ISIS: An Adaptive, Trilingual Conversational System With Interleaving Interaction and Delegation Dialogs

HELEN MENG, P. C. CHING, SHUK FONG CHAN, YEE FONG WONG, and CHEONG CHAT CHAN
The Chinese University of Hong Kong

ISIS (Intelligent Speech for Information Systems) is a trilingual spoken dialog system (SDS) for the stocks 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. The conversational interface is augmented with a screen display that can capture mouse-clicks as well as textual input by typing or stylus-writing. 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. This article reports on our new explorations within the context of ISIS, including: (i) adaptivity to knowledge scope expansion; (ii) asynchronous human-computer interaction by task delegation to software agents; (iii) multi-threaded online interaction and offline delegation dialogs with interruptions for task switching.

Categories and Subject Descriptors: H.5 [**Information Interfaces and Presentation**]

General Terms: Human Factors, Design

Additional Key Words and Phrases: Human-computer spoken language interface, interaction and delegation dialogs

1. INTRODUCTION

Spoken dialog systems (SDS) are becoming increasingly pervasive in our everyday lives for information access. Efforts devoted to the design and development

This research is supported by the Joint Center for Intelligence Engineering between Peking University and The Chinese University of Hong Kong, and partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4326-02E).
Authors' addresses: H. Meng, Human-Computer Communications Laboratory, The Chinese University of Hong Kong, Shatin, NT, Hong Kong SAR, China; email: hmmeng@se.chuk.edu.hk; P. C. Ching, Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong SAR, China; email: pcching@ee.chuk.edu.hk.

of SDS aim to bring the right information to the right people at the right time for a diversity of application domains, for example, finance, air travel, train schedules, and weather. These domains typically involve dynamic information of recurring interest to the user. SDS encompasses a suite of speech and language technologies, which offer a *conversational interface* to dynamic information. They include speech recognition, natural language understanding, dialog modeling, and speech synthesis. Hence the user can present queries to the system by speaking naturally, and the SDS can respond in real-time in synthetic speech. Numerous commercial SDS have been deployed. For example, the SpeechWorks[1] voice-activated stock trading system enables callers to get real-time stock and market information, place orders and trade stocks at any time. The entire human-computer interaction is conducted over a fixed-line/mobile telephone channel—in a screen-less setting. An example is provided in Table I to illustrate the *directed* nature of the dialog interactions, where the system elicits a series of information attributes from the user in a scripted order. Interactions of this style are often known as *directed dialogs* or *system-initiative dialogs*. The system is always in control to guide the dialog at each step and constrain possible user input to a small set of options. Hence the system is able to attain a decent level of robustness in recognizing and interpreting user input in order to achieve a high task completion rate for informational transactions. However, the ease-of-use of these human-computer spoken language interfaces is partially sacrificed due to constraints in interactivity. Hence recent research efforts in the field strive to relax the constraints in system-initiative dialogs.

Mixed-initiative dialog interactions allow both human and computer to influence the dialog flow and thus offer greater flexibility than system-initiative dialogs. If the users could assume complete control of the human-computer interactions and if the system could respond sufficiently to all user requests, we would have *user-initiative dialogs*. However, this remains an elusive goal for existing technologies and systems. Recent efforts have been devoted to the development of mixed-initiative dialog models, in which the computer takes initiative at appropriate times to constrain the user's request in order to progress towards an intended/revised task goal [Horvitz 1999; Levin et al. 2000]. Prominent, mixed-initiative systems [Seneff et al. 1999; Rudnicky et al. 1999; Rosset et al. 1999] involve application domains such as air travel, train travel schedules, hotel and traffic information, and so on.

This article describes an initial attempt to explore three possible directions of evolution for conversational interfaces within the context of an SDS named ISIS in the stocks domain [Meng et al. 2000b, 2001, 2002a]. ISIS is a trilingual system that supports English, Cantonese and Putonghua, the prime languages used in our region. Putonghua (also known as Mandarin) is the official Chinese dialect and Cantonese is a major Chinese dialect predominant in Hong Kong, South China, and many overseas Chinese communities. ISIS resembles a virtual stock-broker who can provide the user with real-time stock market information, manage simulated personal portfolios, and handle simulated financial transactions via mixed-initiative conversational interactions. Consider the

[1]Now Scansoft Inc. (www.scansoft.com).

Table I. An Example Dialog in the Stocks Domain Illustrating the Capabilities of a State-of-the-Art Spoken Dialog System (Source: www.speechworks.com)

System1	Welcome to Lim & Tan Securities voice-activated stock trading system. <tone> Stock trade menu. Please say the name of the stock you'd like to place an order for.
User1	*Singapore Airlines*
System2	Buy or sell?
User2	*Buy*
System3	How many shares would you like to buy
User3	*Two thousand*
System4	Singapore Airlines Corporation. At what price would you <interrupted> . . . ?
User4	<barge in> eighteen eighty
System5	Would you like to use cash or CPF?
User5	*Cash*
System6	OK you want to buy 2000 shares of Singapore Airlines Corporation at eighteen eighty using cash. Your order as been queued. Returning to main menu.

example of the user's request, "*show me the monthly charts*." Instead of showing the monthly charts of *all* existing stocks, the mixed-initiative dialog model in ISIS will prompt the user with "*Please specify a stock name or stock code*." In addition to prompting for missing attributes, the mixed-initiative dialog model in ISIS can also reject an (out-of-domain) request, invoke confirmation sub-dialogs for important dialog turns (e.g., for important transactions) and prompt for user activity when system timeout nears. We have chosen to work in the domain of financial information because of its strong local relevance (since Hong Kong is one of the world's financial centers). Furthermore, the financial domain offers a suitable context for exploring evolutionary enhancements in mixed-initiative SDS development. One possible direction of evolution is *adaptivity*—as new companies continue to be listed at stock exchanges, the conversational interface and its constituent technologies need to be *adaptive* to support automatic expansion of the application's knowledge base. Another possible direction of evolution is *offline delegation*—users of financial applications often have to continually monitor multiple streams of dynamic information to detect specific changes. The information streams may include quotations for different stocks in the user's portfolio and financial indices around the world. Intelligent information systems may seek to reduce the cognitive load of such users by enabling them to specify the preconditions (e.g., increments, decrements and target price/index levels) for which they need to monitor and also to *delegate* the task of monitoring to software agents. A third possible direction of evolution is support for *multi-threaded dialogs*—should a precondition be met while software agents are monitoring various information feeds, the agents will send an alert message to the user. The alert message may arrive while the human and computer are engaged in an online interaction. Under this circumstance, dialog management should offer options that support *interruption* of the online interaction by the alternative dialog thread derived from the alert message, as well as the *return* to the original interrupted workflow. To summarize, a unique feature of this work is the exploration, within the confines of the end-to-end ISIS system, of the *combination* of adaptivity, offline delegation and multi-threaded interactions in coherent, mixed-initiative human-computer dialogs.

While the aforementioned evolutionary developments are explored in concert in the context of ISIS, this work also anticipates the widespread adoption of a new generation of wireless handheld communication devices. These include Internet-ready phones, smart phones and Personal Digital Assistants (PDAs). It is projected that in the next three years there will be 1.4 billion users of wireless handheld devices and the market will grow six-fold.[2] Such rapid adoption is catalyzed by the onset of 3G communication technologies that transmit voice and data simultaneously over a high-bandwidth wireless channel to bring rich media content to mobile users. The new mobile devices have embedded microphones and speakers for voice input and output and may also incorporate combined features and functionalities of the telephone and the computer. The new devices are equipped with small, touch-sensitive screen displays, styluses for textual input via handwriting recognition or virtual keyboards and for pointing/clicking/circling on the screens. This array of interface artifacts will augment the existing conversational interface to enrich interactivity in human-computer multimodal dialogs.

The rest of the paper is organized as follows: Section 2 reviews some previous work in related research areas. Section 3 presents an overview of the ISIS system. Explorations in adaptivity, offline delegation, interruptions and multi-threaded dialogs are described in Sections 4, 5 and 6. Section 7 concludes and presents possible future directions.

2. PREVIOUS WORK

This work explores mixed-initiative dialog modeling combined with adaptivity, offline delegation, and multi-threaded interactions in the context of the ISIS system. This section presents a brief review of related work, including the implementation frameworks for dialog models, handling new vocabularies in spoken language input, and dialog model extensions to support delegation and interruption.

The *state-based framework* is often used for implementing system-initiative dialogs (see Table I). Each dialog turn is modeled as a transition through a state sequence. The entire sequence prompts for the set of information attributes in stepwise order. Expert users may attempt to expedite transactions by providing multiple information attributes in a single dialog turn (e.g., in a natural language query/request). Hence the state-based framework may allow certain states to be bypassed while maintaining the order of information entry. An illustration from the foreign exchange hotline[3] [Meng et al. 2000a] is provided in Table II. While the scripted order for information entry helps achieve a decent level of robustness in task completion, it also constrains interactivity in the dialog. This constraint is relaxed in the *frame-based framework*, which is often used for implementing mixed-initiative dialogs. This framework resembles filling information slots in an electronic form (E-form) and the entries can be input in *any* order. At each dialog turn the system may check for unfilled

[2]Source: International Data Corporation.

[3]This is the CU FOREX system. A demonstration of the direct dialog is available at +852-2603-7884 and a demonstration of the natural language shortcut is available at +852-2603-7830.

Table II. An Example Dialog from the CU FOREX Hotline [Meng et al. 2000a]

System1	Welcome to CU FOREX, a trilingual hotline for real-time foreign exchange inquiries developed at The Chinese University of Hong Kong. Which language would you prefer? Please say Cantonese, Putonghua or English . . . . (same prompt repeated in the other languages)
User1	English
System2	This system can handle a full question for your inquiry. <tone> Main menu. What kind of currency information are you interested in?
User2	*I'd like to know the three-month interest rate of the Eurodollar.*
System3	The quotes we provide are for reference only. Interest rate. Eurodollar for three months is two point four five. <tone> Main menu. What kind of currency information are you interested in?
User3	*How about the exchange rate between the British Pound and the Hong Kong dollar?*
System4	The quotes we provide are for reference only. British pound to Hong Kong dollar. The buying rate is twelve point six zero seven. The selling rate is twelve point six zero one.

slots and prompt the user for associated information attributes. The user may also choose to be cooperative and answer according to the prompt; or uncooperative and provide an alternative information item. In this way both the system and the user can influence dialog progression, which is characteristic of *mixed-initiative* interactions [Goddeau et al. 1996; Aust et al. 1995]. Attempts are underway to achieve even greater flexibility in dialogs to support mixed-initiative negotiations and collaborative problem solving. Such dialogs are the least structured and involve planning and reasoning with knowledge and logic [Sadek et al. 1997]. A comprehensive review can be found in McTear [2002].

There is a critical need for spoken dialog systems to be able to handle new vocabularies in application domains with an expanding knowledge score. However, spoken dialog systems are typically developed for a predetermined and fixed vocabulary. The occurrence of new words, also known as out-of-vocabulary (OOV) words, will inevitably lead to errors. Previous work has studied the OOV problem mainly in the context of speech recognition and language modeling. A generic word model that permits arbitrary phone sequences is used as the OOV word model [Manos and Zue 1997; Bazzi and Glass 2000] to detect new words. This may be used in conjunction with a confidence model to predict recognition errors due to OOV words that are misrecognized as in-vocabulary words [Hazen and Bazzi 2001]. Issar [1996] studied the use of class-based language models that have expandable classes to include OOV words deemed appropriate for the class. Lau and Seneff [1998] and Seneff et al. [1996] applied sublexical linguistic modeling to tackle the OOV problem. The sublexical model captures the relationships among morphs, graphemes, phonemes and phonological rules to be applied to new word detection, letter-to-sound/sound-to-letter generation and eventually to new word acquisition. Automatic acquisition of new information items and nomenclatures, their new vocabularies, spellings, and pronunciations is an important direction for further development. This helps achieve *adaptivity* of conversational interfaces to application domains with growing knowledge scopes.

Thus far, we have seen many examples of using spoken dialog systems for information access. However, it is conceivable that the systems' utility may be extended to offline information monitoring. More specifically, if the user needs to closely monitor changes in a piece of dynamic information, he/she will have to talk to the spoken dialog system rather frequently. The cognitive load of the user may be significantly reduced if the he/she may *delegate* the task of information access/retrieval tasks to software agents that can run continuously. This is exemplified by the multi-domain ORION system developed at MIT [Seneff et al. 2000]. Users can call ORION to enroll with their contact information. They can also call ORION to define a task, for example, to alert the user an hour prior to touchdown of a flight. The system will alert the user at a designated time to deliver the requested information. Such alert messages may at times *interrupt* an online dialog between the human and the computer. A cost-benefit analysis of disrupting the online dialog may be analyzed prior to deciding upon an interruption. Previous work by Horvitz et al. [2003] developed probabilistic user models that incorporate utility values in deliberations about interrupting the user upon receiving alert messages. Related psychological studies investigate the effects of interruption at different phases of the primary, ongoing task [Czerwinski et al. 2000] and propose visualization designs that enhance awareness of multiple, prioritized interruptions/alerts [van Dantzich et al. 2002]. Support for interruptions and multi-threaded dialogs in conversational interfaces calls for a computational theory of discourse that lays out the structures necessary for proper treatment. Such theory is proposed by Grosz and Sidner [1985] for task-oriented dialogs. The aforementioned studies in alerts and interruptions lay the groundwork for developing intelligent systems that can automatically reason whether, when, and how to execute an interruption.

The following presents the ISIS system, a test-bed in which we have implemented a mixed-initiative dialog model and explored adaptivity, delegation, and interruption within the stocks domain.

3. OVERVIEW OF THE ISIS SYSTEM

3.1 The ISIS Domain

The ISIS application domain subsumes real-time stock information inquiries as well as transaction requests. Many subjects were recruited in a survey to provide the various kinds of queries they may wish to present to a financial information system. The survey generated approximately two thousand textual queries in Chinese and English, respectively. These queries were referenced as we defined ten domain-specific task goals that determine the scope of the ISIS domain. The task goals are: QUOTE (asking for a real-time stock quote), NEWS (asking for news about a listed company), TREND (asking about movements of a stock's price), CHART (asking for a graphic display showing recent stock price fluctuations), BUY (seeking to purchase shares of a stock), SELL (seeking to sell shares of a stock), ACCOUNT (asking for the user's portfolio/account information), NOTIFY (setting up the information profile for an alert service), AMEND (amending a previous order of transaction) and CANCEL (canceling a previous order of transaction).

3.2 Core Technologies

ISIS integrates an array of core technologies for speech and language processing. A brief description is provided in the following:

3.2.1 *Trilingual Speech Recognition (SR).* ISIS aims to handle three languages—English, Cantonese and Putonghua (two dialects of Chinese). It integrates an off-the-shelf English speech recognizer as well as two home-grown HMM-based speech recognizers for the Chinese dialects. The Chinese recognizers use acoustic models based on syllable initials (I) and finals (F). Recognition involves a two-pass search—the first creates a syllable lattice and the second traverses the lattice with a language model to produce the output word sequence [Choi et al. 2000].

3.2.2 *Natural Language Understanding (NLU).* The NLU component accepts textual queries derived from typed input, recognized speech,[4] or recognized handwriting. NLU begins with parsing the user's query with a semantic grammar. Since the Chinese language does not have an explicit word delimiter, a Chinese input query in the form of a character sequence is first tokenized into a word sequence by a greedy maximum-match algorithm that references a 1100-word lexicon. Parallel English and Chinese grammars are developed for semantic parsing, and they share a unified set of semantic concepts. Hence parsing identifies a set of semantic concepts in the query and these are fed into a suite of Belief Networks (BN) for task goal inference [Meng et al. 1999]. There are ten BNs in total and each corresponds to a single task goal. Each BN makes a binary decision regarding whether the input query relates to its task goal by generating an *aposteriori* probability that is compared against a threshold value. If all ten BNS vote negative for a given query, it is rejected as out-of-domain (OOD). Alternatively, a query may correspond to a single task goal or multiple task goals. Implementation details of this BN-based NLU framework for ISIS are reported in Meng and Tsui [2000].

Verbalized numeric expressions abound in the ISIS domain—they may correspond to stock codes, stock prices, number of lots or number of shares. This is illustrated by the example query, "*I want three thousand shares of Cheung Kong at one hundred and ten.*" In order to disambiguate among the possible semantic concepts that may correspond to a numeric expression, NLU in ISIS applies a set of *transformation rules.* For example, the transformation rule:

```
<numeric_exp> <share_price> prev_bigram <stock_name> <at>
```

states that a parsed numeric expression (`<numeric_exp>`) should be *transformed* into a share price (`<share_price>`) if it is preceded by the concept bigram (`<stock_name> <at>`). This rule is applicable to our previous example that contains the numeric expression "*one hundred and ten*" and helps label it with the semantic concept of `<share_price>`. The format of the transformation rules resembles the rule templates in Brill [1995] so that we can apply Brill's

[4]Presently the NLU component processes only the top-ranking speech recognition hypothesis. More sophisticated integration techniques between SR and NLU will be pursued as a next step.

transformation-based tagger for part-of-speech (POS) tagging. Brill's work was one of the first attempts to apply transformation-based error-driven learning to natural language processing. His POS tagger first labels every word with its most likely tag and then applies a list of ordered, transformation rules, with a format such as:

```
Change NN to VB when the previous tag is TO
```

(i.e. change the noun to a verb if the word is preceded by the infinitive TO). The most likely tags and transformation rules are automatically derived from annotated corpora. In each iteration, all possible transformation rules that follow a set of rule templates are tried and the one that gives the largest error-correcting improvement is selected. Iterations continue until the incremental improvement falls below a threshold. We have adapted Brill's tagger for transforming semantic tags for semantic disambiguation in handling numeric expressions and out-of-vocabulary words. We found the approach to be directly applicable to tackling the problem of semantic disambiguation.

3.2.3 *Discourse and Dialog.* Discourse inheritance in ISIS uses an electronic form (E-form) for book-keeping [Papineni et al. 1998; Goddeau et al. 1996; Meng et al. 1996]. The information slots in the E-form are derived from the semantic concepts in the NLU grammars. The value of an information attribute obtained from the current user's query over-rides that from the previous query (or queries) in discourse inheritance. A mixed-initiative dialog model is implemented using a turn management table. For example, the turn management table specifies that a query whose task goal is identified to be QUOTE will trigger a response frame labeled QUOTE_RESPONSE in order to use the appropriate template to generate the response text. Additionally, the turn management table specifies the set of information slots that need to be filled before performing each inferred task goal. Should there be necessary but unfilled slots, the dialog model will prompt for missing information. Furthermore, queries inferred to be important transactions, for example, BUY and SELL requests, will trigger a subdialog to confirm all details regarding the transaction prior to order execution. ISIS also has a list of meta-commands, for example, `help`, `undo`, `refresh`,[5] `good-bye`, and so forth, to enable the user to navigate freely in the dialog space. The dialog manager (DM) maintains the general flow control through the sequential process of speech recognition, natural language understanding, database access (for real-time data captured from a direct Reuters satellite downlink) and response generation in synthetic speech, text, and graphics. Speaker authentication is invoked at dialog turns that involve access to account information or execution of a transaction: when the inferred task goals are BUY, SELL, ACCOUNT, AMEND, or CANCEL.

3.2.4 *Spoken Response Generation.* Spoken responses in ISIS need to be generated for three languages. Language generation utilizes a response frame.

[5]The meta-command `undo` deletes the discourse inherited from the immediate past spoken query and is especially useful if the query has speech recognition errors. The command `refresh` deletes the entire inherited discourse to let the dialog start afresh.

The frame specifies the language to be generated, the response type (e.g., QUOTE_RESPONSE), associated information attributes (e.g., `stock_name`, `bid price`, `ask price`, etc.) and their values as obtained from NLU or database access. The response text thus generated is sent to a text-to-speech (TTS) synthesizer. ISIS integrates the downloadable version of the FESTIVAL system [Taylor et al. 1998] for English synthesis. Putonghua synthesis uses Microsoft's SAPI engine (www.microsoft.com/speech/) and Cantonese synthesis uses the homegrown CU VOCAL engine Meng et al. [2002b]. The synthesis quality of CU VOCAL can be optimized for constrained application domains to generate highly natural and intelligible Cantonese speech. This process of domain-optimization has been demonstrated to be portable across Chinese dialects [Fung and Meng 2000].

3.2.5 *Speaker Authentication (SA).* Speaker authentication aims to automatically verify the speaker's claimed identity by his/her voice prior to enabling access to personal financial information and execution of transaction orders. The current SA component is a *text-dependent* system—the system randomly generates a digit string for the speaker to utter. SA uses Gaussian Mixture Models (GMM) [Reynolds 1992] with 16 mixture components. Authentication involves computing the likelihood ratio between the claimed speaker's model and the "world" model. This likelihood ratio is compared with a pre-set threshold, and hypothesis testing is applied to accept or reject the claimant for the purpose of user authentication.

3.3 System Architecture—CORBA

ISIS is a distributed system with a client/server architecture implemented on CORBA (Common Object Request Broker Architecture). CORBA is a middleware with specifications produced by the Object Management Group (OMG) and aims to provide ease and flexibility in system integration. The core technologies described in the previous subsection, for example, SR, NLU, SA, and so on, reside in different server objects (see Figure 1). In addition, there is a server object for tracking timeouts[6] and another encapsulating two software agents responsible for message alerts. The client object is illustrated in Figure 2. CORBA provides the Object Request Broker (ORB) that handles communication between objects, including object location, request routing, and result returning. As illustrated in Figure 1, some server objects are implemented in Java or C on UNIX, while others are in Visual C++ on Windows NT. These server objects extend the stubs/skeletons (i.e. the glue to the ORB from the client/server) to the core speech and language engines. CORBA offers *interoperability* by the Interface Definition Language (IDL) to communicate with the variety of programming languages running on different operating systems. CORBA also offers *location transparency* in that only the names of the server objects need to be known for two-way (sending/receiving) communication and no explicit host/port information is needed. A third advantage of CORBA is *scalability*—a new object

[6]The timeout manager (TM) monitors the time between successive user inputs. If the time duration exceeds a pre-set threshold, TM sends a message to the dialog manager which in turn invokes the response generator to produce system responses such as "*are you there*?"

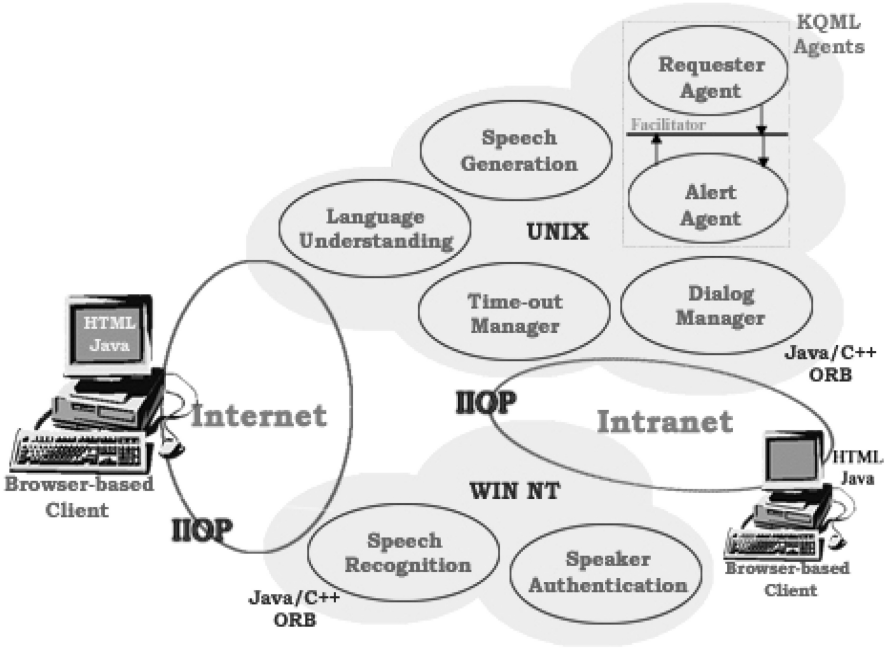

Fig. 1. The ISIS architecture.

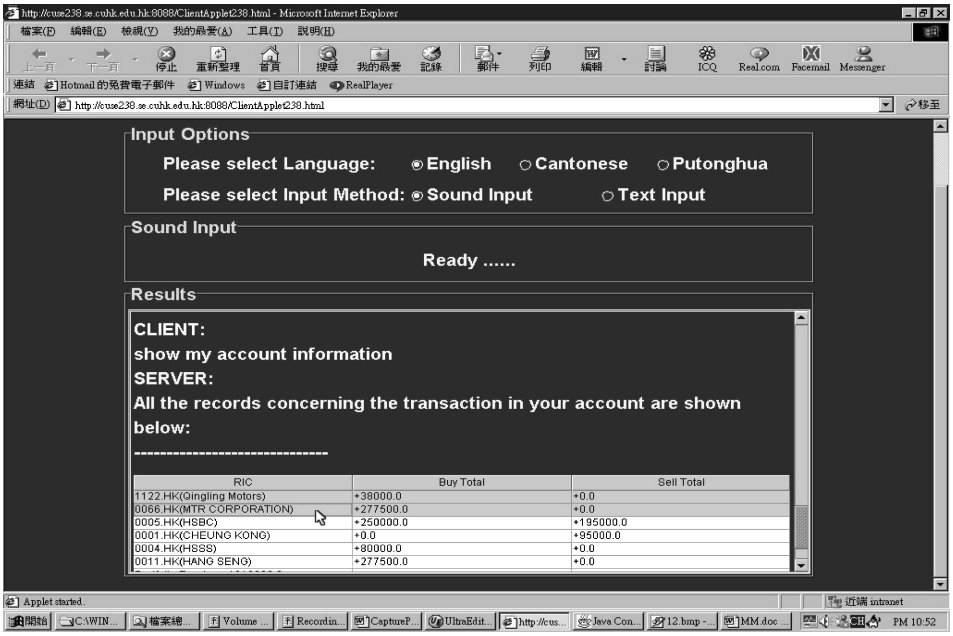

Fig. 2. Screenshot of the client that provides options for language selection and input mode selection (text or voice). Should `Text Input` be selected, a text box will appear and the user can input text by typing or stylus-writing. The client object can also capture click events to support deictic expressions. Clicked items in the table are highlighted.

```
<NLU>
      <goal> buy </goal>
      <ric> 0005.HK </ric>
      <num_lots> 5 </num_lots>
      <price> market </price>
</NLU>
```

Fig. 3. Example of an XML message produced by the NLU server object. The user request was "*Buy five lots of HSBC at the market price please*." "ric" denotes Reuters Instrument Code.

can be added to the system simply by adding its corresponding definition to the IDL, followed by recompilation.

Data are passed among the client and server objects in XML (Extensible Markup Language).[7] The data are labeled with descriptive tags characterizing the server class operation(s). For example, Figure 3 shows the XML output of the NLU component after processing a user request to purchase some stocks. Java applets in the client object can also capture mouse-clicks as events in order to support deictic expressions (see Figure 2). Client methods decode the semantic meaning of the mouse-clicks (e.g., the stock name of the table row that has been clicked) and send the information through CORBA to the NLU server object for subsequent processing.

A reference implementation based on the ISIS system, including software, documentation, and an online demo is available at www.se.cuhk.edu.hk/∼isis/download. Other reference architectures are also available, for example, the GALAXY system [Bayer et al. 2001] that has a hub interacting with several distributed information servers via a hub script that specifies server hosts, ports and functionalities, and the Open Agent Architecture (OAA) [Cohen et al. 1994] that executes the user's request in a distributed manner, by a blackboard server interacting with a community of cooperative software agents running on various machines.

3.4 An Example Dialog in ISIS

The initial prototype of the ISIS system has been developed and subjects have been invited to experiment with the system. Table III shows an example of a typical ISIS dialog and serves to illustrate some of the system's functionalities. A description is included for most dialog turns to explain the underlying mechanisms that support the interaction.

4. ADAPTIVITY TO KNOWLEDGE SCOPE EXPANSION

As mentioned in the introduction, most conversational systems are designed with a *closed* knowledge scope—speech recognizers and language models typically have fixed and predetermined vocabularies; NLU components typically have grammars with a fixed set of rules, categories, preterminals and terminals; and text-to-speech (TTS) synthesis systems typically reference pronunciation dictionaries with a fixed number of entries. Conversely, the knowledge scope of application domains tend to *grow* over time—new companies continue to be

[7]http://w3C.org/XML.

Table III. An Example Dialog from the ISIS System

User	User logs into ISIS by typing in his account code.
ISIS	"*Welcome to ISIS. Please input your query*".
User	"*I would like the exchange rate of the Singapore dollar please*."
ISIS	"*Sorry, I don't understand your request. I can provide only information about stocks such as real-time quotes, news and charts. I can also provide information about a simulated personal portfolio, and handle buy and sell transactions.*" Description: ISIS rejects an out-of-domain query about foreign exchange.)
User	"*I would like to buy three lots of Cheung Kong Holdings please*."
ISIS	"*Please provide the bid price*." Description: Natural language understanding (NLU) infers that the task goal is BUY. Domain-specific constraints show that BUY requires three attributes: the stock name, the number of lots/shares and a bid price. The last one is missing and hence the system prompts for it.
User	我想先看一看每日的走勢圖 Translation: *Can you show me the daily chart first* Description: The user did not answer to the system's prompt for a bid price. The discourse and dialog components offer this flexibility.
ISIS	0001長實請睇 Translation: *0001 (a stock code) Cheung Kong Holdings Ltd. Please check (daily chart shown).* Description: NLU infers the task goal is CHART. Domain-specific constraints in the turn management table show that this requires a stock name and a chart type. The former is missing from the query but is inherited from discourse. The daily chart is shown on the graphical user interface.
User	"*Buy three lots at eight nine dollars please.*"
ISIS	"*Please say the following digits in English to verify your identity: 1 3 7 6 4*." Description: NLU infers that all the necessary attributes for a BUY request have been provided. The Dialog Manager thus invokes speaker authentication (SA) before executing the BUY request.
User	"*One, three, seven, six, four*"
ISIS	"*Please confirm the following action: buy three lots of Cheung Kong Holdings Limited at eight-nine dollars. Please say yes to confirm or no to reject to action*." (SA verified the user's identity and allowed the user to proceed with the BUY request. Transactions are always followed by a confirmation subdialog.)
User	"*Yes*."
ISIS	"*The buy transaction is completed. Please input a new query*."
User	"*Show me the monthly chart of HSBC*."
ISIS	"*The monthly chart of HSBC is shown below*." (graphic shown)
User	"*Bye*."
ISIS	"*Goodbye and see you later*." Description: Bye is a meta-command which can be used to terminate the session at any time.

listed at stock exchanges; new names of people and addresses are added to directory listings; and new products and services may become available. The widening knowledge gap between the conversational interface and the application domain creates an impending need for *adaptivity* in spoken dialog systems.

A direct impact of knowledge scope expansion is the out-of-vocabulary (OOV) problem. New information that needs to be acquired by an *adaptive* interface inevitably involves new words that do not exist in the original vocabularies of the recognizers and their language models, the original set of grammar terminals

in the NLU component, or the original pronunciation dictionaries of the TTS synthesizers. As mentioned earlier, previous work has addressed the OOV problem in the context of SR, language modeling, and letter-to-sound/sound-to-letter generation to augment existing vocabularies with spellings and pronunciations for recognition and synthesis. The current work takes on a slightly different focus in exploring adaptivity in the ISIS dialogs—a newly listed stock (or OOV) detected in an input query will trigger a *subdialog* that seeks to elicit information about the new word from the user and automatically incorporates the word into the ISIS knowledge base. The new stock name may appear as a full name, an abbreviated name, or an acronym (in the case of English). It may also be input in spoken form via speech recognition, or in textual form via typing or stylus-writing on mobile PDAs or portable computers. The current work on OOV handling in ISIS is directly applicable to textual input and spoken English acronyms (which can be recognized by the English recognizer). As mentioned in Chung et al. [2003], OOV handling involves acquisition of the spelling and pronunciation of the new word/name as well as its linguistic usage such as semantic category. We focus on the latter aspect in this work, handling primarily new stock names and temporarily bypassing the problem of OOV spelling/pronunciation acquisition in speech recognition.[8] However, as will be explained later, the method that we use in handling Chinese OOV (c.f. *n-gram grouping*) is conducive to speech recognition of Chinese OOV. This is by virtue of the syllabic nature of the Chinese language, where every written character is pronounced as a spoken syllable.

Automatic incorporation of new stock names in ISIS involves two steps: (i) detecting new stock names and (ii) invoking the subdialog for new stock acquisition. Details are provided as follows:

4.1 Detecting New Stock Names

The detection process takes place in the NLU component. A new stock name is tagged with <oov>. For example, the Artel Group was listed during the time of development at the Stock Exchange of Hong Kong, with a stock code of 0931. Some refer to the company as "Artel". Since the listing was new, the name did not exist in the original ISIS knowledge base. An input query such as "*I'd like to check the news about Artel*" will first undergo semantic parsing in the NLU component to yield the semantic sequence:

Semantic sequence from parser: <dummy> <check> <news> <about> <oov>.

The semantic label <dummy> is used to absorb an arbitrarily long text string while parsing the input query. Such a text string may be a grammatically ill-formed structure, a transcription of spoken disfluencies, or a series of words that does not carry significant semantic content captured by the other semantic labels.

[8]Automatic conversion of a spoken waveform with an unknown word to a grapheme sequence is a problem that merits a focused and dedicated effort. This is a relatively new problem with a few initial attempts for languages including English [Chung et al. 2003], German [Schillo et al. 2000] and Dutch [Decadt et al. 2002].

A set of transformation rules has been written for the purpose of determining whether an OOV word corresponds to a new stock name. The technique is similar to that used for disambiguation among multiple possible semantic categories that correspond to numeric expressions (see Section 3.2.2). The transformation rule that is applicable to the semantic sequence above is:

Transformation rule : `<oov> <stock_name_oov> prev_bigram <news> <about>.`

This rule states that the concept label `<oov>` should be transformed into `<stock_name_oov>` if it is preceded by the concept bigram (`<news> <about>`). Hence the *transformed semantic sequence* becomes:

`<dummy> <check> <news> <about> <stock_name_oov>.`

As a consequence, the OOV "Artel" in the example query is labeled as a new stock name, i.e., `<stock_name_oov>`.

Detection of new stock names in Chinese queries largely follows a similar procedure, except for a procedure known as *n-gram grouping*. Consider another recently listed stock 中國聯通 (China Unicom) that appeared in an input query:

Input Chinese Query: 我想知道中國聯通既股價

(Translation: *I'd like to know China Unicom's stock price*.)
Since this stock name does not exist in the original vocabulary, Chinese word segmentation fails to identify it as a word and tokenizes it into individual characters:

Input Chinese Query after word tokenization: | 我想知道 | 中 | 國 | 聯 | 通 | 既 | 股價 |

Semantic Parser Output:

```
<ask>我想知道</ask>
        <singleton>中</singleton>
        <singleton>國</singleton>
        <singleton>聯 </singleton>
        <singleton>通</singleton>
<particle>既</particle>
<price>股價</price>
```

An *n-gram grouping technique* checks for sequences of contiguous character unigrams and bigrams, which may be indicative of the occurrence of an OOV. These n-grams are then grouped together. To follow up with the example, we obtain:

N-gram Grouping Output:

```
<ask>我想知道</ask>
        <oov>中國聯通</oov>
<particle>既</particle>
<price>股價</price>
```

Hence the semantic sequence from parsing is: `<ask> <oov> <price>.`

Table IV. Example Illustrating the Automatic Acquisition of a New Stock Name in ISIS Through a Spoken Dialog Between the User and the System

User1	*Do you have the real-time quotes of Artel?*
ISIS1	*I do not know of the stock listed as Artel. 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."* ------ 0931.HK – Artel Group 1229.HK – HK-Artfield Group . . . etc.
User2	*It should be zero nine three one.*
ISIS2	*Here are the real-time quotes for Artel Group*. "<quotes shown>" *Do you generally refer to Artel Group as Artel? Please say yes or no.*
User3	*Yes*
ISIS3	*I got it. How else may I help you?*
User4	*Show me the daily chart of Artel please.*
ISIS4	*This is the daily chart of Artel Group.* <chart shown>

The transformation rule applicable to this semantic sequence is:

Transformation rule : `<oov> <stock_name_oov> prevtag <ask> nexttag <price>.`

The rule states that the concept label `<oov>` should be transformed into `<stock_name_oov>` if it is preceded by the tag`<ask>` and followed by the tag `<price>`. Hence the *transformed semantic sequence* becomes:

`<ask> <oov> <price>.`

Consequently the new stock name 中國聯通 (China Unicom) has been detected. It should be noted that the n-gram grouping technique used here is conducive to pronunciation acquisition of new stock names in Chinese speech recognition. This is because the Chinese language is syllabic in nature and each written character is pronounced as a spoken syllable. Since the Chinese speech recognizer used in ISIS is syllable-based, it is possible to map the character sequence tagged with `<stock_name_oov>` to the corresponding syllable sequence transcribed by the speech recognizer to derive a hypothesized pronunciation for the new Chinese stock name.

4.2 Invoking Subdialog for New Stock Incorporation

Detection of a new stock name causes the dialog manager (DM) to invoke a special subdialog that aims to incorporate the new stock into the ISIS knowledge base. The DM begins by triggering a substring match for all possible candidate listings that correspond to the new stock name. For example, the letter string in *Artel* can match "*Artel Group*" and "*Hong Kong Artfield Group*", and so forth. Similarly, the abbreviation 中聯 (for China Unicom) can match 中國聯通 since the characters appear in the same order. The list of possible candidate listings is presented to the user onscreen for selection. Better matches (according to the edit distance) are ranked higher on the list (please see the dialog turn labeled ISIS1 in Table IV).

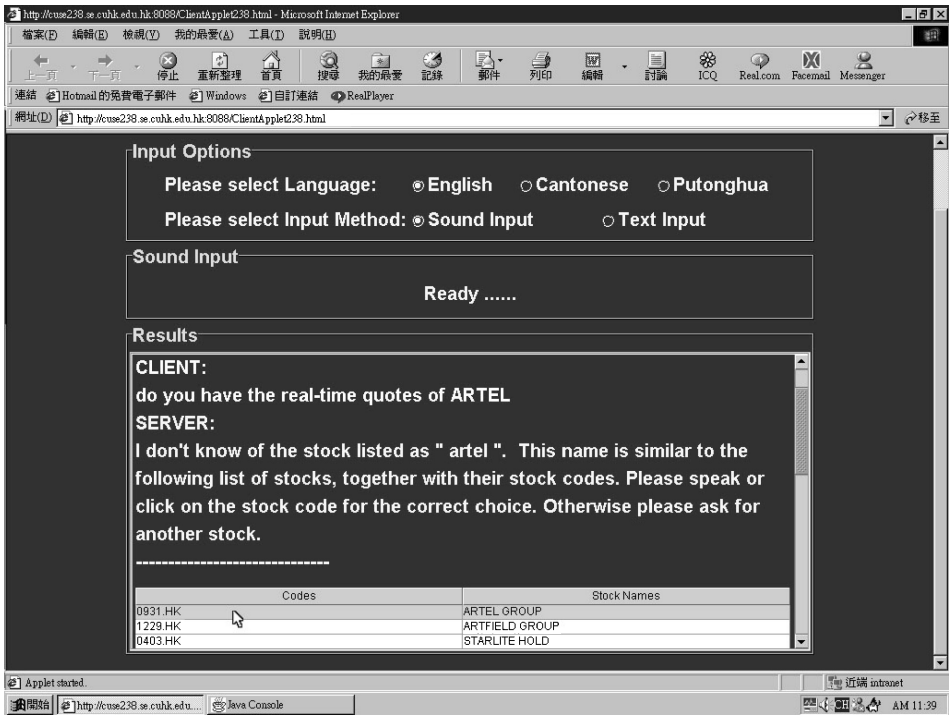

Fig. 4. Screenshot of the ISIS client object presenting information in response to the user's query "*Do you have the real-time quotes of Artel?*" *Artel* is a new stock that did not exist in the original ISIS knowledge base.

Spoken response generation can pronounce the new English stock names by means of letter-to-sound rules. Since the Chinese synthesizers have full coverage of single character pronunciations, new Chinese stock names can be spoken (character by character) in the ISIS system responses.

One may notice from ISIS1 in Table IV that in order to select an item from a short list of new stock names, the user is constrained by the system to use either the stock code or an ordinal reference, for example, *the first one, the next one*, and so on. This is because new stock names contain OOV words that are problematic for speech recognition, but all digits in stock codes are in-vocabulary. However, constraining the user to utter only digits is cumbersome from a usability perspective.[9] This motivated us to enhance the ISIS client object to capture mouse-clicks as events by using Java applets and Java classes. With this small addition of multi-modal capability, deictic expressions that are spoken (e.g., "*It should be this one*") can be augmented by clicking. This is illustrated in Figure 4, where the user has clicked on the selected entry. The mouse action is captured

[9]This constraint may be relaxed in the future by invoking spelling-to-pronunciation generation using letter-to-sound rules and dynamically adding the listed stock names into the recognizers' vocabulary. We have recently been successful in using this method to develop speakable hyperlinks in Chinese speech-based Web browsing.

and the contents of the table are retrieved—this stock information is packaged in an XML message in the NLU component and processed as usual.

Upon completion of this subdialog, the user has confirmed the correct listing for the new stock name. ISIS continues to elicit the preferred ways in which the user will refer to this new stock, including the full name and its abbreviations, for example, *Artel Group, Artel*, and so forth. Subsequently, two new grammar rules are automatically added to the NLU server:

```
stock_name → Artel (Group)
stock_code → 0931.HK.
```

Hereafter the new stock is assimilated in the expanded ISIS knowledge base and the user may refer to the newly listed company by its stock name or stock code.

5. DELEGATION TO SOFTWARE AGENTS

The financial domain has an abundance of time-critical dynamic information; users of financial information systems often have to closely monitor for specific changes in various real-time information feeds, for example, when a target bid/ask price is reached, when the increment/decrement of a financial index exceeds a threshold, and so on. The cognitive load of such users can be greatly reduced if the task of monitoring information may be delegated to software agents. This work involves our initial effort in exploring *asynchronous* human-computer interaction by delegation to software agents within the context of ISIS. The interaction is asynchronous because the human and the computer no longer take alternate dialog turns in completing a task. Instead, the human verbally specifies the task constraints to software agents that will perform the task in the background (i.e. offline). ISIS supports two types of requests for alerts. The first kind is explicit, such as "*Please notify me when HSBC rises above ninety dollars per share*." When the prescribed condition is met at some indeterminate time in the future, the agents will return a notification message to the user. The second kind is implicit—if the user places a buy/sell order with a requested price that differs from the market price, ISIS will inform the user of the difference, offer to launch software agents to monitor the market price and alert the user when the price reaches the requested level. Agent-based software engineering in ISIS uses the Knowledge Query and Manipulation Language (KQML) that provides a core set of speech acts (also known as *performatives*) for interagent communication. Details about the KQML implementation will be presented in the subsections that follow. Alternative foundation technologies also exist for agent-based software engineering. Examples include the use of speech acts with Horn clause predicates for interagent communication in the Open Agent Architecture (OAA)[10] mentioned earlier, as well as the Belief, Desire and Intention (BDI) paradigm [Rao and Georgeff 1995] for structuring the content of the agents' messages in terms of the informational, motivational, and deliberative states of the agent.

[10]www.ai.sri.com.

```
<DM>
   <trigger lang =English status =launchAgent>
   <user_id>005</user_id>
   <action>buy</action> <market>95.0</market>
   <lots>3</lots>
   <stock>0005.HK</stock>
   <share>---</share>
   <price>89</price>
   <time_stamp>Aug_02_2002_14:00:38</time_stamp> </trigger>
</DM>
```

Fig. 5. An example XML message sent by the dialog manager (DM) server object to the requester agent.

5.1 Knowledge Query and Manipulation Language (KQML)

The software agents in ISIS are implemented in KQML. The language is produced by the DARPA Knowledge Sharing Effort.[11] The use of KQML enables implementation of a multi-agent communication feature in ISIS with relatively simple software coding. KQML is both a message format and a message-handling protocol to support run-time information exchange and knowledge sharing among software agents [Finin and Friitzso 1994). It has a three-tiered structure—the outermost *communications* layer specifies the sender and receiver agents; the middle *speech act (or performative)* layer defines the kinds of interactions one may have with the agent; and the innermost layer embeds the *content* of the message. KQML mediates information exchange by an agent called the *facilitator*, which is a software substrate with agent registry that enables the agents to locate one another in a distributed environment.

5.2 Multi-Agent Communication in ISIS

The ISIS implementation uses JKQML (i.e. Java-based KQML). There are three software agents in all—the *requester*, *facilitator*, and *alert* agents. If the user's requested transaction (e.g. "*Buy three lots of HSBC at eighty nine dollars please*") cannot go through due to a mismatch between the requested and market prices, ISIS will trigger the offline delegation procedures. First, a non-blocking XML message (see Figure 5) is sent from the dialog manager (DM) server object to the requester agent. This message encodes the information attribute-value pairs related to the requested transaction.

The requester agent receives this XML message, decodes it and transmits a corresponding KQML message (see Figure 6) through the facilitator agent to the alert agent. The facilitator agent serves to locate the alert agent since the facilitator is an agent registry. In Figure 6, `ask_all` is a performative (speech act) to request for service from all agents. The :`sender` and :`receiver` fields

[11] http://www.cs.umbc.edu/kse/.

```
(ask-all
     :sender requester agent
     :receiver alert agent
     :reply-with messageID
     :language xml
     :content (theaction:buy theuser_id:005 theprice:89
               thetimestamp: Aug 02 2002 14:00:38
               theric:0005.HK   thelot:3    theshare:---))
```

Fig. 6. An example KQML message sent by the Requester Agent to the Facilitator.

```
(tell
     :sender alert agent
     :receiver interpreter
     :reply-with messageID
     :language xml
     :content (theaction:buy  theuser_id:005   themarket:89
               theprice:89
               thetimestamp: Aug 02 2002 14:00:38
               theric:0005.HK   thelot:3    theshare:---
               theresponse: alert!))
```

Fig. 7. An example KQML message sent by the Alert agent to the Facilitator agent.

constitute the communications layer. The :language field specifies the format of the :content parameter. The alert agent receives the KQML message, interprets it and inserts it by SQL into a relational database storing similar requests. The alert agent also keeps track of the alert conditions specified in the database and monitors the real-time data-feed accordingly. The data-feed is a direct satellite downlink from Reuters received via a satellite dish mounted on the roof of our engineering building. If the previously specified condition is met (i.e. HSBC's market price hits 89 dollars per share), the alert agent will send a KQML message (see Figure 7) through the facilitator to alert the requester agent. The performative tell is the expected response to ask_all. In the final step, the requester agent returns a KQML alert message (see Figure 8) to the dialog manager server object. This entire process of multi-agent communication is depicted in Figure 9.

The ISIS system demonstrated that the KQML-based implementation software agents for delegation dialogs, in coordination with the core system implemented with CORBA, is able to keep up with the dynamics of financial information. Hence the client is able to receive alert messages as soon as the pre-specified alert levels of various stock prices are met, without any noticeable delays.

```
<kqml_agent>
     <market>89</market>
     <time_stamp>Aug_02_2002_15:37:02</time_stamp>
     … (other parameters identical to those in Figure 4)
</kqml_agent>
```

Fig. 8. An example XML message sent by the Requester agent to the dialog manager (DM) server object.

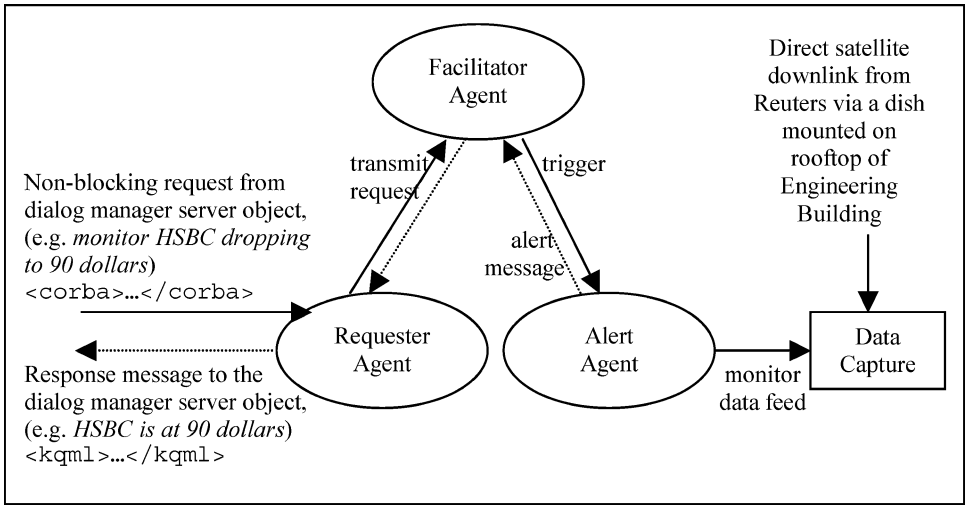

Fig. 9. Multi-agent communication in ISIS. The messages communicated between the agents and the dialog manager (DM) server object are in XML format and wrapped with the indicator tags—the tag <corba> is used if the message originates from the DM; and the tag <kqml> is used if the message originates from the agents.

6. INTERRUPTIONS AND MULTI-THREADED DIALOGS

Delegational interactions involving software agents will generate alert messages that will subsequently interrupt an ongoing dialog—*online interaction* (OI)—between the user and the ISIS system. This raises the problem of maintaining multi-threaded dialogs that correspond to: (i) the current (sub-)task that is in focus in the online interaction, (ii) other concurrent (sub-)tasks being pursued but which are temporarily not in focus; and (iii) one or more pending notifications delivered by the software agents. The user may choose to stop his workflow in the online interaction, bring an alert message into focus to handle the pending task, and then return to the original interrupted workflow. This section reports on an initial investigation of interruptions and multi-threaded dialogs in the context of ISIS. The scope of investigation is constrained to the task structure of the ISIS domain (as described in Section 3.1). The task structure is relatively simple and an intended action may easily be decomposed into a sequence of simple tasks. For example, if the user is considering purchasing the stocks of a particular company, he may seek to obtain the real-time

stock quotation, observe its fluctuation in a chart, check the news of competing companies in the same industry, and finally decide on placing a buy order. As can be seen, multiple tasks pertaining to the user's intent can be handled *sequentially* in the online interaction (OI) dialog in ISIS. Similarly, multiple alert messages resulting from offline interaction (OD) can also be handled one at a time. The situation is more complex in real applications—for example, the user may be checking his appointment calendar while placing a purchase order with ISIS and this already involves concurrent tasks in the OI interaction dialog. Alternatively, multiple alert messages of equally high priority may arrive simultaneously and require immediate attention from the user. Such situations will require a complicated interruption framework. However, as a first step this work aims to identify the necessary mechanisms involved in the simplified interruptions within the context of ISIS. As will be seen, this work draws ideas from previous studies in interruptions and discourse structures.

6.1 Providing Awareness of Alert Messages

As mentioned earlier, there are two types of alert messages that may be delivered by the ISIS software agents. The first kind originates from an explicit request for notification, for example, "*Please notify me when HSBC falls below eighty dollars*." The second kind relates to a previously placed buy/sell order in which the target price and market price differ. Hence the software agent monitors the fluctuating stock price for the requested target and sends a buy/sell reminder when the target price is hit. Alert messages are queued upon arrival to wait for the user's retrieval. This subsection presents interface design considerations in ISIS for the purpose of providing awareness to the user regarding the pending alert(s), avoiding (costly) disruptions to the ongoing task and enabling the user to stop his/her work flow at an instant that he/she deems convenient for switching tasks to handle the alert message(s).

We reference the rich research findings in previous work on designing displays for *ambient* information [Ishii and Ullmer 1997], also known as *peripheral* information [Maglio and Campbell 2000], which presents the challenge of "maximizing information delivery while minimizing intrusiveness". Ambient/peripheral information is not central to the user's ongoing task but provides (important) information regarding other tasks that need to time-share the user's focus and attention. A study by Lim and Wogalter [2000] showed that information positioned centrally or at the top left or bottom right corners tend to better capture the user's attention. Additional design strategies (such as animation, highlighting and color coding) may be used to raise awareness of peripheral information, depending on the relative importance between the interrupted and interrupting tasks. Czerwinski et al. [2000] presented a detailed look at the phase of the ongoing task at the time of interruption and found that some phases are less amenable to interruptions. Consequently, interruptions during these phases are greater detriments to task performance. McFarlane [1999] suggested that such detrimental effects to task performance can be markedly reduced for "negotiated interruptions", where users are given control over when to handle the interruption. Horvitz and Barry [1995] developed a Bayesian model that

Fig. 10. The "Notify" icon indicates the arrival of a notification message.

captures the user's beliefs to evaluate the *value* of various messages that are competing for the user's attention. The model is applied to time-critical display management. van Dantzich et al. [2002] designed a display that can accommodate a large number of notifications within a glance while trying to safeguard the user's attention on the primary task. The circular display is sectored according to the type of notification messages and urgent messages are located closer to the center.

In the ISIS implementation, a dedicated region on the screen display is reserved for icons that indicate the arrival of alert messages. The icon that corresponds to a target price notification (i.e. the first type of alert mentioned above) will cause an icon labeled "Notify" to be placed on screen (see Figure 10). Similarly, the arrival of a buy/sell notification (i.e. the second type of alert mentioned above) will place the icon "Buy/Sell" on screen (see Figure 11). It is possible for both icons to appear simultaneously. The user's eye gaze tends to scan from left to right in the box delineated as "Text Input" (i.e. the text box for typing or stylus-writing) or "Sound Input" (i.e. where the speech recognizer's transcription will appear). By placing the notification icons to the right of this box, the ISIS screen design intends to minimize disruptions to the ongoing workflow: the user's eye-gaze will reach the icon after inputting a request. The notification icons are also moderately close to the center of the screen in an attempt to provide sufficient awareness for the user. As will be elaborated later, the user chooses to handle an alert message by clicking on the notification icons. Hence

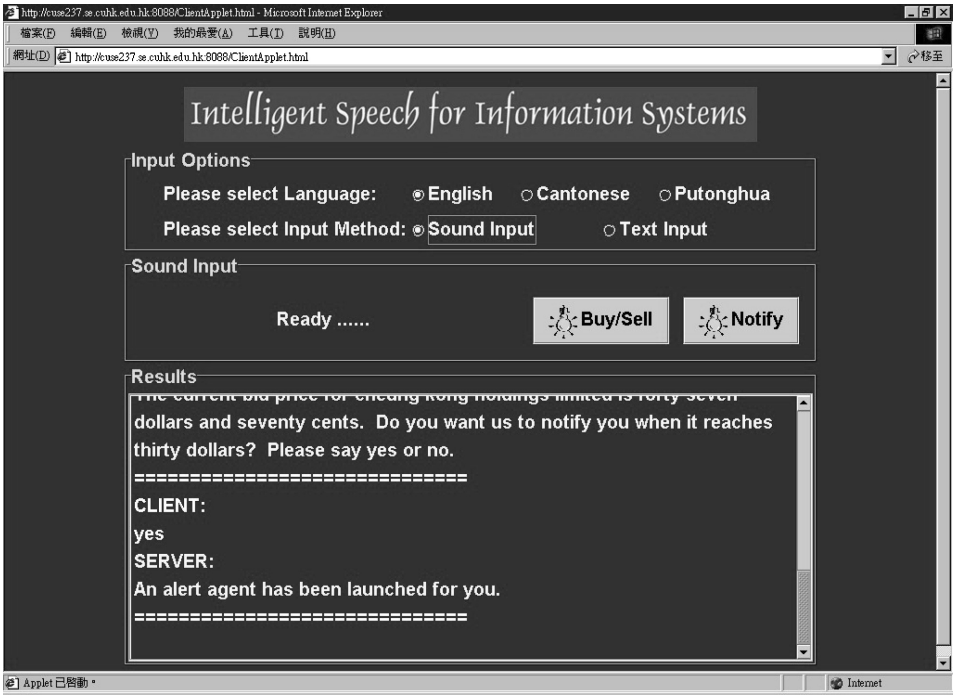

Fig. 11. The presence of both the "Notify" and "Buy/Sell" icons indicate that there are pending alert messages in the queue including notification message(s) and buy/sell reminders.

he/she has full control of when interruptions take place. Alert messages are queued chronologically as they arrive. It is conceivable that if a large volume of alerts are set up by the user, one may need to implement solutions similar to Horvitz and Barry [1995] in evaluating the relative values of the alerts (e.g., according to their financial implications). These values may be used for prioritization and ordering in the queue and managing the notification display similar to van Dantzich et al. [2002]. However, this is beyond the scale and scope of the ISIS system at its present initial stage.

6.2 Interruptions of Online Interaction (OI) Dialogs by Offline Delegation (OD) Alerts

When icons appear to indicate the arrival of alert messages waiting in the queue, the user has the option of clicking on the icons at a convenient time to interrupt the online workflow and bring the alert message(s) into focus to handle the associated pending task(s). Alert messages relating to the icon "Notify" present less interruption to the user's workflow, since a click on the icon triggers delivery of the alert and thereafter the user can immediately resume the original workflow. Alert messages relating to the icon "Buy/Sell" are more complex since they correspond to transactions. Handling a buy/sell reminder involves bringing the transaction into the focus of attention, reinstating the transaction's information attributes and values, allowing the user to make necessary

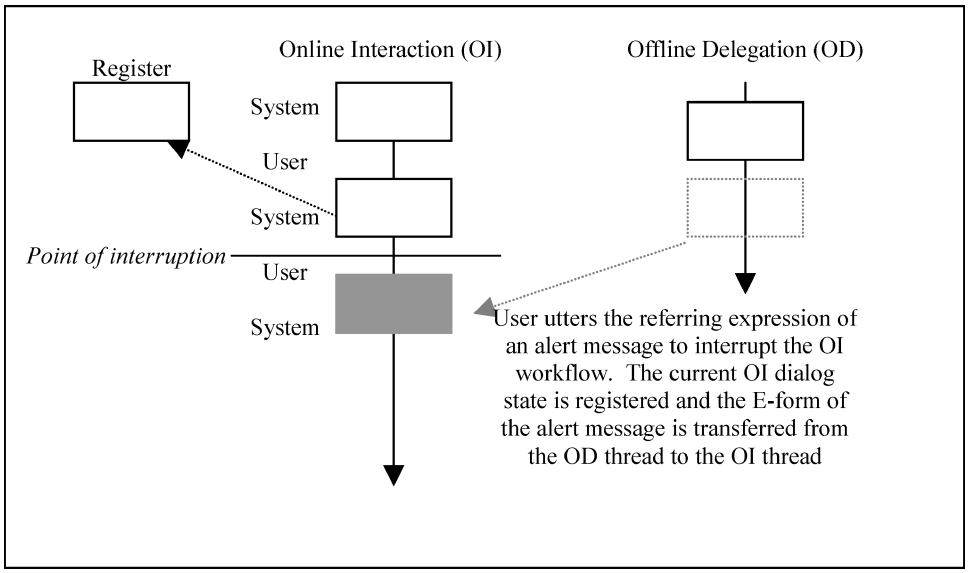

Fig. 12. Structures and mechanisms supporting interruptions in ISIS. Online interaction (OI) and offline delegation (OD) are managed as separate dialog threads. The OI thread stores E-forms in chronological order and the timeline goes downwards. The user and system take alternate dialog turns in the mixed-initiative interactions. Alert messages delivered by software agents queue at the offline delegation dialog thread. The user may interrupt the OI workflow by uttering the referring expression to the alert message. This causes the current OI dialog state to be stored at the register and the alert E-form to be moved from the OD thread to the OI focus.

adjustments to the attributes and invoking the confirmation subdialog for transaction orders. These steps require a series of dialog turns before the user can revert back to the interrupted workflow. Consequently the "Buy/Sell" reminders are more disruptive than the "Notify" alerts. This section describes the underlying structures and mechanisms in ISIS that support task interruption, task switching and task resumption.

ISIS maintains two dialog threads that connect a chronological series of E-forms (or semantic frames), one for OI and the other for OD (see Figure 12). The OI thread maintains discourse of online interactions in which the user and the system take alternate dialog turns in mixed-initiative interactions. The task in focus is represented by the latest E-form in the OI thread. The OD thread stores a queue of alert messages received from the software agents. Each alert message uses its stock name as a default referring expression to itself. Hence in order to interrupt the ongoing workflow and switch to handling an alert message, the user can simply input (or utter) the stock name of the alert message. At this point, the latest E-form on the OI dialog thread that transpired immediately before the interruption is moved into the register (also illustrated in Figure 12), and the E-form of the selected alert message is moved from the OD thread into the focus of the OI thread. Discourse in the E-forms is not inherited across the point interruption such that the information attributes and values corresponding to the original task and the alerted task are kept separately.

Table V. Example Dialog Illustrating the Interruption of an Online Interaction Dialog by an Offline Delegation Alert

User1	*Show me the news of Cheung Kong*
ISIS1	*There is no news for Cheung Kong today* (Buy/Sell icon appears)
User2	(User clicks on the Buy/Sell icon)
ISIS2	*There is one alert message for you regarding a previous buy request on HSBC. If you wish to handle the alert message now, please say HSBC. Otherwise please continue*
User3	*HSBC*
ISIS3	*The stock price of HSBC is at ninety-eight dollars presently. Please confirm your previous request from May 20, 2002 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.*
User4	*Yes"*
ISIS4	*The buy transaction is completed. Please input a new query.*
User5	*Let's go back*
ISIS5	*"Previously you requested to see the past news of Cheung Kong Holdings but there is no news for Cheung Kong today. How else may I help you?"*

Having processed an alert message, the user may choose to handle another alert message in the queue (again by uttering its referring expression), or to revert to the original interrupted task by the command, "*Please go back*." At this point, ISIS restores the registered E-form by moving it back into focus along the OI dialog thread. The original task is thus reinstated and resumes in the OI dialog. Table V shows an example dialog that illustrates interleaving OI and OD interactions.

Previous work by Grosz and Sidner [1985] defines a general computational structure of discourse. The model is a composite of three interacting components—the structure of the linguistic expressions, the structure of intentions, and the attentional state. Linguistic structure involves cue words, phrases, and referring expressions. Intentional structure involves the definition of *discourse purpose* (DP): the objective of the discourse. It also involves the definition of *discourse segments* (DS), each of which has a single intention known as *discourse segment purpose* (DSP). Attentional structure involves a *focus stack* that organizes focus spaces, each of which is assigned to a DP or DSP. The order of stacking reflects the relative salience of the focus spaces. The current ISIS implementation, within the confines of its domain, verifies the necessity of many of the model components of the Grosz and Sidner model. As described previously, it was necessary to provide a handler to every alert message by means of a referring expression; the stock name is used for this purpose. The user's sequence of intended tasks may be gleaned from the E-forms in the OI dialog thread with the latest E-form being the focus of attention. The *register* in the ISIS implementation serves to remember the discourse segment and its purpose (DS or DSP) prior to an interruption. This data structure suffices, as the current cases of interruptions are short and simple (see Section 6.1). However, more complex interruptions are conceivable. For example, the user may be distracted by a later interruption while he/she is processing an earlier one. To handle such multiple alerts will require a more sophisticated register,

for example, in the form of a stack that can store multiple discourse segments. The type of interruption presented in ISIS belongs to the category of *true interruptions* in the Grosz and Sidner [1985] model. True interruptions are kept as separate discourses, just as separate E-forms are kept for the interrupted task in the OI dialog thread and the interrupting task from the OD dialog thread, and there is no information sharing between the E-forms.

7. EMPIRICAL OBSERVATIONS ON USER INTERACTIONS WITH ISIS

We conducted a small-scale experiment involving 15 subjects. The purpose of the experiment was to observe how users interact with the end-to-end ISIS system using dialogs that involve interruptions. The subjects were technical or research staff members who were familiar with the use of computers. They were given a brief verbal description of the capabilities of ISIS with reference to the system overview diagram (see Figure 1), an explanation of the meta-commands (listed in Section 3.2.3) and a three-minute demonstration video on the project's website.[12] The video illustrates a typical interaction session between a user and the ISIS system by providing an example dialog. The subjects were verbally informed that ISIS can support the following user activities: checking for a stock quote and related charts, market trends and financial news from a Reuters data-feed, placing a buy/sell order of a stock, amending/canceling a requested transaction, monitoring the movement of a stock price, and checking the user's simulated account information. Each subject was then asked to formulate a set of tasks in preparation for his/her interaction session that involved managing their simulated portfolios and/or gathering information prior to placing a buy/sell order on stocks that are of interest to them. They were advised to follow through with the completion of the transactions. The subjects were asked to interact with the system by speaking, typing or stylus-writing. All interactions (i.e. user inputs and system responses) were automatically logged by the system. All system operations were automatic except for one Wizard-of-Oz activity in triggering alerts. This use of a manual trigger is due to a practical reason—subjects may set up alert notifications with target stock prices that are different from market prices. There is no guarantee that the target stock prices will be attained within the duration of the interaction session in order for the alert to be generated. Hence the Wizard takes the liberty of overwriting the real-time stocks database such that the target price can be "reached" at an arbitrary instant during the interaction session. This triggers an alert message to the user. Consider the example of a user input, "*Please notify me when TVB rises above four dollars per share.*" After an arbitrary number of dialog turns (capped below ten turns) the wizard overwrites the stock price record in the database to be greater than four dollars. The subject receives an alert immediately afterwards. Subjects were informed that alerts may be artificially triggered by a Wizard.

The fifteen subjects generated fifteen dialog sessions. The input included both English and Chinese queries, but spoken Chinese included only Cantonese

[12]www.se.cuhk.edu.hk/∼isis/.

Table VI. Example Task List Prepared by a Participating Subject

(Start by speaking English). Check the latest information for Sun Hung Kai, e.g. stock price, turnover, market trends, news, etc. to help decide on a purchase price. Then place a purchase order for two lots of Sun Hung Kai.
(Change to speaking Cantonese). Check on the portfolio to see holdings of Hang Seng or another stock. Check the latest information of the stock to decide on a selling price and the number of lots, then place an appropriate sell order for the stock.

utterances since the recruited subject pool did not have native Putonghua speakers. A typical task list prepared by the subjects is shown in Table VI.

The dialog sessions averaged over 19 dialog turns in length. The ISIS system logged a total of 291 dialog turns. After each dialog session, the system log was presented to the subject, who was then asked to examine each system response and mark whether he/she considers it coherent or incoherent with regards to the original subject request. Out of the 291 dialog turns, 259 (about 89%) were deemed coherent by the subjects. The incoherent dialog turns were due to non-parsable input sentences or speech recognition errors, especially for out-of-vocabulary words in spoken input (which ISIS does not support). Based on the system logs and the ISIS task goal definitions (see Section 3.1), we also evaluated whether the intended tasks were completed successfully. Most of the users tried to repeat and rephrase their input requests in order to complete the intended tasks. Other users dismissed the ongoing task and pursued another because they were frustrated by successive errors in the system responses. Among the 120 tasks in total, 101 were successfully completed, 17 were successfully completed after one or more repeated attempts and 2 were dismissed as failures by the subjects. This corresponds to a task completion rate of 98%. These performance measurements suggest that the end-to-end ISIS system is a viable prototype, but there is room for further improvement.

Among the fifteen dialog sessions, the subjects attempted to set up 29 alert messages which in turn triggered 29 alert arrivals (indicated by visual icons). In all but two cases, the subjects chose to *immediately* interrupt the online interaction workflow to handle the alert. The two special cases correspond to the behavior of one subject who chose to complete her task in the OI dialog before switching to the alert message. Having handled the alert, the subjects need to return to the original, interrupted task. We examined the 27 interruptions and found that only 9 of them utilized the explicit spoken/textual request "*go back*" to resume the interrupted discourse (i.e. restoring the discourse state from the register to the OI dialog thread). For the 18 cases that remain, the user simply restarted the interrupted task afresh (3 cases) or initiated another new task afresh (15 cases). User behavior, as observed from these dialog sessions, shows an inclination towards *frequent focus shifting*—they tend to switch quickly from one task to another, especially when there are interruptions by alert messages. One may draw a preliminary conclusion, based on the limited amount of pilot data, that such user behavior may be characteristic of the financial domain where information is extremely time-sensitive and actions need to be taken promptly. Frequent focus shifting presents a challenge in proper handling of discourse contexts, among the interrupted and interrupting tasks.

The use of visual icons to indicate the presence of alert messages in ISIS (see Figure 11) forces the user to *explicitly* communicate an intended focus shift by clicking the icon. This signifies to the system to prepare for discourse handling when the interruption actually occurs. This option presents a simplification advantage and hence is preferred to an alternative that requires the system to automatically infer the occurrence of a focus shift/task switch. The outcome of automatic inference is not guaranteed to be correct every time and thus presents greater ambiguity in discourse handling. The user may also *explicitly* communicate an intended focus shift *back* to a previously interrupted task by the "go back" command. This command also signifies to the system to reinstate the interrupted discourse state. However, we noted that few subjects took advantage of this command and instead went through redundant dialog turns to set up the interrupted task from scratch. Possible remedial measures include—(i) raising the user's awareness of the "go back" command or represent it as a visual icon that appears as a just-in-time reminder [Cutrell et al. 2001]; and (ii) automatically and transparently reinstating the interrupted discourse state immediately upon completion of the interrupting task. Should the usage context evolve such that the typical user tends to be unfocused and frequently shifts his/her focus without prior communication, a more complex discourse interpretation framework, one that involves plan recognition and a focus stack [Lesh et al. 2001], that involves a will be necessary.

8. SUMMARY AND CONCLUSIONS

In this article, we have presented an overview of ISIS, a spoken dialog system for the stocks domain. The system resembles a virtual stock broker, and can provide real-time stock quotes, simulated financial transactions and information about a simulated personal portfolio. The conversational interface is augmented with a screen display that can capture mouse clicks, and users can also input textual queries by typing or stylus-writing. Textual input/output can be in English or Chinese, while spoken input/output can be in English, Cantonese or Putonghua.

ISIS provides the scope for research in a suite of core speech and language component technologies, such as multilingual recognition, understanding, and synthesis, that are integrated with the CORBA architecture. ISIS also provides the system test-bed for exploring several new directions in conversational interface development. The first is *adaptivity* to knowledge scope expansion. The knowledge base of the conversational interface (e.g., vocabularies of the speech recognizer and synthesizer, and the grammars of the natural language understanding (NLU) component) need to match the growing knowledge base of the background information system. For example, new stocks are listed continually at stock exchanges and automatic assimilation of such new information in the ISIS conversational interface is desirable. The current work presents a method of out-of-vocabulary (OOV) word detection in the NLU component, which also labels the semantic class of the new word by a set of transformational rules. If the new word is hypothesized to be a new stock name, then ISIS invokes a special subdialog designed to identify (i) the correct formal listing of the new stock in the financial database; and (ii) the (alternate) name by which the user

prefers to refer to the new stock. This newly acquired information is then automatically encoded in the form of grammar rules that are added to the NLU component.

The second new direction explored in the context of ISIS is asynchronous human-computer interaction, where the user can delegate the task of monitoring a real-time financial feed to software agents. This work presents an implementation of a multi-agent communication framework based on the Knowledge Query and Manipulation Language (KQML). The KQML agents receive requests of the alert services from the user, and work *offline* on the delegated task until the prescribed alert conditions are met, at which time the appropriate alert message is sent back to the main ISIS system. The use of KQML provides a scalable framework and multiple requests for alert services may be monitored simultaneously.

The third new direction explored in ISIS is interruptions in multi-threaded dialogs. ISIS maintains two dialog threads, one for online interaction (OI), in which the human and the system take alternate dialog turns in a mixed-initiative interaction, and another for the offline delegated (OD) alert messages that are delivered by the software agents. ISIS uses visual icons to make the user aware of the arrival of alerts in the offline queue in an effort to minimize disruption to the workflow in the OI dialog. Each alert message also provides a referring expression by using its stock name as default. The user may choose to interrupt his/her workflow in the OI dialog at any point, bring a specified alert message into focus in the OI dialog thread, complete the interrupting task, and then revert to the interrupted task. Fifteen subjects were invited to experiment with the research prototype of ISIS. Empirical observation showed that most subjects will readily interrupt their task at hand (in the OI interaction) to handle an alert message (in the OD dialog thread) as soon as it arrives.

Future work will involve increasing the complexity of the knowledge scope in ISIS, for example, involving multi-domain dialogs and multiple interruption messages. This will enrich the test-bed for exploring multi-threaded dialogs and interruptions in conversational interfaces. We also plan to set up wireless communication infrastructure so that users can interface with the ISIS system in a mobile setting: with smart phones and PDAs. In this context, heavier reliance will be placed on pointing and clicking to enrich our test-bed for exploring multi-modal and mixed-modal interactions [Oviatt et al. 1997; Oviatt and Cohen 2000]. Another possible future direction is to migrate the CORBA/KQML architecture towards a Web Services architecture[13] [Meng et al. 2003], which is an emerging standard based on XML that promotes modular system integration with a high degree of interoperability.

ACKNOWLEDGMENTS

We thank the past and present members of the ISIS team from the Human-Computer Communications Laboratory and Digital Signal Processing Laboratory of CUHK, as well as the National Key Laboratory for Machine Perception

[13]http://www.w3.org/2002/ws/.

of PKU. In particular, we thank Professors Hui-Sheng Chi, Ke Chen, Tan Lee, Ms. Lan Wang, Mr. Man Cheuk Ho and Mr. Kon Fan Low for their participation in this project. We are grateful to Reuters Hong Kong for donating the satellite feed to support this research. We also wish to thank two anonymous reviewers for their helpful suggestions, which include the use of visual icons for notifications in ISIS.

REFERENCES

AUST, H., OERDER, M., SEIDE, F., AND STEINBISS, V. 1995. The Phillips Automatic Train Timetable Information System. *Speech Comm*. *17*, 249–262.

BAZZI, J. AND GLASS, J. 2000. Modeling out-of-vocabulary words for robust speech recognition. In *Proceedings of the International Conference on Spoken Language Technologies (ICSLP-00)*. Beijing. ISCA-Archive, volume 1, 401–404.

BAYER S., DORAN C., AND GEORGE, B. 2001. Exploring speech-enabled dialog with the GALAXY communicator infrastructure. In *Proceedings of the Human Language Technologies Conference (HLT-01)*. San Diego. Morgan Kaufman, 79–81.

BRILL, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*. *21*(4), 543–566.

CHOI, W. N., WONG, Y. W., LEE, T., AND CHING, P. C. 2000. Lexical tree decoding with class-based language models for Chinese speech recognition. In *Proceedings of the International Conference on Spoken Language Technologies (ICSLP-00)*. Beijing. ISCA-Archive, volume 1, 174–177.

CHUNG, G., SENEFF, S., AND WANG, C. 2003. Automatic acquisition of names using speak and spell mode in spoken dialogue systems. In *Proceedings of the Human Language Technologies Conference (HLT-03)*. Edmonton. Morgan Kaufman, 197–200.

COHEN, P. R., CHEYER, J., WANG, M., AND BAEG, S. C. 1994. An open agent architecture. In the *AAAI Spring Symposium Technical Report*. Palo Alto. AAAI Press, 1–8.

CUTRELL, E., CZERWINSKI, M., AND HORVITZ, E. 2001. Notification, Disruption and memory: Effects of messaging on memory and performance. In *Proceedings of Human-Computer Interaction (Interact-01)*. Tokyo. IOS Press, 263–269.

CZERWINSKI, M., CUTRELL, E., AND HORVITZ, E. 2000. Instant messaging and interruption: Influence of task type on performance. In *OZCHI Conference Proceedings 2000*. 356–361.

DECADT, B., DUCHATEAU, J., DAELEMANS, W., AND WAMBACQ, P. 2002. Transcription of out-of-vocabulary words in large vocabulary speech recognition based on Phoneme-to-Grapheme conversion. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP-02)*. Orlando. IEEE, Volume 1, 861–864.

FININ, T. AND FRIITZSON, R. 1994. KQML—A Language and Protocol for Knowledge and Information Exchange. Tech. Rep. CS-94-02, University of Maryland, UMBC, <http://www.mmt.bme.hu/research/ai/lib/kbkshtml/kbks.html>.

FUNG T. Y. AND MENG, H. 2000. Concatenating syllables for response generation in domain-specific applications. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-00)*. Istanbul. IEEE, Volume 2, 933–936.

GODDEAU, D., MENG, H., POLIFRONI, J., SENEFF, S., AND BUSAYAPONGCHAI, S. 1996. A form-based dialog manager for spoken language applications. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)*. Philadephia. IEEE, Volume 2, 701–704.

GROSZ, B. AND SIDNER, C. 1985. Discourse structure and the proper treatment of interruptions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-85)*. Los Angeles. Morgan Kaufman, 832–839.

HAZEN, T. AND BAZZI, I. 2001. A comparison and combination of methods for OOV word detection and word conference scoring. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-01)*. Salt Lake City. IEEE, Volume 1, 397–400.

HORVITZ, E. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI-99)*. Pittsburgh. ACM Press, 159–166.

HORVITZ, E. AND BARRY, M. 1995. Display of information for time-critical decision making. In *Proceedings of Uncertainty in Artificial Intelligence (UAI-95)*. Montreal. Morgan Kaufman, 296–305.

HORVITZ, E., KADIE, C., PAEK, T., AND HOVEL, D. 2003. Models of attention in computing and communication: From principles to applications. *Comm. ACM 46*, 3, 52–59.

ISHII, H. AND ULLMER, B. 1997. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI-97)*. Atlanta. ACM Press, 234–241.

ISSAR, S. 1996. Estimation of language models for new spoken language applications. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)*. Philadephia. ISCA-archive, 869–872.

LAU, R. AND SENEFF, S. 1998. A unified system for sublexical and linguistic modeling using ANGIE and TINA. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-98)*. Sydney. 2443–2446.

LESH, N., RICH, C., AND SIDNER, C. 2001. Collaborating with focused and unfocused users under imperfect communication. In *Proceedings of the International Conference on User Modeling*. Sonthofen. Springer, 63–74.

LEVIN, E., NARAYANAN, S., PIERACCINI, R., BIATOV, K., BOCCHIERI, E., DI FABBRIZIO, G., ECKERT, W., LEE, S., POKROVSKY, A., RAHIM, N., RUSCITTI, P., AND WALKER, M. 2000. The AT&T DARPA communicator mixed-initiative spoken dialog system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-00)*, Beijing, ISCA-Archive, Volume 2, 122–125.

LIM, R. W. AND WOGALTER, M. S. 2000. The position of static and on-off banners in WWW displays on subsequent recognition. In *Proceedings of the Joint Meeting of the Human Factors and Ergonomics Society and the International Ergonomics Association (IEA2000/HFES2000)*. San Diego. 420–423.

MAGLIO, P. AND CAMPBELL, C. 2000. Tradeoffs in displaying peripheral information. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI-00)*. The Hague. ACM Press, 241–248.

MANOS, A. AND ZUE, V. 1997. A segment-based spotter using phonetic filler models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-97)*. Munich. IEEE, Volume 2, 899–902.

MCFARLANE, D. 1999. Coordinating the interruption of people. In *Human-Computer Interaction. Proceedings of Human-Computer Interaction—INTERACT99*, IOS Press, Inc., The Netherlands, 295–303.

MCTEAR, M. 2002. Spoken dialog technology: Enabling the conversational user interface. *ACM Computing Surveys (CSUR). 34*, 1, 90–169, ACM Press.

MENG, H., LEE, S., AND WAI, C. 2000a. CU FOREX: A bilingual spoken dialog system for foreign exchange enquires. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-00)*. Istanbul. IEEE, Volume 2, 229–232.

MENG, H., CHING, P. C., WONG, Y. F., AND CHAN, C. C. 2002b. A multi-modal, trilingual, distributed spoken dialog system developed with CORBA, JAVA, XML and KQML. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-02)*, Denver. 2561–2564.

MENG, H., CHAN, S. F., WONG, Y. F, CHAN, C. C., WONG, Y. W., FUNG, T. Y., TSUI, W. C., CHEN, K., WANG, L., WU, T. Y., LI, X. L., LEE, T., CHOI, W. N., CHING, P. C., AND CHI, H. S. 2001. ISIS: A learning system with combined interaction and delegation dialogs. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech-01)*. Scandinavia. Volume 3, 1551–1554.

MENG H., CHAN, S. F., WONG, Y. F., FUNG, T. Y., TSUI, W. C., LO, T. H., CHAN, C. C., CHEN, K., WANG, L., WU, T. Y., LI, X., LEE, T., CHOI, W. N., WONG, Y. W., CHING, P. C., AND CHI, H. 2000b. ISIS: A multilingual spoken dialog system developed with CORBA and KQML agents. 2000. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-00)*. Beijing. ISCA-Archive, Volume 2, 150–153.

MENG, H. AND TSUI, W. C. 2000. Comprehension across application domains and languages. In *Proceedings of the International Symposium on Chinese Spoken Language Processing (ISCSLP-00)*, Beijing. CD-ROM proceedings.

MENG H., LAM, W., AND WAI, C. 1999. To believe is to understand. In *Proceedings of 6th European Conference on Speech Communication and Technology (Eurospeech-99)*. Budapest. ISCA-archive, 2015–2018.

MENG H., BUSAYAPONGCHAI, S., GLASS, J., GODDEAU, D., HETHERINGTON, L., HURLEY, E., PAO, C., POLIFRONI, J., SENEFF, S., AND ZUE, V. 1996. WHEELS: A conversational system on electronic automobile classifieds. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)*. Philadelphia. IEEE, Volume 1, 542–545.

MENG, H., KEUNG, C. K., SIU, K. C., FUNG, T. Y., AND CHING, P. C. 2002b. CU Vocal: Corpus-based syllable concatenation for Chinese speech synthesis across domains and dialects. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-02)*, Denver. 2373–2376.

MENG, H., LO, T. H., KEUNG, C. K., HO, M. C., HO, M. C., AND LO, W. K. 2003. CU VOCAL Web Service: A Text-to-speech synthesis Web service for voice-enabled Web-mediated applications. In *Proceedings of the 12th International World Wide Web Conference (WWW-03)*, Budapest. www2003.org/cdrom/papers/poster/p056/p56-meng.html.

OVIATT, S., DEANGELI, A., AND KUHN, K. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI-97)*. Atlanta. ACM Press, 415–422.

OVIATT, S. AND COHEN, P. 2000. Multimodal interfaces that process what comes naturally. *Comm. ACM 43*, 3, 45–53.

PAPINENI K. A., ROUKOS S., AND WARD R. T. 1998. Free-flow dialog management using forms. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-98)*. Sydney. 1411–1414.

RAO, A. S. AND GEORGEFF, M. P. 1995. BDI Agents: From Theory to Practice. Tech. Rep. #56, Australian Artificial Intelligence Institute, Melbourne, Australia.

REYNOLDS, D. A. 1992. A Gaussian mixture modeling approach to text-independent speaker identification. Ph.D. thesis, Georgia Institute of Technology.

ROSSET, S., BENNACEF, S., AND LAMEL, L. 1999. Design strategies for spoken language dialog systems. In *Proceedings of 6th European Conference on Speech Communication and Technology (Eurospeech-99)*. Budapest. ISCA-archive, 1535–1538.

RUDNICKY, A., THAYER, E., CONSTANTINIDES, P., TCHOU, C.,SHERN, R.,LENZO, K.,XU, W., AND OH, A. 1999. Creating natural dialogs in the Carnegie Mellon communicator system. In *Proceedings of 6th European Conference on Speech Communication and Technology (Eurospeech-99)*. Budapest. ISCA-archive, 1531–1534.

SADEK, M. D. AND MORI, R. 1997. Dialog systems. In *Spoken Dialog with Computers*. R. de Mori, Ed. Academic Press, 523–561.

SCHILLO, C., FINK, G. A., AND KUMMERT, F. 2000. Grapheme-based speech recognition for large vocabularies. In *Proceedings of the International Conference on Spoken Language Technologies (ICSLP-00)*. Beijing. ISCA-archive, Volume 4, 584–587. Beijing, 2000.

SENEFF, S., LAU, R., AND POLIFRONI, J. 1999. Organization, communication and control in the Galaxy-II conversational system. In *Proceedings of 6th European Conference on Speech Communication and Technology (Eurospeech-99)*. Budapest. ISCA-archive, 1271–1274.

SENEFF, S., CHUU, C., AND CYPHERS, D. S. 2000. Orion: From On-line Interaction to Off-line Delegation. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-00)*. Beijing. ISCA-archive, Volume 2, 142–145.

SENEFF, S., LAU, R., AND MENG, H. 1996. ANGIE: A new framework for speech analysis based on morpho-phonological modeling. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)*. Philadelphia, IEEE, Volume 1, 110–113.

TAYLOR, P., BLACK, A., AND CALEY, R. 1998. The architecture of the festival speech synthesis system. In *Proceedings of the 3rd ESCA/COCOSDA Workshop on Speech Synthesis*. Jenolan Caves, Australia. 147–151.

VAN DANTZICH, M., ROBBINS, D., HORVITZ, E., AND CZERWINSKI, M. 2002. Scope: providing awareness of multiple notifications at a Glance. In *Proceedings of the ACM International Working Conference on Advanced Visual Interfaces (AVI-02)*. Trento, ACM Press. research.microsoft.com/∼horvitz/scope.htm.

Received September 2002; revised June 2003 and February 2004; accepted March 2004