ISIS: A Learning System with Combined Interaction and Delegation Dialogs

Helen Meng[1], Shuk Fong Chan[1], Yee Fong Wong[1], Cheong Chat Chan[1], Yiu Wing Wong,[2]
Tien Ying Fung[1], Wai Ching Tsui[1], Ke Chen[3], Lan Wang[3], Ting Yao Wu[3], Xiaolong Li[3], Tan Lee[2], Wing Nin Choi,[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

This paper presents a progress update of our ISIS[1] trilingual spoken dialog system. As described in [8], this is a conversational system for the stocks domain, and supports interactions in the languages of our region – English and two dialects of Chinese (Mandarin and Cantonese). ISIS provides a system test-bed for our initial explorations with the CORBA architecture, and delegation to KQML (Knowledge Query and Manipulation Language) agents. CORBA offers the advantages of interoperability, scalability and location transparency in client/server systems development. Users can delegate tasks to software agents to help monitor information (e.g. a drop in the price of a pre-specified stock), and generate user alert messages. Our current work presents new research directions in the context of ISIS: (i) automatic incorporation of newly listed stocks into our system's knowledge base; (ii) switching between on-line interaction and off-line delegation in a single dialog thread. We will also report on enhancements in the system's architecture and features (e.g. automatic end-point detection).

1. Introduction

This paper presents a progress update of our ISIS[1] trilingual spoken dialog system. As described in [8], this is a conversational system for the stocks domain, which supports interactions in the languages of our region – English and two dialects of Chinese (Mandarin and Cantonese). ISIS provides a system test-bed for our initial explorations with the CORBA architecture, and delegation to KQML (Knowledge Query and Manipulation Language) agents. CORBA offers the advantages of interoperability, scalability and location transparency in client/server systems development. Users can delegate tasks to software agents implemented in KQML, and the agents can help monitor information (e.g. a drop in the price of a pre-specified stock), and generate user alert messages. Our current work presents new research directions in the context of ISIS: (i) automatic incorporation of newly listed stocks into our system's knowledge base; (ii) switching between on-line interaction and off-line delegation in a single dialog thread. We will also report on enhancements in the system's architecture and features (e.g. automatic end-point detection).

2. System Architecture

Previous work in the development of software infrastructures for dialog systems include [11][2] [2]. Over the past year, we have continued to explore the development of a spoken dialog system based on the CORBA architecture. This middleware resides in between the operating system and the application layer to deliver platform and language independence.

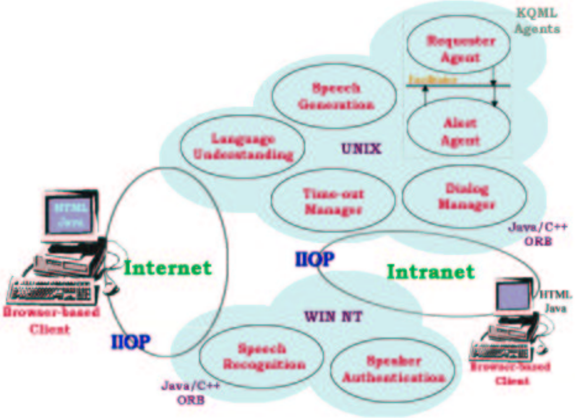

Figure 1. The ISIS System Architecture.

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

[1] ISIS abbreviates Intelligent Speech for Information Systems.

[2] This is the Galaxy Communicator architecture, a reference architecture designed and developed at MIT, and in recent months enhanced and distributed by MITRE Corporation (Bayer et al., 2001).

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. There is also the a pair of software agents implemented in JKQML.

A major change in our current system is to replace the centralized control (and data routing) with a distributed control strategy. The centralized control was previously performed by the Flow Control Manager server object, which is now removed. Current control is distributed as 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 the Flow Control Manager. 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.

Of these server objects, two are new. The Time-out Manager (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 an XML message to its successor, the Dialog Manager (DM) object. DM is also the successor of the Alert agent object and the Language Understanding (LU) object. Hence DM processes the three types of messages 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, which will be described in detail in Section 4.
- 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;[3] (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 object. SG can invoke various speech synthesizers, as shown in the next section.

3. A Learning System

We have begun to develop 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 exchange. 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.

The automatic learning process begins when a user types in[4] 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 [9] 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. In the case of Chinese, OOVs are identified by an n-gram grouping technique together with transformation-based parsing [9]. Subsequent operations are the same as for English.

Referring to the example dialog in Table 2, special care is taken in speech generation. For English, our Speech Generation (SG) server object invokes the Festival text-to-speech synthesizer [12], which employs text analysis and letter-to-sound rules to generate speech for new names. For Chinese, SG can invoke either our concatenative speech synthesizer [5]; or our PSOLA-based text-to-speech synthesizer [7]. The concatenative synthesizer is domain-specific, and uses a bank of pre-recorded domain-specific wave files to generate highly natural speech outputs. It can also generate both Putonghua and Cantonese. Consequently, SG defaults to the concatenative synthesizer for Chinese responses. However, a newly acquired stock name may require waves files absent from the wave bank. Under this situation, SG automatically reverts to the domain-independent Cantonese PSOLA synthesizer. We plan to port our PSOLA synthesizer to handle Putonghua as well.

4. 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*."

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

[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.

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 3).

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 3.

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.

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 new 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 2. Example dialog showing the automatic incorporation of new stocks into the ISIS knowledge base.

User1	**Show me the news of Cheung Kong**
LU produces the E-form (semantic frame): *GOAL: QUOTES ; 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: QUOTES ; 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.*	
System1	There is no news for Cheung Kong today.
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.*	
System2	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.
User2	**HSBC**
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.*	
System3	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.
User3	**Yes**
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).	
System4	The buy transaction is completed. Please input a new query.
User	**Let's go back.**

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.	
System5	Previously you requested to see the past news of Cheung Kong but there is no news for Cheung Kong today. How else may I help you?

Table 3. Example dialog showing the transitions between the online interaction and offline delegation subdialogs.

5. Additional Enhancements

5.1. Automatic End-Point Detection

The ISIS audio recording previously operates with a push-to-talk configuration, and we have enhanced it with automatic end-point detection. 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.

5.2 Constraints Specification

In our implementation, we try to specify constraints and validation by using domain-specific text files. For example, we have text tables that lists mandatory attributes for each information goal to help language understanding; and text files that govern response generation. This should ease further system development.

6. Conclusions and Future Work

This paper presents a progress update of our ISIS system, a trilingual (English, Putonghua and Cantonese) conversational system in the stocks domain. ISIS is developed on the CORBA middleware, and incorporates KQML software agents to support offline delegation. Aside from system enhancements such as the incorporation of automatic endpoint detection, we have begun on two new directions within the ISIS context. The first is to explore the development of a learning system, where the system's knowledge base is automatically expanded through interaction with the user. The second is to explore transitions between online interaction and offline delegation in a single dialog thread. We have presented preliminary mechanisms to achieve these new system capabilities, and much research and investigation remains to be done.

7. 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.

8. References

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

[2] P. R. Cohen, A. J. Cheyer, M. Wang and S. C. Baeg, "An Open Agent Architecture," AAAI Spring Symposium, 1994.

[3] M. Chu and P.C. Ching, "A Cantonese synthesizer based on TD-PSOLA method", Proc. of ISMIP-97, pp.262–7, Taipei.

[4] G. Damnati and F. Panaget, "Adding New Words in a Spoken Dialogue System Vocabulary Using Conceptual Information and Derived Class-based LM," Proceedings of ASRU, 1999.

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

[6] James R. Glass, "Challenges for Spoken Dialogue Systems," Proceedings of ASRU, 1999.

[7] T. Lee, H. Meng, W. Lau, W K. Lo, and P. C. Ching, "Micro-prosodic Control in Cantonese Text-to-Speech Synthesis," Proceedings of Eurospeech 1999.

[8] H. Meng, et al., "ISIS: A Multilingual Spoken Dialog System developed with CORBA and KQML agents," Proc. of ICSLP, 2000.

[9] H. Meng and W. C. Tsui, "Comprehension across Application Domains and Languages," Proceedings of ISCSLP, 2000.

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

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

[12] P. Taylor, A. Black and R. Caley, "The architecture of the Festival Speech Synthesis System," the 3rd ESCA Workshop on Speech Synthesis, pp. 147-151, 1998.