Design and Development Tool for Telephone-based Network Information System

Yukiko Yamaguchi[1], Kazuaki Ito[2], Nobuo Kawaguchi[1],
Shigeki Matsubara[1], and Yasuyoshi Inagaki[2]
[1] Information Technology Center, Nagoya University
[2] Department of Computer Science and Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan

ABSTRACT

Trouble management of the huge network is required for 24 hours opening. The usage of an automatic answering machine is desirable. Since this kind of network information system requires the various components such as speech recognition, speech synthesis, natural language processing, telephone control, networking tools, and so on, it is necessary for the developers to rebuild the system a number of times. We have made a GUI-based tool for assisting the developers to design and rebuild a spoken language system efficiently. We have implemented the tool in Ruby script, including speech synthesis and telephone board control function. Using this tool we have just constructed the first version of the network information system and began the trial operation.

Keywords: Design and Development Tool, GUI-based Tool, Spoken Dialogue System, Campus Network, Telephone-based Information System

1. INTRODUCTION

The campus network of Nagoya University is composed of the 30km optical cable among 120 buildings and 90 subnets. Over 20,000 members (students/staff) rely on the network. When the user within the campus has any network trouble, he/she makes a phone call to the campus network section of Information Technology Center because it is difficult to solve it by oneself. It is required for the staff of the center to examine the situation by using networking tools such as “ping” or “snoop”. If they realize that the campus network has a trouble, they find the cause and respond to the user. However, when the trouble happens in the nighttime or on a holiday, it is not possible to do so. This is the reason that we have decided to develop a telephone-based network information system which respond to network queries automatically.

The system aims to receive the telephone calls from users, to investigate the network situation, and to answer to them. This kind of system requires various kinds of components like speech synthesis, telephone control, network investigation. In developing such a system, it might be required to rearrange the components a number of times. To do the development efficiently, a design and development tool in which a designer can arrange the components with GUI is desirable.

We have implemented a tool for design and development of a telephone-based information system. The tool provides nine basic components, such as speech synthesis, telephone control, and two extension components. A spoken dialogue system can be designed by defining blocks of components, arranging them, and connecting them.

In this paper, we describe the design and development tool in section 2, and its application to a network information system in section 3.

2. DESIGN AND DEVELOPMENT TOOL

2.1 Design of the Tool

A spoken dialogue system requires the basic components such as speech recognition and speech synthesis, and also the special components depending on the task such as the information retrieval and the network investigation. It is desirable that the developer can use the speech synthesis and the speech recognition without the detailed knowledge concerning them. A GUI-based design and development tool provides such an environment as mentioned above. The advantages of the development of a spoken dialogue system using the tool are as follows:

1. The function of each component becomes clear by representing each component as a block.
2. The rearrangement of the components becomes easy.
3. The structure of the entire system becomes easy to understand because the system is graphically designed using GUI.

Therefore, several design and development tools such as LOTOS system[1] and CSLU Toolkit[2] have been developed so far. LOTOS system aims to support developments of city information retrieval system. In LOTOS system, the components such as speech recognition, speech synthesis, condition branch, and database access are expressed as blocks in GUI, and connected by button or mouse operations. It has an expression block in which the designer can implement the

function in Visual Basic. However, the function which can be described in the expression block is limited to operations on the variables.

CSLU Toolkit is the environment to support development of a spoken dialogue system. In RAD (Rapid Application Developer) the designer can create a variety of interactive programs by dragging and dropping dialogue states onto a canvas, connecting them, and configuring them to do things like play audio files, create animated text-to speech, recognize spoken language, or display images. RAD has also ACTION state which executes Tcl/CSLUsh code. The program can be saved in RAD form, and executed only in the environment of CSLU Toolkit.

Then we decides to make a design and development tool which can build a system including the telephone control function and the module that is originally implemented by the designer and output a module which can be executed independently from the tool.

2.2 Implementation of our Tool

We have implemented a new tool on Windows 2000 in Ruby[3]. Ruby is a kind of object-oriented script languages, which has language specification like Perl and works on Windows, Linux and Solaris. It has rich libraries such as text processing, graphics, networking tools and so on. And it is also possible to call programs implemented in C from Ruby script. We have implemented a GUI of our tool by using the Ruby/Tk library. Our tool works on the Ruby interpreter.

Figure 1 shows the main design window of the system. The windows are composed of the component menu, GUI canvas, the block list, the system variable list, and the block information. Initially there is indicated a start block in the GUI canvas. The design by using this tool consists of the following three steps, (1) defining blocks as instances of components, (2) arranging them on the GUI canvas, and (3) connecting them by mouse operations.

This tool has the following nine basic components. We aim a telephone-based system, so we use the telephone control board D/41H and D/4PCI made by Dialogic and implemented its control components in Ruby.

Playback: Play back voice from PCM format data via the telephone board. The designer can specify the following parameters, the upper limit of play back time, the interruption allowance, the branching condition to following block.

Speech Synthesis: Produce voice from Japanese text. We use Document Talker by Fujitsu. The designer can embed variables in Japanese text to be generated. The variables may be given appropriate value in another block. Therefore, the utterance can be changed dynamically by combining with other blocks.

Speech Recognition: Convert telephony voice to Japanese. We have not had the speech recognition function yet. Speech Recognition block can save the input voice in PCM format.

DTMF Input: Receive a DTMF(Dial Tone Multi Frequency) input and store it in a parameter. The designer can specify acquisition number of digit, the upper limit of the acceptance time, and the branching condition to following block.

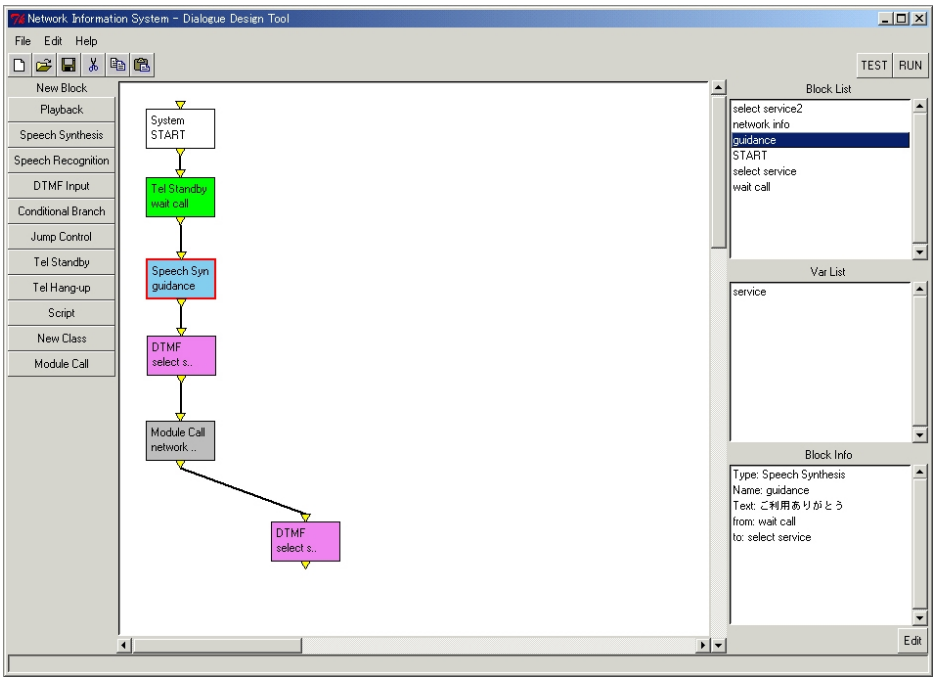

Fig. 1 Design Window

Conditional Branch: Describe conditions to jump to another block. The designers use this block when the branch condition is complex or they want to express the conditional branching explicitly.

Jump Control: Specify the next block name directly. This block can be used when the designer cannot specify the next block by mouse because the system is large-scale.

Module Call: Call a module which was already constructed.

Telephone Standby: Initialize the telephone board and wait for a telephone call. The designer can specify the upper limit of waiting time.

Telephone End: Hang up phone.

The tool also provides the following two enhancing components. These enhancing components enable the function which is made by the designer originally to be treated by the same interface as a basic component.

Script: As a Ruby script, the designer can describe a peculiar function like database exploration and network investigation, etc. The Ruby script is preserved as string data, and will be evaluated at the execution time.

New Class: New class of the operation object. The designer can produce a new component.

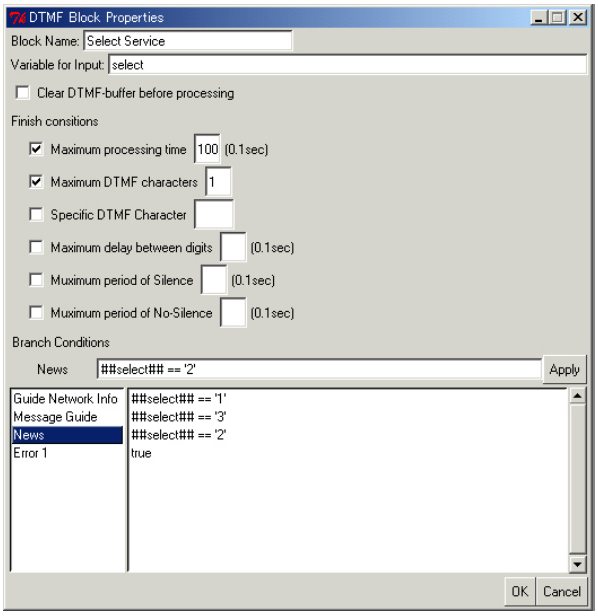

Fig. 2 Specifying of DTMF Input Block

The blocks which construct the spoken dialogue system are achieved by selecting the component and specifying the configuration. Figure 2 shows an example of defining a block of DTMF Input component. The designer specifies the block name, the variable name in which the DTMF input is stored, and the condition to quit DTMF input. For the designer who does not have knowledge concerning the DTMF Input, several typical conditions are prepared. So the designer only has to select one of them on GUI and set a numerical value. In order to specify the following processes, the designer selects the next block in the GUI canvas by mouse operations and specifies the condition in Ruby statement. At this time, the block name is automatically acquired.

The block defined like above is expressed internally by the following three objects.

Information Object: name of the block, type of the block, link to the next block and its condition, etc.

GUI Information Object: the position on the GUI canvas.

Operation Object: the script describing the function of the block.

These objects are instances of the individual class of each component which is inherited the super class that has the common function to all components. Figure 3 shows the relation between the class and the object.

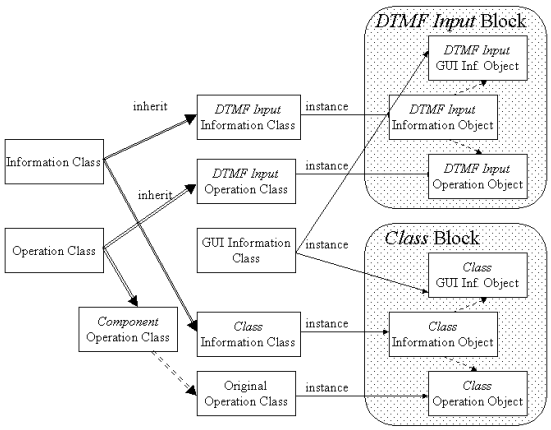

Fig. 3 Relation between Class and Object

The information object not only maintains the specified configuration as shown in Figure 2 but also works as the kernel of the designed system. It has the method to get the operation object and the GUI information object and controls the flow of the processing of the spoken dialogue system. Figure 4 shows the flow. By the method of the information object, the function of the block described in the operation object is executed. From the execution result, the branching condition is judged in the information object, and the processing advances to the following block. The system designed by using this tool starts the processing from the start block, and in each block the processing is similarly done while the next block is set.

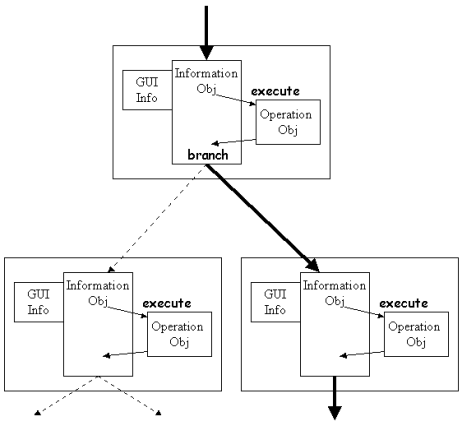

Fig. 4 Flow of Processing of the Designed System

By separating the execution part as an operation object from the control of the flow, the utilization of the extension facility becomes easy. As for the script component, the operation class has the function that evaluates the character string as Ruby script. The script described at the block definition is preserved as character string data in the information object, and will be evaluated at the execution of the operation object. As for the class component, when the block is defined, the designer specifies the original operation class and the block has the operation object as a instance of the original class. The designer can define the operation class originally or enhance the operation class of the other component.

In the development of the spoken dialogue system by using our tool, the designer can check how the system works under two kinds of mode, test mode and execution mode. In the test mode, the designed system executes without actual peripheral I/O. For example, a DTMF Input block gets the input from the keyboard instead of the input from DTMF. Therefore each operation object has two kinds of the operation.

The tool can output the Ruby script which realizes the designed spoken dialogue system and the Postscript file which describes the composition of blocks on the GUI canvas. The Ruby script works only on the tool presently.

3. DEVELOPMENT OF NETWORK INFORMATION SYSTEM

3.1 Construction of Network Information System

The campus network of Nagoya University has a tree topology. This means that the network becomes unavailable when the power supply stops at another building. And as the firewall of campus LAN intercept some kinds of packet from the outside of campus, the user who is using a host in campus network via ISP does not have any investigation method at the situation inaccessible to the host.

Figure 5 shows the notion of the network information system.

A user who has a network trouble makes a phone call to the system, and according to its voice, selects the following service.

1. Information service which offers the schedule of network stop due to network maintenance/construction or power supply stop.
2. Investigation service which examines the network situation.
3. Message service which records a message to the campus network center.

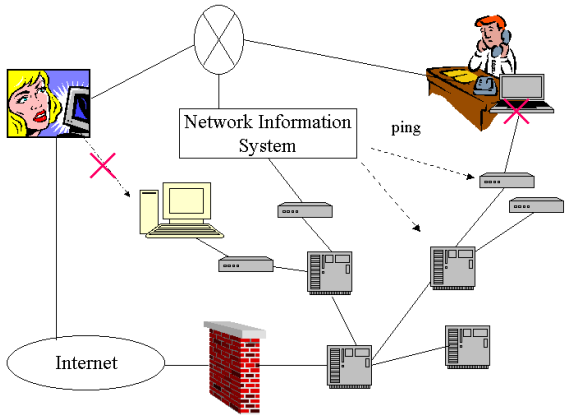

Fig. 5 Network Information System

Figure 6 shows an example of designing of the network information system by using our tool. The system keeps the standby state until receiving a telephone call, and then sends the guidance voice via telephone. The system switches to the block determined according to the DTMF input which the user selects.

When the Information service is selected the system reads network maintenance/construction information from a file and outputs the information voice generated by speech synthesis module.

In the network investigation service, the user inputs IP address by DTMF input. Then the system verifies IP address, investigates the reachabilites from the system to the address and from the system to typical Web site in the Internet.

When the messages service is selected, the system preserves the utterance of the user, and sends a mail to the network managers to notify to receiving the message.

At the end, the user is requested to answer a questionnaire asking about his/her evaluation of the system.

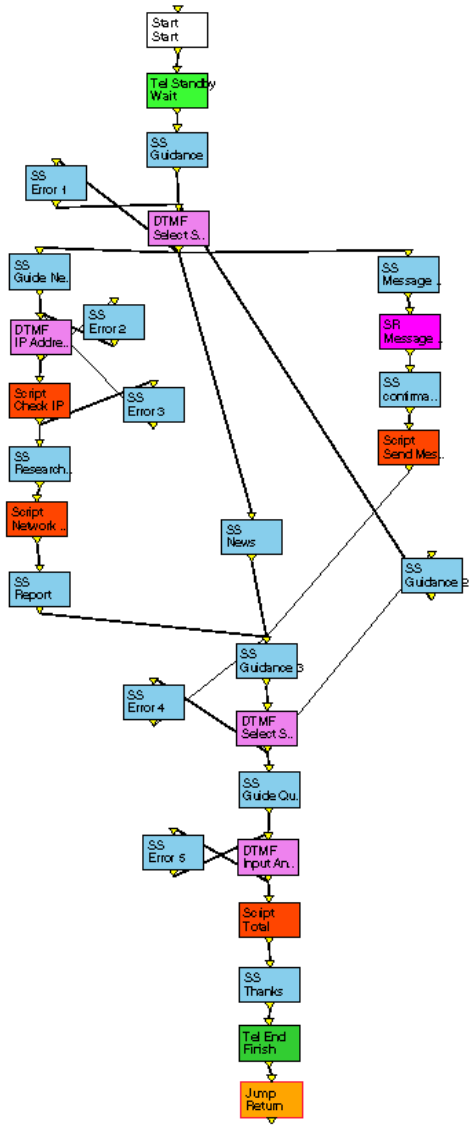

Fig. 6 A Design of Network Information System

Table 1 Composition of the Blocks

Type of block	Number of blocks
Start	1
Speech Synthesis	17
DTMF Input	5
Voice Recording	1
Telephone Standby	1
Telephone End	1
Jump Control	1
Script	5

We have constructed the Network Information System from the 32 blocks. Table 1 shows the composition of the 32 blocks. The speech synthesis blocks output the voice of the guide of the service, the announcement of the network maintenance/construction information, and the report of the network investigation result, etc. The DTMF Input blocks get the selection of the service, IP address, the questionnaire. And we had to implement the following five peculiar function in the script block.

- Verification of input IP address
- Network investigation by ping
- Acquisition network construction information from file
- Mail notification to the network manager
- Total of questionnaire

While we constructs the system by using the tool actually, we realize that it becomes easy to design and to construct the system, and it is helpful for us to be able to examine the structure of the system while designing. Moreover, it is easy to deal with the changes of specification through replacements and substitutions of blocks. On the other hand, it is annoying to set the branching condition in every block.

3.2 Trial Operation of the System

We set up the network information system in Information Technology Center of our university, and began the trial operation. By this trial operation, it has been realized that the system hangs up when the telephone is unexpectedly cut. Therefore we added the function to specify the return block when the telephone is cut at any point of the system. This trial in continuing now, and digs up the problem.

4. CONCLUSION

This paper has described our object-oriented design and development tool for the telephony system. We can design a spoken dialogue system which is combined with nine kinds of basic components and two kinds of enhancing components by using GUI. We can also test and execute it. The tool can output not only Ruby script file which describes the system, also PostScript which describes GUI canvas. However, at present the speech recognition module is not built in and the output Ruby scripts work only on the tool. We continue our researches to overcome these problems.

Using this tool we have just constructed the first version of the network information system. The system can investigate the network reachability from the system to the specified IP address, announces the network stop information and records the message to the network manager. We actually set up the system in Information Technology Center, Nagoya University, and begin the trial operation. We are going to improve the tool and the network information system examining the result of this trial operation.

References

[1] Tomas Nouza, Jan Nouza “Graphic platform for designing and developing practical voice interaction systems”, Proc. of Eurospeech 2001, pp.1287-1290, (2001)

[2] http://cslu.cse.ogi.edu/toolkit/

[3] http://www.ruby-lang.org