SICE: An Enhanced Framework for Design and Development of Speech Interfaces on Client Environment

Prabhat Verma
Computer Science and Engg. Department, Harcourt Butler Technological Institute, Kanpur, Uttar Pradesh, India

Raghuraj Singh
Computer Science and Engg. Department, Harcourt Butler Technological Institute, Kanpur, Uttar Pradesh, India

Avinash Kumar Singh
Computer Science and Engg. Department, Harcourt Butler Technological Institute, Kanpur, Uttar Pradesh, India

ABSTRACT
VoiceXML has emerged as standard for developing IVRS based telephony applications. It provides the requisite robustness and control over dialogue based delivery over telephone line. This paper presents a framework based on VoiceXML specifications for developing web based interactive voice applications. Communications are made using VOIP; therefore, user does not have the dependency on telephony interface as required by existing VoiceXML specifications. Besides, it is able to handle the complex and voluminous information of web. The framework demonstrates the best practice of application design, development and software engineering.

Keywords
VoiceXML, VOIP, Voice User Interface, speech recognition, speech synthesis, SICE

1. INTRODUCTION
The Web has become an important medium for delivering information, and more and more people rely on it for work and entertainment. For example, users like to check e-mails, read news, watch videos, listen to music, ticket reservation and do shopping on the Web. An important domain for voice based web applications is to support the blind or people with other disabilities. Voice based interfaces can empower visually challenged people in many ways e.g. seeking information of their interest or making transactions through web. Unfortunately, real scenario is still not satisfactory altogether. Vast power and resources of web are still not usable for this underprivileged group. Many of the tools available do not cater even the basic need of the target group. The reason is lack of control over content access. Use of VoiceXML, which has been used in IVRS applications over telephone line, may provides the requisite robustness and control over web content. In this paper, we propose a framework, based on VOIP, which could be used to create voiceXML based web applications that provide VUI based interaction in controlled way similar to IVR Systems without requiring the telephone connection for interaction.

1.1 VoiceXML
HTML has roots in publishing. VoiceXML has a programming language background. It has the impression of a programming language: control constructs, variables, event handlers, nested scooping, and so on. At the beginning, VoiceXML was designed to be a programming language easy to learn, lightweight and interpreted for developing VUIs. VoiceXML renders content as speech and interacts with the user using speech recognition and speech synthesis technologies. The way of Voice XML structures for interaction with user is dialogs. A dialog consists of a sequence of prompts spoken by the computer and responses spoken by a person. Responses are given by the person by voice command or key using a keypad. In contrast with GUI windows which are multitasked and two dimensional, VUI Dialogs are sequential and linear by nature. Architecturally, VoiceXML interfaces are event-driven interfaces like GUIs. In a dialog, the computer speaks a prompt and then waits for the user to respond to it. The computer waits until a speech recognition event occurs. A speech recognition event is initiated by the speech recognition engine, which continuously analyzes the user's speech and attempts to match it to expected responses in the dialog. There are a lot of possible speech recognition events, including "recognized responses." "got a responses but didn't recognize it", "no response" and so on. Unlike GUIs and WUIs, where the events that drive the interface are low-level, non-equivocal incidences (button pressed, mouse clicked, and so on), events in VoiceXML interfaces are the result of complex, computation-intensive, possible processing with errors.

1.2 GUIs, WUIs, VUIs
GUI, WUI and VUI represent the major markup language-based browsing interfaces to the internet. GUI is the most fully developed of the three is exemplified by products such us Netscape Communicator and Microsoft Internet Explorer. WUI, the most next developed interfaces, is implemented by "micro browsers" embedded in wireless phones and personal digital assistants (PDAs). VUIs (Voice User Interfaces) are just beginning to appear as browsing interfaces to the internet, and they are driven by the standardization of VoiceXML 1.0. The markup languages for these three types of interfaces are all based on XML. XML is a markup meta language derived from SGML. XML simplifies some of the complex and little-used features of SGML, but it still provides a flexible and extensible base for defining specialized markup languages.

2. EXISTING SPEECH INTERFACE FRAMEWORKS
This section describes the major speech interface frameworks currently available.

2.1 W3C recommended Speech Interface Framework
- **VoiceXML 2.0 [1],** VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of

Web-based development and content delivery to interactive voice response applications.

- **VoiceXML 2.1 [2],** V*oiceXML 2.1* specifies a set of features commonly implemented by Voice Extensible Markup Language platforms, with a small set of widely implemented additional features like Referencing Scripts Dynamically, Using <mark> to Detect Barge-in During Prompt Playback, Using <data> to Fetch XML Without Requiring a Dialog Transition, <data> Fetching Properties, Using <foreach> to Concatenate Prompts and Loop through Executable Content etc.
- **Voice Browser Call Control (CCXML [3]),** CCXML is designed to provide telephony call control support for dialog systems, such as VoiceXML [VOICEXML]. While CCXML can be used with any dialog systems capable of handling media, CCXML has been designed to complement and integrate with a VoiceXML interpreter.
- **State Chart XML (SCXML),** SCXML provides a generic state-machine based execution environment based on CCXML and Harel State Tables. .
- **Speech Recognition Grammar Specification (SRGS) 1.0 [4],** this document defines syntax for representing grammars for use in speech recognition so that developers can specify the words and patterns of words to be listened for by a speech recognizer. The syntax of the grammar format is presented in two forms, an Augmented BNF Form and an XML Form. The specification makes the two representations mappable to allow automatic transformations between the two forms.
- **Semantic Interpretation (SISR) 1.0 [4]**, Semantic Interpretation for Speech Recognition offers semantic interpretation tags that can be added to speech recognition grammars to compute information to return to an application on the basis of rules and tokens that were matched by the speech recognizer.
- **Speech Synthesis Markup Language (SSML) 1.0 and 1.1 [4],** The Speech Synthesis Markup Language Specification is one of the standards and is designed to provide a rich, XML-based markup language for assisting the generation of synthetic speech in Web and other applications. The essential role of the markup language is to provide authors of synthesizable content a standard way to control aspects of speech such as pronunciation, volume, pitch, rate, etc. across different synthesis-capable platforms.
- **Pronunciation Lexicon Specification (PLS) 1.0**, provides the syntax for specifying pronunciation lexicons to be used by Speech Recognition and Speech Synthesis.

2.2 Other Framework for speech Interface

- **SUIML (Speech User Interface Markup Language),** is eXtended Markup Language XML) application [5] that specifies conversations between man and machine [6]. A SUIML document describes a set of objects that represent almost all the information needed in a conversation except for grammars.
- **VoiceBuilder[7],** is framework for building speech applications with no programming effort, based on a markup language and an algorithm using code templates. It was able to generate fully functional speech applications using both system initiative and mixed initiative dialogue strategies. It is a type of Interactive Development Environment(IDE)
- **Web Access by Voice (WAV)[8],** is the integration of many different technologies such as automatic speech recognition, scripts for web navigation, text to speech conversion, with a novel way of extracting information from web via voice in a programmatic manner.This is a utility which provides voice interface on operating system.
- **CoScripter[9],** is a programming by demonstration solution, which helps the blind user perform a pre-define internet task more efficiently. Overall, it is a task oriented voice interface.
- **Spoken Dialogue System (SDS)[10],** is designed for providing automatic dialogue-based voice services accessible through telephone like VoiceXML effectively. It works same as VoiceXML.
- **SpeechPa[11],** is an intelligent speech interface for PA(Personal Assistant) in research and development projects.
- **WebAnima[12],** is a web-based embodied conversational assistant agent which provides conversational interface and ontologies that support semantic interpretation.
- **GeoVAQA [13],** is a restricted Domain Spoken Question Answering system in the scope of the Spanish geography. The system consists of a web based application that allows speech recognition based on HMM model and sends back a concise textual answer.
- **A Voice-Activated Web-based Mandarin Chinese Spoken Document Retrieval System[14],** is an integrated technology for both spoken document retrieval and voice-activated WWW browser with a specific type of speech recognition technique specially design for Chinese language.
- **Florence[15],** a dialogue manager with a more general approach that uses an extensible and flexible framework to combine interchangeable and interoperable dialogue strategies as appropriate to the task. Florence's declarative XML-based language facilitates the development of natural language applications and allows the dialogue author to encapsulate and reuse different algorithms between applications.

Overall, it is clear from the above list that a slew of different XML instances are currently being developed with the goal of making the Web voice-enabled [16,17,18,19,20]. The attention seems to be centered on developing a language in which the website developer or Voice-Web service provider can specify how the user can access Web content. In other words, it is up to the developer to create an XML specification of the speech-based interaction that a user can have with the server. VoiceXML [19] is emerging as the standard and essentially subsumes SpeechML in functionality. VoiceXML provides the programmer with the ability to specify an XML document that defines the types of commands a voice-enabled system can receive and the text that a voice-enabled system can synthesize. Similar to the Java Speech Markup Language, it allows the XML programmer to specify the attributes used to render a given excerpt of text. This may include information about parameters such as rate and volume. Beyond that, VoiceXML,

VoXML, and TalkML provide the infrastructure to define dialogues between the user and the system.

3. DESIGN OF SICE FRAMEWORK

3.1 Enhanced Functionality

The W3C speech interface framework incorporates Voice eXtensible Markup Language (VoiceXML or VXML), speech synthesis markup language (SSML), Speech Recognition Grammar Specification (SRGS), voice browser Call Control XML (CCXML) and Semantic Interpretation for Speech Recognition (SISR). VoiceXML controls how applications interact with a user through interactive voice response over telephone lines; the SRGS offers support for speech recognition; the CCXML provides telephony call control support and other dialog systems, while the SISR defines how speech grammars bind to application semantics [21]. These are able to handle small information via telephone but when the amount of information is in huge quantity and bears complex structure, telephony voice user interface becomes a frustrating tool for user. Being an enhancement over VoiceXML, SICE Framework may be used to design and develop both telephony based as well as web based speech interfaces.

3.2 Platform and Language

Microsoft Windows 7 has been the chosen operating system for this due to its popularity and wide acceptance. Further, due to flexibility provided by it in the development of applications, .Net platform with C# language has been used for development of prototype SICE Framework. Microsoft Language Interface provides better interface in term of functionality and processing power.

3.3 Emphasis on enhancements in Human-Computer interaction

Emphasis is given on better Human-Computer Interaction (HCI) and ease of use through controlled output so that user feels a pleasant experience even during handling interactions of complex nature. The aim is also centered on saving time and efforts in getting information of interest. The detailed schematic diagram and Structure of framework is shown in fig 1 and fig 2 respectively.

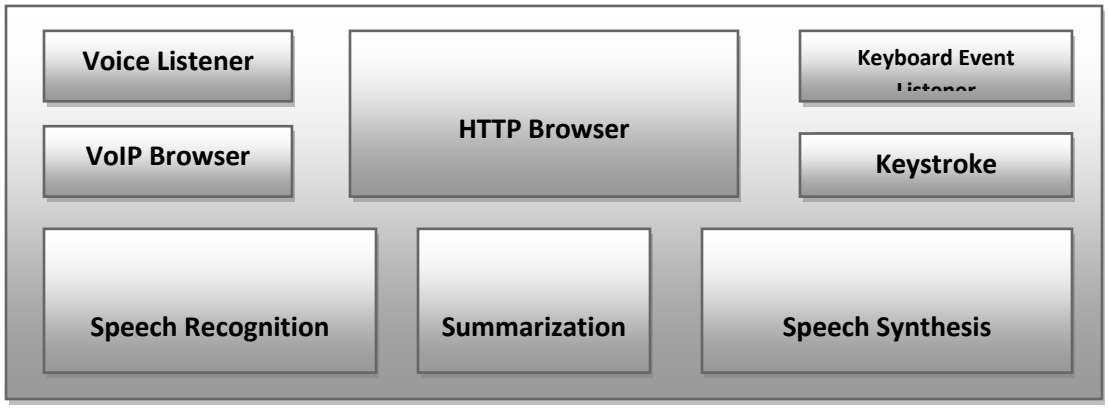

Fig 1: Schematic Diagram

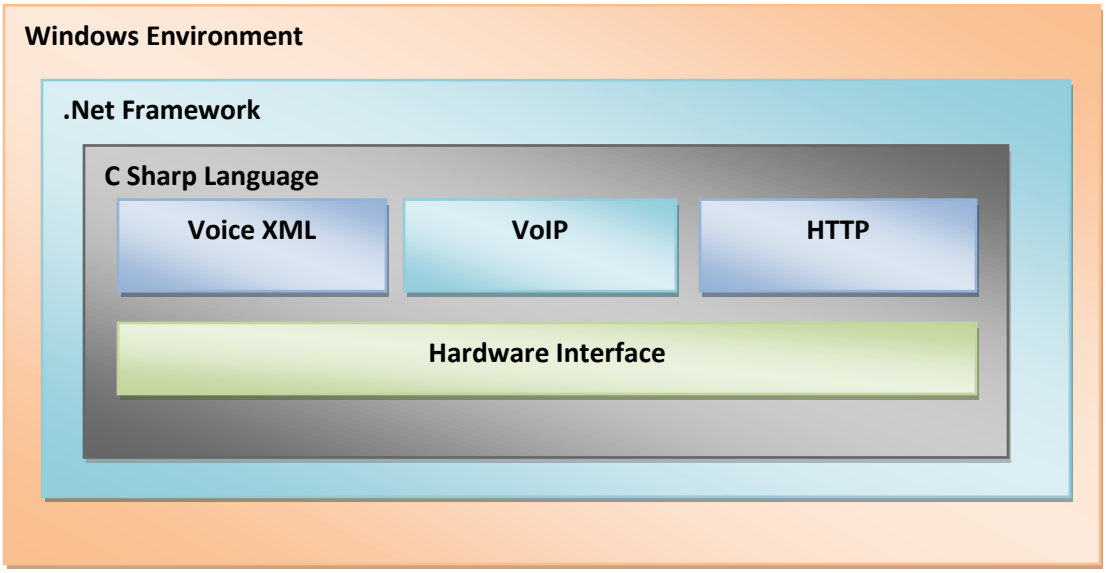

Fig 2: Structure of Framework

In the following subsections, we describe the main components of SICE with their role and responsibilities in details.

3.3.1. HTTP Event Handler

HTTP handles the response made from the request in the form of voice that passes through VoIP Interface and the response in the form of request webpage. It follows the following algorithm after taking request from user over Voice:-

[Assume "|"= OR and URL=Uniform Resource Locator; Request send through VoIP]

- [HTTP Event Handler]
- Start
- Request Page= absoluteURL | abs_path (getting from Speech to Text Conversion on speech server)
- Response = Status-Line; (General-Header;| Response-Header;| Entity-Header); CRLF [Entity-Body];
- Interpret the Response and check validity of page
- If(Page is Valid)
  Display the content in the form of HTML on browser.
- Else
  Shown Message "Some Error on requested webpage"
- End
- [Response Buffer=NULL]

3.3.2. VoIP Event Handler

This module provides two way communications over the internet in a way similar to telephone talk. SICE Framework makes the use of VoIP to take the voice from client side to server side in order to perform processing and returns the response from Speech Synthesizer to HTTP Handler.

3.3.3. Keystroke Handler

Keystroke is the event occurred when a key is presses on keyboard or similar device. The Key stroke handler module generates and processes the request to server similar to VoIP handler when a keystroke occurs.

3.3.4. Speech Recognizer

The speech recognizer resides on server side and always remains in listening mode. However, it is triggered to recognition mode only when a specified word with fixed threshold frequency is occurred. The speech recognition process follows five steps[4]:

a. Audio input: The human voice is transmitted through a microphone connected to a PC with help of standard sound card. The recommended microphone must have the noise cancellation feature so that the actual voice with minimum background noise is received by the speech server.

b. Acoustic processor: The acoustic processor filters out background noise again and converts the captured audio into a series of phonemes.

c. Word matching: The software attempts to match the sounds to the most-likely words in two ways. First, it uses acoustical analysis to build a list of possible matches that contain similar sounds. Then, it uses language modeling (the likelihood that a given word appears between those coming before and after it) to narrow the list to the best candidates. In addition, the word-matching process draws on the user-defined domain (the set of vocabularies, pronunciations, and word-usage models, as well as a model of the user's speech and words). The user can extend the domain by adding new words and can create multiple domains for different applications. Finally, continuous-speech SR examines contextual information to predict what words should come next in the current phrase. This also helps the system to distinguish among homonyms.

d. Decoder: The decoder selects the most-likely word based on the rankings assigned during word matching and assembles the word along with those selected earlier into the most-likely sentence combination.

e. Text output: This module sends the text transcription directly into a separate word processing program which prepares it for the next phase.

3.3.5. Speech Synthesizer

This part of SICE Framework resides on server side providing output (response) over client side in speech form. It works with natural voice especially in Indian context.

3.3.6. Content Summarizer

This component provides the control over the output text content based on the summarized text and keywords extracted from the document using an algorithm based on term co-occurrence. The algorithm mention below and process shown in fig 3:-

- Generate the terms set by removing the meaningless words like articles, preposition etc.
- Calculate the frequency of each term or word and select the first n words.
- Maximum(n)<=Total no of different words calculated in current document
- Calculate set of co-occurred word.
- Generate a Logical graph G = (V,E), where V=first n words and E=set of co-occurred words.
- Create cluster based on frequency ranking or in equal size
- Extract subject term using ,Subject Term = $\sum_{w,w' \in E(G')} C(w,w')$
  Where C(w,w') = R(w|w') /2 + R(w'|w) /2
  And R(w|w') =f(w,w') / f(w'') ,where f(w,w') is the number of co-occurrences of terms, and f(w') is number of occurrences of w'.
- Generate the summarized text based on term co-occurrence graph.
- Convert text in VoiceXML format by inserting <tag></tag> for a paragraph if occurred

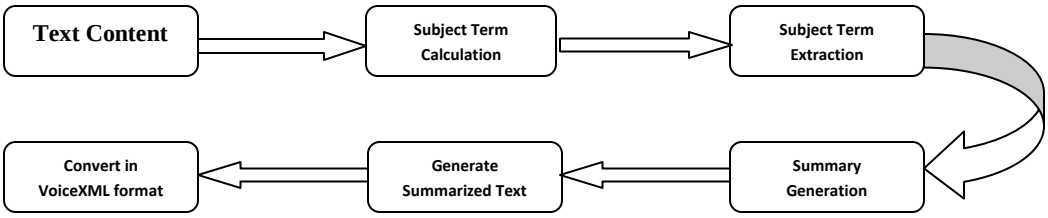

Fig 3: Summarization Algorithm Process

3.3.7. Interface

It integrates the Hardware components i.e. Keyboard, Microphone and Speaker with the SICE Browser so that user can listen and browser can recognize speech and process it by sending request via SICE Browser.

3.3.8. Encrpter-Decrypter

This component is an added security feature in SICE Framework, which provide secure way of transaction/Voice over HTTP and VoIP either in the way of Voice or Text or Keystroke.

4. SPEECH BASED WEB APPLICATIONS DEVELOPMENT USING SICE FRAMEWORK

SICE Framework is good at developing speech based web applications that provide specific functionality to target group e.g. visually challenged. Making query for seat availability/PNR status or exam result may be few such examples. Important is to identify and incorporate those functionalities of the website which would help the target group in substantial way. The working of an application built using SICE framework is demonstrated in the following subsection.

4.1 As a Result Checker

When user says "University Result" (keyword already stored for the URL of this page) in front of SICE Browser, this request is recognized by the Speech Server via VoIP.

As the keyword "University Result" is recognized by server, an event is generated which makes request to actual server of HBTI Website to fetch the required webpage and converts it into VoiceXML format for further processing. The requested page is sent to the user in HTML format also.

In the response user speaks the digit of their roll number as "one, two, three, four, five, six, seven, eight, nine, ten". These digits are recognizing on server that provides input to requested page "University Result".

When all required input is provided then SICE server sends request for the relevant page to University server for result and the result page, after converted into VoiceXML, is sent to the user along with the HTML file. Thus, it starts speaking out the result in a predefined format e.g. "You have obtained 65 marks out of 100 in Data Structures". Figure 4-7 illustrate the process.

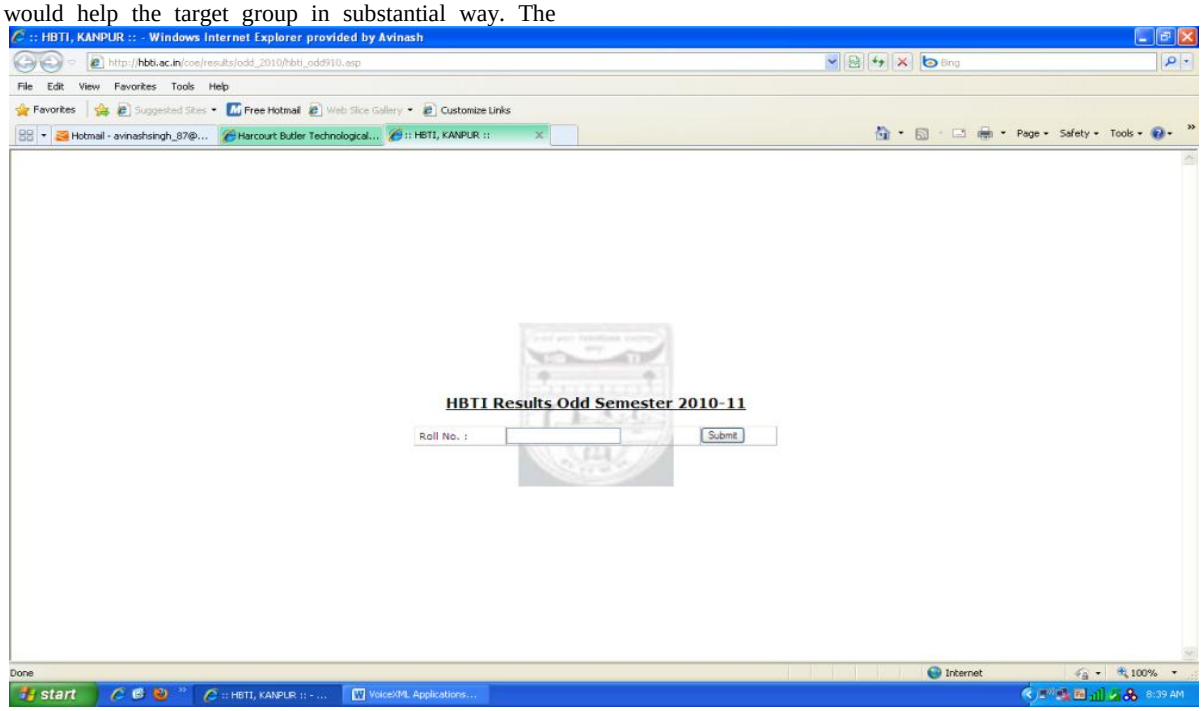

Fig 4: Requested Page

```
1
2  <html>
3  <head>
4  <title>:: HBTI, KANPUR ::</title>
5  <style>
6  a:link{COLOR: #666666; TEXT-DECORATION: none}
7  a:visited{COLOR: #666666; TEXT-DECORATION: none}
8  a:active{COLOR: #666666; TEXT-DECORATION: none}
9  a:hover{COLOR: #666666; TEXT-DECORATION: underline}
10 .s1{font-family: 'Courier New'; font-size:12px}
11 </style>
12 </head>
13 <body topmargin="0" leftmargin="0" style="font family: Verdana; font size:
    8pt; background-image: url('watermark.jpg'); background-repeat: no-repeat,
    background-attachment: fixed; background-position: center">
14 <table align='center' border='0' cellpadding='0' cellspacing='0'
    style='border-collapse: collapse' width='780' height=100%>
15 <tr><td height=96% align=center><BR><BR>
16 <B><u>HBTI Results Odd Semester 2010-11</u></B><BR><BR><table border="1"
    cellpadding="0" cellspacing="0" width="50%" style="border-collapse:
    collapse; font-size: 11px"><FORM METHOD="get" name="f1"><tr><td> Roll
    No. : </td><td><input type="text" name="rollno" maxlength="12" size="20"
    style="font-size: 10px"></td><td><input type="button"
    onClick="document.f1.submit()" size="20" value="Submit" style="font-size:
    10px"></td></tr></FORM></table>
```

Fig 5: Requested Page Source

```
1  <vxml version="2.1">
2    <form id="MainMenu">
3      <field name="RollNumber">
4          Please say your roll number.
5        <!-- grammar -->
6        <grammar type="text/gsl">
7        <![CDATA[
8          ;Match one of the enclosed digits
9          [
10           one two three four five six seven eight nine zero
11         ]
12       ]]>
13       </grammar>
14
15       <!-- when user was silent, restart the field -->
16       <noinput>
17         I did not hear anything.  Please try again.
18         <reprompt/>
19       </noinput>
20       <!-- The user said something that was not defined in our grammar -->
21       <nomatch>
22         I did not recognize that character.  Please try again.
23         <reprompt/>
24       </nomatch>
25     </field>
26     <filled namelist="RollNumber">
27       <if cond="RollNumber == 'one'">
28         <prompt>one entered.</prompt>
29       ..........................................
30       <elseif cond="RollNumber == 'zero'"/>
31         <prompt>zero entered.</prompt>
32       <else/>
33         <prompt>
34           A match has occurred, but no specific if statement
35           was written for it.
```

Fig 6: VoiceXML File of Requested Page

HBTI Results Odd Semester 2010-11

Name:	SHIVAM GUPTA
Father's Name:	SHAILENDRA KUMAR GUPTA
Roll No:	0704586017
Semester :	SEMESTER - 7
Course/Branch:	B. Tech. Leather Technology
Institute Name	Harcourt Butler Technological Institute,Kanpur

MARKS DETAIL

Subject Name	Max. Marks			Marks Obtained		
	Examination	Sessional	Total	Examination	Sessional	Total
Instrumentation and Process Control **[TCH706]**	100	50	150	30	18	48*
Chemical Reaction Engineering **[TCH707]**	100	50	150	10	20	30*
Processing of Leather I **[TLT701]**	100	50	150	73	37	110
Tannery Effluent Treatment **[TLT702]**	100	50	150	79	40	119
Enterpreneurship Development Programme **[ELE-OP]**	100	50	150	62	36	98
Leather Processing Lab II **[TLT751]**	60	40	100	52	30	82
Project (Team Work) **[TLT752]**	0	50	50	-	38	38
Industrial training **[TLT753]**	0	50	50	-	39	39
General Proficiency **[GP701]**		50	50		44	44

CARRY OVER PAPER	TCH706,TCH707,
RESULTS	CP(2)
TOTAL MARKS	608

1) Although utmost care has been exercised in preparation of marks; yet if at any stage any error is detected based on facts; these marks will be treated as null and void and fresh factual marks would be given.
2) If it is detected at any stage that a student appeared in the examination in violation of admission /examination rules/norms, the statement of marks given herein will be treated as null and void.

Fig 7: Final Page after providing the required input

5. CONCLUSIONS

The SICE framework presented in this paper can be used for design and development of speech based web supporting applications. Advantages are manifold: It provides flexibility like IVRS without requiring telephony. Complex web data can be conveniently handled using customized two way dialogue based access system in a controlled way. The framework is more effective in developing speech based web access systems for dedicated functionalities rather than developing generalized speech interfaces. Website providers themselves can use the framework to provide speech based access to the visually challenged selectively for more important functionalities/ portions. The enhanced functionalities shall be a great contribution to the society in general and to the visually challenged in particular.

6. ACKNOWLEDGMENTS

This research is a part of the Major Research Project entitled "Design and Development of Web Browser for Visually Challenged" funded by the University Grants Commission, New Delhi running in Computer Science & Engineering Department of Harcourt Butler Technological Institute, Kanpur.

7. REFERENCES

[1] VoiceXML 2.0 http://www.w3.org/TR/voicexml20/ accessed 26th March, 2011)

[2] VoiceXML 2.1 http://www.w3.org/TR/voicexml21/ (accessed 26th March, 2011)

[3] W3C Voice Specification http://www.w3.org/Voice/

[4] J. Markowitz, Using Speech Recognition, Prentice-Hall,Upper Saddle River, NJ, 1996.

[5] W3C: Extensible Markup Language (XML) 1.0 (Third Edition), Feb 2004. http://www.w3.org/TR/2004/REC-xml-20040204/

[6] Montiel-Hern´andez, J. and Cuay´ahuitl, H., "SUIML: A Markup Language for Facilitating Automatic Speech Application Development," in proceedings of theMICAI'04 (WIC), Mexico City, Mexico, Apr 2004.

[7] Heriberto Cuay´ahuitl, Miguel ´Angel Rodr´ıguez-Moreno, and Juventino Montiel-Hern´andez,"VoiceBuilder: A Framework for Automatic Speech Application Development"

[8] Himanshu Chauhan, Pankaj Dhoolia, Ullas Nambiar ,Ashish Verma, "WAV: Voice Access to Web Information for Masses"

[9] J. P. Bigham, T. A. Lau and J. W. Nichols, "TrailBlazer: Enabling Blind Users to Blaze Trails Through the Web", submitted to International Conference on Intelligent User Interfaces, Florida,2009.

[10] Stanislav Ondáš and Jozef Juhár, "Development and Evaluation of the Spoken Dialogue System Based on the W3C Recommendations"

[11] E. C. Paraiso, and J.-P. A. Barthes; "An Intelligent Speech Interface for Personal Assistants in R&D Projects". In: CSCWD 2005 - The 9th IEEE International Conference on CSCW in Design, Coventry - UK, v. 2. pp. 804-809, 2005.

[12] Emerson Cabrera Paraiso, Yuri Campbell, Cesar A. Tacla.," WebAnima: A Web-Based Embodied Conversational Assistant to Interface Users with Multi-Agent-Based CSCW Applications"

[13] *Jordi Luque, Daniel Ferrés, Javier Hernando, José B. Mariño and Horacio Rodríguez.," GeoVAQA: A VOICE ACTIVATED GEOGRAPHICAL QUESTION ANSWERING SYSTEM".*

[14] Hsin-min Wang, Berlin Chen, Liang-jui Shen, and Chao-chi Chang." A Voice-Activated Web-based Mandarin Chinese Spoken Document Retrieval System".

[15] Giuseppe Di Fabbrizio, Charles Lewis. "Florence: a Dialogue Manager Framework for Spoken Dialogue Systems".

[16] Java Speech, http://java.sun.com/products/java-media/speech/ (accessed 22nd May, 2011).

[17] SpeechML, http://www.alphaworks.ibm.com/formula/speechml.

[18] TalkML, http://www.w3.org/Voice/TalkML/.

[19] VoiceXML, http://www.voicexml.org/.

[20] VoXML, http://www.voxml.org/.

[21] J. Daly, M. Forgue, Hirakawa, World Wide Web Consortium Issues VoiceXML 2.0 and Speech Recognition Grammar as W3C Recommendations, available online at: http://www.w3.org/2004/03/voicexml2-pressrelease (accessed 26th March, 2008).

.