SPICE

A Multimodal Conversational User Interface to an Electronic Program Guide

Andreas Kellner, Thomas Portele
Philips Research Laboratories Aachen

Abstract: This paper describes a conversational user interface to an Electronic Program Guide (EPG). It tries to mediate between the intricate functionality of devices on the one side and the user's abstract intention on the other side. In contrast to simple keyword-based voice command interfaces, conversational user interfaces allow the user to interact with services and devices without specific knowledge of keywords or menu structures by means of natural language input and co-operative dialogue. The SPICE prototype system described here combines a spoken dialogue interface with a touch-screen display to allow for natural and efficient multimodal interaction in various usage scenarios.
This paper explains the underlying concepts of the SPICE multimodal conversational user interface and their realization in a prototype application

Keywords: Spoken Dialogue, Multimodal Interface, Conversational System

1. 1. INTRODUCTION

Spoken dialogue systems for well-defined goal-oriented tasks have been used in commercial applications over the telephone for a few years now [Aust et al. 1995, Kellner et al 1997, Walker et al. 2001]. In most cases, these systems have been specifically designed for a single application. Often, these applications follow the same simple interaction scheme: In the '*information gathering'* phase, the user interactively provides values for a number of pre-defined application slots (e.g. '*origin'*, '*destination'*, '*date'*, and '*time'* for a timetable information system). After all necessary slots are filled and confirmed by the user, the system accesses a structured database

with these search criteria and retrieves matching entries. In the successive '*information presentation*' phase, the user can navigate through the results.

Dialogue-based user interfaces for mobile and living-room environments however, have to provide a more flexible interaction mechanism that is able to:

- support information browsing in addition to goal-oriented slot filling,
- combine multiple applications in a plug-and-play fashion,
- offer a choice of modalities for input and output and allow for separate as well as joint use of those modalities,
- allow real conversational behaviour, and
- deal with unstructured content.

In order to explore possible solutions to these challenges, the SPICE (Speech Interfaces for Consumer Electronics) system was developed at Philips Research. SPICE is a prototype of a conversational user interface to an Electronic Program Guide that supports the navigation in a large TV program database. The selected programs can be used for programming a VCR and controlling a TV set. The user can interact with the system by means of spoken dialogue in combination with touch-screen input on a hand-held graphics display.

The remainder of this paper is organized as follows: In chapter 2, we present the most important features that constitute the conversational user interface of the SPICE system. In chapter 3, we describe the application set-up. Finally, in chapter 4, the underlying system architecture and the technical realization of its components is explained.

2. FEATURES OF THE MULTIMODAL CONVERSATIONAL USER INTERFACE

In the living room of the future, different interconnected devices will provide a variety of complex, interrelated digital services such as TV-on-demand, navigation in information spaces, or audio-visual communication. New user interface paradigms are therefore necessary to mediate between intricate functionalities and their users. In the SPICE prototype system, natural language interaction in combination with a touch screen display allows for powerful, flexible and easy to learn interaction that turns the interface into a communication partner that co-operates with the user to fulfil various tasks. The main features of this interface are:

2.1 Natural language input

The user is able to express her wishes and intentions in her own words without worrying about the correct translation onto the corresponding device commands to accomplish a certain task. Consequently, the user does not have to remember specific pre-defined keywords, but uses natural language to express herself. In contrast to the pre-defined menu structure of today's devices, it is also up to the user to decide what information to give and in which order. It is also possible to give multiple commands and/or information items in a single utterance (e.g. 'What entertainment shows are on BBC1 tonight?').

2.2 Direct access to content

In many applications in the consumer electronics domain, the user often wants to navigate through large content spaces such as songs in a CD collection or movies in a TV program guide. Due to the limited 'interaction bandwidth' of conventional interfaces such as remote controls or small-vocabulary voice-control systems, the individual entries (e.g. titles) can today only be selected indirectly by specifying filters like 'genre', 'artist', or 'time' or by navigation up and down a long list. The conversational speech interface of SPICE, in contrast, allows direct access to the information items the user actually wants.

An additional problem occurs, because in many applications that deal with large content, 'fuzzy' input (e.g. '*second James Bond'* instead of '*James Bond 007: From Russia with Love'*) is difficult to handle because possible alternative formulations have to be modelled explicitly in the grammar or database.

By using large-vocabulary speech recognition and natural language understanding capabilities in combination with info-retrieval technology, it is possible to navigate in unstructured information sources such as the program descriptions (e.g. 'Is there any movie about wildlife in Africa this week?').

2.3 Cooperative dialogue

In conversational user interfaces, the interaction between the user and the system becomes a two-way communication [Smith&Gordon 1997]. While normally the user is in full control of the dialogue, the device can also take the initiative and guide the user through a complex task (e.g. programming a VCR), offer some suggestions for content-selection based on the user's preferences, or provide assistance (e.g. a help function) when problems

occur. If disambiguation is necessary SPICE proposes that the user specifies the information item that has maximal discriminative power for the possible alternatives. (e.g. ‘On which day do you want to record the show?’).

2.4 Choice and combination of modalities

The full potential of man-machine interaction can only be obtained in multi-modal systems because some of the modalities may not be appropriate in certain usage contexts (e.g. hands-free usage or loud background noise). Studies report that interfaces allowing multimodal input like speech and pointing are preferred by users since users have a good intuition about when to use different modes and when to switch modality which leads to a better performance, faster error recovery and less frustration [Cohen et al. 1998, Oviatt 1999].

In SPICE, pointing input can be used separately or combined with speech input for efficiently selecting information from lists currently displayed on the screen (e.g. “Give me more information on THIS ONE”, “What else do you have on THAT channel?”). Furthermore, the combination of the two modalities improves the robustness of the recognition due to redundancy in the two input modes [Oviatt 1999].

3. THE SPICE EPG PROTOTYPE

An Electronic Program Guide (EPG) allows the user to navigate through a TV program database and to retrieve background information for the shows and/or schedule them for recording or later viewing. With the increasing number of TV stations, EPG systems become more and more popular and are already integrated in many TV sets or set-top boxes. Today, navigation in an EPG is in most cases performed with a remote control by following pre-defined menu trees.

The SPICE system demonstrates a multimodal conversational user interface to an electronic program guide. The system allows the user to navigate in a database of three weeks of TV program data covering 10 channels and roughly 7000 titles. The user can find entries from this database by various search criteria (‘*date’*, ‘*time’*, ‘*genre’*, ‘*title’*) or by searching the unstructured program description texts. For the selected items the user can obtain background information or schedule a title for viewing or recording.

The program database was taken from an Internet source. The entries were automatically processed (i.e. ordered into title, actor, description) and converted into phonemic representations for the recognizer lexicon. The

system's knowledge sources such as lexica and language models can be updated automatically for the dynamic program data.

The user interacts with the system by natural language input, by pointing on the touch-screen, or by a combination of both. The system displays the requested information on the screen of a small hand-held device (see Figure 1).

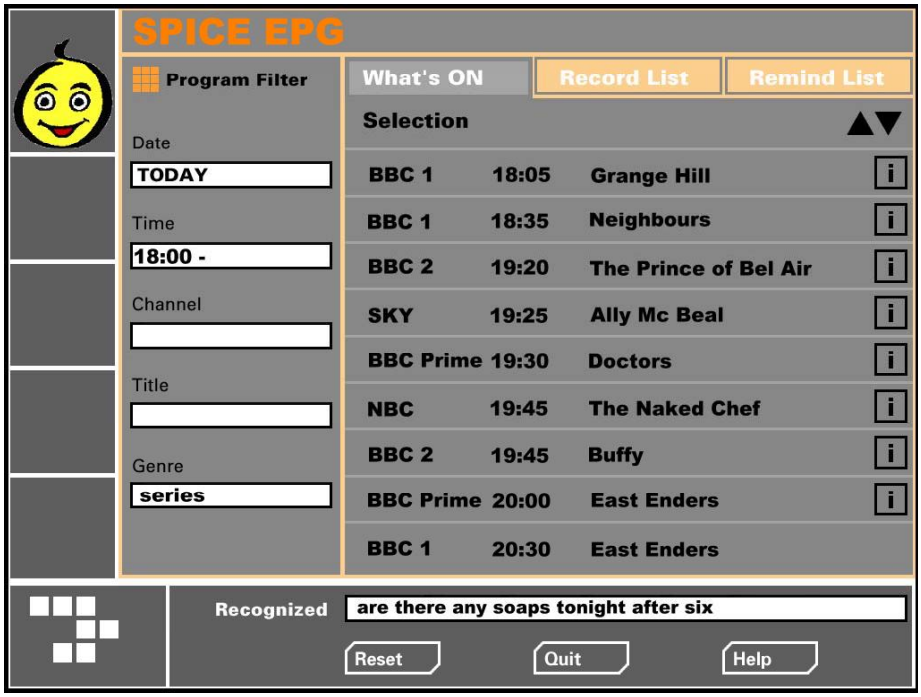

Figure 1. SPICE display.

In addition to the information output and feedback on the system's recognition result, the screen shows a small picture of an animated face that represents the system state. In this way, the user is always informed about the status of the interaction. Furthermore, the visualization of the system's personality as a communication partner makes the spoken interaction with an otherwise 'dead' device more natural.

4. SYSTEM DESCRIPTION

Figure 2 shows the overall architecture of the SPICE system. The current research prototype consists of several modules most of which are described in more detail in [2].

4.1 Speech Recognition

The Philips large vocabulary continuous speech recognizer analyses the incoming speech signal and recognizes the word sequence spoken by the user. In contrast to simple keyword interfaces, conversational user interfaces require a large vocabulary that can be updated dynamically to allow for the recognition of words from a non-stationary database.

The recognition vocabulary consists of more than 14.000 words, 95% of which are extracted from the program titles or descriptions.

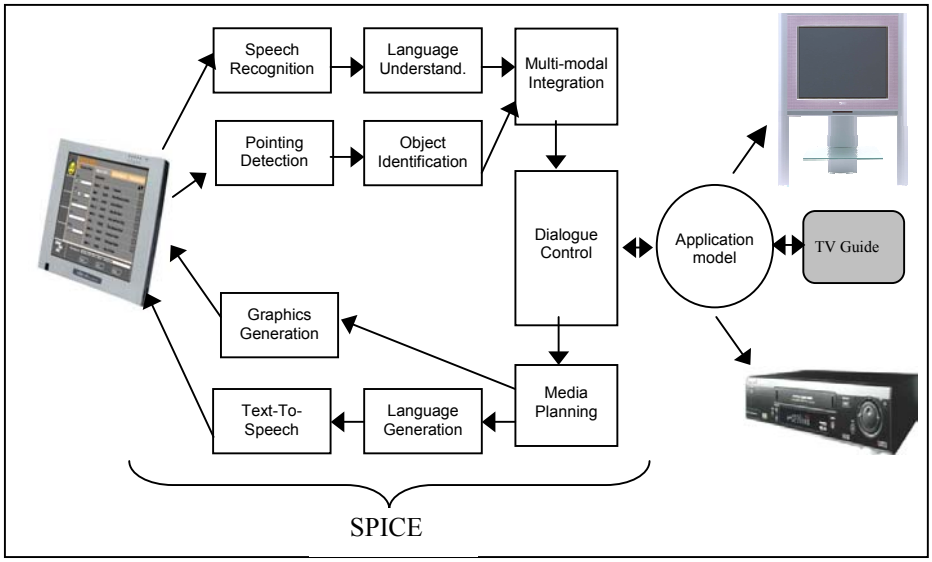

Figure 2. Overview of the SPICE architecture.

The recognizer is a state-of-the-art HMM recognizer (16000 densities) with a MFCC feature extraction and a bigram language model. It was trained with 80 hours of data collected for standard dictation tasks. The recognizer generates a word lattice as a compact representation of different alternative word sequences.

For the SPICE system, supervised adaptation was performed with in-domain material (about 15 minutes) for 5 male non-native speakers. On this set of speakers, the adaptation reduced the error rate by 50 % relative.

4.2 Natural Language Understanding

This module analyses the spoken input and extracts all the semantic information that is relevant in the given application (so called concepts). In addition to the standard parsing, it identifies descriptive phrases like incomplete titles or description of content in the input and retrieves matching candidates from the (movie-) database.

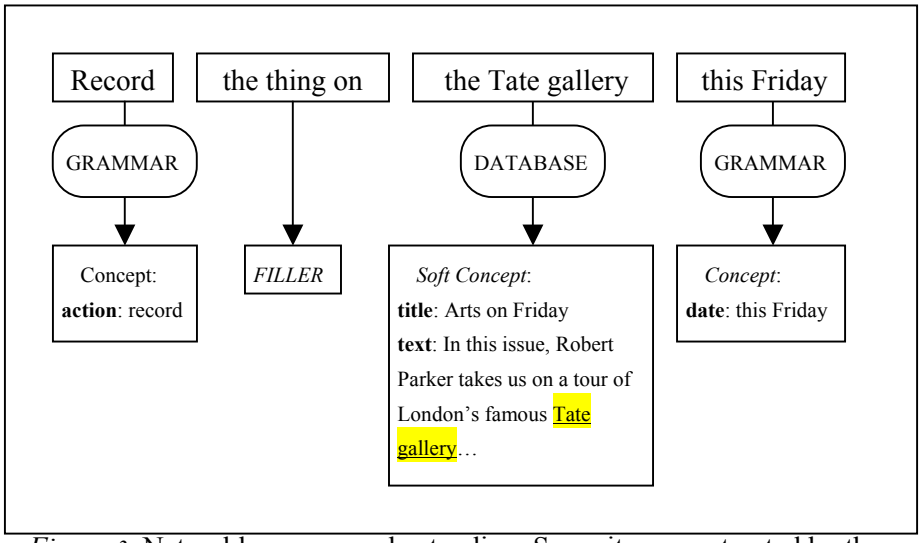

Figure 3. Natural language understanding: Some items are treated by the grammar, others are ignored (mapped to FILLER), while fuzzy expressions are mapped onto database entries with info retrieval methods.

A stochastic context-free grammar is applied to the word lattice delivered from the recognizer. The rules specified in the grammar do not have to cover the complete user utterance. They represent meaningful phrases in the utterance, so called concepts. The concepts are extracted from the user utterance by a top-down chart parser that allows for island parsing. In addition to the concepts, there is an alternative model for meaningless phrases, a so-called filler-model. A more detailed explanation of stochastic context-free grammars and filler models can be found in [Aust et al. 1995, Souvignier et al. 2000]

The grammar includes references to the EPG program database and therefore allows the integration of dynamic data. While this provides for a representation of information that is presented in a structured way (so called *concepts*), this approach cannot be used for phrases that describe the content of a programme item or specify information items (like titles) in a fuzzy or incomplete way. The standard grammar formalism was therefore extended by 'soft concepts'.

The main idea is to proceed in a two-step approach by first identifying phrases carrying a specific type of information without explicitly consulting the database and computing their semantic information in a second step.

The first step is realised by applying competing stochastic language models.

If a phrase is scored with different information specific language models it will get high scores if it represents the respective type of information and low scores if it belongs to a totally different topic. As one competitor, a general language model is used and a phrase is only accepted as information bearing if its score with respect to one of the special language models is considerably higher than that with respect to the general model.

In a second step, the semantic contents attached to the 'soft concepts' are computed. Information retrieval methods are applied to find the database entries that best match the information bearing phrases. Again, information specific retrieval indices are used which are constructed from the titles and descriptions in the database. This means that a phrase identified as describing a title is only matched against true titles occurring in the database and one can thus view the info retrieval as translation process from fuzzy titles to proper titles.

The semantic content is finally simply found as one or more field values of the database entries corresponding to the retrieved titles or descriptions.

The overall mechanism is illustrated by an example in Figure 3.

4.3 Multimodal Integration

In parallel to the speech input, the user can interact with the system by tapping on a touch-screen display. The multimodal integration combines the semantics represented in the speech input and in the pointing input into a coherent semantic representation of the user input. In the SPICE prototype, pointing input can be used to select information that is currently displayed on the screen. This can be a single program item or part of the information of this item (like "What else do you have on THAT channel?"). The system's internal processing of the pointing input is done in two steps:

The graphics display manager detects a pointing event (i.e. the user tapped onto something on the screen). From this event, the object identification module creates a number of alternative semantic representations. Each of these refers to a set of information items representing the desired user input. In addition, a timestamp is created that allows to coordinate the pointing event with the speech signal.

The two input modalities can occur simultaneously (e.g. pointing and speaking at the same time) or sequential (e.g. first speaking then pointing). In order to combine and understand these inputs, the time integration pattern

and the chronology of the input events are important. The implementation of the integration algorithm thus requires exact time-stamps for the input events, a time-sensitive grouping of events according to temporal constraints (for example a pointing event is only related to language input which lies within a time-lag of less than 4 seconds), and the determination of co-reference between deictic words - reference words like “this”, “that”, or “these” - and pointed selections.

To give an example, let the user ask the SPICE system “What movie is on this channel?” while pointing to an item, which contains the time, channel, title, and category of a program. The recognizer produces two interpretations: “What movie is once channel” as the best hypothesis, with “once channel” mapped to the FILLER concept, since it is not interpretable; and “What movie is on this channel” as second best hypothesis with “this channel” mapped to a DEICTIC concept that has to point to a channel item. The first hypothesis cannot be unified with the pointed item; therefore the second best, for which “this channel” will be unified with ‘BBC1’, will be rescored and the best joint hypothesis becomes “What movie is on THIS CHANNEL[channel = ‘BBC1’].”

4.4 Dialogue Management

The dialogue manager is the central module of the system. It maintains the system's internal knowledge stack, interacts with the actual applications (e.g. TV, VCR, or EPG-database), and decides about the next action of the system. The browsing mode of interaction demands the integration of pertinent navigation and relaxation techniques.

The dialogue manager is responsible for maintaining the system's internal knowledge (belief). This means that in each turn, it has to combine the information contained in the new utterance with the system belief of the previous utterance. In the current system, the belief is stored in a number of semantic slots.

The dialogue manager has to handle phenomena like

- **Corrections**: The user overwrites a value of a slot with a new one.
- **Verifications**: The user (implicitly or explicitly) verifies the value of a slot.
- **Combinations**: Two different values for a slot can be combined into one (e.g. "3:00" + "afternoon" = "3 pm").
- **Contradictions**: Contradictions between different slots (e.g. day = 30, month = February) have to be detected. In such a case, the system has to trigger a disambiguation question.

- **Disambiguation**: The user's intention cannot uniquely be identified from her input (e.g. because of contradictory or missing information). In this case, the system has to take the initiative and prompt the user for the correct interpretation.
- **Changing User Intention**: In a browsing application with changing user goals, the system has to decide which information items provided in previous user turns are still valid and which should be discarded.
- **Relaxation**: The user can find no matches in the database for the search criteria specified. The system has to decide which of the criteria can be relaxed or completely removed so that it can return the most relevant information to the user.

One big issue in dialogue management is how to describe a spoken language dialogue. In simple applications, this can be done by means of a finite-state-network. For mixed-initiative interactions, where both, the system and the user control the dialogue flow, this is not possible any longer. The number of different dialogue states would simply become too large to handle.

For the Philips spoken language dialogue systems, a special High-Level Dialogue Description Language (HDDL) was developed [Aust et al. 1995]. This allows the specification of the dialogue for a complex application in an abstract way by describing the task (in this case the slots that have to be filled by the system) and the basic high-level dialogue flow. This language has been used successfully for various slot-filling dialogues like timetable information or directory assistance [Aust et al. 1995, Kellner et al. 1997]. It may, however not be perfectly suited to describe interactions as can be expected for applications like SPICE. One of the big differences between the automatic inquiry systems for which HDDL was developed and the conversational dialogues in SPICE is that the interactions in automatic inquiry systems are very goal oriented. The user has a specific goal in mind (e.g. a journey she wants to make) and knows all the information required for that goal (e.g. date, time, departure city, arrival city). All the system has to do is to collect the values for the required information slots, access the database and present the information returned by the database. In SPICE, the situation is different. In most cases, the user does not have a specific goal in mind. Instead, she wants to browse through the EPG database (rather than searching for a specific item) and changes her goal very frequently without explicitly telling the system. Although the SPICE dialogue is formulated in HDDL, possible extensions and other paradigms are currently under investigation.

4.5 Media Planning, Language Generation and Spoken Output Generation

Once the system has computed the current status of the application and the required next steps, it has to decide, which information it wants to present to the user.

The SPICE media planning strategy is to supply content information and system state feedback via the visual channel (see below), and to use the acoustic channel only for clarification dialogues.

The template-based language generation [Portele 2000] draws from a set of output templates with optional variable parts where the current value of system slot variables can be filled in. The generator has to concatenate several of these templates and replace the variables (like e.g. CHANNEL) by the correct values.

The current SPICE system uses pre-recorded phrases for spoken output.

4.6 Graphics Output

In addition to spoken feedback, the system displays information on the hand-held screen. In the SPICE system, the graphical user interface (see figure 1) consists of a general feedback area and an application area.

The application area displays either a selection of program items matching the user's search criteria, a screen with background information to a specific program, or a list of all programs that are scheduled for recording or reminding.

The feedback area helps the user to understand the system's reactions and it's current state of activity. It shows the user input as understood by the system and a small picture of a cartoon character that represents the system state. Different facial expressions of the 'Smiley' icon depict the current processing phase of the system (e.g. 'idle', 'waiting-for-activation', 'listening', 'recognizing', 'checking database'). In addition, the Smiley supports the interaction with the user by showing specific expressions in case of misunderstandings, questions, and suggestions by the system.

The results of a user study that was recently carried out with the SPICE system in our lab show, that the animated character is liked very much by the users because it helps to understand what the system is doing at any point in the interaction.

5. CONCLUSION

Conversational user interfaces offer a number of challenges on top of traditional telephone-based inquiry systems. The SPICE system, which offers a multimodal conversation user interface to an electronic program guide, was built to investigate these issues and develop the technology necessary for real-world applications.

This paper describes the setup of the SPICE prototype and the underlying speech & language technology, focusing on the most important aspects of multimodal conversational user interfaces as the next generation of spoken dialogue systems.

6. REFERENCES

Aust, H., Oerder, M., Seide, F. and Steinbiss, V.: The Philips automatic train timetable information system. Speech Communication, 17(3-4):249-262, 1995

Souvignier, B., Kellner, A., Rueber, B., Schramm, H. and Seide, F..: The thoughtful elephant: Strategies for spoken dialog systems. IEEE Transactions on Speech and Audio Processing, 8(1):51-62, 2000

Cohen, P., Johnston, M., McGee, D., Oviatt, S., Clow, J. and Smith, I.: The efficiency of multimodal interaction. Proceedings of the International Conference on Spoken Language Processing, pages 249-252, 1998

Kellner, A., Rueber, B., Seide, F. and Tran, B.: PADIS - An Automatic Telephone Switchboard and Directory Information System. Speech Communication, 23(1):95-111, 1997

Oviatt, S.: Mutual disambiguation of recognition errors in a multimodal architecture. Proceedings of CHI 99 Conference on Human Factors in Computing Systems, pages 576-583, 1999

Portele, T.: Natural language generation for spoken dialogue. Proceedings of the International Conference on Spoken Language Processing, III, pages 310-313 , 2000

Smith, R. and Gordon, S.: Effects of variable initiative on linguistic behaviour in human-computer spoken natural language dialogue. Computational Linguistics, 23:141-168, 1997.

Walker, M., Passonneau, R. and Boland, J.: Quantitative and Qualitative Evaluation of DARPA Communicator Spoken Dialogue Systems. Proceedings of 39th Annual Meeting of the Association for Computational Linguistics, 515-522, 2001