Design and Implementation of Enhanced Real Time News Service Using RSS and VoiceXML

Hyeong-Joon Kwon[1], Jeong-Hoon Shin[2], and Kwang-Seok Hong[1]

[1] School of Information and Communication Engineering,
Sungkyunkwan University,
300 Chunchun Dong, Jangan-gu, Suwon, Kyungki-do, 440-746 Korea
katsyuki@skku.edu, kshong@yurim.skku.ac.kr
http://hci.skku.ac.kr
[2] School of Computer and Information Communications Engineering,
Catholic University of Daegu,
300 Geumnak 1-ri, Hayang-eup, Gyeongsan-si, Gyeongsangbuk-do, 712-702 Korea
only4you@cu.ac.kr
http://only4you.or.kr

Abstract. In ubiquitous computing, most people need to track various sources of news using various devices, but it becomes difficult once there are more than a handful of sources. This is the reason for this is that users have to navigate to each page, load it, remember how it's formatted, and find where they last left off in the list. To solve these problems, many service providers provide RDF Site Summary documents. In this paper, we propose a newly designed news service using RSS and VoiceXML. RSS is an XML format that supports the syndication of news stories and similar content. There are several different formats for XML syndication that are referred to as RSS. Since RSS is in XML format, turning it into VoiceXML is easy and the synergy benefits of binding RSS and VXML are great. VoiceXML is a non-proprietary, web-based markup language for creating vocal dialogues between humans and computers. In this paper, we first focus on binding RSS and VXML, and RSS feed parsing. As a result of this research, we implement enhanced real time news service. People can use our service with their wired and wireless phone at any time, at any place. Also, we validate usability by comparing a typical RSS service scenario and typical VoiceXML service scenario, and calculating user's satisfaction.

1 Introduction

The World Wide Web is a huge collection of different sites, and are all updated at different times. The web has quickly grown from a modest hypertext system of interest to computer researchers to a ubiquitous information system that includes virtually all human knowledge. To determine whether or not a site has been updated, a user would have to go to each site and attempt to recollect what information they saw previously, and then find what data, if any, is new. Most users of the web have a collection of sites they frequent to get information. But, nowadays, it is difficult to get the necessary data from the vast scale of web sites. That's because it exists too much information in the web. Users want to get the right data from web sites at the right

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part I, HCII 2007, LNCS 4557, pp. 677–686, 2007.

time with the minimum amount of effort. These kinds of user's needs motivate the change of web sites into a more standardized structure. An increasingly large amount of data is structured, stored, and sent over a network using XML. One such example is Really Simple Syndication (RSS) [1], [2].

RSS has gained popularity thanks to the increased use of web pages and news subscriptions. A site can make its updates available through an RSS feed, a file located within the site that contains the most recently added items in XML format. An RSS feed is written in XML. A feed comprises a channel, which has a title, link, description, etc, followed by a series of items. The real benefit of RSS, apart from the speed of looking at many different sites, is that all the feeds are chosen by the user. No user has the power to set their agenda, and crucially no-one can intervene to send spam. With thousands of sites now RSS-enabled and more on the way, RSS has become perhaps the most visible XML success story to date [2].

In this paper suggests a prototype of a dialog system combining VXML (VoiceXML), which is the W3C's standard XML format for specifying interactive voice dialogues between human and computer, and RSS(RDF Site Summary or Really Simple Syndication), which a representative technology of the semantic web for syndication and subscription of updated web-content. The merits of the proposed system are as follows: 1) It is a new method that recognizes spoken content using wire and wireless telephone networks and then provides content to users via STT(Speech-to-Text), TTS(Text-to-Speech) and visual environment using RSS, 2) It can apply advantages of RSS, where the subscription of updated content is converted to VXML, without modifying traditional methods to provide RSS service, 3) In terms of users, it can reduce restrictions on time-space, in search of content provided by RSS, because it uses wire and wireless telephone networks, not the internet environment. 4) In terms of information provider, it does not need special equipment, design of difficult STT and TTS algorithm, for syndication of new content using speech recognition and synthesis technology. We implemented a news service system using VXML and RSS for performance evaluation of the proposed system. In the experiment results, we estimated the response time and the speech recognition rate in subscription and search of actual content, and confirmed that the proposed system can provide content provided using a RSS Feed.

2 Related Works

In this chapter, we introduce core technologies for our research, that is, RSS and VoiceXML. Then, we introduce basic service architecture for speech applications.

2.1 RSS

Prior to RSS, several similar formats already existed for syndication, but none achieved widespread popularity or are still in common uses today, as most were envisioned to work only with a single service. These originated from push and pull technologies. Two of the earliest examples are "Backweb" and "Pointcast" [3].

Between 1995 and 1997, Ramanathan V. Guha at Apple Computer's Advanced Technology Group developed the Meta Content Framework (MCF). The MCF was a specification for structuring metadata information about web sites and other data, and the basis of Project X (aka Hot Sauce), a 3D flythrough visualizer for the web. When the research project was discontinued, Guha left Apple and went to work at Netscape, where he adapted MCF to use XML and created the first version of the Resource Description Framework (RDF). Then, in 1997 Microsoft created the Channel Definition Format for the Active Channel feature of Internet Explorer 4.0, however, the feature never became popular [2]. Really Simple Syndication (RSS) is a lightweight XML format designed for sharing headlines and other Web content. It can be interpreted as a constantly updated and distributed "What's New" summary for your site. Originated by UserLand in 1997 and subsequently used by Netscape to fill channels for Netcenter, RSS has evolved into a popular means of sharing content between sites (including the BBC, CNET, CNN, Disney, Forbes, Motley Fool, Wired, Red Herring, Salon, Slashdot, ZDNet, and more). RSS solves the myriad problems webmasters commonly face, such as increasing traffic, and gathering and distributing news. RSS can also be the basis for additional content distribution services [2], [3], [4].

2.1.1 Configuration

RSS files are often labeled as XML. RSS version 1.0 is also RDF (any version), which, again, is important only because an RSS file may be labeled as RDF. RSS files (which are also called RSS feeds or channels) simply contain a list of items. Usually, each item contains a title, summary, and a link to a URL (e.g. a web page). Other information, such as the date, creator’s name, etc., may also be included. The most common use for RSS files is for news and other reverse-chronologically ordered websites such as blogs. For example, a particular page on Fagan Finder has a change log, which is also available in RSS format. An item’s description may contain all of a news article, blog post, etc., or just an extract or summary. The item’s link will usually point to the full content (although it may also point to the item linked by the content itself). Figure 1 shows tree types of core elements of RSS 2.0 [2].

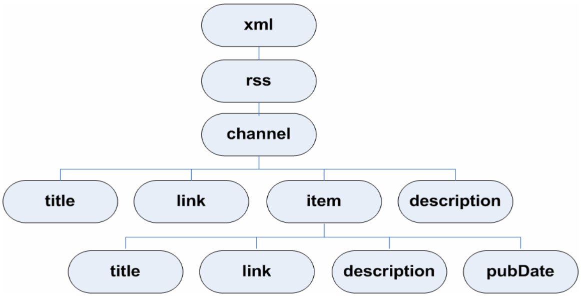

Fig. 1. RSS 2.0 core elements – Tree type

2.1.2 RSS Syntax

RSS is used to share content between websites. With RSS, we can register our content with companies called aggregators. First, create an RSS document and save it with

Table 1. Simplified version of an actual RSS feed

RSS Format in English	Example RSS Feed in English	Example RSS Feed in XML
Title	Joe's Breakfast News	<title>Joe's Breakfast News</title>
Link	www.joe.com/news/	<link>http://www.joe.com/news/</link>
		<item>
Item Title Link	Item Orange Juice Voted Best Fruit Juice www.joe.com/news/orange-juice.html	<title>Orange Juice Voted Best Fruit Juice</title> <link>http://www.joe.com/news/orange-juice.html</link> </item>
		<item>
Item Title Link	Item Acme Introduces New Flakes 'n' Nuts Cereal www.joe.com/news/flakes-n-nuts.html	<title>Acme Introduces New Flakes 'n' Nuts Cereal</title> <link>http://www.joe.com/news/flakes-n-nuts.html</link> </item>

an .xml extension. Then, upload the file to your website. Next, register with an RSS aggregator. Each day the aggregator searches the registered websites for RSS documents, verifies the link, and displays information about the feed so clients can link to documents of interest [2], [3].

2.2 VoiceXML

VoiceXML is the HTML of the voice web, the open standard markup language for voice applications. VoiceXML harnesses the massive web infrastructure developed for HTML, to make it easy to create and deploy voice applications. Similar to HTML, VoiceXML has created substantial business opportunities [1].

VoiceXML 1.0 was published by the VoiceXML Forum, which is a consortium of over 500 companies, in March 2000. The Forum then handed over control of the standard to the World Wide Web Consortium (W3C), and now concentrates on conformance, education, and marketing. The W3C has recently published VoiceXML 2.0 as a Candidate Recommendation. Products based on VoiceXML 2.0 are already widely available. While HTML assumes a graphical web browser with display, keyboard, and mouse, VoiceXML assumes a voice browser with audio output, audio input, and keypad input. Audio input is handled by the voice browser's speech recognizer. Audio output consists both of recordings and speech synthesized by the voice browser's text-to-speech system [1].

VoiceXML takes advantage of several trends:

- The growth of the World-Wide Web and its capabilities.
- Improvements in computer-based speech recognition and text-to-speech synthesis.
- The spread of the WWW beyond the desktop computer.

2.3 Service Architecture for Speech Applications

Figure 2 shows service architecture for speech applications using VoiceXML 2.0. In this paper, we use HUVOIS, which contains ASR, TTS and VoiceXML interpreters, provided by Korea Telecom [1], [5].

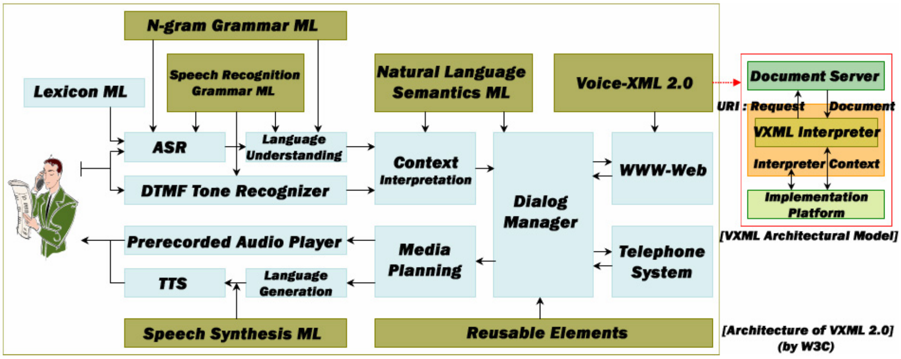

Fig. 2. Service architecture for speech applications using VoiceXML

3 VoiceXML Dialogue System Based on RSS

XSLT is designed for use as part of XSL, which is a style sheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary [6].

XSLT is also designed to be used independently of XSL. That is to say, XSLT could be used as a XML transformation language. However, our proposed system doesn't use XSLT for the purpose of speech applications. For speech applications, we use an HTML parser, the reason for this is that the RSS feeder doesn't include complete content so that its concept is summary, we parses the HTML document that is source of content. To solve these problems, we propose the VoiceXML dialogue system, based on RSS, with a server-side script [6].

3.1 Features

In this chapter, we propose a VoiceXML dialogue system based on RSS. This proposed system provides content which is then provided to the RSS feeder, through a telephony network. The system can provide simultaneously visual RSS service and voice applications using VoiceXML. The system uses a RSS feeder, thus as a result, the proposed system doesn't need an additional DB for VoiceXML service. Figure 3 shows the relationship between components in the proposed system.

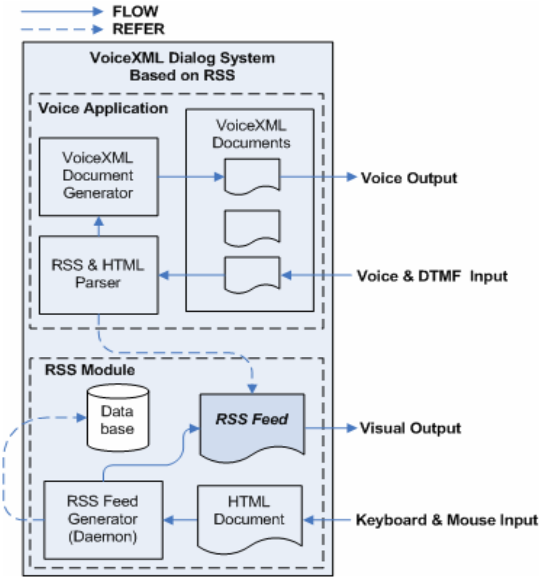

Fig. 3. Components' relationship in the proposed system

3.2 System Components

Our proposed system acts dynamically using scripts language based on Mixed Initiative Forms. In this system, the XML parser and HTML parser are additionally implemented for WEB service using traditional RSS.

3.2.1 Generating RSS Feeds

It is recommended to generate RSS feeds using script languages such as ASP, JSP or PHP [7]. To gather content for the each element in an RSS feed, we use the question and answer method. Then, we record content in the field with elements compatible with RSS version and text format. We use a formatted field for the XML, RSS, channel, title, link and description elements. As shown in Figure 4, in the case of item element, we use the recursive loop algorithm.

```
for (each record in resultSet from database table)
  for (each field in currentRecord)
      if ([column_name] != "pubDate")
       writeline "<[column_name]>", "<[column_content]>", "<[column_name]>"
      endif
  endfor
endfor
```

Fig. 4. Algorithm for generating RSS feeds

The above mentioned algorithm shows, a recursive question and answer method for each element and field.

3.2.2 XML Parser

In this paper, we use the XML parsing algorithm using the DOM. Figure 5 shows the details of the parsing algorithm [8].

```
xmldomVariable <= RSS_FEED_URL
 for (each element in xmlDomVariable)
   if ([element_name] = requiredElements)
     writeline [element_name], [element_content]
   endif
 endfor
```

Fig. 5. Algorithm for XML parsing using the DOM

As shown in Figure 5, our system searches the contents of RSS feed line by line, and finds the necessary element. After that, it records the name of the element and the content of the element. Occasionally, these kinds of algorithms bring about inefficient results. To overcome this inefficient result, we suggest an improved XML parsing algorithm using the DOM. Figure 6 shows the improved XML parsing algorithm using the DOM [8].

```
xmlDomVariable <= RSS_FEED_URL
currentItemCount <= 0
maxItemCount  <= Output Number

 for (each element in xmlDomVariable)
   if ([element_name] = requiredElements)
     if ([element_name] = [item] and
          currentItemCount < maxItemCount)
          currentItemCount ++
          writeline      [element_name]
          writeline      [element_content]
     endif
   endif
 endfor
```

Fig. 6. Improved XML parsing algorithm using the DOM

3.2.3 HTML Parser

Our proposed system has an additional HTML parser for the purpose of using when description elements are incomplete. Generally, description element consists of a part of all content and all content is linked via hypertext. To solve these problems, we implemented in the proposed system the ability to extract necessary content from the hyper linked documents. Figure 7 shows the HTML document parsing algorithm.

```
textString <= Text in html document.
stHeadString <= Insert startpoint string.
stTailString <= Insert endpoint string.

stringArray_1st    Enumerate textString with stHeadString
stringArray_2st    Enumerate stringArray_1st(1) with stTailString
stringResult       Delete HTMLTAG in stringArray_2nd(0)

 Return StringResult
```

Fig. 7. HTML document parsing algorithm using annotation

Figure 7 shows the usage of annotation for extracting essential content from the HTML documents.

4 Experimental Results

To verify the performance of proposed system, we implemented a news service using an RSS feed (http://www.chosun.com/rss/rss.xml) compatible with the RSS 2.0 standard. We use the response time and speech recognition rate as indicators.

4.1 Experimental Condition

Table 2 shows the experimental condition and hardware specification. As shown in Table 2, we use an Intel Xeon CPU and Intel Dialogic D/120JCT-LS as a telephony card. The speech recognition and synthesis engine are provided by Korea Telecom (KT HUVOIS).

Table 2. Hardware and software specification

Component	Description
CPU	Intel Xeon 3.0Ghz Dual
RAM	2GB
CTI Board	Intel Dialogic D/120JCT-LS PCI (12CH)
OS	Windows 2000 Professional Service Pack 4
IIS Ver.	5.0
ASR and TTS	KT Huvois

4.2 Performance Evaluation

Ten testee were selected for performance evaluation and learning courses were finished with several kinds of proposed systems. Performance evaluation was conducted 10 times for each subject. The evaluation was based on speech recognition rate, has shown Table 3, and response time.

Table 3. Response Time

Trial Number	Response Time	Spoken Language
1	Within 1 sec	il:b∂n
2	Within 1 sec	i:b∂n
3	Within 1 sec	sam:b∂n
4	Within 1 sec	sa:b∂n
5	Within 1 sec	o:b∂n
6	Within 2 sec	juk:b∂n
7	Within 1 sec	i:b∂n
8	Within 1 sec	sam:b∂n
9	Within 2 sec	sa:b∂n
10	Within 1 sec	o:b∂n
11	Within 1 sec	il:b∂n
12	Within 1 sec	i:b∂n
13	Within 1 sec	sam:b∂n
14	Within 1 sec	sa:b∂n
15	Within 2 sec	o:b∂n

Table 4 shows the speech recognition rate. Testees, designated as testee (A1, A2, …) were people experienced using a voice-enabled service; testees designated as testee (B1, B2, …) were not experienced using voice-enabled service. If voice-enabled service continues to be popular, related research is needed to resolve the deficiencies relating to people's lack of experience of using this kind of system.

Table 4. Speech recognition rate

Category	Testee	False Rcognition	Recognition Rate(%)
Experienced User	A1	1 / 10	90
	A2	2 / 10	80
	A3	1 / 10	90
	A4	1 / 10	90
	A5	2 / 10	80
Non Experienced User	B1	3 / 10	80
	B2	3 / 10	70
	B3	4 / 10	60
	B4	2 / 10	80
	B5	3 / 10	70
Description	Total 10 person	Total 22 / 100	Average 79

5 Conclusions

In this paper, we proposed a prototype of speech application service using VoiceXML and RSS feed. The key benefit of our proposed system is that it doesn’t need any change in traditional service architecture. We conducted performance evaluation to verify the usefulness of the proposed system. As shown in the experimental results, our proposed system can provide stable service without delay. Another key benefit of our system is that users could use the provided service on the basis of audio-visual environment. In terms of system architecture, our proposed system provides services with minimum server load. RSS and VoiceXML are some of the most popular technologies. Due to the importance of these technologies, we hope to provide useful reference to researchers of web technology and web architecture.

Acknowledgment. This research was supported by MIC, Korea under ITRC IITA-2006-(C1090-0603-0046).

References

1. McGlashan, S., et al.: Voice Extensible Markup Language Version 2.0 Specification (2004), http://www.w3c.org/TR/voicexml20
2. Winer, D.: RSS 2.0 Specifications, Technology at Harvard Law (2004) http://blogs.law.harvard.edu/tech/rss
3. Finder, F.: (2004) http://www.faganfinder.com/search/rss.php
4. Atomenabled. (2004) http://www.atomenabled.org
5. KT Voice Recognition Service R&D Team.: HUVOIS Solution Manual R1.1, KT Service Develop Laboratory (2003)
6. VOX POPLUI.: (2005) http://www.voiceingov.org/blog?page_id=8
7. Vankayala, R.R., et al.: Dynamic Voice User Interface Using VoiceXML and Active-Server Pages. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, Springer, Heidelberg (2006)
8. W3C, D.O.M.: interest group.: Document Object Model (2005) http://www.w3.org/DOM/
9. Tsai, M.-J.: The VoiceXML Dialog System for the E-Commerce Ordering Service. In: IEEE Proceedings of the Ninth International Conference (2005)
10. Shin, J.-H., et al.: Simple and Powerful Interface E-Leaning System Using VXML: Design and Implementation of Web and PSTN Linked Efficient Learning System. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, Springer, Heidelberg (2006)
11. De Sutter, R., et al.: Enhancing RSS Feeds: Eliminating Overheard through Binary Encoding. In: IEEE Proceedings of the Third International Conference (2005)
12. Andersson, E.A., et al.: Early Adopter VoiceXML. Wrox (2002)
13. Finder, F.: (2004) http://www.faganfinder.com/search/changes/
14. Finder, F.: (2004) http://www.faganfinder.com/search/changes/?display=rss