http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (21): 6213-6217, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6213.6217

Research on the Universals of Voice Interactive Interface

Wang Shan and YI Ding
Computer Engineering College, Shenzhen Polytechnic College, Shenzhen 518055, China

Abstract: This study presents the universals of voice interactive interface. The design of voice interactive interface is separated from the business logic so that the implementation of voice interactive application is simplified. A voice interactive interface model and the description of its knowledge base are proposed. The key technology to realize this model is given with an example.

Key words: Terms-voice interactive interface, voice interactive application, knowledge base, state transfer network, limited natural language recognition

INTRODUCTION

The traditional application system uses keyboard or mouse to input and screen display to output. Speech technology provides a feasible way to achieve real sense of man-machine dialogue (Gao *et al*., 2012). When Voice interactive function is integrated into existing human-computer interactive system as a complement of traditional man-machine interface, it makes the process of Human-computer Interaction (HCI) more natural and humanized (Fujita and Kato, 2011). Voice interaction is especially useful in the application which blind person uses.

This voice interaction contains two aspects: one is speech recognition technology that computer can understand human speech and the application is operated by speaking, the other is speech synthesis technology that traditional output of text can be converted to voice output. There are many examples such as NVID project in (Paper *et al*., 2004) and iPhone's Siri. The U.S. Navy has launched a Naval Voice Interactive Device (NVID) project that creates a lightweight, portable computing device that uses speech recognition to enter shipboard environmental survey data into a computer database and to generate reports to fulfill surveillance requirements (Paper *et al*., 2004). Siri is available for iPhone products. Siri is the intelligent personal assistant that helps users get things done just by asking. It allows users to use voice to send messages, schedule meetings, place phone calls and more. Siri understands human natural speech and it asks users questions if it needs more information to complete a task.

In the future, there will be more and more applications with voice interactive technology like Siri to come out (Yong *et al*., 2004).

CHARACTERISTICS OF VOICE INTERACTIVE INTERFACE AND ITS DESIGN ELEMENTS

Some limitations about language expression are needed in order to improve speech recognition rate. In addition, the factors influencing speech technology include the following aspects.

- Unpredictable errors. It is impossible to know in advance what errors would be made by speech recognition and users often don't like to use such products with uncertainty
- The difference between spoken language and written language
- The control right of user. If users feel no control, a more emotional response would be occurring

The form of man-machine dialogue, which is technically feasible, includes question and answer mode, command mode, limited natural language dialogue etc. Question and answer mode, command mode can be viewed as a simple example of limited natural language dialogue. In this study, the limited natural language dialogue will be discussed as an example.

The following questions will be researched that how to realize the man-machine dialogue in limited natural language.

- The dialogue management. The context of dialogue forms the internal state of the system. State and state transfer is the basis of dialogue management
- The semantic recognition. Firstly the user's voice is identified into the text and then natural language understanding made. To understand the semantics, it is needed to study semantic grammatical structures according to user intent (Munzlinger *et al*., 2007)

Corresponding Author: Wang Shan, Computer Engineering College, Shenzhen Polytechnic College, Shenzhen 518055, China

Many voice application systems mix these two questions with their business logics. When the business logic complexity increases, the system interface design complexity will increase too. It will make the system difficult to extend and maintain. To solve these problems the research and implementation of voice interactive interface universals will be discussed next.

THE VOICE INTERACTIVE INTERFACE MODEL (VIIM)

Extraction of the Topic from Utterances in a Spoken Dialog System studies an extraction method based on rules but there is no application and implementation discussed (Wang and Jiang, 2004).

In this study, a Voice Interactive Interface Model (VIIM) and its description of knowledge base are proposed. The key technology to realize the VIIM is provided with an example. The model is suitable for many kinds of voice interactive application systems such as some public information service systems and so on.

VIIM, shown in Fig. 1, consists of knowledge base, voice parts, dialogue management. The knowledge base is the core of VIIM. It defines knowledge and tasks which are associated with the application and is used to support system's voice interaction and dialogue management. The voice parts collects sound signal and identifies the user intent with the support of knowledge base, thus sends the user intent to dialogue management. With the support of knowledge base, dialogue management controls dialogue context, activates the corresponding task to execute.

Knowledge base associated with voice interaction involves two aspects of content: one is user intent and its semantic grammatical structure, the other is conversation context planning. To support the semantic recognition, using semantic frame set defines the user intent and its semantic grammatical structures associated with application. To support dialogue control, using State Transfer Network (STN) defines conversation context associated with application. The STN is a hierarchy nested characteristic collection of state transfer diagram. The semantic frame set and STN consists of the knowledge base which supports voice interaction.

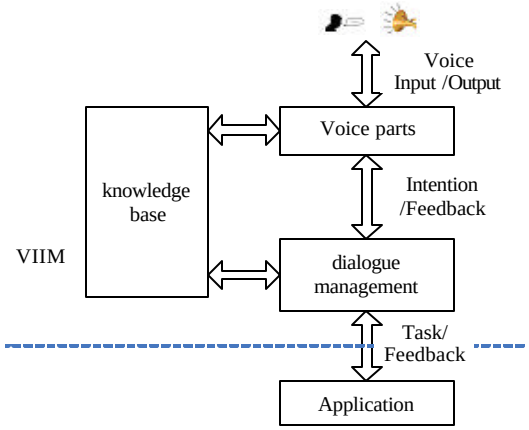

Fig. 1: Voice interactive interface (VIIM)

THE KNOWLEDGE BASE OF VILM

The knowledge base, as the core of VIIM, defines the knowledge and task associated with application and is consisted of semantic frame set and STN.

The semantic frame: The semantic frame, which supports semantics recognition of the limited natural language, is a collection of semantic features. Each semantic feature represents for special user intent. It defines the user intent and its semantic grammatical structures which is context-free, i.e. (user intent, parameter 1, 2,...).

Here is an example of query about someone's phone number, address or work unit. The query intent described with limited natural language is expressed firstly. There are many different ways of expression for the same intent from formal to colloquial expression. A variety of combinations of such expression are shown in Table 1 but the grammar structure abstracted is the same, i.e. (user intent, parameter 1, 2). Secondly, the content of the grammatical structure can be divided into static and dynamic content. The dynamic content parameter 1 'who' and parameter 2 'what' can be extracted out and their sub-grammatical structure can be defined. If the user said, 'look up Jack's address?' This sentence can be identified. The result of analysis is that the user intent is 'query', 'who' is for Jack and 'what' is for address.

State Transsfer Network (STN): Based on the conversation context related with application, STN designs and defines system state and task. The dialogue management, supported by the STN in knowledge base, realizes the context control and activates the corresponding task to execute.

Table 1: An example of the user intent and its semantic grammatical structure

Intent: Query	Parameter 1: Who	Parameter 2: What
Would you like to query	Peter	Phone
Can you help me query	Jack	Address
Look up	Mike	Work unit
Tell me		

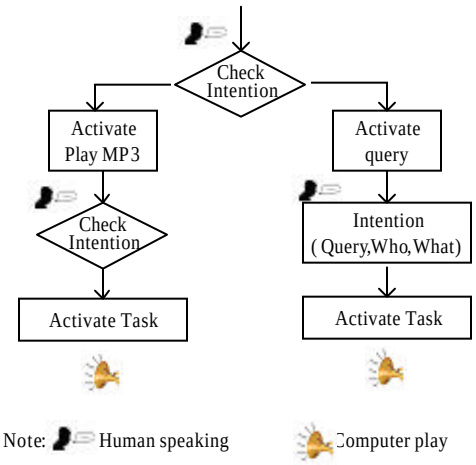

Fig. 2: Man-macine dialogue

Table 2: S0 state table

Event	Task	Next state
Query	Activate query task	S1
Play MP3	Activate play MP3 task	S2

Event: Recognizing the user intent

STN is described with a hierarchy nested characteristic collection of state transfer network diagram. STN is consisted of state collection, state table collection and the collection of state transfer network diagram.

Each state has a state table with three columns: event, task and next state, i.e., Table 2. The state table describes the relationship among the current state, event and next state. If an event happens at current state, the corresponding task must be activated to execute, thus the system transfers into next state. The event here means that a user intent has been identified successfully.

State Transfer Network (STN) is a kind of network diagram built up by state tables, i.e., Fig. 3. The arrowhead line is connected from current state to next state, each line corresponds to the event which makes the transfer of state when an event is triggered.

An example is shown in Fig. 2. The function is chosen by speech firstly: Query or play MP3; then user intent is identified by speech recognition and activates corresponding task execution. The work flow of Fig. 2 corresponds to two layers of STN respectively, as shown in Fig. 3.

The first layer state collection includes S0, S1, S2. S0 is 'initial' state; S1 is 'query' state; S2 is 'play MP3' state.

Each state has its state table, i.e., the state table of state S0 is shown in Table 2.

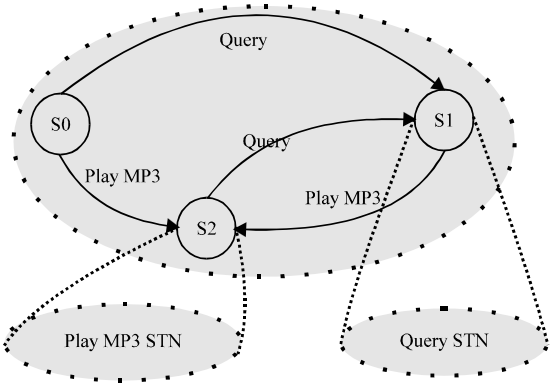

Fig. 3: The STN of Fig. 2

THE KEY TECHNOLOGIES TO REALIZE VIIM

When a voice application is to be developed, different voice engine and development platform should be considered to use according to the requirement of application and the features of the technology. The key technologies to realize VIIM with Microsoft Speech SDK and VB.NET are discussed below (Kang and Yoo, 2007).

The definition of semantic frame with XML: Schema language: The formalized grammar of XML Schema language is used to define the semantic frame set and to save in the form of XML file.

The definition of XML schema for Table 1 with Microsoft Speech SDK is listed below.

```
<RULE NAME="query" TOPLEVEL="ACTIVE">
   <L>
      <P>Would you like to query </P>
      <P>Can you help me query</P>
      <P>Look up</P>
      <P>Tell me</P>
   </L>
   <RULEREF NAME="who"/>
   <O>μÄ</O>
   <RULEREF NAME="what"/>
</RULE>

<RULE NAME="who">
   <LN PROPNAME="WHONAME" PROPID="whoname">
     <PN VAL="0">Peter</PN>
     <PN VAL="1">Jack</PN>
     <PN VAL="2">Mike</PN>
   </LN>
</RULE>

<RULE NAME="what" >
   <LN PROPNAME=" WHATDETAIL " PROPID=" whatdetail ">
     <PN VAL="0">telephone number</PN>
     <PN VAL="1">address</PN>
     <PN VAL="2">work unit</PN>
   </LN>
</RULE>
```

The row (user intent, parameter 1, 2) extracted from Table 1 corresponds grammatical structures (query, who, what). XML schema defines it as <RULE NAME="query" TOPLEVEL="ACTIVE">. Among them, the Dynamic content is 'who' and 'what'. The sub-grammatical structures for 'who' and 'what' are defined to <RULE NAME="who"> and <RULE NAME="what">. So if the user said 'look up Jack's address', the sentence should be identified that the intent is 'query', who is for 'Jack' and what is for 'address' in XML schema.

The XML file, saved in form of text file, is called by Speech recognition program to support limited natural language speech recognition. The developer of the application system can easily update or add the content of XML schema at any time according to the dialogue content so that the effect of human-machine dialogue is improved continuously.

The realization of STN in knowledge base: The STN consists of a series of collections which are realized with Microsoft.NET collection technology (Kang and Yoo, 2007). The collections contain many classes of objects. ArrayList is a class of collection with imitation of a dynamic array. Items of array are the instances of object. The instance of the class can be used to save any type of a variable. When objects are added or decreased into the collection, the size of the collection should be vary according to the requirement.

The data items in one row of state table can be defined as 'statetable' class.

```
Public Class State Table
   Delegate Sub ActionDef()   'delegate, task
   Public events As Integer      'event
   Public actions As ActionDef   'task
   Public nextstate As Integer   'next state
End Class
```

Delegate technology of Microsoft. NET is used here (Kang and Yoo, 2007). Delegate is type-safe, object-oriented references to methods. By using delegate, you make your method accessible to others and therefore more extensible. The address of any procedure or method can be gotten that these procedure or method can be called by that delegate. A delegate named 'ActionDef' is defined here and the data type of class field actions is delegate of ActionDef. The task defined in state table is assigned to actions. So the meaning of one row in state table is that when the event from class field actions happens, the task of actions will be executed. Only this knowledge is saved in state table.

The real execution is driven by the finite-state machine.

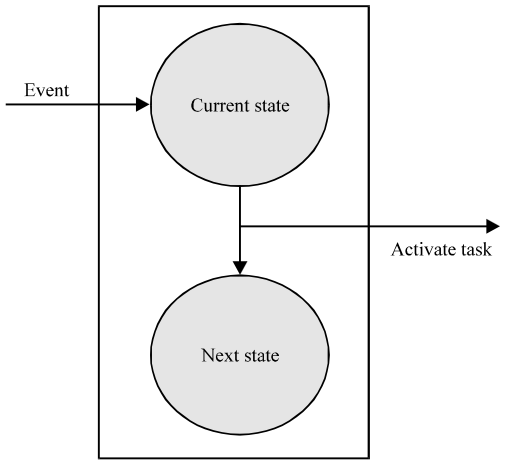

Fig. 4: The finite-state machine

The implementation and features of the finite-state machine in dialogue management: The finite-state machine is the core of dialogue management. The dialogue management records the current state. If a certain event is triggered, the finite-state machine, shown in Fig. 4, should activate the task defined in the state table related with current state in STN and then transfer into the next state. The finite-state machine perform the state transfer according to STN when an event is triggered, no care for the connotation of state. It drives task to activate without care of the meaning of the task.

The technology of Microsoft. NET's delegate is used to realize the task driven by finite-state machine. That address of tasks is stored in state table in knowledge base. The real execution here is driven by the finite-state machine. The finite-state machine calls the method of delegate object which is referenced by actions to perform the real task to execute.

It makes the control flow automatically and efficiently by using Microsoft. NET delegate to realize the finite-state machine. It brings about the separation between the control code of system and the knowledge related with application. Thus a lot of control code is removed. When applications change, the knowledge base is needed to be maintained only. The control code keeps the same so that developers do not need to code and maintain for the complex control logic.

THE VOICE APPLICATION SYSTEM BASED ON VIIM

When VIIM is used in the voice interactive application, the workflow is shown here. The voice parts collects sound signal and identifies the user intent with the support of knowledge base, thus sends the user intent

Table 3: The comparison between voice application system based on VIIM and traditional voice application system

	Voice application system based on VIIM	Traditional voice application system
Architecture	Clear	Fuzzy
Expandability	Easy	Difficult
Reusability	Reusable,	No reusable
Maintainability	Easy	Difficult

to dialogue management. The dialogue management module records the current state and sub-state of system and transfer into next state when dialogue happens according to the definition of STN.

The core of dialogue management is the finite-state machine whose automatic mechanism is independent of the dialogue, shown in Fig. 4. If a certain event is triggered, the finite-state machine should activate the task related with current state in knowledge base and then transfer into the next state. Now the next state will become the new current state. The separation between the control code and the knowledge makes the finite-state machine to deal with the complexity effectively, control the workflow automatically.

The feature of voice application system based on VIIM is that the separation between voice interface layer and the application layer (Table 4). Only the knowledge base is related with a special application, dialogue management irrelevant to the special application. Due to the support of knowledge base, dialogue management can carry out complete self-control of inner state and task and solve the division of function hierarchy and collaboration about interface layer and application layer well.

CONCLUSION

Compared the voice application system based on VIIM with traditional system, VIIM solves the problems well concerned with the separation and collaboration between voice interface layer and application business layer. So the application system based on VIIM has good expandability. When the complexity of business increases, the complexity of interface design does not increase. It is necessary to maintain the knowledge base only when the content of business varies while the control program does not change.

The universal VIIM can be used in all kinds of voice interactive system such as flight query, telecom business query etc. It provides a new idea and attempt for the research of humanized PC.

ACKNOWLEDGMENT

This study was supported by Science and technology project in Shenzhen (06KJce037) (001CJC008).

REFERENCES

Wang, B. and M.H. Jiang, 2004. Extraction of the topic from utterances in a spoken dialog system. Comput. Eng. Appl., 1: 58-60.

Kang, B.S. and G.J. Yoo, 2007. Efficient voice user interface system using voice XML and ASP.NET 2.0. Parallel Proces. Technol. Lecture Notes Comput. Sci., 4847: 608-616.

Paper, D.J., J.A. Rodger and S.J. Simon, 2004. Voice says it all in the Navy. Communi. ACM, 47: 97-101.

Munzlinger, E., F. da Silva Soares and C.H.Q. Forster, 2007. Evaluation of a multi-user system of voice interaction using grammars. Human-Computer Interact., 4663: 452-455.

Gao, H.X., H. Xuan and S.G. Li, 2012. Speech technology innovation on traditional interaction research. Popular Literat. Art, Vol. 9.

Yong, H., D. Xu and G.Z. Dai, 2004. Research on development of speech user interface. Comput. Sci., 31: 1-4.

Fujita, K. and T. Kato, 2011. Design and development of eyes- and hands-free voice interface for mobile phone. Human Centered Design, 6776: 207-216.