction

For independent agents and agents working in multi-agent systems alike, the context in which an action is performed often determines the success of that action. In conversation, it is essential that the proper utterances are placed in the proper context. In addition to allowing those utterances to be interpreted properly, grouping utterances by topic allows the conversation to be coherent. In other kinds of collaborative work, placing actions in the proper context is also important. By grouping actions together with others that require the same important features of a context, the actions can be achieved efficiently. Also, agents can be more helpful if they recognize another's action as being in the context of a larger plan. However, since in the real world new goals may arise after an agent has begun to execute its plan, agents must be able to place goals in the proper context as they arise.

In this paper, we present our approach to placing actions within the proper context. This includes *focusing attention* on the proper action to be executed next. Actions that can be used to achieve an agent's current goals are grouped together by important, predictable features of the context. At the time that the next action must be chosen, potential actions are evaluated based on how well they fit into the current context and the priority of the goal that they will be used to achieve.

We begin in Section 1 with a discussion of JUDIS [1], a dialogue system in which the approach was first implemented to create coherent conversations between a multi-agent system and a user. In Section 2, we discuss our plans for extending this approach to general problem solving, as well as conversation, in collaborative agents.

2 Fitting Goals into Conversation Contexts

The approach described in this paper was first implemented in JUDIS [1]. JUDIS controls conversation for a distributed problem solver in the domain of menu planning [2]. As each problem solver works, goals might arise to get information from or to give information to the user. These goals can be achieved by adding utterances in the conversation. However, because different problem solvers may be working on different parts of the menu at the same time, the utterances cannot simply be included in the conversation in the order in which they arise. Instead, JUDIS must find the

proper context for the utterance. In JUDIS we are concerned only with using the context to make the conversation coherent.¹ Consequently, the context of the current discourse can be represented by the current topic.

To group utterances together, JUDIS uses a *template* for conversation. The template is built from conversation schemas represented using Conversation MOPs [3; 4]. Initially, the template is a very general structure, predicting the opening and closing of the conversation in detail and merely the expectation that there will be a middle that will be used to discuss a meal. At the end of the opening, JUDIS asks the user for information about the meal that is being planned. As goals arise from problem solvers² JUDIS adds detail to the middle of the conversation.

In JUDIS , utterances are added to the template through a conversation schema that can include both goals. For example, consider a situation in which a traditional planner working on the appetizer wants to suggest gazpacho and sends this goal to JUDIS . Assume this is the first goal. JUDIS knows that the discussion of the meal will include a discussion of the appetizer. It adds this goal to the appetizer topic. If no other goal were to enter the system before JUDIS began discussion of the appetizer, it would simply tell the user that it would recommend gazpacho.

Now suppose, that at the same time that the planner is working on the appetizer, a case-based reasoner is working on the main course. It is trying to apply a case in which someone became sick from the main course of stuffed peppers because he was allergic to green peppers. The case-based reasoner will only suggest this main course if it knows that it can avoid the failure of the remembered case, so it sends a goal to JUDIS to find out if anyone is allergic to green peppers. It will not ask JUDIS to suggest stuffed peppers until it knows the failure can be avoided. Assume that this goal has arisen just after JUDIS has switched the topic to the appetizer, but before making the suggestion of gazpacho. To add the question about green peppers, JUDIS extends the topic of the appetizer to include a discussion of the ingredients in the appetizer. This is done using the conversation schema for a list.

Now, suppose the planner is trying to determine if the user would like to include parsley in the gazpacho. It creates a goal to ask the user about parsley. Since parsley is an ingredient of gazpacho, the utterance will be included as part of the list already in the template.

The conversation is being conducted as new goals arise and must be added to the template. To help JUDIS participate in the on-going conversation without frustrating the user, the template always represents a coherent conversation. However, the order in which utterances will be said is not completely determined by the template, so JUDIS cannot simply focus attention on the next utterance in the template to select the next utterance. There are two main reasons for this. First, the schemas themselves do not completely determine which utterances will be said when. Instead, conversation schemas often provide only a partial ordering for their scenes. They also have optional scenes which may or may not be included in the conversation. Second, there may be reasons why a schema might be violated. An urgent utterance, such as warning of an immediate safety hazard, may cause a speaker to abandon a conversation schema entirely. There are also cases where speakers must move back to previous topics to incorporate goals that were not known when those topics were originally covered. In some cases, a speaker may need to move ahead in the conversation to handle a high priority goal quickly.

In JUDIS , we see the problem as a tension between convention, as represented by the schemas

¹Using the context to interpret utterances is beyond the scope of this work.

²Due to the delayed implementation of the distributed system, JUDIS worked with the problem solvers on only small problems. To test JUDIS for more complex streams of goals, streams of conversation goals from each problem solver and streams of goals constructed following the problem solving approach of the problems solvers were used.

in the template, and intention, as represented by the priorities of the active goals. JUDIS combines these influences in the form of activation sent to utterances in the template. If the utterance is ready to be said based on conventions of conversation, it receives a fixed amount of activation. An utterance is ready to be said if two conditions are met. First, the schema which contains the next utterance must also contain the previous utterance, or the schema which contains the previous utterance must have completed and the schema which contains the next utterance must be ready for execution. Second, all mandatory schemas and utterances which must precede the utterance must have been executed.

The activation based on intention is given to utterances depending on the priorities of the goals which they are meant to achieve. For very high priority goals, the priority of the goal will outweigh the template and the utterance will be said immediately. In less extreme cases, the priority of the utterance will help choose between utterances that are ready to be said in a schema with partially-ordered steps. Intention-based activation also allows optional utterances in schemas to be included in the conversation. The details of our method of passing activation is discussed in more detail in [5].

3 Exploiting Context to Focus Attention in General Problem Solving

We are beginning to design a system which will take the approach to focusing attention used in JUDIS and apply it to general problem solving. The system will work in the domain of Autonomous Oceanographic Sampling Networks (AOSNs) [6]. In AOSNs, underwater vehicles and non-mobile sensor platforms collaborate to collect data for scientific missions. Agents in the AOSN are given a mission by a scientist or group of scientists. New goals can arise, for example, in response to unexpected events in the environment or as scientists make new requests. As in JUDIS, we fit the actions required to satisfy the new goals into an appropriate context created by executing the existing plan.

Contexts are too rich for us to try to match potential contexts expected in the plan with the perfect context for executing a new action. Instead, we must group actions together using only particular features of the context. From our domain-dependent work with JUDIS, we have learned that two characteristics should apply when deciding the features of the context that should be used for the grouping. Goals should be grouped together by features of the context that are costly to acquire and which can be predicted with a great deal of certainty. When we apply this principle to problem solving in AOSNs, we group actions together if they take place at the same location or if they use the same resources that are costly to acquire. We call these resources, *high-cost resources*.

For general problem solving, there is some difficulty in predicting the new actions that must be fit into the plan. In JUDIS, the problem solvers' goals to get or to give information were easily associated with individual utterances.³ For the general problem solver, hierarchical schemas are used to store plans that can be used to achieve goals (similar to those in Orca [7]). Important features of the context can be stored in a generalization of plans to achieve the goal. For example, if an agent had a plan to acquire a book, the most general schema could indicate that the agent needs to be at a library or a bookstore to get the book.

In JUDIS, the new actions were added to the template by linking them through a detailed

³Of course, there are times when an utterance can satisfy several goals or when a single goal can be satisfied with several utterances. JUDIS did not handle these cases.

discourse schema. This is because we were concerned with conversation structure being followed to create a coherent conversation. However, it is time consuming both to add this much detail when a new goal becomes active and to change the plan if it fails. For these reasons, and because plan coherence is less important in general problem solving, we simply create nodes for high-cost resources and for locations.

When a new goal arises, it is added to the agenda. When the problem solver focuses attention on the goal, it finds a schema which can be used to satisfy that goal and places the actions of the schema on the agenda. If the schema has information about the need for a high-cost resource or for the vehicle to be at a specific location, that information can be used to help the problem solver fit actions into the proper context. So, for example, when the goal to read a book is replaced on the agenda by the schema to acquire a book, the problem solver has the information that the agent must be either at a bookstore or a library. If both nodes exist, the schema for acquiring a book at the given location can be linked to the nodes. If neither exist, new nodes can be formed to represent the library or the local bookstore. However, if one node exists, the problem solver uses the prediction that the agent will be in the proper context for one of the schemas. For example, if the agent will be at the library for a meeting, it will be in the proper context for acquiring a book at the library. In addition to being linked to these organizing nodes, the act remains linked to the schema on the agenda.

There are several details that must be worked out for real-world planning. For example, we must make sure that preconditions for any of the actions associated with the node are not violated by the others. For example, an after hours meeting at the library would *not* provide the proper context for getting a book. We also need to decide how to determine that actions can be removed from a node because they have been satisfied elsewhere.

We, again, use an activation model to focus attention on an action on the agenda. Activation comes from three sources. The priority of the goal contributes to the activation of each action that will achieve it. There are also two kinds of context which contribute to activation: the schema's connection to features in the predicted context through the organizational nodes and the context that the schema provides through the schema that it is part of. The last is similar to the JUDIS's convention-based activation and is meant, in part, to help other agent's understand the acting agent's current plan. It is especially important for agent's to follow their schemas when engaging in conversation. As we work out the details of activation, we will take this into account. Activation can also be modified by contextual schemas. This is done for goal priorities in Orca and will be extended to other forms of activation later in our work.

4 Conclusion

We are developing an approach to placing newly-arising goals in context and focusing attention on those goals as an agent executes its plans. The approach was implemented to be part of a user interface to a distributed problem solving system. We are currently extending this approach for more general problem solving in a multi-agent system.

References

- [1] E. H. Turner. Organizing dialogue from an incoherent stream of goals. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING-92)*, pages 338–344, Nantes, France, 1992.

- [2] R. E. Cullingford and J. L. Kolodner. Interactive advice giving. In *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, pages 709–714, Atlanta, Georgia, 1986.
- [3] K. Kellermann, S. Broetzmann, T.-S. Lim, and K. Kitao. The conversation MOP: Scenes in the stream of discourse. *Discourse Processes*, 12(1):27–61, 1989.
- [4] E. H. Turner and R. E. Cullingford. Using conversation MOPs in natural language interfaces. *Discourse Processes*, 12(1):63–90, 1989.
- [5] E. H. Turner. Integrating intention and convention to organize problem solving dialogues. Technical Report GIT-ICS-90/02, School of Information and Computer Science, Georgia Institute of Technology, 1990. Ph.D. thesis.
- [6] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6(3), 1993.
- [7] R. M. Turner and R. A. G. Stevenson. ORCA: An adaptive, context-sensitive reasoner for controlling auvs. In *Proceedings of the 7th International Symposium on Unmanned Untethered Underwater Submersible Technology (AUV '91)*, 1991.