

Browser 연동에 의한 Multi-Modal 영영사전 검색 시스템의 구현에 관한 연구

장준식*, 윤재석*

대전대학교 컴퓨터공학과 *

요 약

개인용 컴퓨터나 무선 전화, PDA의 기술 발전으로 이들이 인터넷에 접속이 용이해짐에 따라, 사용자들은 좀 더 다양하게 인터넷 상의 정보를 검색할 수 있는 방법을 찾고 있다. 본 연구에서는 GUI Browser와 Voice Browser를 연동하여 각 Browser들의 인터페이스를 동시에 사용할 수 있는 시스템을 설계·구현해 보았다. 이 때 Java RMI(Remote Method Invokation)를 이용하여 각 Browser 인터페이스들 사이에 통신이 가능하도록 하였다. 구체적인 시스템으로써 VUI와 GUI를 통해서 검색이 가능한 Multi-Modal 영영사전 어플리케이션을 구현하여 보았다.

키워드

Java RMI, Multi-Modal, VoiceXML

1. 서 론

일반적인 Web Browsing은 개인용 컴퓨터나 노트북 등에 의해서 가장 많이 이루어지고 있으며, 셀룰러 폰이나 PDA와 같은 소형 기기에서의 인터넷에 접속이 점점 늘어가고 있다. 또한 Wireless Application Protocol(WAP)이 가능한 무선 전화는 키패드로 인터넷을 검색하고 그 내용을 소형 디스플레이를 통해서 볼 수 있게 하였다. 음성 어플리케이션 및 전화 통신 기술의 발전으로 인해 VoiceXML[1]과 같은 음성 인터페이스로 웹을 Browsing할 수 있는 시스템에 대한 표준들이 많이 연구되고 있다[2][3]. VoiceXML은 전화를 이용한 음성 인터페이스를 통하여 VoiceXML로 작성된 인터넷 사이트를 Browsing할 수 있게 하였으나 HTML이나 WML과 같은 그래픽 인터페이스를 제공하지 않아 텍스트나 그래픽 정보를 함께 제공하는 기능을 가지고 있지 않다.

오늘날의 음성 통신 기기의 많은 부분을 차지하는 모바일 디바이스의 사용은 데이터의 입력의 어려움과 음성 인식시에 오류 가능성 등을 내포하고 있다. 또한 음성 데이터가 가지는 직렬성으로 인해 데이터의 양이 많을 경우 정보 전달에 많은 시간이 소요된다. 이처럼 각 시스템의 플랫폼 상에 브라우저들은 각자 장단점을 가지고 있다. 이로 인해서 각 브라우저를 상호 보완할 수 있게 하고, 다양하게 상호작용할 수 있는 시스템의 필요성이 제기되었다.

본 연구에서는 표준 입출력 모드(키보드, 마우스,

디스플레이)를 지원하는 웹 Browser와 음성인식 및 음성합성을 지원하는 Voice Browser를 통합하는 Multi-Modal 시스템의 구조를 제시하였고, 이것의 구현 예로 영영 사전 검색 어플리케이션을 구현하였다. 이 시스템의 구조는 현존하는 HTML과 VoiceXML에 새로운 기능을 추가하는 대신에 특정 마크업 언어(HTML, VoiceXML)를 구현하는 각 플랫폼들 간에 미들웨어 어플리케이션을 두어 각 모드를 동기화하는 방법을 선택하였다. 각 브라우저간의 동기화를 위해 필요한 이벤트 메시지 교환 및 각 브라우저들의 상태를 저장하는 방법으로 Java RMI[5] 모델을 제안하였다. 이 방법은 Java RMI 통신이 가능하도록 구현된 미들웨어적인 기능을 하는 서버에 각 클라이언트 Browser 객체들의 상태 정보를 보관해 둔 후에 각 클라이언트 Browser 객체들로부터 받은 이벤트 메시지를 저장하거나 각 클라이언트로 새로운 메시지를 다시 전송할 수 있게 함으로써 Multi-Modal Interaction 시스템의 각 모드를 문서 레벨에서 동기화할 수 있도록 하였다. 이때 영영 사전의 컨텐츠는 HTML 및 VoiceXML로 각각 만들어질 수 있게 하고 VoiceXML로 변환 및 각 모드들의 템플릿 모델을 정의하고 생성하기 위한 방법으로 서블릿[6]을 사용하였다.

2. 본 론

2.1 Multi-Modal Voice Browsing System

웹을 음성인식이나 DTMF Key Tone으로 Bro-

wsing하기 위해서 개발된 VoiceXML을 기반으로 한 Voice Browsing 시스템은 기존의 음성 어플리케이션의 개발에 비해 개발이 용이하고 개발에 필요한 다른 많은 컴포넌트들의 개발 노력을 줄일 수 있게 하였다[2][3]. WML이 셀룰러 폰과 같이 소형 입력 장치에서 텍스트 정보나 그래픽 정보를 표현할 수 있게 하였지만 운전 중일 때처럼 시야가 제한된 상황에서는 음성 인식이나 음성 출력이 더 효율적인 경우가 있을 수 있게 되었다. 이런 시스템은 소형 무선 전화 상에서도 웹 상의 정보를 소리로 얻을 수 있도록 하였지만 그래픽 인터페이스를 제공하지 않아 음성으로 전달하는데 적절하지 않은 정보를 제공하는 것에서는 한계성을 가지고 있다. 이 때문에 사용자들은 단일 모드가 아닌 음성이나 관례적인 디스플레이, 키보드, 마우스 등을 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 입출력 모드(dtmf, keyboard, small screen pointing device, 등등)가 혼합된 시스템으로 정의할 수 있다[1]. Multi-Modal Interaction 시스템에서는 음성 모드에서 사용하는 음성 어플리케이션의 기본적인 음성 입출력 인터페이스는 TTS(Text To Speech) 합성기와 입력 인터페이스로 ASR(Automatic Speech Recognizer)를 사용한다.

Multi-Modal Browsing을 위한 데이터와 표현 로직으로 접근 가능한 방법이 2가지가 있는데, 먼저 XML을 기반으로 하여 Multi-Modal 입출력을 제공하는 데이터와 표현 로직을 동시에 다룰 수 있는 특정 마크업 언어를 만드는 것이 있다. 이 방법은 W3C에 의해 권장되고 있는 방향이고 아주 강력하지만, 이런 구조를 이루기 위해서 기존의 HTML들이 모두 새로운 마크업 언어로 변경되어야 하고 새로운 Browser가 필요하다는 점에서 실현성에 문제가 있다[4].

또 다른 방법은 본 연구의 영영사전 어플리케이션에서 사용한 방법으로 현재의 VoiceXML 어플리케이션 시스템과 연결하는 방법이다. 특정 HTML 문서에 대응되는 VoiceXML을 생성하도록 함으로써 Multi-Modal이 가능하도록 하는 것이다.

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 RMI는 자바 어플리케이션 환경에서 수행되도록 설계되었고 동일한 Java virtual machine(JVM) 환경에 있는 것처럼 작동한다. 프로그래밍 언어를 Java만을 써야 하는 제약성이 있지만 Java 플랫폼의 객체 모델의 이점을 활용할 수 있다. RMI는 이와 유사한 CORBA, DCOM, EJB 등과는 달리 객체 또는 컴포넌트 간에 정보 전달을 지원하는 별도의 미들

웨어 시스템을 요구하지 않는다.

서버 측에는 원격 인터페이스와 실제 구현 객체들을 가지고 있으며, 클라이언트는 RMI를 통해서 서버의 실제 구현 부분과 통신하는 것이 아닌 인터페이스를 통해서 통신한다. 그림1은 Java RMI 구조를 나타낸다. 서버와 클라이언트는 직접 통신하는 것이 아니라, 동일한 원격 인터페이스를 가지고 서버로 객체를 보내거나 서버에서 객체를 반환 받는다.

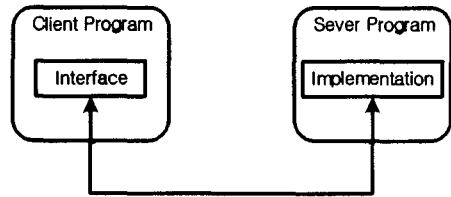


그림 1 Java RMI 구조

3. Java RMI 통신 모델을 적용한 Multi-Modal 영영사전 검색 System

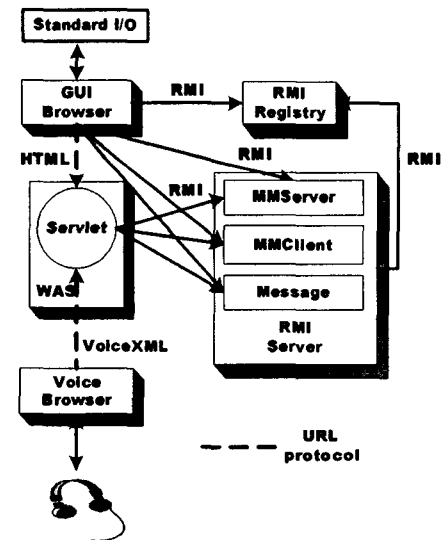


그림 2 Multi-Modal 영영 사전 시스템 구조

본 연구에서 제안한 Browser의 통합에 의한 Multi-Modal 영영 사전 검색 시스템은 기본적으로 Java RMI가 가지는 분산 시스템의 구조를 바탕으로 설계하였다. 이 시스템은 각각의 Browser의 정보를 그에 맞는 클라이언트에게 전송하는 미들웨어 역할을 하는 Multi-Modal RMI 서버 컴포넌트와 HTML 문서를 볼 수 있는 GUI Browser 컴포넌트, 음성 통신 모드를 지원하는 Voice Browser 컴포넌트로 구성하였다. 그림2는 본 연구에서 제안한 Java RMI 통신 모델을 적용한 Multi-Modal 영영사전 검색 시스템의 구조를 나타낸다. 본 연구에서

는 이와 같은 구조를 적용하여 두 가지 모달로 동시에 접근 및 제어가 가능한 영영 사전 어플리케이션을 구현하여 보았다.

본 연구에서 제안한 Multi-Modal Voice Browsing 시스템은 사용자가 인터넷 상의 HTML 문서를 보면서 동시에 Voice 입력으로도 브라우징이 가능하게 하기 위해 고안하였다. GUI Browser 상에 나타나는 HTML 문서를 적절한 해석 과정을 거쳐 Voice Browser에 입력이 가능한 VoiceXML 문서로 변환함으로써 Voice 입출력이 가능하도록 하는 것이다. 이런 두 가지 모드 상에서 사용자에게 발생된 이벤트를 상호간에 동기화 시키기 위한 방안으로 Java RMI를 도입하였다. Java RMI를 이용하여 각 브라우저 간의 통신 인터페이스를 정의하고 이 인터페이스를 통하여 각 모드간의 통신이 이루어지게 함으로써 각각의 모드가 동기화될 수 있는 것이다. HTML 문서를 VoiceXML로 변환하거나 음성 인식을 위한 소프트웨어들은 모두 서버측에 위치하게 함으로써 클라이언트가 가지는 부담을 최소화할 수 있다. 또한 VoiceXML의 장점을 인하여 음성 어플리케이션의 개발이 용이해진다.

Java RMI는 원격지의 객체와 통신을 위해서 반드시 그 객체들의 인터페이스들을 정의해야 한다. 원격지와 통신이 원격지의 실제 구현 객체와 이루어지는 것이 아니라 그것의 proxy 역할을 하는 인터페이스와 통신하기 때문이다. 원격지의 객체 찾는 것은 그림2의 rregistry에 등록된 객체의 lookup을 통해서 이루어진다. RMI 서버에는 실제 객체들의 인터페이스를 정의해야 하는데, 이 인터페이스들은 클라이언트 객체를 저장과 관리를 정의하는 RMI MultimodalServer 인터페이스, 각 클라이언트의 상태의 저장과 변경을 정의하는 RMI MultimodalClient 인터페이스, 클라이언트와 서버 사이에 주고 받는 메시지의 타입을 정의하는 Message 인터페이스로 구성하였다. RMI MultimodalServer를 실제로 구현한 RMI MultimodalServerImpl이 rregistry에 등록되고 각 클라이언트 어플리케이션은 각각의 인터페이스를 통해 이벤트 메시지를 전송한다.

인터넷 상의 간단한 HTML 문서를 볼 수 있도록 GUI Browser를 Java Swing으로 작성하였다. GUI Browser에는 RMI Server와 통신할 수 있는 인터페이스를 가지고 있도록 하였다. 음성 입출 모드 측의 Voice Browser는 Nuance Voice Web Server 플랫폼[7]을 사용하였다. 이 Nuance 플랫폼은 음성 인식, TTS(Text-To-Speech)등의 기능을 제공하는 플랫폼이다.

GUI Browser나 Voice Browser로 전송되는 각각의 HTML과 VoiceXML 콘텐츠의 모델은 서블릿[6]을 통하여 정의하도록 하였다. 이 서블릿이 구동되는 환경은 Tomcat[8]을 사용하였다. 그림3에는 이 두 종류의 모델을 정의하는 HS(Http Servlet)들이 있다. DictionaryView는 영영사전의 데이터를 HTML로 생성하고 HTML로 VoiceXML은 Dicti-

onaryView나 다른 간단한 HTML문서의 필터링 과정을 통해서 VoiceXML로 변환하였다. 이때 Voice Browser가 동적으로 생성된 VoiceXML 문서를 실행하면서 TTS(Text to Speech) 합성기[7]를 통해서 GUI Browser 상에 보이는 콘텐츠를 들을 수 있게 하였다.

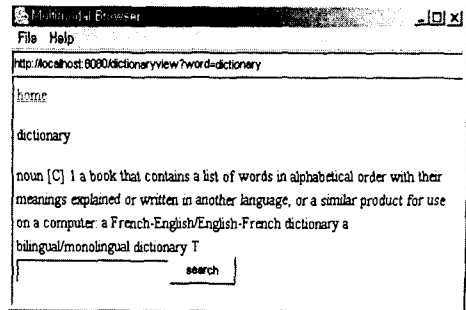


그림 4 GUI Browser

본 연구에서는 Java RMI 통신 모델을 적용한 Multi-Modal 영영사전 시스템을 구현하였다. 이 어플리케이션은 인터넷으로 영어 단어를 검색하는 것으로서 영영 사전에서 선택한 100가지의 무작위로 영어 단어를 데이터베이스에 등록하였다. 클라이언트 어플리케이션 중에서 GUI Browser 어플리케이션은 그림4와 같다. 그림4는 검색 창에서 영어 단어 'dictionary' 를 입력하고 'search' 버튼을 클릭한 후에 검색된 내용을 보이고 있다. 이때 사용자가 접속한 웹사이트의 주소가 RMI MultimodalServer 어플리케이션에 전송된다. 이것으로 이 사용자가 어떤 웹 주소를 입력했는지를 다른 클라이언트 측 어플리케이션도 알 수 있게 된다. 다른 클라이언트 측 어플리케이션인 Voice Browser에게 현재 입력한 단어의 의미를 출력하라는 명령을 음성으로 보내면 그 웹 주소의 HTML문서는 필터링 과정을 거쳐 VoiceXML로 변환되어 음성 데이터로 사용자에게 전달된다.

여러 사용자가 이 시스템에 접속하게 될 경우 어떤 사용자가 어떤 웹사이트에 접속했는지에 관한 정보를 구별할 수 있어야 각 사용자마다 정확한 정보를 전달할 수 있다. 이것을 위해서 각 사용자가 처음에 GUI브라우저에 접속할 시에 자신의 ID를 입력하도록 하였다. 이 ID에 의해서 RMI MultimodalServer에는 각 클라이언트 별로 고유한 RMI MultimodalClient 객체를 생성하도록 하였다. 다른 클라이언트 측의 어플리케이션인 Voice Browser로 접속할 시에 ID를 말하면 GUI 어플리케이션에 의해 생성된 RMI MultimodalClient를 ID로 찾을 수 있게 하여 사용자 별로 각각의 모드를 동기화할 수 있도록 하였다.

각 모드의 클라이언트 어플리케이션과 RMI 서버 상에 전송되는 메시지의 변경 및 확장이 쉽도록 하기 위해 Message 인터페이스를 정의하고 실제

내용은 RMI 서버 측에서 가지고 있도록 하였다. 이것으로 Message의 구현 내용이 추후에 변하더라도 인터페이스가 변하지 않는 한에는 각 모드의 클라이언트 어플리케이션을 수정하지 않아도 되는 효과를 얻을 수 있었다. 또한 Voice Browser 측에서 등록된 영어 단어 중 하나를 말하면 그 단어를 검색하고자 한다는 것이 RMI 서버의 RMIMultimodalClient에 전달되고 이와 동시에 GUI Browser 어플리케이션으로도 전달되어 그 단어가 검색된 결과를 사용자가 검색 창에 입력하지 않더라도 그 페이지로 이동하도록 하였다.

본 연구에서 구현한 Multi-Modal 영영 사전 검색 시스템은 테스트 환경으로 일반 데스크탑 PC를 사용하였으며, 검색 대상인 영어 단어는 영영사전에서 임의의 단어 100개를 선택하여 구성하였다. 각 클라이언트 측 보기 모델을 각각 HTML과 VoiceXML로 정의함으로써 어플리케이션 간의 호환성을 높였다. 이 시스템은 각 클라이언트들을 구별하기 위한 방법으로 각 클라이언트가 입력한 ID에 의해 클라이언트 객체를 생성하도록 하였고, 클라이언트 측의 시스템에 음성 인식이나 음성 합성 등과 같은 무거운 엔진들을 포함하지 않으며 또한 클라이언트가 접속 후에 각 클라이언트에게 보내는 메시지 전송과 관리 부분도 RMI 서버가 대신하도록 하였다.

3. 결론

본 연구는 일반적인 표준입출력과 더불어 음성 입출력이 가능한 Multi-Modal 인터페이스를 가능한 영영사전 검색 시스템에 관하여 연구해 보았다. 현재 많이 사용되고 있는 HTML과 VoiceXML을 사용하는 각 시스템을 통합하는 방법으로 Java RMI 모델을 적용하여 각 클라이언트의 상태를 서버에 저장하고 각 클라이언트에게 메시지를 전송하는 방법을 선택하여 각 모드간의 동기화된 통신이 이루어지도록 하였다. 이러한 시스템의 한 예로 HTML문서를 필터링하여 동적으로 VoiceXML로 변환하여, 영영 사전 검색 사이트에서 영어 단어 검색을 Multi-Modal로 가능하도록 구현하여 보았다.

본 연구에서는 HTML이나 VoiceXML의 페이지 문서 수준의 동기화로써 Multi-Modal을 구현하였다. 이 시스템은 상대적으로 구성이 복잡한 데스크탑 PC를 대상으로 하는 웹사이트보다는 콘텐츠의 구성이 간단한 웹사이트가 좀 더 효율적으로 적용될 수 있을 것이다. 차후에는 이와 같은 시스템을 Java RMI가 적용 가능한 셀룰러 폰이나 PDA와 같은 소형 시스템인 Thin-Client 상에서 효율적으로 운용할 수 있는 방법에 관한 연구가 필요할 것이다.

참고 문헌

- [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 Llitjos 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>"
- [8] Apache Jakarta Project, "<http://jakarta.apache.org/>"