Flexible Conversation Management using a BDI Agent Approach

Wilson Wong, Lawrence Cavedon, John Thangarajah, Lin Padgham

School of Computer Science and IT, RMIT University, Melbourne, Australia
{wilson.wong,lawrence.cavedon,john.thangarajah,lin.padgham}@rmit.edu.au

Abstract. We describe a BDI (Belief, Desire, Intention) goal-oriented architecture for a conversational virtual companion embodied as a child's Toy, designed to be both entertaining and capable of carrying out collaborative tasks. We argue that the goal-oriented approach supports both structured conversational activities (e.g., story-telling, collaborative games) as well as more "free-flowing" engaging dialogue with variation and some unpredictability. BDI plans encode the knowledge required for the structured engagements, with the use of multiple plans for conversational goals providing variation in the interactions.

1 Introduction

A conversational virtual companion [1] must be able to engage the user, which in turn requires the ability to support both structured conversation-based activities (e.g., story-telling, collaborative games) as well as more "free-flowing" chatty dialogue. Unlike task-based dialogue, the purpose of what we refer to as *Conversational Activity* is not simply to successfully perform a task (e.g., book a flight), but to actually engage and entertain the user over an extended period.

We describe a BDI architecture for a conversational agent that supports both task-oriented dialogue as well as more "chatty" conversations. The BDI agent model has been used successfully in a range of applications (e.g., see [2, Ch. 10]) that require a mix of reactive behaviour and goal-directed reasoning; this design model supports different means for achieving a goal depending on context and other factors [3]. The mixed reactive/proactive model enables the management of coherent conversational activity while still being responsive to unexpected user input. Scripted BDI plans provide knowledge of how to perform different types of *Conversational Activities*, while appropriate *Knowledge Bases* contain information about entities associated with those activities. BDI agent-based approaches to dialogue management have been previously proposed (e.g., [4, 5]); however, these have typically been for task-oriented conversations (e.g., accessing email or managing an appointment). A novelty of our approach is the use of the BDI framework to provide variability in the way a goal is progressed, as well as in the conversational content.

The content for our agent appears in the form of *Conversational Fragments*, which are effectively templates of adjacent pairs of utterances. These are dynamically assembled into sequences to construct conversations. However, unlike chatbots, the conversation is strongly guided by the plans of the Conversational Activity which provides the narrative framework and coherence typically missing from chatbot-generated dialogue; our approach also supports *mixed-initiative*.

2 Architectural Overview

Our conversational infrastructure is implemented in the context of an interactive Toy, designed as a virtual companion for children. The Toy contains a *Dialogue Manager* (`DM`) which is composed of a *Conversation Management* (`CM`) component that interacts with an *Activity Management* (`AM`) component. These are both implemented using a BDI agent-oriented methodology.

The `AM` selects and instantiates specific *Conversational Activities* which direct the structure and the kind of content for the Toy utterances, while the `CM` manages the specific details of choosing utterances and interpreting input. The `CM` has dialogue processing strategies built as plans. For example, there are plans designed to handle errors or low-confidence results from speech recognition; plans to handle utterance content and update the information state; and plans to manage concurrent conversational threads and select which of a number of candidate responses to output. The full system involves automatic speech recognition (ASR) and other inputs, which we do not describe here.

The `CM` is designed to be multi-domain and extensible via Conversational Activity modules. These modules are designed to guide conversation around particular activities within a content domain and encapsulate the plans and data required for this. A conversational activity module contains: a knowledge-base segment; a set of conversational fragments; a collection of plans to handle the particular conversational activities of the module; and for each top level conversational activity an input grammar which specifies the form of the input to be interpreted as a trigger for starting this activity. The input grammar is specified using regular expressions that can be matched against user input. For example the input grammar for the story-telling activity is: "`* tell * story *`" which results in instantiation of a goal to initiate a story-telling activity.

The input handling component of the `DM` analyses and extracts weighted keyphrases, topics, sentiments and requests from the user inputs. We use the Stanford Parser [6] for part-of-speech tagging, Morphadorner[1] for lemmatisation, and the dictionary-based approach for detecting sentiment [7] and request. It analyses whether the input is a response that (1) matches one of a set of templates for continuation of the current conversational activity (`OK`), (2) is a specific request, i.e., matches an input grammar to some conversational activity (`Specific-Request`), (3) is an expression of negative sentiment but without any specific request (`Negative`), or (4) is not able to be understood (`Not-Understood`). The analysed input is then provided to the `AM` for decision as to what to do next, which can be to continue with the current activity, or to abort/suspend it in which case either a new Conversational Activity will be instantiated, or an existing suspended one will be resumed.

[1] http://morphadorner.northwestern.edu

In generating output the `CM` receives weighted contextual information that has been built up from both inputs and utterances during the interaction, as well as information provided by the plans within the current Conversational Activity. It uses the *Fragment Library* to find appropriate utterance templates, instantiates any variables, using both contextual information and the *Knowledge Base* and then strategically determines the response to be uttered. The *Fragment Library* contains Conversational Fragments, which are pre-scripted pieces of dialogue, which may be tagged as relevant to particular goals, and may contain both input and response variables (e.g. `$FOOD`, `$ANIMAL` etc), to allow scripting of more general purpose fragments. A generic response variable `$ETC` is used when any response is considered acceptable. The use of Conversational Fragments avoids the need for full natural language generation and allows the Toy to generate quite flexible interactions by choosing amongst relevant fragments in a non-deterministic, but nevertheless guided manner.

When interacting with the child, the Toy suggests possible Conversational Activities such as a cooking game/role play, a story, a quiz, etc. These activities are represented as BDI goal-plan structures (i.e., a set of plan templates in the agent's plan library) which guide the different aspects of the activity and the selection of fragments for the Toy to utter in pursuit of that activity. Importantly, the specific utterances are **not** part of the activity structure. Rather, it includes goals and plans to move the conversation in particular directions. The plans can provide contextual information which is used by the `CM` to select appropriate outputs. Analysis of the child's input also provides data that is used to determine how to progress the activity. For example keyphrases from the input may help to guide plan selection within a particular activity. This notion of Conversational Activity helps to keep the dialogue cohesive, while allowing flexibility. It also meets the requirement that an engaging interaction should provide interesting tasks for the child while staying controlled by them. We note that activities can be resumed or paused to allow switching between them, either to follow the child's topic requests or to insert personalised contributions, for example.

3 Activity Management

Central to our architecture is the library of goal-plan structures for directing coherent interaction with the user. Our architecture assumes a BDI plan library, with plans that have a *trigger* (the goal[2] they will achieve), a *context condition*, which determines the situation under which this plan is to be used, and a *body* which contains the plan code, which we can think of as *plan-steps*. Some of these steps will be subgoals, which trigger the selection of plans to achieve them. This gives rise to a goal-plan tree where a goal can have many possible plan options to achieve it and a plan may contain many (sub)goals. As analysed in [3], this provides a very large number of possible executions within a relatively compact

[2] Goals are often called and implemented as *Events* in BDI agent implementation platforms.

structure. According to the example in [3], a goal-plan tree with depth 3, 2 plans per goal, and 4 subgoals per plan will result in over 2 million executions.

In our case, this equates to 2 million different potential conversations resulting from a single activity tree of this size. It is this which we exploit to achieve the desired variability, while retaining coherent, goal-oriented dialogue. As variability itself is a key aim in our design, we require multiple plans that can be applicable to each sub-goal. These are then chosen somewhat randomly to avoid the child obtaining the same responses to their own inputs.

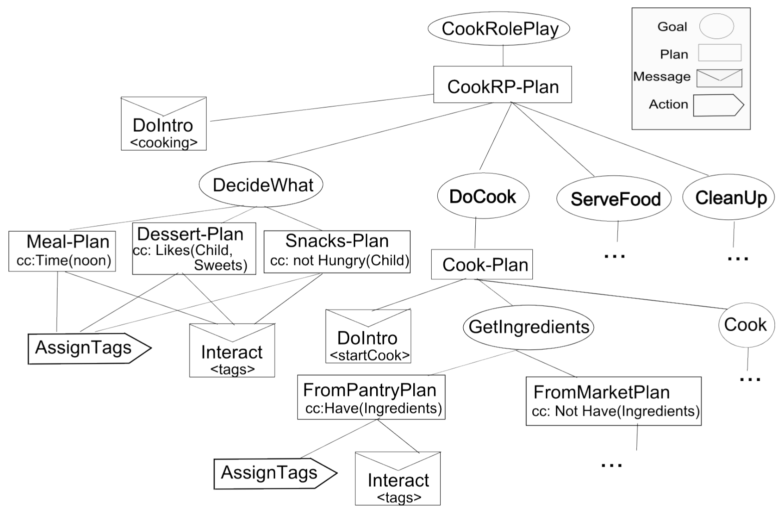

Fig. 1. Example activity: Cooking role play

Processing Conversational Goals: Figure 1 shows a (partial) example goal-plan tree for the cooking role-play activity in the Toy. In this we see that the top-level goal has a single plan which guides the structure of the activity. It is possible to have different plans to choose from at the top level, providing even more variety. This plan has a sequential set of subgoals, each with a set of plans to choose from, and so on. Prior to executing the subgoals the plan first sends a `DoIntro` message [3] to the `CM` carrying information about the current activity (Cooking) and triggers a plan in the `CM` to select a suitable introductory fragment for this activity. This introductory fragment (without any expected response) will be prefixed to the next system output. It will then decide what to cook using one of the plans that achieve the `DecideWhat` subgoal. This involves performing some interactions with the user, by assigning suitable tags (`AssignTags` action) that will be used when selecting output fragments and posting an `Interact` message to the `CM` to perform the interaction. This results in the `CM` determining an output fragment and analysing the user response, which is then provided

[3] The `DoIntro` message is essentially a subgoal that is performed by another agent.

back to the plan in the form of keyphrases and a response category. Assuming the response is `OK`, when the plan has completed its interactions, it decides (based on the keywords collected) what food it believes is going to be prepared and the activity progresses onto the next subgoal `DoCook`, which is managed in a similar way. The `DoIntro` goal assists in smoothly moving between sub-activities.

Managing Activities: An important capability of the Toy is to be responsive to the user. The agent has to be able to drop the current Conversational Activity if this is inferred from user input. This may occur via an explicit demand (e.g., *"I want to do something else"*), detecting that the child has drifted away from responsiveness, or opportunistically (e.g., to insert a joke pertinent to the previous utterance). The intention-based BDI approach incorporates a framework for suspending plans (and thereby activities), including infrastructure for reasoning about whether to resume or terminate a suspended activity later on. This provides a seamless approach to inserting "digressions", which are common in natural conversation flow. The BDI architecture thus provides the framework for transitioning between Conversational Activities. However, we require such transitions to be conversationally smooth and coherent. To facilitate this we use a *semantic relatedness measure* [8], and *transition fragments*. The latter are specially designed fragments to provide the conversational glue between activities (e.g. *"OK, lets finish the* `$FROMACTIVITY` *now and do something else."*).

4 Example Interaction

Figure 2 shows an example interaction with the Toy. The agent started with the chat activity at Line 4 and the keywords from the subsequent user inputs and system outputs were collected to form the context. We will highlight several features discussed above using the line numbers:
`Line 13`: Select the cooking roleplay based on context;
`Line 15`: `CookRP-Plan` guides conversation; `GetIngredients` subgoal invoked;
`Line 16`: Explicit request from user matches trigger for story telling activity;
`Line 17`: Cooking activity suspended; Agent chooses story using relatedness;
`Line 24`: Story ends; Cooking activity is resumed using a transition fragment;
`Line 27`: Question detected; the question-answering activity invoked;
`Line 36`: Cooking activity ends; Agent selects new Conversational Activity.

Figure 3 shows two conversations that evolve from the same start point, and use the same Conversational Activity structure. However, the choice of different plans to achieve the subgoals, as well as different fragments to output for the `Interact` goal, results in different interactions (i.e., variability).

5 Conclusion

We have described a goal-oriented approach to managing Conversational Activities between a user and a conversational virtual companion. We have performed

```
04: SYSM: do you like chocolate?
05: USER: i love chocolate.
06: SYSM: i love it too. i wish i could eat it for breakfast...
07: USER: chocolate for breakfast? yuck!
08: SYSM: what do you eat for breakfast instead of chocolate?
09: USER: toast.
10: SYSM: what do you like to eat on your toast.
11: USER: ice cream.
12: SYSM: yum yum! delicious.
13: SYSM: do you want to make some ice cream?
14: USER: yeah.
15: SYSM: do you have the recipe with you?
16: USER: can you please tell me a story?
17: SYSM: sure. susie the ice cream moose was...
...
23: SYSM: that's the end of the story.
24: SYSM: let's continue with our cooking. so, do you have the recipe?
25: USER: no.
26: SYSM: you'll need milk and chocolate.
27: USER: where does chocolate come from?
28: SYSM: chocolate comes from cacao trees.
29: SYSM: let's continue with our cooking, do you need to get out a pan?
...
34: USER: it's done!
35: SYSM: yum yum! your ice cream is ready.
36: SYSM: now, let's talk about the zoo, ok?
```

Fig. 2. An example conversation.

```
          USER: i really like cooking.
(a)       SYSM: cooking is great.
          USER: yeah. i agree.
          SYSM: what do you want to cook for your friends.

          USER: i really like cooking.
(b)       SYSM: if we do some cooking, we can share it with  your friends.
          USER: i love cooking and sharing.
          SYSM: let's decide what to cook.
```

Fig. 3. Examples showing variability in system responses.

an initial evaluation, with human subjects rating the system for coherence, interesting variability in outputs, and overall appropriately sustained focus over a conversation. Overall rating by the subjects was positive, notwithstanding relatively limited coverage in terms of content and knowledge base. We have recently developed techniques to automatically mine content from web forums [9], which should significantly alleviate this problem. We are also developing a model of *proactive engagement*, whereby the conversational agent actively monitors user engagement [10] and applies conversational strategies when required.

Acknowledgment

This work is partially supported by the Australian Research Council and Real Thing Entertainment Pty. Ltd. under Linkage grant number LP110100050.

References

1. Cavazza, M., Camara, R., Turunen, M.: How was your day? a companion ECA. In: Proc. AAMAS, Toronto (2010)
2. Wooldridge, M.: Introduction to MultiAgent Systems. 2nd edn. Wiley & Sons (2009)
3. Winikoff, M., Padgham, L.: Developing Intelligent Agent Systems: A Practical Guide. Wiley Series in Agent Technology. Wiley and Sons (2004)
4. Nguyen, A., Wobcke, W.: An agent-based approach to dialogue management in personal assistants. In: Proc. IUI, San Diego, California (2005)
5. vanOijen, J., vanDoesburg, W., Dignum, F.: Goal-based communication using bdi agents as virtual humans in training: An ontology driven dialogue system. In: Proc. AAMAS Workshop on Agents for Games and Simulation, Toronto (2010)
6. Klein, D., Manning, C.: Accurate unlexicalized parsing. In: Proceedings of the 41st Meeting of the ACL. (2003)
7. Tang, H., Tan, S., Cheng, X.: A survey on sentiment detection of reviews. Expert Systems with Applications **36 Issue 7**(Pages 10760-10773) (2009)
8. Macias-Galindo, D., Wong, W., Thangarajah, J., Cavedon, L.: Coherent topic transition in a conversational agent. In: Proceedings of the 13th Annual Conference of the International Speech Communication Association (InterSpeech), Oregon, USA (2012)
9. Wong, W., Thangarajah, J., Padgham, L.: Health conversational system based on contextual matching of community-driven question-answer pairs. In: Proc. CIKM Demo Track, Glasgow (2011) 2577–2580
10. Castellano, G., Pereira, A., Leit, I., Paiva, A., McOwan, P.: Detecting user engagement with a robot companion using task and social interaction-based features. In: Proc. ICMI, Cambridge, USA (2009)