Lessons from the Development of a Conversational Interface

Marianne Hickey, Paul St John Brittan
Internet and Mobile Systems Laboratory
HP Laboratories Bristol
HPL-2001-77
September 14th , 2001*

E-mail: {marianne_hickey, paul_brittan}@hpl.hp.com

speech, dialogue, wizard of oz, MIT GALAXY

The design of an effective mixed initiative dialogue system still presents great challenges. This paper reports on the experiences gained in the design and implementation of an experimental spoken dialogue system, MIZIK, which revolves around a new domain, the music charts. It describes the processes we went through to: determine the development approach for a robust system; specify the scope of the domain; select an appropriate architecture and speech and language technology; collect training data specific to the domain and the target user population and, finally, to develop the experimental system. The paper concludes with a number of key lessons learnt during these processes, many of which are equally applicable to the design and development of any conversational speech interface.

* Internal Accession Date Only
Approved for External Publication
Eurospeech 2001, Aalborg, Denmark. Sept 3-7 2001.

Lessons from the Development of a Conversational Interface

Marianne Hickey and Paul St John Brittan

HP Laboratories
Filton Road, Stoke Gifford, Bristol, BS34 8QZ, UK.
{marianne_hickey, paul_brittan}@hpl.hp.com

Abstract

The design of an effective mixed initiative dialogue system still presents great challenges. This paper reports on the experiences gained in the design and implementation of an experimental spoken dialogue system, MIZIK, which revolves around a new domain, the music charts. It describes the processes we went through to: determine the development approach for a robust system; specify the scope of the domain; select an appropriate architecture and speech and language technology; collect training data specific to the domain and the target user population and, finally, to develop the experimental system. The paper concludes with a number of key lessons learnt during these processes, many of which are equally applicable to the design and development of any conversational speech interface.

1. Introduction

We are interested in what it will take to enable wide-scale deployment and use of speech and language services. If spoken dialogue systems are to become widely used they must be robust. That is, they must work reliably and effectively under the range of demands that real-life situations and users will make. There are different ways of viewing and achieving robustness. One option is to put the system in control, breaking the dialogue into lots of small steps and at each step only allowing the user to speak from a very limited grammar. Another way of looking at robustness is to design the spoken dialogue system such that it can handle the variety of ways in which people will want to interact with it in real situations. That is, design a mixed initiative system. With this approach, the user or the system can take control in the dialogue, changing the course of the interaction. Secondly, the user should be able to phrase utterances in as unconstrained a way as possible – a more subtle form of initiative. To make the development of mixed initiative systems tangible, the dialogue needs to be constrained to a well-specified domain. That is, with an ideal system, the human can say anything within the current domain of the dialogue, whenever s/he chooses and the system should respond in a helpful way. In addition, there needs to be a way of moving gracefully between domains.

Our overall aim is to find out what it takes to implement mixed initiative systems for real: for new domains; with developers who are not highly specialized in speech and language technology; where there are many users, and where there are many domains and a user needs to navigate between them. In this paper, we concentrate on the development process of a spoken dialogue system in one new domain. The aim here is not to develop the speech and language processing engines. Rather, we are using existing engines to learn about the design and development processes for a new domain and to produce a set of guidelines. This paper reports on our experiences in developing an experimental spoken dialogue system, MIZIK, based around a music domain. It highlights the lessons learnt during this process, which apply to any domain.

2. Approaches to developing a robust system

To produce a spoken dialogue system that is robust to the way people are likely to use it, development should be based on spontaneous data that is representative of a realistic dialogue between a naïve user and the system. Such data can be used to develop and train the speech recognizer, language understanding, dialogue management and language generation. We looked at different ways of developing our system using realistic data. Domain-specific data is needed for much of the system development and, as this is a new domain, the process necessarily involved collecting the data.

- One way is to develop an end-to-end bootstrap spoken dialogue system, initially based on textual data collected from a small, expert user group. The next step is to get people to use it, log the data and use it to develop and train the system. The process continues with iterative cycles of data collection and system development. With each cycle, as the system becomes more reliable, the user group is expanded and includes more naïve subjects.

- Another method is to use a Wizard of Oz paradigm [1] to collect the data from naïve subjects. Consider a system accessible via a telephone. A subject calls what s/he believes to be an automatic spoken dialogue system. At the other end of the line, a person acting as the Wizard interprets the subject's spoken request and enters, for example by typing, a representative query into a Wizard System. The Wizard System then processes the query and gives a spoken response to the subject. The interaction is logged and the data collected used to develop and train an end-to-end spoken dialogue system.

The approach we took was to start with a Wizard of Oz exercise, primarily because it was an efficient way to collect consistent training and test data. This in itself is not the only step that is necessary to develop a robust system. Once a spoken dialogue system is implemented using the data collected in a Wizard-of-Oz exercise, it needs to be developed further, using the iterative approach of collecting data from people using it and training the system with that data.

3. System components and architecture

To design and experiment with a mixed initiative dialogue system in the music domain, three platforms were required.

- MIZIK: the end-to-end spoken dialogue system.
- MIZIKWizard: a platform to collect training data.
- MIZIKEvaluation: for performance evaluation.

In addition, we required the spoken dialogue system to be readily extensible for further work on multimodality. This paper describes the first two of these platforms, MIZIK and MIZIKWizard. We chose to base the work around the GALAXY architecture from MIT Spoken Language Systems Group [2]. GALAXY is a hub and spoke architecture, where the programmable hub mediates the interactions between various servers. This was a good architecture for our purposes. Firstly, it is straightforward to implement new platforms: for example, to add new functionality in the form of GALAXY servers, and to reuse servers in different systems. It also enables us to experiment with alternative speech and language technology very easily.

To develop MIZIK, shown in Figure 1, we wanted to use technology designed for mixed initiative dialogue. For this reason, MIZIK includes Speech Recognition (SUMMIT), Language Understanding (TINA) and Language Generation (GENESIS) engines from MIT Spoken Language Systems Group [3]. The dialogue manager is also based on MIT's Turn Management approach. For TTS we are currently using Acuvoice AV2001 [4] or Festival [5]. MIZIK has an Audio Server to connect a caller to the system and the application back-end is based around an Oracle database.

MIZIKWizard includes a graphical user interface (GUI) for the Wizard to enter a query: the implementation is based on the GUI server developed for WebGALAXY [6]. Language Understanding is different to that in the MIZIK system, as the Wizard operator interprets, rather than transcribes, the caller's requests. Turn Management and Language Generation have the same functionality as is required by MIZIK, with extensions designed specifically for the Wizard of Oz exercise. MIZIKWizard also includes the Database, Audio and TTS servers.

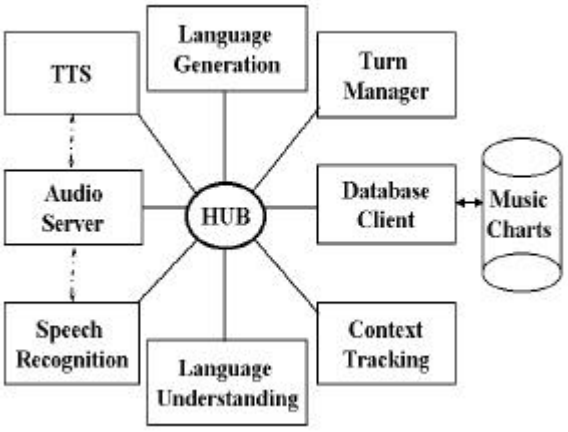

Figure 1: MIZIK architecture

4. Specifying the domain

A key part of developing a spoken dialogue system is to first of all clearly articulate the domain. System development follows from a full definition of the domain. Our aim was to allow people to interact with a music service over the telephone. Music is potentially a very large domain. We brainstormed around this topic, collecting example queries from a group of colleagues. A range of capabilities was considered. The service might include finding out information, such as: album or single titles, artists, prices, reviews, chart position, trivia, music events, etc. Additionally a user might select music to play, purchase music or related merchandise, and so on. Such a service can be split up into many smaller domains and indeed needs to be divided into smaller tasks in order that it is feasible to implement a spoken language user interface. Using the initial collection of example utterances, we selected a domain covering the music charts, as this was tangible, while still presenting some interesting challenges. For example, one challenge presented by this domain is the large number of unusual words contained in artist names and titles of music.

There are three major influences on the scope of the domain:

- The content that is available, for example, the size and content of the database.
- The service that you want to provide, for example, represented by the range of user queries you want the system to be able to handle.
- Pragmatic issues, including the performance capabilities of the technology and development issues. These factors will tend to limit the scope of the domain.

Having selected a domain, it was scoped using a combination of content driven, service driven and pragmatic approaches. Regarding content, our initial database contained information on best sellers, the current chart and latest entries. Specific data for each title included artist name, position and price. To help scope the domain from a service driven approach, we collected a further set of bootstrap utterances (in text) to find out how people would want to interact with a music chart service. On the basis of these utterances, the initial database was augmented with reviews of music. Considering the pragmatic issues, we limited the domain to exclude utterances containing the title of a piece of music, as this would have given the system too much freedom. To keep things simple to start with, a static database was used, as the resulting static domain will ease comparisons between versions of MIZIK as it is developed and evaluated. The result of the domain definition exercise was that MIZIK would handle queries about artists, prices, chart positions and reviews. For example, the user can ask for music by an artist or for the price or a review of music by an artist.

A set of database queries was designed, which determines the pieces of information a caller needs to provide. With this, we can define the set of attributes that can represent user queries, the values those attributes can take, and which combinations of attributes can be used to form valid database queries. There is an attribute for each piece of information the user may need to provide within the domain. The combination of all attributes and values represents the scope of the domain. Table 1 shows the set of attributes and values for the MIZIK domain. Table 2 shows example utterances and how they are represented by attributes and values. All of these, apart from the last, contain enough information to form a database query.

Table 1: Attributes and values for MIZIK

Attribute	Values	Database query
pos_cat	best_selling latest current	query best sellers query latest titles query current chart
pos_val	Number (one to forty)	query on the chart position
prod_orig	artist name	query on artist name
prod_var	album single	query for album/single
action	get_title get_price get_review get_pos	get the title get the price get a review get the chart position

Table 2: Example utterances and their attribute-values

Example Utterance	Attribute-value pair
"Find a review of the best selling album by All Saints."	pos-cat: best_selling prod-orig: all_saints prod-var: album action: get_review
"What are the new releases?"	pos-cat: latest action: get_title
"How much is the best selling album by The The?"	pos-cat: best_selling prod-orig: the_the prod-var: album action: get_price
"How many albums are there by New Order?"	prod-orig: new_order prod-var: album action: get_title
"What is the current Coldplay single?"	prod_orig: coldplay prod_var: single pos_cat: current action: get_title
"Get me a review."	action: get_review

5. Data collection and system development

5.1. MIZIKWizard system

MIZIKWizard development began by considering turn management, the database interface and the language generation for the English language responses. The turn manager is based around a simple set of rules: the rules fire according to the contents of an eform, which is a set of attribute-value pairs. The MIZIK eform can include the attribute-value pairs shown in Table 1. MIZIKWizard includes functionality to: detect any eform that contains enough information for a database query; produce a SQL query from it (this makes use of GENESIS); get the query results from the database and formulate an appropriate reply. There are also help responses; for example, for those e-forms that did not contain sufficient information for a database query. All of this can be reused later in the MIZIK system.

A protocol was designed for the Wizard-of-Oz interactions, which was tailored such that the Wizard operator would need little training. The Wizard interactions were designed to investigate what subjects would like to find out from a music service as well as to collect data specific to our smaller domain of the music charts. The protocol requires each subject to ask around ten questions about music. For the first four questions, the subject can ask any question about music. After this, the subject is prompted to ask specifically about the UK music charts. A grammar and semantics were designed specifically for MIZIKWizard, using the domain-specific sample utterances that were previously collected. Turn management, language understanding and generation were augmented specifically for the Wizard protocol.

5.2. Wizard data collection and analysis

We advertised for subjects from the general public via radio, email and flyers, targeting young adults. Subjects were invited to call a toll-free telephone number and to ask the system ten questions about music. The subjects' utterances and the system responses were recorded for later analysis. Callers were rewarded with cinema tickets; once these were dispatched, their identities were not recorded.

Over 4000 utterances were recorded and transcribed. Some callers were not able to complete ten questions, whereas others asked more than ten. Also, there were numerous repeat calls. The data includes 355 callers: 56% were male and 44% were female. The average age of the callers was 23 years, which is ideal for the target population of the MIZIK domain. The majority had British accents (4.4% had foreign accents): of these, many had regional accents that were local to Bristol and surrounding areas, where we advertised for subjects. There was considerable variation in the type of line and location from which calls were made. Considering all calls to the system, there were 270 landlines, 31 mobile phones, 25 cordless phones and 15 public phones. The data invariably contained background noise due to the various locations from which the calls were made. Much of the data contained spontaneous speech effects such as filled pauses (*um* and *uh*) and partial words. Where these occurred, the transcriptions were labeled accordingly.

The utterances were classified as in-domain or out-of-domain according to whether they could be represented by the MIZIK attributes, see Tables 1 and 2. For example, the utterance *"Tell me how long Sting has been in the charts"* cannot be represented by the attributes and would be classified as out-of-domain. The out-of-domain utterances, whilst not needed for immediate development of MIZIK, are important for future work. They provide an insight into the sort of expectations that callers have of a music service and could be used to expand the MIZIK domain, implement new domains or to develop domain-switching capabilities. The Wizard-of-Oz exercise resulted in a corpus of approximately 2500 in-domain utterances, 1500 of which were used to develop and train an initial version of MIZIK. The rest of the corpus was reserved for evaluation and future development.

The results of the Wizard-of-Oz exercise support the observation that it is essential to collect domain specific data from a range of people from the target user group. Figure 2 illustrates some of the diversity of the in-domain questions that were asked by subjects.

"What's David Gray's album like?"
"Where is Moloko's song in the chart?"
"Could you give me the price the CD price of the um current number one?"

Figure 2: Example utterances from the Wizard-of-Oz

5.3. MIZIK development

Using the training data, a SUMMIT recognizer was developed for the MIZIK domain. The resulting vocabulary is 950 words, about 550 of which are artist names. The vocabulary was designed using the artist names in our database, the words in the training set and the words that are covered by the MIZIK grammar. We developed an automated process to obtain the vocabulary. Pronunciations are drawn from MIT's SLS-PRONLEX dictionary and phonological modeling is achieved using these along with a set of phonological rules. Pronunciations that cannot be obtained in this way, which applies to many of the artist names, were entered manually. MIZIK makes use of a class bigram and trigram language models. Currently there are three classes: artist names, chart position and number. MIZIK uses acoustic models from the MIT JUPITER system [3].

For language understanding, the training data was used to develop a grammar and the semantic mapping rules that produce a semantic frame from a parse. Figure 3 shows an example semantic frame. Context tracking was implemented using TINA and the transformation from a semantic frame to an e-form was implemented with GENESIS. Turn management and language generation components from MIZIKWizard were reused in the MIZIK system, with some modification.

```
{c get_title
  :topic {q product
          :quantifier "def"
          :pred {p position_category
                 :topic "current" }
          :pred {p by_product_originator
                 :topic {q product_originator
                         :name "coldplay" } }
          :pred {p product_variant
                 :topic "single" } }
  :domain "hp" }
```

Figure 3: Example semantic frame for *"What is the current Coldplay single?"*

6. Conclusions

Music provided a compelling domain within which to explore the design and implementation of a mixed initiative system. During the development of an experimental spoken dialogue system, MIZIK, we learnt a number of fundamental lessons, which we believe are applicable to the design of any mixed initiative system.

- A key to developing a spoken dialogue system is to first of all clearly articulate the domain – this eases subsequent data collection and implementation of the system.
- A staged process can help to define the domain: first decide on the topic; then, define the exact scope, using a combination of content, service and pragmatic approaches.
- Decide on a set of attributes and values to cover all of the information used in the system: this will form the basis of the e-forms.
- For the data collection process, design an advertising strategy around the target user population.
- A well-designed Wizard-of-Oz protocol enables the use of Wizard operators who need little training and have no prior experience of speech and language technology.
- Match the number of Wizard operators to the predicted number of callers and to the advertising strategy. A call center is ideally suited to handling large and varied demand.
- To collect training data, give people a good motivation to call, with an incentive of a reward or information that is important to them.
- As an observation during transcription of the data, we found that someone from within the target user population produced more accurate transcriptions.
- Realistic, domain specific data is needed to train and develop a spoken dialogue system - the diversity of utterances we collected supports this observation.
- GALAXY offers a flexible and extensible architecture for implementing a variety of platforms, for example for data collection and a live system. Additionally, it offers the facility to experiment with alternative speech and language technology components.

These lessons, combined with the MIT GALAXY architecture, have provided us with a useful springboard to experiment in future with mixed initiative systems and multi-modality. Our intention is to use the data collected during the Wizard-of-Oz exercise to evaluate the MIZIK system and to extend it to experiment with Multimodal session management.

7. Acknowledgements

We would like to thank the staff in MIT's Spoken Language Systems Group for all their assistance and advice during this project. Particular thanks go to T.J. Hazen, Joe Polifroni and Stephanie Seneff. Also, we would like to acknowledge the contributions of staff and student interns at HP Laboratories, including: Guillaume Belrose; Annelies De Bruine; Yu Chien Chan; Stephen Hinde; Jane Mather; Michael McTernan; Jean-Francois Pillou; Daniel Willcox.

8. References

[1] Whittaker S.J. & Stenton S.P., "Wizard of Oz studies and the design of Natural Language systems", Proc. of the 4th EACL, 1989, pp. 116-122

[2] Polifroni J. and Seneff S., "GALAXY-II as an Architecture for Spoken Dialogue Evaluation" Proc. 2nd Int. Conf. on Language Resources and Evaluation (LREC), Athens, Greece, May 31-June 2, 2000.

[3] Zue V. et al., "JUPITER: A Telephone-Based Conversational Interface for Weather Information," IEEE Trans. on Speech and Audio Proc., V.8, N.1, Jan 2000.

[4] Fonix Corporation, http://www.fonix.com.

[5] Taylor P.A., Black A. and Caley R., "The Architecture of the Festival Speech Synthesis System", Third ESCA Workshop in Speech Synthesis, 1998, pp. 147-151.

[6] Lau R., Flammia G., Pao C. and Zue, V., "WebGALAXY: Beyond Point and Click - A Conversational Interface to a Browser," Proc. 6th Int. World Wide Web Conf., April 1997, pp. 119-127.