

Java RMI를 이용한 Multi-Modal Browsing 시스템에 관한 연구

장준식*, 윤재석*, 김국보*
대전대학교 컴퓨터공학과*

A Study on the Multi-Modal Browsing System Using Java RMI

JoonSik Jang, JaeSeog Yoon, GukBoh Kim
Dept. of Computer Engineering, Daejin Univ.

요 약

개인용 컴퓨터나 무선 전화, PDA(Personal Digital Assistants)의 기술 발전으로 이들이 인터넷에 접속할 수 있게 되었으며, 사용자들은 좀 더 다양한 방법으로 인터넷을 Browsing할 수 있는 방법을 찾고 있다. 본 연구는 표준 입출력 모드와 음성 인식과 음성 합성 모드를 동시에 적용할 수 있는 Multi-Modal Browsing 시스템을 연구하여 보았다. 이때 각 모드간에 동기화를 위한 이벤트 메시지 교환과 각 모드의 상태 저장 방법으로 Java RMI 모델을 제안하였다.

1. 서론

일반적인 웹 Browsing은 개인용 컴퓨터나 노트북 등으로 이루어지고 있으며, 셀룰러 폰이나 PDA와 같은 소형 기기에서도 인터넷에 접속할 수 있게 되었다. Wireless Application Protocol(WAP)이 가능한 무선 전화는 다이얼 톤 키패드로 인터넷을 검색하고 그 내용을 소형 디스플레이를 통해서 볼 수 있게 하였다. 또한 전화 통신 기술의 빠른 발전으로 인해 VoiceXML[1]과 같은 음성 인터페이스로 웹을 Browsing할 수 있는 시스템들이 많이 연구되고 있다[2][3]. VoiceXML은 사용자가 전화 상에서 음성 인터페이스를 통하여 VoiceXML로 이루어진 웹을 Browsing할 수 있게 하였다. 하지만 HTML이나 WML과 같은 그래픽 인터페이스를 제공하지 않는다.

오늘날의 대부분의 모바일 디바이스의 사용은 데이터의 입력의 어려움과 음성 인식시에 오류 가능성 등을 내포

하고 있다. 또한 음성 데이터의 직렬성으로 인해 데이터의 양이 많을 경우 정보 전달에 많은 시간이 소요된다. 이로 인해 각 모드들을 상호 보완할 수 있게하고, 다양한 상호 작용 모드를 사용할 수 있는 시스템의 필요성이 제기되었다.

본 연구에서는 표준 입출력 모드(키보드, 마우스, 디스플레이)를 지원하는 웹 Browser와 음성인식 및 음성합성을 지원하는 Multi-Modal Browsing 시스템의 한 예로 영 어사전 검색 어플리케이션을 구현하였다. 현존하는 HTML, WML 또는 VoiceXML에 새로운 기능을 추가하는 대신에 특정 마크업 언어(HTML, VoiceXML)를 구현하는 각 플랫폼들 간에 중앙 서버를 두어 각 모드를 동기화하는 방법을 선택하였다. 서버에 각 클라이언트의 상태 정보를 보관해 두었다가 클라이언트로부터 메시지를 받아 저장 및 각 클라이언트로 새로운 메시지를 다시 전송하게 함으로써 Multi-Modal의 각 모드를 문서 레벨에서 동기화 하였다.

각 모드간의 동기화를 위한 이벤트 메시지 교환 및 각 모드의 상태 저장 방법으로 Java RMI[5] 모델을 제안하였다. 이때 영어 사전의 컨텐츠(HTML)를 VoiceXML로 변환하고 각 이벤트 메시지를 RMI 서버와 통신을 시키기 위해 서블릿[6] 기술을 이용하였다.

2. Multi-Modal Browsing

2.1 Multi-Modal Voice Browsing System

웹을 음성이나 DTMF Key Tone으로 Browsing하기 위해서 개발된 기존의 VoiceXML은 음성 어플리케이션의 개발에 비해 개발이 용이하고 개발에 필요한 다른 많은 컴포넌트들의 개발 노력을 줄일 수 있게 하였다[2][3]. 이런 시스템은 소형 무선 전화 상에서도 웹상의 정보를 검색할 수 있도록 하였지만 그래픽 인터페이스를 제공하지 않아 많은 정보 전달시에 많은 시간이 소요되는 경우가 있다. 또한 인터넷 익스플로러와 같은 웹 Browser는 PC 기반의 시스템에서는 효율적으로 사용할 수 있으나 셀룰러 폰이나 PDA와 같이 작은 입력 장치를 가진 시스템에서는 사용하기 어렵다. 이 때문에 사용자들은 단일 모드가 아닌 음성이나 관계적인 디스플레이, 키보드, 마우스 등이 혼합해서 사용할 수 있는 Multi-Modal 시스템의 필요성을 느끼게 되었다.

Multi-Modal은 하나 이상의 음성 모드(음성 인식, 음성 합성, 미리 녹음된 음성)와 하나 이상의 표준 입력력 모드(dtmf, keyboard, small screen pointing device, 등등)가 혼합된 시스템으로 정의할 수 있다[1]. Multi-Modal 시스템에서 음성 모드를 위한 음성 어플리케이션은 Voice로 인터페이스하기 위해 기본적으로 출력 인터페이스로 TTS(Text To Speech) 합성기와 입력 인터페이스로 ASR(Automatic Speech Recognizer)을 포함한다.

Multi-Modal Browsing을 위한 데이터와 표현 로직으로 접근 가능한 방법이 2가지가 있는데, 먼저 XML을 기반으로 하여 Multi-Modal 입출력을 제공하는 데이터와 표현 로직을 동시에 다룰수 있는 특정 마크업 언어를 만드는 것이 있다. 이 방법은 W3C에 의해 장려되고 있는 방향이고 아주 강력하지만, 이런 구조를 이루기 위해서 기존의 HTML들이 모두 새로운 마크업 언어로 변경되어야 하고 새로운 Browser를 필요로한다는 점에서 실현성에 문제가

있다[4].

두번째 방법은 본 연구에서 구현한 방법으로 현재의 VoiceXML 시스템과 연결하는 방법이다. 특정 HTML 문서에 대응되는 VoiceXML을 생성하도록함으로써 Multi-Modal이 가능하도록 하는 것이다. 각 모드의 상태를 서버에 저장하고 이것을 각 클라이언트에 전송하기 위한 방법으로 Java RMI 모델을 제안하였다.

2.2 Java RMI

Java Remote Method Invocation(RMI)[5] 기술은 JDK 1.1부터 적용되어서 Java 환경에서 원격 컴퓨터 또는 프로그램간에 통신을 할 수 있는 기능을 제공하도록 기본적으로 포함되어 있다. Java RMI는자바 어플리케이션 환경에서 수행되도록 설계되었고 동일한 Java virture machine(JVM) 환경인 것으로 가정한다. 프로그래밍 언어를 Java만을 써야하는 제약성이 있지만 Java 플랫폼의 객체 모델을 이점을 활용할 수 있다. RMI와 유사한 CORBA, DCOM, EJB 등과는 달리 객체 또는 컴포넌트간 정보 전달을 지원하는 별도의 미들웨어 시스템을 필요로 하지 않는다.

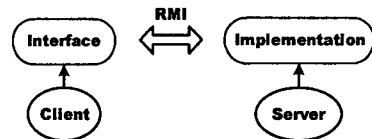


그림 1. Java RMI Communication

서버측에는 원격 인터페이스와 실제 구현 객체들을 가지고 있으며, 클라이언트는 RMI를 통해서 서버의 실제 구현 부분과 통신하는 것이 아닌 인터페이스를 통해서 통신한다. 그림1과 같이 서버와 클라이언트는 동일한 원격 인터페이스를 가지고 서버로 객체를 보내거나 서버에서 객체를 반환 받는다.

2.3 Java RMI 모델을 적용한 Multi-Modal Voice Browsing 시스템

본 연구에서 구현한 Voice Browsing 시스템은 기본적으로 분산 구조를 가지고 있다. 서버에 저장된 각각의 모드의 정보를 그에 맞는 클라이언트에게 전송하는 구조를

가진다. 그림2은 본 연구에서 제안한 Java RMI 모델에 의한 Multi-Modal Voice Browsing 시스템의 구조를 나타낸다. GUI Browser는 그림3와 같이 Java Swing으로 웹상의 HTML 문서를 볼 수 있도록 작성하였다. GUI Browser에는 RMI Server와 통신할 수 있는 컴포넌트가 포함되어 있다. RMI 서버에는 클라이언트 객체를 저장하고 관리하는 MMServer객체, 각 클라이언트의 상태의 저장과 변경을 담당하는 MMClient객체, 클라이언트와 서버 사이의 메시지의 형태를 정의하는 Message객체를 가지고 있다. Message 객체는 프라퍼티로 메시지의 타입과 메시지 콘텐츠를 가지고 있다.

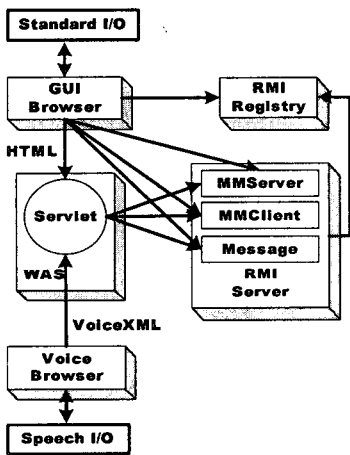


그림 2. Multi-Modal Voice Browsing 시스템 구조

클라이언트가 그림3의 GUI Browser를 실행하면 각 클라이언트를 식별하기 위한 ID를 입력하도록 하였다. 이 ID는 서버에서 각 메시지를 보낼 클라이언트를 구별하기 위해서 사용된다. MMServer객체는 각 클라이언트 객체들을 고유ID와 클라이언트 객체 쌍으로 저장한다.

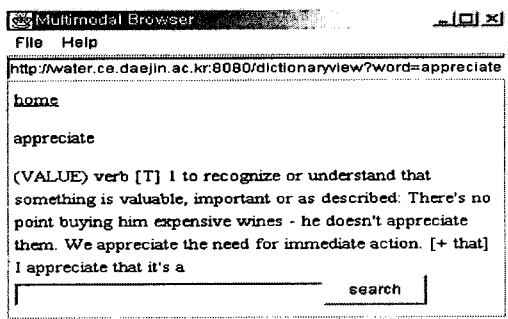


그림 3. Multi-Modal을 제공하는 GUI Browser

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- copyright 2001 by Daejin University MAS Lab -->
<vxml version="2.0">
  <link
next="http://203.237.81.198:8080/servlet/HTML2VoiceXML">
  </link>
  <link
next="http://203.237.81.198:8080/servlet/HTML2VoiceXML?url=http://water.ce.daejin.ac.kr:8080/dictionaryview">
  </link>
  <var name="url"/>
  <form id="dictionary">
    <block>
      <prompt>
        <sentence>dictionaryview</sentence><break
size="large"/><sentence>appreciate</sentence><break
size="large"/><sentence>value verb t to
recognize or understand that something is
valuable important or as described theres no point
buying him expensive wines he doesnt
appreciate them we appreciate the need for
immediate action that i appreciate that its
a</sentence><break size="large"/>please say new
word</prompt>
    </block>
    <grammar src="/dictionary.grammar#DicGrammar"/>
    <field name="word" slot="word">
      </field>
      <filled>
        <assign
                                name="url"
                                expr="http://203.237.81.198:8080/
dictionaryview?word='+ word'/">
      </filled>
      <submit
next="/servlet/HTML2VoiceXML"
method="post" namelist="url"/>
    </filled>
  </form>
</vxml>
```

그림 4. 그림4의 화면을 디스플레이하기 위한 HTML을 VoiceXML로 변환한 결과

음성 입출력 모드를 실행하기 위한 Voice Browser는 Nuance System[7]을 사용하였다. HTML을 필터링하는 컴포넌트를 작성하여 서블릿에 포함시키고, 필터에 의해 생성된 데이터를 동적으로 VoiceXML로 변환하였다. 이때 Voice Browser가 동적으로 생성된 VoiceXML을 실행하면서 TTS(Text to Speech) 합성기를 통해서 GUI Browser 상에 보이는 콘텐츠를 들을 수 있게 하였다. 또한 Voice Browser는 본 연구에서 준비한 100가지의 특정 영어 단어 중에 하나를 말하면 그 단어에 관한 URL 정보를 RMI 서버로 Message 객체를 통해서 보내기 위해 서블릿을 요청한다. 이 서블릿은 요청될 때 마다 MMServer의 내의 각 클라이언트 객체의 상태를 검색한다. 이때 각 클라이언트 객체의 상태가 변하면 이것을 다른 모드에 적용한다. 이것

은 그림3의 텍스트필드에 키보드로 한가지 단어를 입력하고 검색 버튼을 마우스로 클릭한 것과 같은 효과를 발생시킨다. MMServer 객체는 메시지를 받으면 메시지를 보낸 객체의 ID를 식별하여 그 메시지를 전달할 MMClient 객체를 선택하고 이 객체를 생성시킨 클라이언트의 GUI Browser가 그 메시지를 받아서 그 처리를 하도록 하였다. 이러한 과정에서 GUI Browser는 새로운 웹페이지로 이동하게 된다. 그림3은 appreciate 단어를 말했을 때 GUI Browser 상에서 나타난 화면이고, 이와 동시에 Voice Browser에서 appreciate 단어에 관한 설명을 TTS로 사용자에게 전달할 수 있게 하기 위하여 그림4와 같은 VoiceXML 문서가 자동으로 생성되도록 하였다.

웹 페이지 상에 있는 하이퍼링크 객체 중에서 이미지가 아닌 텍스트로 연결된 객체는 VoiceXML의 Grammar로 변형되도록 하고, 이것으로 사용자가 웹페이지상에 보이는 하이퍼링크를 말로 하면 마우스를 클릭한 것과 같은 동작을 하도록 하게 된다. 이로서 본 연구에서 구현한 영어사전 검색 어플리케이션 사이트 뿐만 아니라 다른 특정 웹사이트의 Browsing의 가능성도 열어 보았다. 그림5는 Altavista 사이트에서 news란 음성 입력후에 News 사이트로 이동한 화면이다.

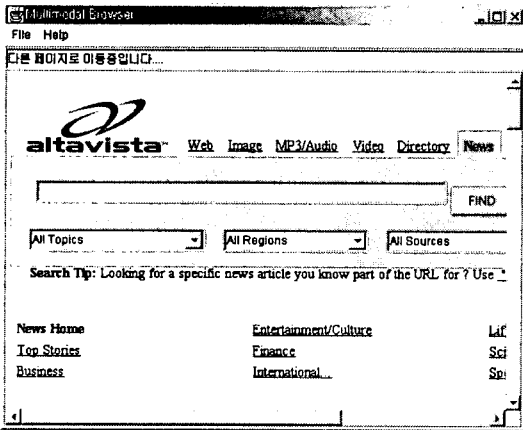


그림 5. Altavista 사이트 내의 News 사이트로 이동한 화면

본 연구에서 구현한 Multi-Modal Browsing 시스템은 테스트 대상인 영어 단어 100개를 검색할 수 있는 웹사이트를 구축하여 실험하였으며, 알타비스타 또는 야후와 같은 사이트도 테스트 대상으로 하였다. 이 시스템은 클라이

언트상의 시스템에 음성 인식이나 음성 합성 등과 같은 무거운 엔진들을 포함하지 않는다. 또한 클라이언트가 접속 후에 각 클라이언트에게 보내는 메시지 전송과 관리 부분도 RMI 서버가 대신하고 있다. 각 클라이언트들을 구별하기 위한 방법으로 각 클라이언트가 입력한 ID에 의해 클라이언트 객체를 생성하도록 하였다.

3. 결론

본 연구는 일반적인 표준입출력과 더불어 음성 입출력이 가능한 Multi-Modal Browsing이 가능한 시스템에 관하여 연구해 보았다. Java RMI 모델을 적용하여 각 클라이언트의 상태를 서버에 저장하고 각 클라이언트에게 메시지를 전송하는 방법을 선택하여 각 모드 간의 동기화된 통신이 이루어지도록 하였다. HTML문서를 필터링하여 동적으로 VoiceXML로 변환하여, 영어 사전 검색 사이트에서 영어 단어 검색을 Multi-Modal로 가능하도록 구현하여 보았다.

본 연구에서는 HTML이나 VoiceXML의 페이지 문서 수준의 동기화로서 Multi-Modal을 구현하였다. 차후에는 이와 같은 시스템을 셀룰러 폰이나 PDA와 같은 소형 시스템인 Thin Client 상에서 효율적으로 운용할 수 있는 방법에 관한 연구가 필요할 것이다.

[참고문헌]

- [1] Voice Browser Activity, <http://www.w3.org/Voice/>
- [2] 장준식, 윤재석, 김국보, "VoiceXML을 사용한 음성 인식 항공 정보 시스템에서의 JAVA Beans Component 구조에 관한 연구", 한국멀티미디어학회 춘계학술대회, 2002년 5월, p105-111
- [3] Bennett C, Font Lilit s A, Building VoiceXML-based Applications Proceedings of ICSLP 2002 (Denver, Colorado), pages 2245-2248.
- [4] Caccia G, Cicchitto A, Lancini R, Multimodal browsing , ASRU '01 IEEE Workshop on, 2001, Page 272 -275
- [5] Java Remote Method Invocation (RMI), <http://java.sun.com/products/jdk/rmi/>
- [6] Java Servlet Technology, <http://java.sun.com/products/servlet/>
- [7] Nuance Developers Network, <http://extranet.nuance.com>