Towards Dynamic Multi-Domain Dialogue Processing

Botond Pakucs

Centre for Speech Technology - CTT
Royal Institute of Technology - KTH
Stockholm, Sweden
botte@speech.kth.se

Abstract

This paper introduces SesaME, a generic dialogue management framework, especially designed for supporting dynamic multi-domain dialogue processing. SesaME supports a multitude of highly distributed applications and facilitates simultaneous adaptation to individual users and their environment. The dynamic multi-domain dialogue processing is supported through the use of standardised and highly distributed domain descriptions. For fast, runtime handling of these domain descriptions a specially developed, dynamic plug and play solution is employed. In this paper, a description of how SesaME's functionality is evaluated within the framework of the PER demonstrator is also presented.

1. Introduction

There is a growing interest for employing speech interfaces in mobile and ubiquitous computing environments. In these new environments, the users' requirements on the speech interfaces can be expected to increase. For providing intuitiveness in real life situations, it is desirable to allow the users to transparently and seamlessly switch between several services, topics and tasks within one dialogue. Accordingly, providing support for a multi-domain approach appears to be necessary.

Several spoken dialogue system frameworks support a separation of application dependent dialogue management from generic dialogue feature handling. Such systems are employed within single domains, or a static set of predefined domains. In mobile environments, however, a seamless access to locally available services and devices should be provided. Accordingly, a dynamic handling of the services and devices is required. Thus, the support for the multi-domain dialogue processing should also be dynamic.

In mobile environments, the dynamic support for a wide range of topics and tasks within one and the same dialogue is a major challenge. In dynamically changing environments the user should be able to initiate a new task while waiting for some other specific task to be completed. The user should also have the possibility to cancel a previously issued command or to change the parameters of some previously initiated service. Furthermore, the system itself should be able to interrupt an ongoing dialogue and direct the user's attention to higher priority events taking place in the user's immediate environment. Means to coordinate and control multiple concurrent appliances and services should also be provided.

To achieve a dynamic multi-domain dialogue processing, a multi-domain speech understanding is required. Thus, a dynamic handling of the speech recognition, the linguistic resources, and the distributed dialogue management capabilities has to be supported. A solution has been developed to support the dynamic adaptation of the language model to the current discourse context without grammar recompilation [1]. For integrating higher order linguistic knowledge with domain-independent speech recognition, a two-stage architecture has been proposed in [2]. For the domain of networked home devices an automated plug and play based reconfiguration of dialogue system components has been developed [3], with special focus on the management of the distributed linguistic information, the handling of distributed grammars and semantic interpretation.

In this paper the focus is on the dynamic handling of the dialogue management capabilities. The next section will elaborate on the basic concepts and the functionality of the SesaME architecture. The dynamic multi-domain dialogue management is described in Section 3. Following that, the support for flexible and adaptive interaction is briefly discussed. Finally, the evaluation of SesaME within the framework of the PER (Prototype Entrance Receptionist) demonstrator is also described.

One of the major challenges for spoken dialogue systems is to make systems easier to adapt and port to new domains [4]. The SesaME framework targets this challenge. The major contribution of this work is the dynamic plug and play handling of the distributed domain descriptions. SesaME supports the automatic, runtime extension of the system functionality with new topics and tasks without the need of manual intervention.

2. SesaME Architecture

SesaME, shown in Figure 1, is a generic, task-oriented dialogue manager which features a blackboard- and agent-based modular architecture. Thus, the SesaME framework follows the current trend of agent-based dialogue system design and is comparable with other agent-based systems such as the TRIPS [5] or the Jaspis [6] architectures.

SesaME relies on, but is not dependent of, the ATLAS generic speech technology platform [7]. The ATLAS platform provides high-level primitives for basic speech I/O, but access to low-level data is also facilitated. The ATLAS platform includes support for the ACE speech recogniser [8], which features dynamical adaptation of the language model. The communication with the ATLAS platform is done through a specially developed ATLAS API layer, which can be adopted to other speech technology platforms and components.

2.1. Interaction Manager

Generic application independent features of the dialogue management are handled by the *Interaction Manager* (IM). The most important task performed by the IM is to supervise the interaction with the user. The aim is to detect end evaluate generic

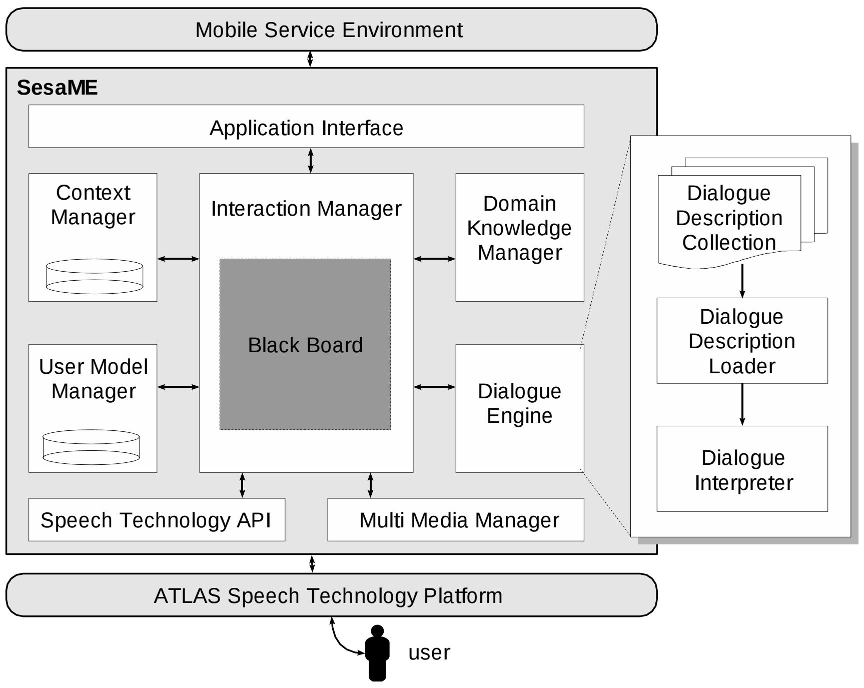

Figure 1: SesaME system architecture overview.

dialogue management phenomena such as "no new information supplied by the user during last turn", managing time-outs etc. The IM is also responsible for the meta dialogue management, such as providing information on available services and domain tasks. Performing error detection and handling is also one of the major tasks of the IM. The IM also keeps track of dialogue history, and orchestrates the various knowledge models.

Facilitation of the interaction between system components is performed by the IM through a central backboard-based mechanism. The blackboard holds the current *information state* [9] available to the dialogue system. However, this representation is not formalised; the information state is merely a collection of the information available to the dialogue system. The update of the information state is event-based, where events can be dialogue moves, internal events, or changes in the user's external context.

2.2. Program flow control

Beside the blackboard, the IM is composed of a collection of autonomous agents. These agents are taking care of one well-defined atomic task each. The communication between the agents is facilitated by the central blackboard where a *subscribe and notify* mechanism is used. All of the agents are subscribing to the required pieces of data. Whenever the subscribed data is updated the subscriber is notified. Fetching the data values, performing the suitable operations and writing the results back to the blackboard is autonomously carried out by the agent itself. Accordingly, no message passing, nor any central orchestration of the agents is necessary. Actually, no direct communication takes place between the agents and therefore no agent communication language is required.

Agents may perform some plain information refinement or they may generate a hypothesis about some specific detail in the current dialogue. An advantage of this solution is that several agents may work on the same problem in a competitive way. The competitive agents' output is considered to be hypotheses. The evaluation of the delivered hypotheses and the further propagation of the final result is carried out by specially designed *decision agents*.

Each of the agents is running in a separate thread, allowing a massive parallel processing of the available information. This parallel processing together with the event-based updating of the information state facilitates an asynchronous dialogue processing [10]. For instance, if the processing of the user's last utterance takes long time, a time-out signal triggers a specialised feed-back agent which, without knowing anything about the problem produces a system utterance for notifying the user about the system status: *"Please hold on."*.

The activity in the agents is triggered by a set of preconditions or constraints. These preconditions can be system-internal or external events such as time-outs, sensory inputs or data becoming available for further manipulation. This solution allows the system to take initiative or to direct the user's attention toward some more important events. This triggering mechanism also simplifies the synchronisation problem often encountered in agent-based dialogue systems [10]. Whenever a synchronisation problem may arise (e.g. no information is available or several values are available for the same feature) the problem has to be solved inside the individual agent.

None of the autonomous agents has access to the central representation of the problem. There is no central representation and there is no central agent responsible for planing or coordination. All agents are assuming that there exist some other agents delivering data and information necessary for further processing. However, none of the agents knows about the assumed agents or their internal states. The system is organised in layers of autonomous agents. The basic ideas behind this layered, subsumption architecture are similar to the ideas presented by Brooks [11] and employed in mobile robot systems.

3. Dynamic Multi-Domain Approach

One of the key-issues in the SesaME architecture is to support a multi-domain approach. The locally available domain descriptions, including dialogue descriptions and grammars has to be dynamically loaded and to be activated on the fly. For handling these requirements, a dynamic plug-and-play functionality of the dialogue management capabilities has been developed [12].

In SesaME, most of the operations related to the plug-and-play functionality are carried out by the *Dialogue Engine* (DE). Synchronisation and communication with the mobile service environment is taken care of by the *Application Interface* (AI). Whenever new changes occur in the service environment (e.g. new services become available or existing services disappear etc.), the AI dynamically updates the *Dialogue Description Collection* (DDC). All currently available dialogue descriptions are stored in the DDC. Beside the different task- and domain-specific dialogue descriptions, DDC also contains resident application-independent dialogue descriptions used for error handling or for meta-dialogues such as providing information on available services.

Procedurally, the internal plug and play functionality can be divided into three main parts, the *identification of the task & topic* and the associated dialogue description, the *activation* of the identified dialogue description, and the actual *dialogue management*.

At the current stage, the identification of the correct dialogue description is based on topic vectors and keywords extracted from the dialogue descriptions. However, the context and the user models or plan-based mechanisms could also be used for this purpose.

For interoperability reasons, the dialogue descriptions has to be described in some standardised way. The dialogue description formalism employed in SesaME is a slightly modified version of the VoiceXML markup language[1]. This modification was necessary for allowing additional system prompts used during adaptations (see next section), and keyword lists used for the topic detection. The use of VoiceXML-based dialogue descriptions facilitates rapid application development and effectively shields the developers from low-level implementation details.

When activating the appropriate dialogue description, a fast solution [12] is used for translating the dialogue descriptions to internal data structures. During this translation process, different VoiceXML data structures such as prompts, semantic information, links etc. are preserved and data structures appropriate for frame based dialogue management are generated.

The actual dialogue management is the main task of the *Dialogue Interpreter* (DI). The interpretation of the user's last utterance, planning and executing the system's next utterance is performed based on the previously generated internal data structures. The dialogue management process does not follow the VoiceXML specifications. Accordingly, VoiceXML is only used as a dialogue description language.

4. Content-Based Adaptation

In SesaME, a flexible dialogue management, and simultaneous adaptation to an individual user and to the user's current situation is supported [13].

After an interaction with the user, every utterance is represented as a feature vector containing feature-value pairs of all relevant information (such as topic, start time of the utterance, length of the utterance, user choices etc.). The only common property of the features in the feature vector is the co-occurrence. These feature vectors forms the basic elements of the user model. For performing the user modelling, a content-based solution [14] is employed. The user model is represented as a vector-space model and a cosine-metric matching function is used for predicting the users behaviour.

[1]For more info see: http://www.w3.org/TR/voicexml20/

The *Context Manager* keeps track of the current context. During a new interaction, based on available contextual information, similar interactions are retrieved from the user model. These retrieval results can be used to predict specific features of the ongoing interaction, and to achieve adaptation to the current context.

For example, based on earlier interactions with a voice controlled elevator it may be possible to detect that the user's most frequent choice of semantic object was the "fifth floor" when answering to the standardised prompt: *"Which floor would you prefer?"*. Thus, it is possible to predict that the user may want to take the elevator to the fifth floor. By using the additional prompt supported in SesaME, it is possible to ask the user a more natural question: *"Fifth floor, as usual?"* instead of the impersonal standardised prompt:

```
<prompt> Which floor would you prefer?  </prompt>
<alt-prompt>
<value expr=''predicted-floor''/> floor as usual?
</alt-prompt>
```

If there are no similar interactions, or no obvious patterns are present in the previous interactions (such as a CD purchasing task), then the default standardised prompt is used.

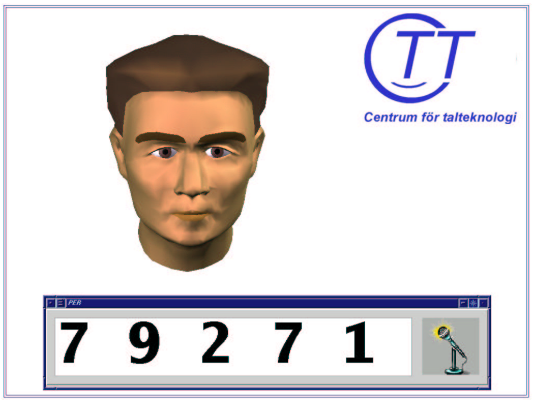

Figure 2: A screen shot of the PER interface.

5. Evaluation

Before evaluating SesaME as a generic, adaptive dialogue manager in dynamic multi-domain environments, it is necessary to evaluate it as a traditional domain-dependent dialogue manager. In the first application of SesaME, the focus is on the evaluation of the plug-and-play functionality. This evaluation is conducted within the framework of the PER project [15].

5.1. PER

PER, Figure 2, is an animated-agent based automated receptionist located at the entrance of our department. The system features a multilingual speaker verification system for Swedish and English. Originally, the application's functionality was streamlined for allowing fast and robust access for employees.

The functionality of PER has been extended to allow handling of external visitors as well. The interaction with the visitors relies on the SesaME dialogue manager and makes use of VoiceXML-based dialogue descriptions. PER features several different dialogue descriptions associated with different types of visitors and visitor goals such as expected personal guests, seminar visitors, or students attending lectures. The handling of these dialogue descriptions, the topic-based identification of the correct dialogue description, the activation of the identified dialogue description and the dia-logue management, is performed according to the description provided in the previous sections.

The domain-dependent data necessary for the dialogues is stored in an external back-end server and is made available for manipulation through a web-based interface. In this way, easy access and manipulation is provided for the employees, and the data upon which PER operates and upon which the dialogue descriptions are generated is always kept up to date.

In the framework of the PER demonstrator, SesaME has been proved to be functional. Additionally, the multi-domain approach allows the incremental updating of the system with new tasks. We plan, for instance, to add dialogue descriptions to help to guide the visitors in the building.

5.2. The Butler

Currently, the development of new application, the Butler, is conducted, for further evaluating the dynamic multi-domain support in SesaME.

The Butler will provide telephone-based, multi-domain information services around the campus area such as commuter train timetables, menu information for the nearby lunch-restaurants, information on time and location of meetings and seminars at the department. The back-end information for all of these services is based on available web-based services. The domain descriptions, necessary for the Butler and SesaME, will be generated and processed dynamically at runtime.

6. Conclusions

In this paper, SesaME, a framework for dynamic multi-domain dialogue processing was presented. SesaME features a highly flexible and modular agent-based architecture and in this way a gradual extension of the system functionality is supported.

One of the major challenges for spoken dialogue systems in mobile environments is to support the automatic runtime extension of the system functionality with new domain topics and tasks. The SesaME framework targets this challenge by supporting a dynamic multi-domain approach by employing highly distributed domain descriptions and a dynamic plug and play solution. Accordingly, a fast application development is also facilitated.

Some parts of the SesaME architecture are still under development. However, it has already been successfully employed in the framework of the PER demonstrator. This application indicates that it is feasible to support a dynamic multi-domain approach through the presented dynamic plug-and-play solution.

7. Acknowledgements

This research was carried out at the CTT, Centre for Speech Technology, a competence centre at KTH, supported by VINNOVA (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organisations. This work was also sponsored by GSLT, the Swedish Graduate School of Language Technology and by the European Union's Information Society Technologies Programme under contract IST-2000-29452, DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces).

8. References

[1] Mehryar Mohri and Fernando C. N. Pereira, "Dynamic compilation of weighted context-free grammars," in *Proceedings of ACL '98*, Montréal, Québec, Canada, 1998.

[2] Grace Chung, Stephanie Seneff, and Lee Hetherington, "Towards multi-domain speech understanding using a two-stage recognizer," in *Proceedings of Eurospeech '99*, Budapest, Hungary, Sept. 1999, pp. 2655–2658.

[3] Manny Rayner, Ian Lewin, Genevieve Gorrell, and Johan Boye, "Plug and play speech understanding," in *Proceedings of 2nd SIGdial Workshop on Discourse and Dialogue*, Aalborg, Denmark, Sept. 2001.

[4] James R. Glass, "Challenges for spoken dialogue systems," in *Proceedings of 1999 IEEE ASRU Workshop*, Keystone, CO, USA, Dec. 1999.

[5] James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent, "An architecture for a generic dialogue shell," *Natural Language Engineering*, vol. 6, no. 3-4, pp. 213–228, Dec. 2000.

[6] Markku Turunen and Jaakko Hakulinen, "Jaspis - a framework for multilingual adaptive speech applications," in *Proceedings of ICSLP 2000*, Beijing, China, 2000.

[7] Håkan Melin, "ATLAS: A generic software platform for speech technology based applications," *TMH-QPRS, Quarterly Progress and Status Report*, vol. 42, 2001.

[8] Alexander Seward, "A tree-trellis n-best decoder for stochastic context-free grammars," in *Proceedings of ICSLP 2000*, Beijing, China, Oct. 2000.

[9] Staffan Larsson and David R. Traum, "Information state and dialogue management in the TRINDI dialogue move engine toolkit," *Natural Language Engineering*, vol. 6, pp. 323–340, Sept. 2000.

[10] Nate Blaylock, James Allen, and George Ferguson, "Synchronization in an asynchronous agent-based architecture for dialogue systems," in *Proceedings of 3rd SIGdial Workshop on Discourse and Dialogue*, June 2002.

[11] Rodney A. Brooks, "Intelligence without representation," *Artificial Intelligence Journal*, vol. 47, pp. 139–159, 1991.

[12] Botond Pakucs, "VoiceXML-based dynamic plug and play dialogue management for mobile environments," in *Proceedings of ISCA T&R Workshop on Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany, June 2002.

[13] Botond Pakucs, "SesaME: A Framework for Personalised and Adaptive Speech Interfaces," in *Proceedings of EACL-03 Workshop on Dialogue Systems*, Budapest, Hungary, Apr. 2003.

[14] Ingrid Zukerman and David W. Albrecht, "Predictive statistical models for user modeling," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 5–18, 2001.

[15] Botond Pakucs and Håkan Melin, "PER: A speech based automated entrance receptionist," *Presented at the Nordic Computational Linguistic Conference, NoDaLiDa 2001*, 2001, Available at: http://www.speech.kth.se/˜botte/.