From GEMINI to DiaGen: Improving Development of Speech Dialogues for Embedded Systems

Stefan W. Hamerich

University of Hamburg
Department of Informatics
Natural Language Systems Division
Hamburg – Germany

Harman/Becker Automotive Systems
CoC Speech & Connectivity
Speech Services
Ulm – Germany

shamerich@harmanbecker.com

Abstract

In this paper DiaGen is presented, a tool that provides support in generating code for embedded dialogue applications. By aid of it, the dialogue development process is speeded up considerably. At the same time it is guaranteed that only well-formed and well-defined constructs are used. Having had its roots in the EU-funded project GEMINI, fundamental changes were necessary to adopt it to the requirements of the application environment. Additionally within this paper the basics of embedded speech dialogue systems are covered.

1 Introduction

The EU funded research project GEMINI (Generic Environment for Multilingual Interactive Natural Interfaces) aimed at the development of an Application Generation Platform (AGP) to semiautomatically generate multimodal dialogue applications for database access (Hamerich et al., 2004a). At the end of the project, two telephony applications had been successfully deployed: a banking application for a Greek bank, and a citizen care application for a German city. The former has been used by several thousand customers (Hamerich et al., 2004b).

Based on the ideas and concepts of GEMINI a new tool named DiaGen has been developed, which improves the development process for dialogue applications with regard to certain aspects.

This paper is structured as follows: First the basic ideas of the GEMINI AGP are introduced. Next the characteristics and peculiarities of embedded speech applications are explained. This is followed by a description of the concepts of GEMINI which had been a starting point for the development of DiaGen. The core of this paper follows: a detailled description of the DiaGen tool. Finally the conclusion and outlook are presented.

2 The GEMINI AGP

The GEMINI AGP provided support for the semi-automatic creation of phone-based dialogue applications. The development process had several layers. Through the different phases of a layer the application developer was guided by a wizard and had to use specialised assistants for each phase.

The first starting point was a rough abstract dialogue model, which has been enriched step by step through all phases until finally dialogue model was completed. All models are completely written in a language specifically developed for the purposes of GEMINI covering both, dialogue description and data modelling (Hamerich et al., 2003; Schubert and Hamerich, 2005).

Originally the GEMINI AGP was designed for phone-based or web-based applications. Therefore the final outcome of the AGP was VoiceXML or xHTML, according to the initial selection of the application developer.

The three layers of the platform are described in depth in (d'Haro et al., 2006).

3 Automotive Speech Dialogues

Speech dialogues for cars are embedded solutions running under real-time operating systems with very

Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, pages 92–95,
Columbus, June 2008.

low memory and CPU power (Hamerich, 2005).[1] Next to these hardware requirements customers from automotive industry demand very explicit specifications to understand the complete dialogue flow and see its connections to the graphical/haptical HMI (human machine interface) in a car. Therefore special algorithms and tools are used, to develop and run speech dialogues on such embedded systems. In consequence Harman/Becker has a proprietary dialogue description language developed especially for being used on embedded environments (Hamerich and Hanrieder, 2004). The Generic Dialogue Modelling Language (GDML) is designed as a compiled language to save memory and CPU resources. This makes sense, since dialogues within a car are still closed applications.

Speech control for cars is available to the end customer since 1996 (Heisterkamp, 2001). Today many car manufacturers offer speech control systems. Typical applications in a car are voice control of telephone, tuner and navigation system. Direct control of media files using their meta-data (e.g. ID3-Tags) by saying e.g. "play title 'Bad' by 'Michael Jackson'" is a feature currently under development (Wang and Hamerich, 2008).

In spite of several tools and libraries, dialogue development for automotive applications is mainly still manual work.

4 Porting Ideas from GEMINI to DiaGen

Since the GEMINI AGP showed that advanced speech dialogue applications can be created fast and easy it was straightforward to attempt to transfer at least some of the possibilities from the AGP into the world of embedded speech dialogues. However the following features need to be changed for the new tool:

- Speech dialogues in cars do not access a database; instead the devices are controlled directly by the speech dialogue. Therefore DiaGen does not need a database interface but should instead offer a flexible way to integrate access to external devices.

- When starting development with the AGP first a rough dialogue specification has to be provided, which for every new application needs to be given again (except the library approach is used, which makes only sense for very similar applications). It would make sense to provide a sample dialogue at the start of dialogue development for embedded applications, containing the most common interfaces and allowing faster creation of new applications from this starting point.

- When using the AGP for dialogue development, there was no consistency check for speech grammars and their connection to the dialogue. This should be improved with DiaGen.

- Since highly customised applications are demanded, code is still written by hand. Nevertheless dialogue designers are supported with several tools and libraries. Therefore the new tool should fit into the existing tool chain, but should also allow for manual editing or at least fine-tuning of the code. Since it was experienced from GEMINI that generating VoiceXML from the models coded in the GEMINI modelling language was hard work, it was decided to directly work on the runtime language for the new tool. This minimises efforts for the generation components and on the other hand allows for easy editing of code files. That means for the new tool no generator component is needed. Instead the compiler needed for the embedded dialogue descriptions should be added to DiaGen, to allow for integrated development.

- Since the creation of a phone-based dialogue system requires specialised handling for different situations (e.g. for database access, output generation, etc.) several specialised wizards have been created forming the AGP. Since development for a speech control system is quite different it does not make sense, to have several assistants. Therefore DiaGen integrates all the needed functionality into one tool.

[1] Generally embedded systems comprise other highly integrated systems as well. Since the approach for embedding speech dialogue systems described here can work on such systems as well, the term 'embedded' is used as a generalisation.

5 DiaGen

As already described above, DiaGen was developed as a new tool, based on the experiences made within the GEMINI project. The key idea of DiaGen is to ease development of speech dialogues for automotive applications. The main point here is not only to speed up coding of dialogue scripts but additionally to support the development of correct, consistent, and user-friendly dialogue applications.

The main differences between DiaGen and the GEMINI AGP are already described above. In this section the most outstanding properties of the final tool are discussed in detail.

5.1 Modelling Language

Since effort for generating runtime code from development models was a big issuee within GEMINI and it is often required to change code details even in a late phase of development, it was decided for DiaGen to work directly on GDML. This allows DiaGen to offer manual editing at any development stage.

5.2 Integration

For a GDML developer, there are daily tools to work with. These are the grammar and dialogue compiler and a testing and debugging tool. These tools all have been integrated into DiaGen. For each tool, DiaGen allows to set configuration parameters as well as to compile and debug directly in the environment.

5.3 Project Model

One of the main features of DiaGen is a complete project model, which contains all project files and runtime configuration settings. Loading this model into DiaGen allows easy compiling, testing and editing of the complete application.

The model can be extended by editing the contained files using DiaGen. Additionally DiaGen also offers the possibility to add predefined routines or methods to the model, allowing for a library usage.

Another advantage of the model is the complete coverage of variables, functions, prompts, etc. This speeds up the development process quite a lot, since the tool automatically proposes allowed argument values for a function call. And if a variable has not been defined in the current context, this can just be done by a simple click on the respective button. This feature was already available in parts with the GEMINI AGP.

5.4 Sample Application

As already mentioned in section 4 development for a new application with DiaGen starts with a sample application. This saves time since setting up a new running application with correct configuration settings by hand can be a lengthy process. If instead a complete running system is copied and stripped down, this costs time as well. Starting with a small sample application therefore is much more efficient.

The sample application can easily be updated and maintained, therefore even new configuration settings or techniques can be adopted.

5.5 Device Interface

To control devices by speech, their interface must be accessible for the dialogue. This in GDML generally is done with the concept of system calls for details see (Hamerich and Hanrieder, 2004). New system calls can be created using DiaGen or just be added to an existing DiaGen project. When a system call is needed, it can just be selected from a list, saving time for lookup. Of course all the advantages of the project model (sec. 5.3) apply for system calls and their arguments and results as well.

5.6 Grammar Tag Consistency

GDML (like VoiceXML) uses semantic grammar tags to identify user utterances. These tags are even independent of the used language making GDML dialogues complete language independent. This gives bigger flexibility and minimises efforts for porting a dialogue application to another language.

To initiate a dialogue reaction, a specified tag has to be delivered from the parser. For each tag a dialogue action inside the dialogue code itself is needed. In this case consistency of these tags in grammar and dialogue script is of highest importance. As already mentioned the GEMINI AGP did not ensure this consistency automatically. This led to high efforts when developing an application with the AGP. To minimise these efforts and disable potential errors the consistency shall be ensured automatically by DiaGen.

To do so DiaGen offers a special view of the grammar. For each grammar rule or combination of rules all possible grammar tags are shown. Selecting a tag automatically constructs a complete switch-case statement for all possible alternatives and ensures consistency between grammar and dialogue.

5.7 Usage of DiaGen

DiaGen has been developed to allow fast creation of flexible speech dialogues for automotive applications. See Figure 1 for possibilities of its context menu. It was used successfully for a proactive dynamic traffic information application based on Traffic Message Channel (TMC) messages. This application has already been described in (Hamerich, 2007). Since the tool is still in its testing phase, it is currently used for prototypical development only.

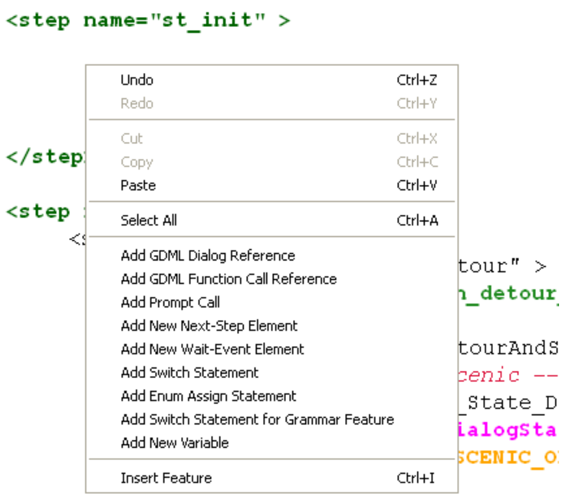

Figure 1: *Context menu of DiaGen within GDML dialog step.*

6 Conclusion

In this paper DiaGen was presented. A tool to improve the development process of embedded speech dialogues as used for automotive systems. Major improvements offered by usage of DiaGen are speed-up of coding and verified code consistency. DiaGen results partly from the experiences collected within the GEMINI project. But since GEMINI concentrated on phone-based and multimodal applications, several changes have been necessary for embedded dialogues, which have been described.

7 Future Work

As pointed out the tool is currently used to develop a pilot application. As feedback from the work on the pilot application, DiaGen is constantly being updated. At a later development stage of DiaGen it will be evaluated to be used for product development as well.

References

L.F. d'Haro, R. de Córdoba, J. Ferreiros, S.W. Hamerich, V. Schless, B. Kladis, V. Schubert, O. Kocsis, S. Igel, and J.M. Pardo. 2006. An Advanced Platform to Speed up the Design of Multilingual Dialog Applications for Multiple Modalities. *Speech Communication*, 48(6):863–887.

S.W. Hamerich and G. Hanrieder. 2004. Modelling Generic Dialog Applications for Embedded Systems. In *Proc. ICSLP*, pages 237–240, Jeju, Korea.

S.W. Hamerich, Y.-F. H. Wang, V. Schubert, V. Schless, and S. Igel. 2003. XML-Based Dialogue Descriptions in the GEMINI Project. In *Proc. 'Berliner XML-Tage 2003'*, pages 404–412, Berlin, Germany.

S.W. Hamerich, R. de Córdoba, V. Schless, L.F. d'Haro, B. Kladis, V. Schubert, O. Kocsis, S. Igel, and J.M. Pardo. 2004a. The GEMINI Platform: Semi-Automatic Generation of Dialogue Applications. In *Proc. ICSLP*, pages 2629–2632, Jeju, Korea.

S.W. Hamerich, V. Schubert, V. Schless, R. de Córdoba, J. M. Pardo, L. F. d'Haro, B. Kladis, O. Kocsis, and S. Igel. 2004b. Semi-Automatic Generation of Dialogue Applications in the GEMINI Project. In *Proc. SIGdial*, pages 31–34, Cambridge, USA.

S.W. Hamerich. 2005. Speech Dialogue Systems for Cars - an Overview. *SDV – Sprache und Datenverarbeitung*, 29(2):107–118.

S.W. Hamerich. 2007. Towards Advanced Speech Driven Navigation Systems for Cars. In *Proc. IE*, pages 247–250, Ulm, Germany.

P. Heisterkamp. 2001. Linguatronic – Product-Level Speech System for Mercedes-Benz Cars. In *Proc. HLT*, pages 1–2, San Diego, USA.

V. Schubert and S.W. Hamerich. 2005. The Dialog Application Metalanguage GDialogXML. In *Proc. EUROSPEECH*, pages 789–792, Lisbon, Portugal.

Y.-F.H. Wang and S.W. Hamerich, 2008. *Dybkjær, L. and Minker, W. (Ed.): Recent Trends in Discourse and Dialogue*, chapter Designing Speech-Controlled Media File Selection for Automotive Systems, pages 25–43. Springer, Dordrecht, Netherlands.