퍼지 및 지능시스템학회 논문지 2004, Vol. 14, No. 6, pp. 771-775

An Interactive Voice Web Browser Usable as a Multimodal Interface in Information Devices by Using VoiceXML

MinSeok Jang

Dept. of Computer Information Science, Kunsan National Univ.

Abstract

The present Web surroundings is mostly composed of HTML(Hypertext Mark-up Language) and thereby users obtain web informations mainly in GUI(Graphical User Interface) environment by clicking mouse in order to keep up with hyperlinked informations. However it is very inconvenient to work in this environment comparing with easily accessed one in which human's voice is utilized for obtaining informations.

Using VoiceXML, resulted from XML, for supplying the information through telephone on the basis of the contemporary matured technology of voice recognition/synthesis to work out the inconvenience problem, this paper presents the research results about VoiceXML VUI(Voice User Interface) Browser designed and implemented for realizing its technology and also the VoiceXML Dialog designed for the purpose of the browser's efficient use.

Key words : VoiceXML, Voice Web Browser, Interactive Voice Interface, Multimodal Interface, GVUI

1. Introduction

In the present or forthcoming Internet world, we need wired or wireless information devices including PC, PDA, phone, handheld PC, etc. to retrieve web-based information via their embedded multimodal interfaces such as Voice, DTMF, Keyboard, Mouse, Pen, and Visual Interface, etc [1]. VoceXML [2, 3] is designed to make Internet content and information accessible via voice and phone. VoiceXML standardization is in progress by four major enterprises including AT&T, IBM, Lucent Technology, Motorola. In May 2000, its version 1.0 specification is submitted to W3C, and in 2004, version2.0 spec. is being standardized in state of PR(Proposed Recommendation).

The existing speech recognition and synthesis engines can be integrated with VoiceXML so that the popular GUI interfaces can be replaced with VUI(Voice User Interface). VoiceXML helps the developer to focus on the dialog scenario instead of voice engines [4]. Thus VoiceXML is able to address the issues of the existing mechanic and electronic voice system and web based voice system [5].

However, the existing systems integrated with VoiceXML focuses mainly on VoiceXML related speech engines and developing tools (IBM [6]: Voice Toolkit, MS [7, 10]: Speech API, Nuance [8]: Speech Recognition Engine, Voice Platform, Voice Web Server, SpeechWorks [9]: OpenSpeech server). Furthermore they provide only VUI to clients so that it is not convenient to the clients, especially in the Web environments. That is, they do not support the popular HTML information of the World Wide Web. Besides, voice only interfaces make the users feel so complicated to acquire information. Therefore, we present our product VoiceWeb V1.0 in this paper to address the issues of the existing VUIs with the solution GVUI (Graphical & Voice User Interface); we apply XML Island technique to a server site in order to integrate HTML with VoiceXML; for a client site, we implement the voice web browser that supports a VoiceXML and HTML text. The existing VoiceXML systems provide only VUIs that make the dialog scenario be more complicated to give a same amount of web information than GVUIs because they could handle not text and graphical information but only information accessible via voice.

접수일자 : 2004년 9월 1일
완료일자 : 2004년 10월 15일
감사의 글 : 본 결과물은 정보통신부의 정보통신기초기술연구지원사업(정보통신연구진흥원)으로 수행한 연구결과입니다.

2. Review

The existing speech information systems applying speech recognition/synthesis technologies are broadly classified into two ways, mechanically electronic and web-based one. In the former, ARS, CTI, VAD(Voice Activated Dialing), PDA with speech recognition function, etc have been commercialized. But they mostly are partially coupled with speech recognition/synthesis technique, and furthermore their given information is fixed and restricted, and their related tools are not universally available. And the system is difficult to extend the function because only expert is able to construct and maintain it. Thus it takes much more cost than the latter to utilize the former. The web-based way

which can solve the above problems is also classified into two ways on whether it applies VoiceXML or not. STG's WebGALAXY [11], IBM's Annotation-based Transcoding System [12], etc are non-VoiceXML applications which provide its one-way voice interface transforming the HTML text information into speech format, or operating the menu or link by voice. On the contrary, VoiceXML has the following characteristics which are reason for being able to give flexible and efficient interface solving the above problems and thus helping developers to concentrate only on making out dialog scenario [4];

- separating users' interactive code from service logic which property relieves developers from low-level programming or resource management.
- integrating and giving voice service and DB service in client/server environment.
- because VoiceXML web server executes various service logics and thus enables to produce the VoiceXML documents dynamically, it is possible to serve the updated information very rapidly.
- the existing web environment is usable in VoiceXML one.

Furthermore complex scenario is expressible because it is possible for VoiceXML to allow script code and computational function in the document, and thus complex information treatment is available.

VoiceXML supporting system is comprised of application, DB, VoiceXML gateway, speech recognizer and synthesizer(TTS; Text To Speech) which give all together voice service. For example, from the existing computer vendors to the speech recognition related ones, they produce kinds of systems, which are IBM's voice toolkit in its websphere server [6, 13], MS's speech API 5.0/5.1, Nuance's Nuance 8.0 speech recognition engine, NVP(Nuance Voice Platform), Nuance's Voice Web Server 2.0, Speechworks's OpenSpeech Server. But they are mainly concerning VoiceXML related engine, developing tools, and processing most functions on server side. Therefore present VoiceXML-based systems have the following problems; At first, their user interface is very inconvenient to use, because they are operated only in GUI way not VUI one. Secondly, the served information is restricted, that is, the existing unlimited HTML information is not fully given in the systems, because VoiceXML itself has the other properties than HTML. Thirdly, complicated and interactive services is not able to be provided. Because, in the existing systems, their information is requested and responded only to voice and thus complicated and programmable construction of dialog(sequence, selection, iteration elements) are not possible.

To solve these problems, in this paper, universal GVUI is designed and implemented on client side which is able to give HTML information too. On server side, by applying "XML Island" technique in making out VoiceXML document, the existing HTML document's information is accessible and also interactive dialog construction method is suggested. This method will also provide the natural interface for information acquisition with users by giving a basis for building up the interface of wired or wireless information devices as embedded system.

3. Development Method/Procedures

3.1. Design and implementation of a client side GVUI Browser

The proposed browser's architecture and its behavior are as follows;

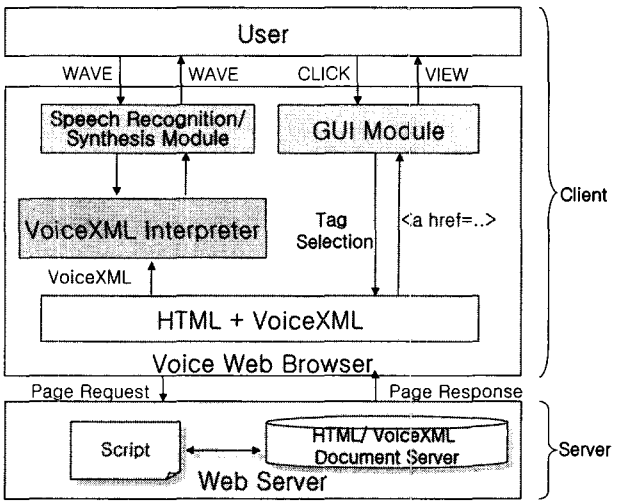

Fig. 1. Architecture of GVUI Browser

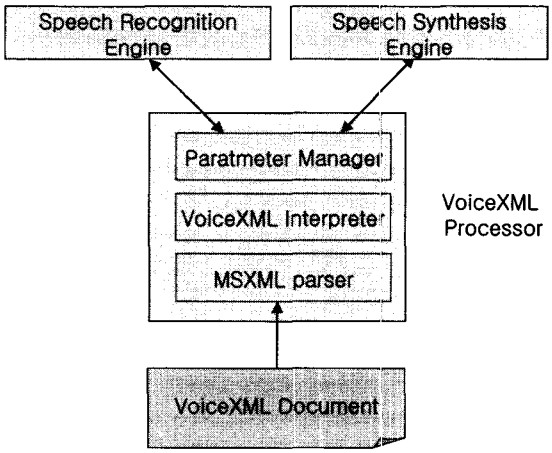

Fig. 2. VoiceXML Processor's Functional Diagram

At first MSXML parser in VoiceXML processor checks whether the VoiceXML document is valid against VoiceXML ver 1.0 DTD or not, and then VoiceXML interpreter performs the related functions using DOM. At this time, parameter manager manages parameters used in VoiceXML document and the additive information of

commands being processed by ECMAScript, and it executes the related functions reciprocally with VoiceXML interpreter. VoiceXML tags recognized in this browser are shown in Table 1 which are extracted after the unnecessary things are removed from VoiceXML 1.0 spec.

Table 1. Recognized Tag

Classification Type	Element(Tag)
Dialog	form, field, option, menu, choice
Branch	Goto
Condition	if, else, elseif, option
Voice	block, break, prompt
Connect	Link

The operation procedure of VoiceXML processor is as follows;

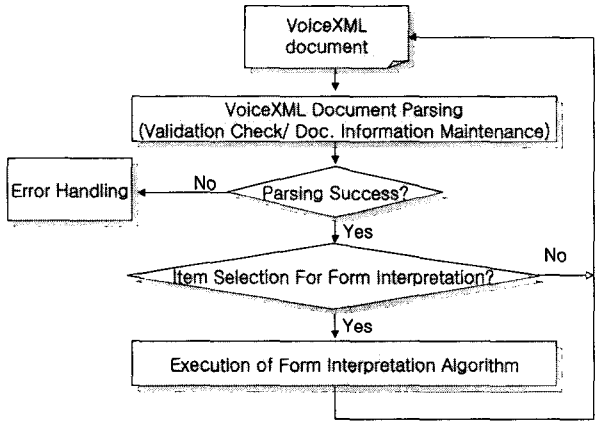

Fig. 3. Operation Procedure of VoiceXML Processor

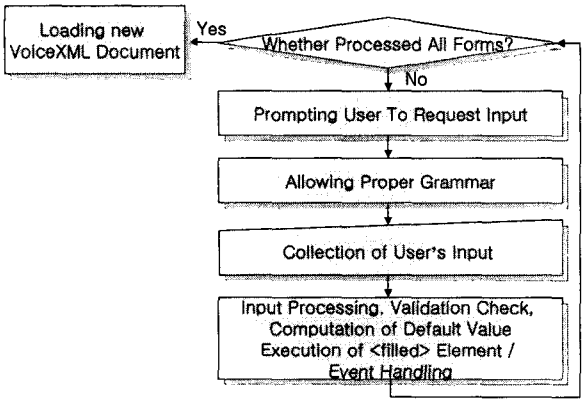

Fig. 4. FIA

At first, it parses the VoiceXML document, that is, decides the validity and then initializes and stores the parameter information of the loaded document. After that, according to the necessity of interaction, it runs FIA(Form Interpretation Algorithm) [14] or executes the transition to another documents.

In case of the document with multi-form, all forms are executed, or the another form or new document is executed at certain form according to the user's input. In case with one form element, <prompt> element requests a response or provides a certain information to users and the next process goes on. And during the processing, user's speech input is received through <filled> element and checked for validation, and then the next element is continuously performed.

3.2 Applying the XML Island technique

This paper has the existing HTML way accommodate the VoiceXML one naturally by using XML Island technique [15]. XML Island is the technique which helps HTML to supply more affluent applications by controlling data inserted in HTML source through various kinds of script code utilizing XML DOM. The voice browser receives the HTML document(comprised of script and DB) from web server and extracts VoiceXML document which is XML island part encapsulated in the HTML source. The document is processed by speech recognition/synthesis module. Here the anchor tag(ex; <a href="...">) in HTML are browsed as like the existing method and therefore the proposed method accommodates the HTML source. This method gives voice interface supporting the existing web to users, and also has developers construct their web site with ease only by being familiar with simple VoiceXML grammar. As shown in the example document, a variety of dialog scenarios can be made using <filled>, <field> elements.

```
<HTML>
<TITLE>·····</TITLE>
<HEAD>·····</HEAD>
<BODY>
<XML ID="vxml">
<vxml version = "1.0">
<form>
  <field name="selection">
  <prompt> Answer your choice out of weather,
        traffic, news information! </prompt>
  <filled>
    <if cond="selection=='weather'">
        <goto nextitem = "weather"/>
    <elseif cond="selection=='traffic'"/>
        <goto nextitem = "traffic" />
    <elseif cond="selection=='news'"/>
        <goto nextitem = "news" />
    <else/>
    </if>
  </filled>
  </field>
  <field name = "weather">
  <prompt>Today's weather will be given
```

```
        ···</prompt>
  <link next = "weather.html"/>
  </field>
  <field name = "traffic">
  <prompt>The present traffic information will be
        given ··· </prompt>
  <link next = "traffic.html"/>
  </field>
  <field name = "news">
  <prompt> Today's main news will be given
        ...</prompt>
  <link next = "news.html"/>
  </field>
</form>
</vxml>
</XML>
</HTML>
```

3.3 Construction of Interactive Interface

This section shows a method of making out VoiceXML documents in cases of menu selection scenario, one of the mostly used in web environment. In the present portal web site, the property of the site is not recognized easily and it is very difficult to find out its particular menu and to go on surfing mainly because of its many menus and splendid graphics. On the contrary, the voice interface leads users spontaneously to surf web sites on basis of request/response. In addition to that, the method is designed to support not only VUI but also the existing GUI. And thus it operates reciprocally with both ways of interfaces. This speech scenario operates as in Fig. 5.

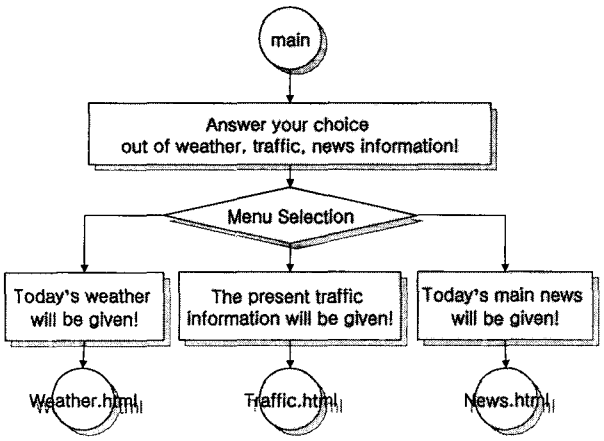

Fig. 5. Speech Scenario's Flowchart

Because in the past VoiceXML service, text or graphical information is not given, its structure must be complicated to solve the problem. But the proposed GVUI is able to express those information in the above simple structure. It is possible because the scenario is dependent on HTML page.

4. Results

4.1 System Implementation and Its Environment

Table 2. GVUI Browser's Developing Environment

Item	Contents
OS	Windows 2000 Server Service Pack 3
Developing Programming Language	C# (.NET Framework)
XML Parser	MSXML 3.0
Developing API	-speech recognition; SAPI 4.0, MSAGENT 2.0 -Korean TTS; L&H's TTS 3000

The programming language, C# on Dot Net framework is used which gives the platform independent developing environment as like Java.

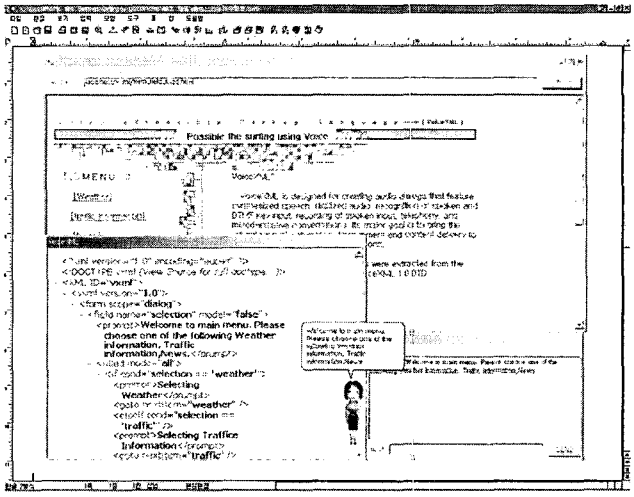

Fig. 6. Constituent Windows of GVUI Browser

The browser consists of several windows including the one showing HTML document, the one showing VoiceXML part, and the one representing voice interface visually. Here, speech input can be inserted in parallel with text input. And the output is given in both forms of speech and text with a character as shown in Fig. 6.

4.2 Comparison Results

Table 3. Differences with the existing VoiceXML-based Speech Recognition Browser

Items	Proposed GVUI	Existing VoiceXML-based Tools and Servers
Products	VoiceWeb V1.0	IBM: Voice Toolkit MS:

		Speech API 5.0/5.1 Nuance: Nuance 8.0 Speech Recognition Engine, Nuance Voice Platform(NVP), Voice Web Server 2.0, SpeechWorks: OpenSpeech Server
Location of Processor	Client Side	Server Side (burden on Server Side)
Transmission Format	Text	Speech (burden on both Network and Server)
Client devices	All wired or wireless Information Devices	Wired Telephone
Interface Type	GVUI (Graphical + Voice) ∴ Convenient	VUI(only voice) ∴ Inconvenient
Passing Network	Web	PSTN
Support of HTML	O	X
Amount of Information	Unlimited	Restricted
Extensibility of PL	O (∵ .NET Platform)	△
Dialog Scenario	Simple	Complicated

5. Conclusions

The existing VoiceXML-based applications provide information services through telephone via PSTN. Their browser reaches the level of only simulating VoiceXML and giving an one-sided, restricted service and thus is insufficient to practice multimodal interface. This paper suggested the method which solves the main disadvantages including unsatisfactory interactive interface, refusal of HTML information, and also shows the efficient resulted browser. It will give a basis of interactive multimodal interface accessible to a unlimited size of web information.

6. References

[1] W3C Multimodal Interaction Activity, http://www.w3.org/2002/mmi/

[2] VoiceXML Forum, http://www.voicexml.org

[3] VoiceXML 2.0 Spec., http://www.w3.org/TR/voicexml20/

[4] Peter j. Danielsen, "The Promise of a Voice-Enabled Web", IEEE Computer, VOL.33, NO.3, pp.104-106, Aug. 2000

[5] Snowshore Networks, http://www.snowshore.com

[6] IBM Voice Toolkit, http://www-4.ibm.com/software/speech/enterprise/vtoolkit.html

[7] Microsoft Speech, http://www.microsoft.com/speech/

[8] Nuance, http://www.nuance.com/prodserv/prodnuance.html

[9] SpeechWorks OpenSpeech Server, http://www.speechworks.com/products/speechrec/index.cfm

[10] Microsoft Agent, http://www.microsoft.com/agent/

[11] R.Lau, G.Flammia, C.Pao, and V.Zue, "WebGALAXY: Beyond Point and Click a Conversational Interface To a Browser", in Proc. Sixth International World Wide Web Conference (M.R. Genesereth and A. Patterson, eds.), Santa Clara, CA, pp. 119-128, Apr 1997

[12] Chieko Asakawa et al, "Annotation Based Transcoding for Nonvisual Web Access", Proc. ASSET'00, pp.172-179, Nov. 2000

[13] IBM, http://www-4.ibm.com/software/webservers/appserv

[14] Stephen Breitenbach, et al, Early Adopter VoiceXML, Wrox Press Inc., p.300, Aug. 2001

[15] W3C DOM Requirements, http://www.w3.org/TR/DOM-Requirements

저 자 소 개

MinSeok Jang

-Feb. 1989; Bachelor Degree from Dept. of Electronic Eng., Yonsei Univ.

-Aug. 1991; Master Degree from Dept. of Electronic Eng., Yonsei Univ.

-Aug. 1997; Ph.D Degree from Dept. of Electronic Eng., Yonsei Univ.

-Sep. 1997 ~; Associate Professor at Dept. of Computer Information Science, Kunsan National Univ.

-Research Interests; Protocol Engineering, Software Engineering, Web-based Technology(XML)

Phone : +88-63-469-4557
Fax : +88-63-469-4560
E-mail : msjang@kunsan.ac.kr