ISIS: A Trilingual Conversational System with Learning Capabilities and Combined Interaction and Delegation Dialogs

Helen Meng[1], Yee Fong Wong[1], Cheong Chat Chan[1], Shuk Fong Chan[1], Chi Kin Keung[1], Kai Chung Siu[1],
Tien Ying Fung[1], Wai Ching Tsui[1], Ke Chen[3], Lan Wang[3], Ting Yao Wu[3], Xiaolong Li[3], Tan Lee[2], Yiu Wing Wong[2],
P. C. Ching[2] and Huisheng Chi[3]
[1]Human-Computer Communications Laboratory,
[2]Digital Signal Processing Laboratory,
The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong SAR, China
[3]National Key Laboratory for Machine Perception,
Peking University
Beijing, China
{hmmeng@se.cuhk.edu.hk}

Abstract

ISIS, which abbreviates Intelligent Speech for Information Systems, is a trilingual spoken dialog system (SDS) for the financial domain. It handles two dialects of Chinese (Cantonese and Putonghua), as well as English – the predominant languages in our region. The system supports spoken language queries regarding stock market information and simulated personal portfolios. Real-time information is retrieved directly from a dedicated Reuters satellite feed. ISIS provides a system test-bed for our work in multilingual speech recognition and generation, speaker authentication, language understanding and dialog modeling. Furthermore, ISIS supports our new explorations in: (i) CORBA's *interoperability* and *scalability* for SDS development; (ii) *asynchronous* human-computer interaction by delegation to KQML software agents; (iii) switching between online interaction and offline delegation in a single dialog thread, and (iv) automatic incorporation of newly listed stocks to expand our system's knowledge base.

1. Introduction

ISIS (Intelligent Speech for Information Systems) is a trilingual spoken dialog system (SDS) for retrieving real-time market information and managing simulated personal portfolios. As exemplified by many prominent systems developed previously, e.g. [1, 11, 13, 15], an SDS delineates a scope for research on spoken language technologies. ISIS provides a system test-bed for our research in multilingual speech recognition and generation, speaker authentication, language understanding and dialog modeling.

ISIS is a trilingual system, supporting English, Cantonese and Putonghua – the predominant languages used in our region. The system resembles a virtual stockbroker, which can provide the user with real-time stock market information and personal portfolio information. It also handles simulated financial transactions. The financial domain is of particular interest to our region, which is one of the world's financial centers. Furthermore, the financial domain provides the necessary complexity for our research in spoken dialog systems. For example, number expressions abound and they need to be disambiguated and understood; new companies continue to be listed in the stock exchanges, which demands that our system be extensible to accommodate the new information.

This paper reports on our new explorations in: (i) CORBA's *interoperability* and *scalability* for SDS development; (ii) *asynchronous* human-computer interaction by delegation to KQML software agents; (iii) switching between online interaction and offline delegation in a single dialog thread, and (iv) automatic incorporation of newly listed stocks to expand our system's knowledge base.

2. CORBA

We have chosen to implement ISIS based on CORBA (Common Object Request Broker Architecture). CORBA is a suite of specifications produced by the Object Management Group (OMG), and aims to provide ease and flexibility for distributing components. Based on object orientation, CORBA's components can be shared among applications. This middleware also allows distributed object applications to *interoperate* across platforms through the network, by providing the IDL (Interface Definition Language) to communicate with different programming languages running on multiple operating systems. In addition, CORBA provides the ORB (Object Request Broker) which is a class library that handles communication between the IDL and the objects. ORB handles object location, request routing and result returning.

The ISIS implementation consists of a host of object-oriented server classes (speech recognition, language understanding, speech generation, etc.), that communicate with the client classes (the text I/O GUI and the speech I/O GUI) to handle user input over the Internet. The server classes run on multiple platforms – the UNIX and Windows NT operating systems, and programming languages including Java, C and Visual C++. Hence the IDL helps achieve *interoperability*. By altering the IDL, new classes can be added and tested without affecting the existing architecture. This enhances the *scalability* of the architecture. Furthermore, the implementation does not require that all server classes be started for development. For example, without starting the speech recognizers, users can still access the ISIS system via the text input GUI. Additionally, by virtue of the ORB, and CORBA's IIOP (Internet and Inter-ORB Protocol), our browser-based client can communicate with the server classes over the Internet or local Intranets, while maintaining *location transparency* of the objects.

3. Delegation to KQML Software Agents

As mentioned previously, this work includes our initial exploration with asynchronous human-computer interaction by delegation to KQML (Knowledge Query and Manipulation Language) software agents [3]. KQML is both a message format and a message-handling protocol to support information exchange and knowledge sharing among software agents. It enables a multi-agent communication with simple coding. KQML provides a basic architecture for us to implement the *user alert function* by means of information exchange among software agents. A non-blocking user query can result in a system reply (or an indeterminate number of replies) at irregular times in the future. In the mean time, the user can continue with the interaction session. We refer to this as *asynchronous* communication. Information exchange is mediated by a *facilitator*, which maintains a database of local services for routing messages.

4. System Architecture

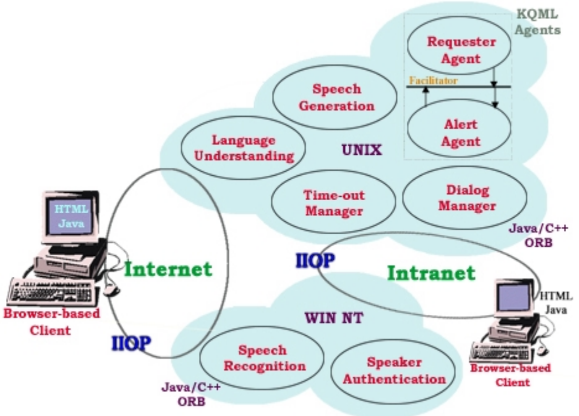

Figure 1. The ISIS System Architecture.

Figure 1 illustrates the architecture of ISIS. The server objects and the browser-based client object communicate with one another via the intranet or Internet with the IIOP (Internet InterORB Protocol). The client supports text I/O and audio I/O by incorporating the applet with Java Sound API. The six server objects include speech recognition, language understanding, speech generation, speaker authentication, dialog manager, and the time-out manager. Some are implemented in Java or C in the UNIX platform, others are implemented in Visual C++ on the NT platform. These server and client classes inherit CORBA stubs and skeletons to communicate with each other by passing messages through the IDL. Additionally, there is a pair of KQML software agents that communicate via the facilitator to handle the user's price alert requests. When triggered, the Requester agent issues a price alert request to the Alert agent which tracks the market price continuously. When the user's specified price is met, the Alert agent activates the Requester agent to alert the user. The Requester agent then sends the alert message to the Dialog Manager.

Data is passed between the server/client classes in the format of XML (EXtensible Markup Language).[1] The data is labeled with descriptive semantic tags characterizing the server class operations. For example, Figure 2 shows the output of the natural language understanding (NLU) component based on the input *"buy five lots of HSBC at the market price please."*

```
<NLU>
   <BUY>
      <STOCK TYPE = ric STATUS = iv> 0005.HK </STOCK>
      <PRICE TYPE = equal> market </PRICE>
      <LOT TYPE = equal> 5 </LOT>
   </BUY>
</NLU>
```

Figure 2: Example of a message produced by the language understanding server class. Input query *was "buy five lots of HSBC at the market price please."* *"ric"* denotes Reuters Instrument Code. *"iv"* denotes in-vocabulary.

With a distributed control strategy, each object (server / client) keeps track of its successor object(s) in the processing pipeline. Hence the current design avoids having to develop a very complex server object. This design is also more robust, because the system will not be paralyzed immediately upon glitches in any centralized control server. Instead, under the situations where a server object has problems, the other objects can still proceed with their processes and complete some of the tasks in the pipeline. For example, the Dialog Manager (DM) is the successor of the Time-out Manager (TM), the Alert agent and the Language Understanding (LU). TM (see Figure 1) monitors the time between successive user's inputs. If the time duration exceeds a pre-set threshold (i.e. the user has been silent for a while), TM sends a message to the DM object. As mentioned earlier, the Alert agent monitors the real-time financial data feed continuously and initiates an alert message when the user's pre-specified price is met. The LU server object transforms the user's query into a semantic frame and sends it to the DM. DM processes the messages from its three predecessor objects differently:

- If the message is received from TM, DM invokes its response generation procedure to produce the system response, "Are you there?" and then repeats the last system response.
- If the message is received from the Alert agent, DM handles it as an offline delegation subdialog, as will be described in detail in Section 7.
- If the message is received from LU, DM invokes a series of procedures / steps: (1) check for missing attributes in the semantic frame (E-form); (2) inherit discourse concepts; (3) validate at the first checkpoint;[2] (4) access information/database; (5) validate at the second checkpoint; and (6) generate a response frame.

The successor of DM is the Speech Generation (SG) server. SG can invoke various speech synthesizers, as will be described later.

[1] http://www.w3.org/XML

[2] Checkpoints one and two perform various kinds of validation, as we will see in the example dialogs that follow.

5. Spoken Language Technologies

ISIS integrates a plethora of speech and language technologies, together with dialog modeling techniques. We will describe them in the following.

5.1. Speech Recognition

Speech recognition (SR) in ISIS aims to handle three languages – English, Cantonese and Putonghua (two dialects of Chinese). We have integrated an off-the-shelf English speech recognizer into our system. We have also developed preliminary versions of the Cantonese and Putonghua recognizers. These HMM recognizers use acoustic models based on the initials (I) and finals (F) of the dialects. The Chinese recognizers use a two-pass search, where the first pass creates a syllable lattice, and the second traverses the lattice with a language model to produce recognized word sequences.

5.2. Language Understanding

We applied our framework based on Belief Networks for natural language understanding [9]. Our approach involves parsing the user's query (which may be spoken or typed) by means of a semantic grammar. The semantic concepts obtained are then fed into a suite of Belief Networks (BN) for communicative goal inference. This approach was previously applied to the English sentences in the ATIS domain.

We have extended this natural language understanding (NLU) framework to handle Chinese as well as English queries, and ported from ATIS to the stocks domain. Details are reported in [8]. Chinese input queries are first tokenized into a word sequence based on a 1100-word lexicon and a greedy algorithm. Parallel English and Chinese grammars were developed for semantic parsing, and they share a unified set of semantic concepts. We have also identified ten domain-specific goals for the stocks domain,[3] which include REAL-TIME QUOTES, NEWS, ORDER_AMENDMENTS, BUY, SELL, PORTFOLIO_INQUIRY, etc. Hence ten BNs were developed for goal inference. Should all ten BNs vote negative for a given query, it is rejected as out-of-domain (OOD).

The stocks domain has additional complexities for natural language understanding. Verbalized numbers abound in the domain-specific queries, and they can refer to stock codes, prices, number of lots, number of shares, etc. Our NLU component needs to disambiguate among these semantic categories. Consider the query example:

Buy HSBC, three hundred lots, at a hundred and five per share.

Verbalized numbers are parsed to obtain their numeric values, and the number expressions are classified into the appropriate semantic category by considering *both* left and right contexts in transformation-based rules for parsing [2]. For example, "*a hundred and five*" above is identified as a SHARE_PRICE. As will be described later, the same strategy is also useful for identifying the semantic category of out-of-vocabulary words, e.g. new stock names. Presently, our language understanding component only uses the *single* best recognition hypothesis is used for understanding. We plan to incorporate better SR/NLU coupling strategies as a next step.

5.3. Spoken Response Generation

System responses in ISIS need to be generated in three languages as well. For English, we generate the response text and send it to the FESTIVAL system [16], which has been integrated into ISIS. Language generation for English takes a response semantic frame as input. The frame specifies the language (i.e. English), the response type (e.g. STOCK_QUOTES_RESPONSE), related information attributes (e.g. STOCK_NAME, BID/ASK PRICES, etc.) and their values. Based on this response frame, the English response grammar is run in generative mode to produce the response text, to be used in text-to-speech synthesis. A similar approach is used for Cantonese response generation, and the response text is sent to CU VOCAL [5], a domain-independent corpus-based concatenative Cantonese text-to-speech synthesizer. CU Vocal ensures that in a contiguous pair of selected units the right coarticulatory context of the left neighbor is compatible with the left coarticulatory context of the right neighbor [4]. The methodology used in CU Vocal can be domain-optimized to produce spoken responses with enhanced naturalness. This methodology has been ported to Putonghua in a domain-optimized context for Putonghua response generation in ISIS.

5.4. Speaker Authentication

During an interaction, the user may attempt to access general stock market information, or personalized portfolio information. Inferred goals such as BUY, SELL, PORTFOLIO_INQUIRY, etc., from language understanding will trigger the *speaker authentication* component. The objective of speaker authentication (SA) is to automatically verify the speaker's claimed identity by his/her voice. Hence it provides a secured gateway for the user to access personal financial information and conduct financial transactions. The current SA component is a *text-dependent* speaker verification system, i.e. the system provides a digit string for the speaker to utter. Previous work indicates that the *Gaussian Mixture Model* (GMM) performs well for SA [12]. So a GMM-based SA with 16 mixture components is adopted for ISIS.

Another critical problem for speaker verification is the decision-making procedure. In ISIS, we adopt the hypothesis testing approach. We use the likelihood ratio between the claimed speaker's model and a background model from other speakers. After comparing this likelihood ratio with a preset threshold, the speaker authentication component makes the binary decision of acceptance or rejection.

5.5. Dialog and Discourse

Discourse inheritance is achieved by using an electronic form (E-form) model [10]. Semantic concepts from the current user's query take precedence over previous query (queries) for NLU. A mixed-initiative dialog model is used, and it can prompt for missing information, offer price alert services, invoke confirmation sub-dialogs, etc. The system also has a list of meta-commands e.g. HELP, GOOD-BYE, UNDO, etc., to allow the user to navigate freely in the dialog space.

In addition, there are two novel features in the ISIS dialog model: (1) It can incorporate newly listed stocks through a dialog interaction in order to expand the system's knowledge base. (2) The dialog model can also switch

[3] This is based on data collection from appropriate subjects for task definition.

between online interaction and offline delegation. These will be discussed in the next two sections.

6. A Learning System

We are developing ISIS into a learning system that can automatically expand its knowledge base. This is a desirable feature for our application domain because new stocks are continually added to the listing at the stock exchanges. For example, the Mass Transit Railway Corporation was listed at the Stock Exchange of Hong Kong recently, under the name MTR Corporation according to our dedicated Reuters feed. The company is commonly referred to as MTR in Hong Kong. Since the listing is new, none of these names exist in the ISIS knowledge base.

User1	**Do you have the real-time quotes of MTR?**
NLU produces the E-form in XML format: *<QUOTES><STOCK_NAME_OOV>MTR</STOCK_NAME_OOV></QUOTES>* *The checkpoint in DM (step 3) detects the <STOCK_NAME_OOV> tag in the SQL generation step, and invokes a procedure (step 4) that probes Reuters' name listings with a substring matching algorithm. Better matches are ranked higher in the displayed list.*	
System1	I do not know of MTR. This name is similar to the following list of stocks, together with their stock codes. Please speak the stock code for the correct choice. Otherwise please ask for another stock. ---------- 0066: MTR CORPORATION 1005: MATRIX HOLDINGS 0375: YGM TRADING etc.
User2	**It should be zero zero six six**
DM interprets the spoken code correctly by the context inheritance (step 2). The following subdialog attempts to match the commonly used stock name with the listed name.	
System2	Here are the real-time quotes for MTR (Quotes shown) Do you generally refer to MTR CORPORATION as MTR? Please say yes or no.
User3	**Yes**
Upon user confirmation, DM calls LU to add the rules STOCK_NAME→MTR and STOCK_CODE→0066 to its grammar.	
System3	I got it. How else may I help you?
User 4	**Show me the latest news of MTR please.**
LU's new grammar rule is now capable of interpreting queries regarding the newly listed company.	
System4	There is no news for MTR today.

Table 1. Example dialog showing the automatic incorporation of new stocks into the ISIS knowledge base.

We begin by tackling the problem at the natural language level.[4] The automatic learning process begins when a user types in an input such as, "Do you have the real-time quotes of MTR?" Our Language Understanding (LU) server object identifies that MTR is Out-Of-Vocabulary (OOV), and employs a transformation-based parsing technique [2] to infer a possible concept category for the word, and tagged it as <STOCK_NAME_OOV>. The sub-dialog that follows is shown in Table 1. The underlying operations are explained in italics in the table. The subdialog confirms with the user the stock code that corresponds to the new stock name, and automatically incorporates the new grammar rules in the LU component. Thereafter the new stock name has been incorporated into the ISIS knowledge base. In the case of Chinese, OOVs are identified by an n-gram grouping technique together with transformation-based parsing. Subsequent operations are the same as for English.

Referring to the example dialog in Table 1, special care is taken in speech generation. For English, our Speech Generation (SG) server object invokes the Festival text-to-speech synthesizer [16], which can use text analysis and letter-to-sound rules to generate English speech for new names. For Cantonese, SG can invoke our CU VOCAL domain-independent Cantonese text-to-speech synthesizer to generate a newly acquired stock name. We plan to attain domain-independence in Putonghua synthesis as well.

7. Combining Interaction and Delegation Subdialogs

As shown in Figure 1, users of ISIS can delegate tasks to a pair of software agents implemented in KQML. KQML is both a message format and a message-handling protocol to support information exchange among agents. A non-blocking request from the user query is sent to the Requester agent (see Figure 1), which communicates the message to the Alert agent through the Facilitator. The typical request is for the Alert agent to monitor a specified stock price hitting a particular price point. Users may launch an agent with an explicit request, e.g. "*Please notify me when Cheung Kong Holdings rises to ninety two dollars per share*." Alternatively, a buy or sell request for which there is a mismatch between the ask/bid price and the market price will also cause an agent to be launched. When the specified condition is met, the Alert agent will send an alert message through the Facilitator and the Requester and back to the user (see Table 2).

Hence the dialog model in ISIS allows transition between a mixed-initiative online interaction (OI) and an offline delegation (OD). This is achieved by maintaining two lists of E-forms (semantic frames). The first one (L1) shows the discourse history, and the second (L2) represents the alert message(s) received from the Alert agent. An example dialog is shown in Table 2.

It should be noted that the E-forms belonging to the offline delegation subdialogs are marked in order that discourse inheritance does not mix with E-forms belonging to online interaction in the dialog thread.

[4] At the moment we bypass the problem of recognizing of out-of-vocabulary (OOV) words, which is a research problem in and of itself.

<table>
<tr><td><b>User1</b></td><td><b>Show me the current news of Cheung Kong</b></td></tr>
<tr><td colspan="2"><i>LU produces the E-form (semantic frame):
GOAL: NEWS ; STOCK_NAME: Cheung Kong
DM begins by appending the E-form to L1, performs database access (step 4) and augments the E-form with the retrieved information (RESULT)to become:
GOAL: NEWS ; STOCK_NAME: Cheung Kong; RESULT: nil
DM's second checkpoint (step 5) validates the presence of RESULT and updates the E-form with STATUS: success to indicate that the user's request has been fulfilled.
Response Frame Generation (step 6) generates a response frame to send to the Speech Generation (SG) object, which produces the initial part of the system response.</i></td></tr>
<tr><td>System1</td><td>There is no news for Cheung Kong today.</td></tr>
<tr><td colspan="2"><i>At this instant the Alert agent sends a message to DM, regarding a previous buy request for HSBC.
DM invokes the following series of steps upon receiving a message from the Alert agent.
(1) It appends the message frame to L2.
(2) It detects that the previous user request is fulfilled (STATUS: success) in the L1 E-form.
(3) It invokes the Response Frame generation step to produce the following system response (System2). Notice that the stock name (HSBC) is used as an anchor word to refer to its corresponding alert message.
(4) It sets the value of a global variable α=1 which marks the possible switch from the online interaction dialog to the offline delegation subdialog.</i></td></tr>
<tr><td>System2</td><td>There is one alert message for you regarding a previous buy request on HSBC. If you want to handle the alert message now, please say HSBC. Otherwise, please continue.</td></tr>
<tr><td><b>User2</b></td><td><b>HSBC</b></td></tr>
<tr><td colspan="2"><i>LU produces the E-form (semantic frame):
GOAL: OOD ; STOCK_NAME: HSBC
DM receives this message from LU. Then it invokes the first checkpoint (step 3) to validate that the α is set (and then resets it); and HSBC is the anchor word for an alert message in L2. This step also stores the current E-form in L1 to a register. It then removes the message frame from L2 and appends to L1. It also updates the E-form with STATUS: to_confirm, to prepare for a typical transaction's confirmation subdialog. Step 6 then generates the following response (System3). As such we are switching to the offline delegation subdialog.</i></td></tr>
<tr><td>System3</td><td>The stock price of HSBC is at ninety-eight dollars presently. Please confirm your previous request from March 20, 2001 at 2:00PM. You wish to buy three lots of HSBC at the requested price of ninety-eight dollars per share. Please say yes to confirm or no to reject the action.</td></tr>
<tr><td><b>User3</b></td><td><b>Yes</b></td></tr>
<tr><td colspan="2"><i>LU produces an E-form, and sends it to DM. The usual procedures as described in Section 2 are invoked. The system then responds as follows (System4).</i></td></tr>
<tr><td>System4</td><td>The buy transaction is completed. Please input a new query.</td></tr>
<tr><td><b>User</b></td><td><b>Let's go back.</b></td></tr>
<tr><td colspan="2"><i>LU treats this as a meta-command. DM's first checkpoint (step 3) detects this and restores the latest dialog state (for online interaction) from the register to L1. Response Generation (step 6) then presents a summary of this dialog state. As such we have switched back to an online interaction subdialog.</i></td></tr>
<tr><td>System5</td><td>Previously you requested to see the current news of Cheung Kong but there is no news for Cheung Kong today. How else may I help you?</td></tr>
</table>

Table 2. Example dialog showing the transitions between the online interaction and offline delegation subdialogs.

8. Recent Development

A recent development in ISIS is automatic end-point detection, to enhance the previous push-to-talk configuration. The end-point detection algorithm references three quantities measured from the input signal: energy, zero-crossing rate and periodicity. These are measured for every frame with a 10ms frame shift. If the energy level exceeds a pre-set threshold, we begin to monitor for periodicity / zero-crossings for 15ms. If either is found, we assume the instant of threshold crossover is the start-point of speech activity. Periodicity is used to detect voiced segments, and zero-crossing rate is used to detect fricatives. If several consecutive frames show no periodicity for 0.5s, or the energy level drops below a threshold, we assume that we have found the endpoint. We also allow margins of 0.3s on both sides (start and end points) to guarantee that the whole speech segment is extracted.

To evaluate our automatic endpoint detection algorithm, we used ten minutes of Cantonese recordings with acoustics from an office environment. The recordings have hand-labeled endpoints as references. 98% of the endpoints were detected within 0.017s duration of the reference boundaries. We also added noise to the speech data (SNR=20dB) and repeated the experiment. In this case, 85% of the end points are detected within 0.024s of the reference boundaries. Hence we conclude that this endpoint detector is fit for use in an office environment.

9. Conclusions and Future Work

This paper describes the design and development of ISIS, a trilingual spoken dialog system for the financial domain. The system has been under development for nearly two years. We have reported on our preliminary exploration with a CORBA implementation and KQML software agents for user alert functions. We present the novel features in spoken dialog systems development in ISIS. The system can automatically incorporated newly listed stock names (out-of-vocabulary words) into its knowledge base. It can also support online interaction and offline delegation, and the dialog model allows the user to switch freely between the two. We have also incorporated one of the first Cantonese corpus-based concatenative speech synthesizer into ISIS to produce highly natural Cantonese responses.

10. Acknowledgments

This research is supported by the Joint Center for Intelligence Engineering between Peking University and the Chinese University of Hong Kong. We thank Reuters Hong Kong for donating their satellite feed to support our research.

11. References

[1] Bayer S., Doran C., George B., “Exploring Speech-Enabled Dialog with the GALAXY Communicator Infrastructure,” Proceedings of HLT Conference, 2001.

[2] Brill, E., "Automatic Grammar Induction and Parsing Free Text: A Transformation-based Approach," Proceedings of ACL, 1993.

[3] Finin, T., R. Friitzson, D. McKay and R. McEntire, “KQML – A Language and Protocol for Knowledge and Information Exchange,” Technical Report CS-94-02, University of Maryland, UMBC

[4] Fung T. Y. and H. Meng, "Concatenating Syllables for Response Generation in Domain-Specific Applications," Proceedings of ICASSP, 2000.

[5] Meng, H., et al., www.se.cuhk.edu.hk/cuvocal

[6] Meng, H. et al., “ISIS: A Learning System with Combined Interaction and Delegation Dialogs” Proceedings of Eurospeech, September 2001.

[7] Meng H., et al., “ISIS: A Multilingual Spoken Dialog System developed with CORBA and KQML agents,” Proceedings of ICSLP, 2000.

[8] Meng, H. and C. Tsui, “Comprehension Across Application Domains and Languages,” Proceedings of ISCSLP, 2000.

[9] Meng H., W. Lam and C. Wai, "To Believe is to Understand," Proceedings of Eurospeech, 1999.

[10] Meng H., et al., “WHEELS: A Conversational System on Electronic Automobile Classified, Proceedings of ICSLP, 1996.

[11] Papineni K. A., Roukos S. and Ward R. T., “Free-Flow Dialog Management using Forms,” Proceedings of ICSLP, 1998.

[12] Reynolds, D. A., “A Gaussian mixture modeling approach to text-independent speaker identification,” Ph.D. thesis, Georgia Institute of Technology, 1992.

[13] Rosset, S., S. Bennacef and L Lamel, “Design Strategies for Spoken Language Dialog Systems,” Proceedings of Eurospeech, 1999.

[14] Rudnicky, A. et al., “Creating Natural Dialogs in the Carnegie Mellon Communicator System,” Proceedings of Eurospeech, 1999.

[15] Seneff, S., et al., "Organization, Communication and Control in the Galaxy-II Conversational System,” Proceedings of ICSLP, 1998.

[16] Taylor, P. et al., "The architecture of the Festival speech synthesis system", in Proceedings of the 3rd ESCA Workshop on Speech Synthesis, pp.147-151, 1997.