A Multi-threading Extension to State-based Dialogue Management

Tina Klüwer & Hans Uszkoreit
German Research Center for Artificial Intelligence
Alt-Moabit 91c, 10559 Berlin, Germany
tina.kluewer@dfki.de, hans.uszkoreit@dfki.de

1 Introduction

This paper describes a dialogue manager, which provides support for multiple, interleaved conversation threads. Multi-threaded dialogues are frequently initialized by humans (Shyrokov et al., 2007), (Yang et al., 2008). Interleaved dialogue threads differ from embedded dialogue threads insofar that they allow for threads being alternated entangled. Although multi-threaded conversations are a frequent human behavior, support for multi-threaded conversations in dialogue systems is very rare. One example is (Lemon et al., 2002), who describe a possibility to integrate multi-threading into an Information State Update model. However, (Yang et al., 2008) criticize (Lemon et al., 2002), because they neglect to signal conversation switches made by the system. The system described in (Nakano et al., 2008) is able to manage multiple tasks through several expert components for every task. However, experts cover fine-grained tasks such as "understanding a request for weather information". They do not capsulate substructures of a dialogue and are therefore not comparable to conversation threads.

This paper presents a state-based dialogue manager, which supports multi-threaded behavior and offers conversation switch markers.

2 Multi-threaded Dialogue Support

Input to the dialogue system is by default interpreted in the context of the currently active dialogue thread. If the current thread is found to be inappropriate, the dialogue manager needs to replace the active thread with either an until now inactive thread or with an active but paused thread. Dialogue Management is based on a finite-state graph. The finite-state automaton is described by a hierarchical state-transition diagram including Harel's state charts. In our dialogue manager conversation threads are special types of supernodes.

Conversation threads can occur in three different conditions: active, paused and inactive. Analog to the activation, termination and pausing of conversation threads, the underlying graph interpreter activates, terminates and pauses the belonging thread supernodes.

Empirical research has stated that the change of conversation threads by the system can easily become confusing to the user (Heeman et al., 2005), especially if the system does not provide a discourse marker to notify the change.

Therefore, the described system provides "bridging utterances" to indicate a thread switch. They consist of two parts: The first one is a more general reference to the newly activated or reactivated thread (mostly through verbalizing the topic of the selected thread), the second one the repetition or rephrasing of the last utterance which was made by the system to reestablish common ground.

3 Selection of Dialogue Threads

In the **system initiative** scenario the dialogue manager has to decide which dialogue thread constitutes an appropriate continuation of the conversation, e.g. after a dialogue thread was finished and the conversation pauses. The system can choose between reactivating a paused dialogue thread or activating a new thread. Two criteria are used for the selection: time information (since when an active thread is paused) and importance information (how important is the thread for the overall conversation).

For **user initiative**, all incoming user utterances are by default at first processed in the context of the currently active thread. If the active thread fails in offering a valid transition to a new state, the dialogue manager selects a dialogue thread which fits to the incoming utterance. This can be a paused thread or a thread which has to be freshly initialized. The selection process is led by the matching values for the topic of the utterance as well as the recognized dialogue act and domain. If more than one thread with matching values for topic, domain and a valid dialogue act can be found, the selection process continues with the measures for importance and time.

4 Evaluation of Thread Selection

The dialogue manager was evaluated through conversation logs from user experiments. Each utterance in the conversation logs was manually annotated with thread function information. Thread functions include the opening of new threads, the reinitialization of paused threads and the selection of threads according to user utterances.

Unfortunately, in the evaluation experiments the users did not make use of the interleaved dialogue possibilities, but only used embedded dialogue threads. However, since the system does not differentiate between embedded or interleaved threads, we expect the system to also provide good support for interleaved conversations.

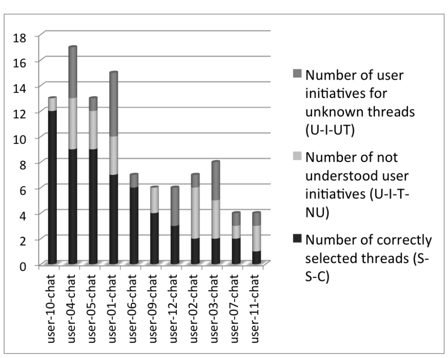

Figure 1: User initialized dialogue threads and system reaction

In general, the evaluation shows that the thread selection works very well.

Figure 1 shows the division of the system's thread selections as reaction to thread initialization by the users. There were no incorrectly selected threads by the dialogue manager, but a number of problems originating from failures of the NLU component of the system (U-I-T-NU) and some cases in which the users initialized dialogue threads unknown to the system (U-I-UT). In total 23 of 102 user's attempts to initialize new dialogue threads were not understood by the input analysis (25,48%).

Thread selection for system initiative also works very well. There were only 16 errors in 157 thread selections. Most of the errors (13 of 16) are caused by a missing behavior in the selection algorithm, which did not consider the number of already uttered rejections by the user.

The system reinitialized 63 paused threads either because of a user utterance or as system initiative. The number of reinitialized threads per conversation differs from 29 (the highest number) to 8 (the lowest number of reinitialized threads). All threads for reactivation were correctly selected.

Acknowledgments This work was supported by the project "Sprinter", funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS12006A.

References

Peter A. Heeman, Fan Yang, Andrew L. Kun, and Alexander Shyrokov. 2005. Conventions in human-human multi-threaded dialogues: a preliminary study. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 293–295, New York, NY, USA. ACM.

Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. 2002. Multi-tasking and collaborative activities in dialogue systems. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue - Volume 2*, pages 113–124, Stroudsburg, PA, USA. ACL.

Mikio Nakano, Kotaro Funakoshi, Yuji Hasegawa, and Hiroshi Tsujino. 2008. A framework for building conversational agents based on a multi-expert model. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. ACL.

Alexander Shyrokov, Andrew Kun, and Peter Heeman. 2007. Experimental modeling of human-human multi-threaded dialogues in the presence of a manual-visual task. In *Proceedings of the SIGdial 2007*, Antwerp, Belgium.

Fan Yang, Peter A. Heeman, and Andrew Kun. 2008. Switching to real-time tasks in multi-tasking dialogue. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 1025–1032, Stroudsburg, PA, USA. ACL.