모바일 환경을 위한 멀티모달 미들웨어의 설계 및 구현

박성수(KT), 박성찬(KT), 안세열(KT), 김원우(KT), 구명완(KT)

<차 례>

1. 서론
2. 기존의 연구
2.1 개요
2.2 멀티모달 인터페이스의 분류 및 정의
2.3 멀티모달 인터페이스 표준
3. 미들웨어 구조
3.1. 개요
3.2. 멀티모달 실행 프레임워크
3.3 모달리티 콤포넌트
4. 동작과정
4.1. EMMA의 생성과 해석
4.2. 멀티모달 통신
4.3. 상황인지 인터페이스
5. 실험
6. 결론

<Abstract>

Design and Implementation of Multimodal Middleware for Mobile Environments

Seong-Soo Park, Sung-Chan Park, Seyeol Ahn, Won-Woo Kim, Myoung-Wan Koo

W3C announced a standard software architecture for multimodal context-aware middleware that emphasizes modularity and separates structure, contents, and presentation. We implemented a distributed multimodal interface system followed the W3C architecture, based on SCXML. SCXML uses parallel states to invoke both XHTML and VoiceXML contents as well as to gather composite or sequential multimodal inputs through man-machine interactions. We also hire Delivery Context Interface(DCI) module and an external service bundle enabling middleware to support context-awareness services for real world environments. The provision of personalized user interfaces for mobile devices is expected to be used for different devices with a wide variety of capabilities and interaction modalities. We demonstrated the implemented middleware could maintain multimodal scenarios in a clear, concise and consistent manner by some experiments.

* Keywords: Speech-oriented multimodal interface, Multimodal system architecture.

1. 서 론

무선 인터넷 환경이 도입되면서 PDA 혹은 스마트폰과 같은 휴대형 소형 기기들의 사용이 급등하였다. 소형 기기들의 하드웨어 성능은 데스크탑 PC와 거의 대등해져 이미 많은 수의 PC 웹 응용 프로그램이 모바일 환경에서 동작하고 있다. 한 가지 문제는 기기가 작아지면서 키보드와 마우스 등의 전통적인 인터페이스 방식을 고수하기가 어려워졌다. 모바일 환경의 특성상 단말과 응용 프로그램은 직관적, 환경 적응적 또는 선택적인 상호작용을 필요로 하게 되었음에도 불구하고 오히려 사용자 인터페이스는 나아지지 않았다.

이에 따라 사용자로 하여금 기기를 효율적으로 다룰 수 있는 대안의 인터페이스 기술이 필요하게 되었다. 음성과 키 입력 두 가지 이상의 모달리티를 사용하는 멀티모달 인터페이스는 이러한 전통의 인터페이스 한계를 극복하는 기술로 떠오르고 있다. 멀티모달 인터페이스는 주어진 환경에 따라 다수의 모달리티를 선택적 또는 복합적으로 사용함으로서 입출력의 효율성과 편의성을 높인다.

멀티모달 미들웨어(이하 미들웨어)란 서비스 제작자가 멀티모달 응용 서비스를 개발할 수 있게 해 주는 소프트웨어다. 웹 서비스 개발자는 미들웨어가 제공하는 마크업 언어를 이용하면 멀티모달 응용 프로그램을 손쉽게 만들 수 있다. 만약 미들웨어가 존재하지 않는다면 서비스 제작을 위해 멀티모달 어플리케이션에 직접 코드를 추가해 넣어야 한다. 미들웨어는 사용자와 어플리케이션 사이에 독립적으로 번역기 역할을 함으로써 구조의 콘텐츠, 뷰, 제어가 기능적으로 분리된다.

이 논문에서는 W3C(World Wide Web Consortium) 소프트웨어 아키텍처의 새로운 표준[1]을 따르는 멀티모달 인터페이스를 구현하였다. 본 시스템은 시나리오에 따라 다양한 형태의 멀티모달 서비스가 가능하도록 미들웨어 구조로 제공된다. 멀티모달 시나리오 제작용으로는 해럴(Harel) 상태차트 기반의 대화 모델링 언어인 SCXML(State chart XML)[2]을 도입하였다. SCXML은 대화 매니저(Interaction Manager)에서 동작하며 대화매니저는 SCXML로 작성된 시나리오 스크립트를 불러온 다음 XHTML, VoiceXML, SVG와 같은 XML 마크업을 처리하는 모달리티 콤포넌트를 구동한다.

미들웨어의 구성요소는 모달리티 콤포넌트, 인터랙션 매니저, DCI(Delivery Context Interface)[3], 세션 데이터 관리 모듈이다. 인터랙션 매니저, DCI, 세션 데이터 관리 모듈을 통합하여 멀티모달 실행 프레임워크라 한다. 인터랙션 매니저는 사용자와 시스템 사이의 상호작용을 하는 데 있어 이들 모달리티 콤포넌트를 제어, 감독할 뿐 아니라 사용자로부터 전달된 단순결합 또는 보조결합 멀티모달 입력을 처리하거나 외부 모듈과 접속한다. DCI는 보다 개인화된 서비스가 가능하도록 개인별 프로파일 정보와 단말의 환경 정보가 저장된 DB이다. 세션 데이터 관리 모듈은 세션 관리를 위한 데이터 지원 및 관리를 수행한다.

본 논문의 의의는 SCXML, VoiceXML, XHTML 등의 마크업을 지원하는 미들웨어를 구현한 것이다. 미들웨어는 단순결합 멀티모달 뿐만 아니라 보조결합 멀티모달 인터페이스를 지원한다. VoiceXML, XHTML, SVG 마크업은 단말이 가진 음성 또는 GUI(Graphic User Interface)의 개별 모달리티에 대한 인터페이스 시나리오를 기술하며, SCXML 마크업은 이들을 제어, 감독하는 멀티모달 시나리오를 기술한다. 제작된 스크립트에 의해 모달리티 컴포넌트는 멀티모달 서버에 플러그인 방식으로 동작한다. 몇 가지 테스트 결과 미들웨어는 명확, 간결, 일관된 방법으로 사용자가 시스템과의 다양한 상호작용과 멀티모달 시나리오를 수행할 수 있음을 보였다.

본 논문의 구성은 다음과 같다. 2장의 기존의 연구에서는 멀티모달 시스템의 구성과 사례를 알아본다. 3장의 미들웨어 구조에서는 구성요소와 시스템 전반의 구조를 설명한다. 4장과 5장에서는 동작과정과 실험을 통해 미들웨어를 평가하고 6장에서 결론을 맺는다.

2. 기존의 연구

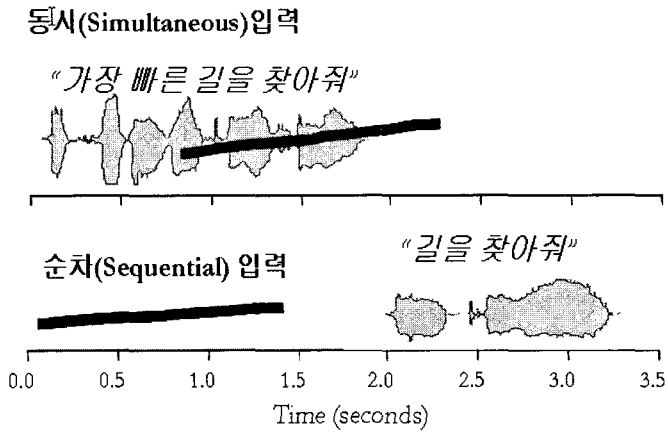
2.1. 개요

멀티모달이란 다수의 모달리티(modality)란 의미로서, 모달리티란 인간의 시각, 청각, 촉각, 미각, 후각 등 각각의 감각채널을 모델링하여 기계장치로 전환된 채널이다. 또한 각 모달리티를 종합하여 교류하는 것을 멀티모달 상호작용이라 한다. 멀티모달 입력의 예로는 음성, 펜(촉수), 키보드, 마우스, 터치스크린 입력, 입술 운동 입력, 제스처 입력, 안구 이동 입력 등이다. 출력의 예로는 아이콘, 표, 테이블과 같은 그래픽 요소와 이어콘(earcon), 효과음 등의 사운드 요소, 그리고 쇼크 등의 진동요소가 있다. 본 논문에서 멀티모달 인터페이스는 모달리티의 도움을 빌어 입력 콘텐츠를 얻은 다음 중앙서버의 프로그램이 이를 분석, 가공 등을 거쳐 디스플레이 장치 등의 출력 모달리티로 사용자에게 결과를 전달한다. 미들웨어는 이러한 모든 입출력 모달리티를 활용하여 웹기반 멀티모달 상호작용을 제공하기 위한 시스템이다. 미들웨어는 다수의 인터페이스를 지원하기 때문에 각 인터페이스는 서로 동기화되어 모달리티 사이의 전환이 가능해야 한다. 예를 들어 비 음성 모달리티를 사용하다가 음성 모달리티로 변경할 수 있으며 반대도 가능하며 동시에 사용하는 것도 가능하다. 본 논문에서는 휴대용 단말인 PDA를 대상으로 입력의 경우 음성과 펜(또는 키패드), 출력의 경우 그래픽과 음성합성을 가정하였다.

2.2. 멀티모달 인터페이스의 분류 및 정의

멀티모달 인터페이스의 분류는 시간적 관점과 결합 성격에 따라 다음과 같이 분류된다. 시간적 관점에 따라 순차(sequential)와 동시(simultaneous) 멀티모달, 결합 성격에 따라 단순(alternate)결합, 보조(supplementary)결합 멀티모달이 있다[4]. 시간적 관점의 멀티모달은 시간 축에서 겹침 현상 유무에 따른 분류이다. [4]에 따르면 순차 모달리티는 어떤 시점에 활성화된 모달리티가 하나라는 의미로 정의돼 있다. 그러나 본 논문에서는 2가지 모달리티가 항상 활성화되어 있다고 가정하였다. 여기에서 순차는 활성화 여부가 아닌 입력의 측면에서 단지 시간 축에서 겹치지 않는 멀티모달의 의미[5]로 사용되었다. 따라서 순차 모달리티 입력은 시간 축에서 겹치지 않는 반면 동시 모달리티 입력은 겹치게 된다.

결합 성격에 따른 멀티모달은 단순결합과 보조결합이 있다. 보조결합은 복합(composite) 멀티모달이라고도 하며 결합된 모달리티 사이에서 협력으로 정보를 얻는 방식이다. 활성화된 각 모달리티로 부터 얻은 정보는 부분(partial)정보로서 서로 통합 해석되어야 비로소 완전(full)정보가 된다. 반면 다수의 모달리티가 활성화될 때 단순 결합 입력은 완전한 의미를 가진 정보로 해석하여 처리한다. 단순결합의 경우 다른 모달리티로 동기화(synchronization)가 진행된다. 예를 들어 "뒤로 가기"라는 음성 입력은 완전정보로서 비 음성 출력 모드로 정보가 동기화되어 사용자는 음성인식의 결과를 화면 출력으로 확인할 수 있다.



<그림 1> 시간에 따른 보조결합 멀티모달

<그림 1>은 보조결합 멀티모달의 예이다. 첫 번째 그림은 동시입력, 두 번째 그림은 순차입력을 나타낸 것이다. 지도상에서 경로 긋기가 발생하면 인터랙션 매니저는 나머지 입력정보가 완성될 때까지 지정된 대기시간 동안 음성 모달리티의

입력을 기다린다. 이런 상태를 "wait and hook"이라고 한다. 대기시간 이내에 입력이 없으면 먼저 입력된 정보는 초기화된다.

2.3. 멀티모달 인터페이스 표준

W3C에서는 미들웨어 플랫폼에 대한 표준화 작업을 진행 중이며 음성과 비 음성 입출력 장치를 지원하는 모바일용 웹 응용 프로그램을 개발하기 위한 언어도 함께 제시하고 있다. 본 미들웨어는 W3C 소프트웨어 아키텍처의 멀티모달 표준의 제안에 따라 분산형으로 설계되었다. 클라이언트의 경우 모바일 윈도우 운영체제 하에서 동작하는 단말용 브라우저가 펜 클릭, 음성 인식 모듈 이외에 합성음을 처리할 수 있게 개발되어 사용자로 하여금 멀티모달 웹 조작이 가능하도록 하였다. 본 미들웨어는 다음에 열거되는 여러 가지 표준들을 참고하거나 재사용하여 모듈 사이에는 독립적이고 느슨한(loosely-coupled) 관계를 갖도록 하였다.

2.3.1. VoiceXML

VoiceXML[6]은 음성으로 웹 정보를 탐색하는 응용 프로그램을 개발하기 위한 환경을 제공하는 대화형 마크업이다. 합성음과 음성인식 그리고 DTMF 키 입력을 처리하고 사용자 시스템 사이의 혼합 주도형 대화 시나리오를 처리하기 위해 고안되었다. 현재까지 W3C에서는 VoiceXML 2.1 버전이 실제 응용 프로그램을 위해 권장되어 있고 VoiceXML 3.0이 차세대 VoiceXML의 표준으로 제시되어 있다. 차세대 버전에서는 현존하는 대화 시나리오와 미디어 제어의 개선에 주안점이 두어졌다. 여기에 데이터, 흐름, 다이얼로그를 기능적으로 분리하고 비동기 외부 이벤트를 용이하게 처리할 수 있는 구조를 갖추었다.

VoiceXML은 음성기반의 현존하는 모달리티 콤포넌트 언어 중에서 가장 우수한 후보 중 하나이다. 응용 프로그램 서버는 음성 인터페이스를 위해 VoiceXML로 작성된 도큐먼트를 주재(hosting)하고 비 음성 인터페이스를 위해 XHTML로 작성된 도큐먼트를 주재한다. 음성 브라우저는 음성 인터페이스를 구동하여 VoiceXML 인터프리터로 하여금 VoiceXML로 작성된 도큐먼트를 HTTP로 패치하여 음성 상호작용을 위한 적절한 행동을 취하게 한다. 또한 음성 브라우저는 음성인식 장치(ASR), 음성합성장치(TTS), SIP/RTP 등 음성 스트림을 전송하고 제어하기 위한 사용자 에이전트를 탑재하고 있다.

미들웨어 관점에서 VoiceXML과 같이 사용자와 상호작용을 담당하는 부분을 모달리티 콤포넌트라 부른다. 멀티모달 실행 프레임워크는 모달리티 콤포넌트를 등록하거나 제거하는 등의 저작 기능을 수행한다. 모달리티 콤포넌트의 수행은 블랙박스로 모듈화되어 이벤트와 메시지 통신만으로 서버와 의사소통을 한다. 본 미

들웨어는 XHTML, VoiceXML, SVG를 채택하였다.

2.3.2. SVG

SVG(Scalable Vector Graphics)는 2차원 그래픽을 표현하기 위해 XML을 기반으로 만들어진 언어로서 W3C에 의해 제안된 XML 그래픽 표준이다. XML의 개방성, 상호 운용성 등의 장점을 모두 수용하며, SMIL, GML, MathML 등 다른 XML 언어들과 결합시켜 다양한 웹 어플리케이션으로 응용할 수 있다. 본 미들웨어에서는 지도 검색을 위해 탑재되었으며 지도의 픽셀을 벡터방식으로 처리하여 해상도와 관계없이 지도 좌표를 확대 또는 축소할 수 있다.

2.3.3. X+V

현재 2개 이상의 모달리티를 허용하는 멀티모달 시스템이 다양하게 제시되어 있다. 그 중 마크업을 사용하는 하나의 제안은 X+V[7]의 멀티모달 시스템이다. X+V란 IBM에서 개발한 형태로 VocieXML이 XHTML에 내포된 구조이다. 그러나 이 방법은 입출력 동기화와 단순결합 멀티모달을 비교적 쉽게 처리할 수 있는 반면 보조결합 멀티모달의 처리가 어렵고 VoiceXML의 일부 기능이 제한된다. 또한 XHTML과 VoiceXML이 완전히 분리되지 않아 기존의 단일 모달리티 모드에서 동작하던 XHTML과 VoiceXML 시나리오를 상기 X+V 형식에 맞게 수정해야 한다.

2.3.4. EMMA

EMMA[8]란 Extensible Annotation Markup Language의 약자로서 멀티모달 상호작용에 있어 사용자 입력을 XML 마크업을 표현하기 위한 W3C의 규격이며 현재도 차기 버전 작업이 진행 중이다. EMMA는 음성, 펜, 키보드, 필기체 등 사용자의 입력을 받아 환경 정보를 메타 데이터에 실어 XML스타일로 정형화한 데이터 구조로서 멀티모달 시스템에서 서로 다른 콤포넌트 사이의 데이터 교환이 가능하도록 해준다. 구체적인 동작과정은 4장에서 상술한다.

2.3.5. DCI(Delivery Context Interface)

서비스의 목적은 개인 또는 환경 변수에 따른 다양한 서비스를 단일 미들웨어에서 가능하게 만드는 것이다. 동적 실제 환경에 적응하기 위해서는 다양한 콘텍스트를 수집, 이용하여 개인별로 맞춤형 서비스의 능력을 갖춰야 한다. 미들웨어는 콘텍스트 정보를 동기(synchronously) 또는 비동기(asynchronously)방식으로 얻는

다. 동기 방식의 경우 미들웨어는 DCI가 탑재되어 있다. 미들웨어의 인터랙션 매니저는 DCI와 인터페이스 동작으로 질의문을 보내고 단말 정보를 얻어 온다. 단말 정보는 배터리 잔량, 전파 수신세기 등 실시간 정보가 DB형태로 디바스 프로파일 [9]에 저장되어 있다. 비동기 방식의 경우 미들웨어는 외부 서비스 번들과 통신할 수 있는 I/O프로세서를 가지고 있다. 외부 번들은 온톨로지 등 특정 도메인의 지식 서비스와 연결되어 상황에 따른 비동기식 서비스를 제공한다. 예를 들어, 외부 번들이 날씨 서비스 번들인 경우 사용자가 고속도로를 운전하면서 전방에 안개 지역이 나타날 때 비동기적 신호로 긴급메시지를 통보해 준다. 통상 미들웨어로는 RDF 데이터 형식의 비동기식 정보가 전달되어 서블릿 번들에 의해 XHTML, VoiceXML 등 PDA의 브라우저로 하여금 시청각 마크업으로 변환되어 사용자에게 표시해 준다.

2.3.6. SCXML

SCXML은 비동기식(Asynchronous) 이벤트 기반 범용(general-purpose) 상태기계 언어로 버전 1.0이 출시되었다. SCXML은 모달리티 콤포넌트를 지원하는 웹기반 모바일용 응용 프로그램 수행이 가능한 시나리오이다. 기존의 단일 모달리티를 처리하는 XHTML과 같이 개발자가 SCXML을 조작함으로써 멀티모달 시나리오를 처리할 수 있다.

SCXML은 VoiceXML에서 전화망의 호 처리를 담당했던 CCXML과 해럴 상태 차트의 개념을 결합하여 XML 스타일의 기계 판독이 가능한 구조로 거듭났다. 해럴 상태 차트는 UML 모델링 언어의 근간으로서 SCXML은 UML의 기본 개념 이외에 몇 가지 요소가 추가되었다. 예컨대, 기존의 상태(state), 전이(transition), 이벤트(event), 조건(condition) 등의 기본 개념 이외에 조건(conditional) 전이(transition), 포대기형(nested) 상태, 외부 콤포넌트 구동 그리고 병렬(parallel) 상태 기능이 추가되었다.

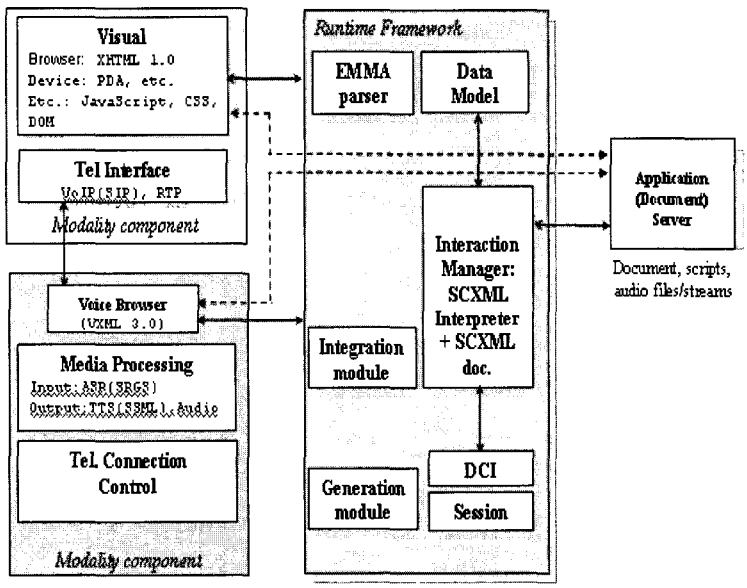
전이의 경우 무조건 전이와 이벤트 전이 뿐 아니라 감시 조건(guard condition)에 의해서도 현재 상태에서 다른 상태로 이동이 가능하여 정교하고 복잡한 상태 기계의 다양한 분기를 지원한다. 포대기형 상태는 부모상태(superstate)가 자식상태(substate)를 재귀적으로 포함하여 구조를 단순화 또는 캡슐화한다. 히스토리 기능은 가장 최근 방문했던 상태를 SCXML이 기억함으로써 예상치 못한 종료 등 예외 상황에 잘 대처할 수 있게 하였다.

SCXML은 동시에 둘 이상의 상태가 활성화됨을 의미하는 병렬상태를 지원한다. 멀티모달 프로세싱을 위해 SCXML은 VoiceXML과 XHTML페이지를 두 가지 병렬 상태로 맵핑한 다음 동시에 이들 상태를 활성 상태로 만들어 메시지를 사용자에게 전달하거나 입력 값을 수집한다. 단순결합 또는 보조결합 멀티모달 입력을

처리하기 위해 병렬 상태의 두 가지 속성이 사용된다. 첫째, 병렬상태로 동시에 활성화된 상태들 중 어느 하나에서 상태 밖으로 빠져 나오는 조건이 성립하면 나머지 활성 상태도 자동적으로 종료, 소멸된다. 둘째, 활성화된 어느 상태 영역에서 다른 상태의 위치를 엿볼 수 있는 <In> 연산자가 사용되어 활성화된 상태들에 대한 자식 상태의 움직임을 서로 감지할 수 있다.

3. 미들웨어 구조

3.1. 개요



<그림 2> 미들웨어 구조

본 미들웨어의 구조는 W3C의 멀티모달 상호작용 프레임워크의 제안을 참고하여 제작되었다. <그림 2>에 도시된 바와 같이 미들웨어는 모달리티 콤포넌트, 런타임 프레임워크, 외부 모듈(응용 프로그램 서버)로 구성되어 네트워크 전반에 걸쳐 재사용 가능한 마크업을 처리한다.

다수의 모달리티를 다루는 경우, 개별 모달리티를 위한 플랫폼과 달리 음성, 펜, 디스플레이 또는 촉각 인터페이스를 통한 멀티모달 I/O를 운용할 수 있는 새로운 플랫폼 디자인이 필요하다. 멀티모달 실행 프레임워크는 모바일 환경의 단말을 구동하기 위한 필수 요소로서 블랙박스로 모듈화된 모달리티 콤포넌트를 제어 감독한다. 실행 프레임워크의 인터랙션 매니저는 이들 모달리티 콤포넌트와 통신

을 위한 저작 도구가 필요하다. 따라서 SCXML 마크업으로 기술된 시나리오를 불러오기 하여 초기화한다. 또한 라이프 사이클 이벤트를 발생시켜 원격으로 동작하는 클라이언트와 미들웨어 사이의 통신을 수행한다. 이외에 모달리티 콤포넌트와 DCI 같은 다른 콤포넌트를 비동기 방식으로 연결해 주는 역할을 한다. VoiceXML, XHTML, SVG 등이 단일 모달을 위한 언어라면 X+V, WML, SCXML은 단일 모달리티들을 협력, 동기화하는 멀티모달을 위한 언어이다.

3.2. 멀티모달 실행 프레임워크

멀티모달 실행 프레임워크는 데이터의 논리적 구조를 담당하는 세션 데이터 관리 모듈, 모든 입출력을 관장하는 입출력 프로세서, 대화 흐름을 제어하는 상태 차트 XML 실행 엔진, EMMA해석기, 상태 차트 XML해석기, 각종 환경 정보 또는 사용자 선호 정보를 담고 있는 DCI로 구성돼 있다. 모달리티 콤포넌트에서 전달된 원시 입력 데이터는 EMMA해석기에서 상태 차트 XML 실행 엔진에서 직접 다룰 수 있는 콘텍스트로 변환된다.

3.2.1. 인터랙션 매니저

인터랙션 매니저는 런타임 프레임워크의 구성 요소로서 다른 콤포넌트가 발생하는 모든 이벤트들을 다룬다. 또한 외부 모듈과의 통신을 위해 로컬 변수들을 관리한다. 이벤트들에 대응하는 방식은 특정 마크업에 기술된 시나리오에 따라 이루어진다. 인터랙션 매니저에서 채택한 멀티모달 언어는 State Chart XML(이하 SCXML)로서 SCXML 해석기, 모델 생성기, I/O 프로세서 등으로 구성돼 있다.

SCXML 실행엔진의 액션 핸들러는 상태기계가 움직일 때 각 위치에서 해당 명령이 수행되도록 한다. 상태기계가 움직일 때의 시점은 상태 진입, 상태 종료, 전이, 상태 내부 등이 있다. 각 시점에서 사전 정의된 액션 핸들러가 수행된다.

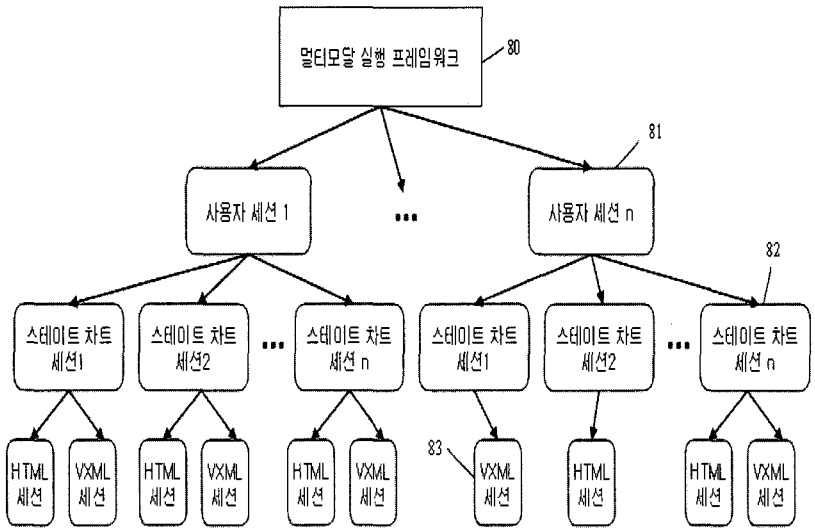
3.2.2. 세션 데이터 관리 모듈

세션 데이터 관리 모듈은 세션이 유지되는 동안 응용 프로그램 레벨의 데이터를 제공하고 변수에 대한 상태를 관리, 유지하는 장치이다. 인터랙션 매니저는 시나리오 제어 로직에 따라 변수들을 생성, 갱신, 소멸시킬 수 있으나 모달리티 콤포넌트에서는 직접 접근이 불가능하다. 본 미들웨어는 SCXML 스펙에서 제공된 기본적인 로컬 변수만을 사용하였다. 이 변수들은 모달리티 콤포넌트의 활성화 여부에 따라 생존 주기를 갖는다.

세션은 새로운 사용자가 접속할 때마다 세션 아이디를 부여하여 세션이라는

고유 영역을 생성하며 상호작용에 따라 파라메터를 교환하거나 세션을 생성, 유지, 종료하는 등 사용자별 멀티모달 실행 프레임워크와 모달리티 콤포넌트 사이의 논리적 대응 관계를 정의한다.

<그림 3>은 세션의 계층구조로서 실행 프레임워크(80)는 최상위 세션이다. 세션은 멀티모달 실행 프레임워크와 모달리티 콤포넌트 영역 양쪽에서 위치한다. 사용자 세션 초기화는 멀티모달 실행 프레임워크 플랫폼에 의해 진행된다.



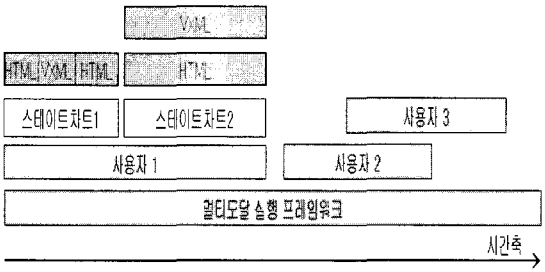
<그림 3> 세션의 구조

응용 프로그램이 시작되면 멀티모달 프레임 세션은 사용자 인증을 거쳐 사용자 세션을 생성하고 응용 프로그램의 최상위 SCXML 마크업을 읽어 들여 스테이트 차트 세션(82)을 초기화한다. 다시 XHTML, VoiceXML 브라우저를 운용하기 위한 소프트웨어를 적재하면서 (83)을 초기화한다. 그리고 매번 초기화 시점에 각각의 콤포넌트 세션이 시작되도록 수행 메시지를 보낸다. 각각의 콤포넌트 세션(XHTML, VoiceXML세션)은 마크업의 메시지로부터 종료시점을 알려오거나 (예, <exit> tag) (83)이 중지(Halt) 이벤트를 보낼 때 종료된다. 이처럼 (81)은 (82)의 운용에, 다시 (82)는 (83)의 운용에 필요한 라이프 사이클 이벤트를 생성하고 비동기 통신을 한다. 상위 세션이 종료되면 모든 각 하위세션은 자동적으로 종료된다.

사용자 세션(81)은 사용자와 상호작용을 하는 전 과정에서 유지되는 반면 하위 스테이트 차트 세션(82)은 사용자 세션이 수행되는 동안만 발생된다. 따라서 하위 세션은 상위 세션 시간 축 이내에 놓이게 된다. 따라서 시간 축으로부터 VXML 세션(83)은 스테이트 차트 세션(82)보다 짧을 수 있으며 상 하위 세션은 각기 다른 지속시간(duration)을 가진다.

예를 들어, <그림 4>의 스테이트 차트 세션 1에서 GUI 상호작용만으로 시작되

는 응용 프로그램이 있어 HTML세션이 생성되었다고 하자. 서비스 도중 사용자가 GUI를 음성모드로 바꾸면 응용 프로그램에서 음성 입출력이 지속되는 동안 HTML세션은 활성화를 멈춘다. 이어 음성 입출력을 마치고 GUI로 바꾸면 HTML 세션에 의해 상호작용이 계속된다. 이 경우 스테이트 차트 세션 1은 3가지 모달리티 콤포넌트 세션으로 만들어진다.



<그림 4> 세션의 생성과 소멸

3.3. 모달리티 콤포넌트

모달리티 콤포넌트는 멀티모달 상호작용이 가능한 응용 프로그램 소프트웨어를 구동하여 사용자와 입출력이 생겨나는 부분으로서 클라이언트에 해당된다. 비음성 입출력의 경우 XHTML 1.0 브라우저에서 처리한다. XHTML은 시각 정보를 보여 주는 브라우저로서 여기에서 자세한 설명은 생략한다. 음성입력의 경우 음성 신호를 전 처리할 수 있는 마이크, 음성특징 추출기와 합성음을 출력할 수 있는 디코더, 스피커에서 처리된다. 한편, 마이크에서 전달되는 신호 중 음성이라고 판단되는 입력에 대하여 일정시간이 지나면 음성입력의 시작과 끝점추출을 하고 음성특징 추출기를 거쳐 음성 브라우저의 음성인식기로 인식을 요청하는 과정은 통상의 음성 입출력 시스템을 따른다. 음성 출력의 경우는 스피커, 디코더, 인코더, TTS에서 처리된다. VoiceXML 인터프리터에 의해 음성을 PDA에 돌려주어야 하는 경우 VoiceXML 스크립트 내에서 음성출력이 텍스트 형태로 저장된다. TTS에서 음성으로 변환된 후 인코더와 PDA의 디코더를 거치면 최종 합성된 음성이 스피커로 출력된다.

서버는 음성인식의 후 처리를 담당할 음성 브라우저와 멀티모달 실행 프레임워크로 나눌 수 있다. 음성 브라우저의 경우 음성인식과 합성의 분산 처리 정책에 따라 서버에 포함되었다. 음성 브라우저는 VoiceXML 인터프리터와 ASR, 음성합성을 담당할 TTS 엔진과 합성 인코더로 구성된다. VoiceXML 인터프리터는 인식 결과에 따라 VoiceXML 스크립트를 실행하여 음성인식 또는 합성을 지시하고 필

요에 따라 이벤트를 멀티모달 실행 프레임워크로 전송한다. ASR는 VoiceXML 인터프리터로부터 받은 인식 대상 단어리스트와 음성데이터를 이용하여 인식 점수를 계산하여 가장 높은 점수를 얻은 인식 대상 단어에 대하여 결과를 다시 VoiceXML 인터프리터로 전달한다.

PDA에서 발생한 입력에서 음성의 경우는 마이크 입력을 분석하여 음성 특징 추출기로부터 특징 데이터를 획득하여 후 처리를 위해 음성 브라우저로 전송하는 분산형 음성인식을 이용한다. 단말의 전 처리 과정은 끝점추출과 특징추출이 일어나고 압축된 정보가 서버로 전달된다. 서버의 음성 브라우저는 후 처리를 담당하여 ASR에서 인식 결과를 얻은 후 EMMA 데이터 형식으로 바꾸어 인터랙션 매니저로 전달한다.

비음성인 경우 XHTML브라우저는 키 입력을 받아 XHTML 브라우저에서 EMMA 형식으로 변환하여 멀티모달 실행 프레임워크로 전송한다. 여기서 전송 형식은 상시 또는 비상시의 특정 통신 프로토콜에 국한되지 않고 1-1 통신이 가능한 임의의 네트워크에 의해 연결된다. 멀티모달 출력에서 음성의 경우는 음성 브라우저의 TTS로부터 인코더에서 음성콘텐츠를 인코딩하여 PDA로 전송하면 디코더에서 디코딩 과정을 거쳐 최종 합성음을 스피커로 내보낸다.

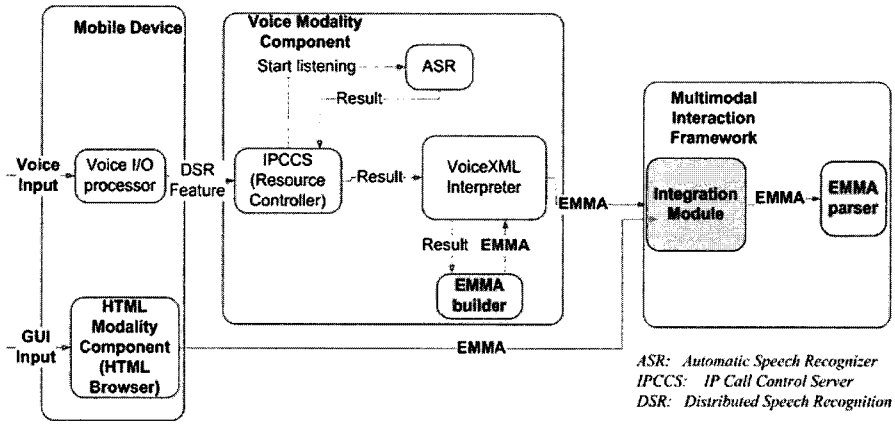
4. 동작과정

4.1. EMMA의 생성과 해석

본 미들웨어는 멀티모달 입력에 대하여 XML 코드로 변환된 EMMA를 이용하였다. 각종 메타데이터로 포장되어 XML 스타일로 변환된 EMMA는 멀티모달 인터랙션(MMI) 프레임워크 또는 멀티모달 실행 프레임워크에서 클라이언트 도메인으로부터 전달 가능한 콘텍스트 파라메터로서 취급된다. 모바일 단말의 각 클라이언트 도메인은 사용자 입력에 대하여 상호작용하는 채널을 제공한다. <그림 5>는 각 클라이언트 도메인, 즉 모달리티 콤포넌트에서 발생된 입력에 대하여 EMMA로 변환되어 멀티모달 인터랙션 프레임워크에 전달되는 과정을 보인 것이다.

음성입력의 경우, 사용자로부터 발생된 원시 음성 신호는 모바일 응용 프로그램의 실행 시에 음성 I/O 프로세서에서 처리된다. 원시 음성 신호로부터 DSR(Distributed Speech Recognition) 특징이 추출되어 RTP(Real-time transport protocol)에서 자원 제어기인 IPCCS(IP call control server)로 넘겨지고 ASR(Automatic Speech Recognition)서버는 음성인식 결과를 내놓는다. VoiceXML 해석기는 인식 결과를 EMMA 빌더로 보내어 XML 형식의 EMMA 데이터를 되돌려 받은 뒤 멀티모달 인터랙션 프레임워크의 통합 모듈로 보낸다.

비 음성 입력의 경우, XHTML 브라우저의 GUI 입력은 PDA 브라우저에서 바로 EMMA 형식으로 변환되어 간단히 멀티모달 인터랙션 프레임워크의 통합모듈로 전달된다.



<그림 5> EMMA를 이용한 사용자 입력의 생성과 해석

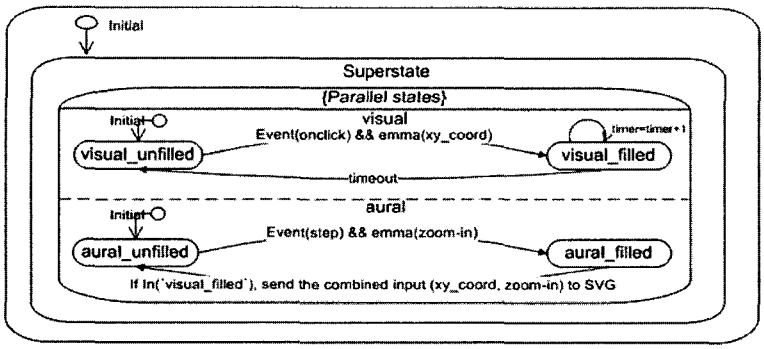
4.2. 멀티모달 통신

다수의 모달리티로부터 들어온 EMMA입력은 실행 프레임 워크의 통합 모듈에서 처리된다. EMMA파서는 EMMA XML 트리를 순회하면서 JEXL 표현언어의 해쉬 맵(hash map) 형태의 속성과 값의 쌍으로 정렬된 리스트로 변환한다. JEXL 표현 언어로 변환된 사용자 입력은 SCXML 스크립트 안에서 자유롭게 기술이 가능하며 SCXML은 이들 값을 외부 콘텍스트로 간주하여 세션 데이터 관리 모듈에서 관리한다. SCXML 도큐먼트가 SCXML 파서에 의해 해석되면 자바 오브젝트 모델이 생성된다. 이어서 사용자 입력에 따라 결정적(deterministic) 방식으로 상태기계가 동작한다. SCXML이 사용자에게 2가지 모달리티로 입력을 허용한다면 2개의 병렬 상태가 생성되어 활성화된다. 본 시스템의 병렬상태는 XHTML과 VoiceXML 모달리티 컴포넌트로 맵핑되었다.

4.2.1. 멀티모달 입력 처리

사용자가 지도상에서 한 지점을 클릭하면서 "확대(zoom-in)"라고 발성하는 시나리오를 가정해 보자. 사용자로부터 얻어지는 입력은 보조결합 멀티모달로서 각 입력된 부분 정보가 해석되고 이에 따라 각 모달리티의 상태가 전이한다. <그림 6>

는 지도 확대 축소의 SCXML의 동작을 다이어그램으로 나타낸 그림이다. 이와 동등한 SCXML 마크업은 <그림 7>에 표기되어 있다.



<그림 6> 보조결합 멀티모달 입력 처리를 위한 SCXML 다이어그램

보조결합 멀티모달 입력 동작을 상세히 설명하면 다음과 같다. 멀티모달 실행 프레임워크가 XHTML과 VoiceXML의 입력 상태를 나타내는 2개의 병렬상태 visual과 aural을 모델링하여 활성화한다. XHTML 브라우저와 VoiceXML브라우저가 접속되면 사용자 입력대기 상태가 된다. 각 모달리티에서 사용자 입력이 발생하면 조건 전이에 의해 그 이벤트의 속성이 부분정보인가 완전정보인가를 판별한다. 그 정보가 완전정보이면 다른 모달리티의 입력 대기 없이 병렬상태를 종료하고 다른 쪽 모달리티에 동기화 신호를 보낸다. 예컨대, 개별 모달리티에서 통합이 필요 없는 "처음으로" 또는 "돌아가기" 등 음성 명령은 자체로서 완전 정보의 의미를 지닌다. 따라서 이 경우 다른 쪽 모달리티에 동기신호를 보내 출력 값을 동기화시킨다. 상기 <그림 6>과 <그림 7>에서 이 부분의 묘사는 생략되었다.

XHTML 입력 이벤트의 속성이 지도의 좌표이면 이 입력은 부분정보로서 visual_unfilled에서 visual_filled로 전이하고 좌표 값은 SCXML 로컬 변수에 저장된다. 마찬가지로 입력 수신 대기 상태의 다른 쪽(음성) 모달리티에서 타임아웃 이내에 부분정보(음성입력 "zoom-in")를 수신하면 aural_unfilled에서 aural_filled로 전이한다. 병렬 상태의 모든 자식 상태는 각각 visual_filled와 aural_filled에 존재하게 된다. 이 때 aural의 엿보기(In) 기능에 의해 SCXML은 병렬 상태를 초기화하면서 각 부분정보들을 통합한다. 만일 입력 이벤트가 발생하지 않고 타임아웃이 발생하면 값이 채워지지 않은 상태로 초기화된다.

타임아웃이 발생하는 상호 대기 시간 임계값(threshold)는 5초~10초 사이에서 고정된 값을 사용하였다. 따라서 모달리티 콤포넌트를 통해 입력된 다중 입력들은 그 해석에 따라 다른 모달리티로 동기화 되거나 완전 정보로 통합된다. 이와 같이 XML 마크업에 의해 여러 형태의 모달리티 입력 처리가 가능하다.

```
<state id="system_on">
    <parallel id="parts">
      <state id="visual">
       <initial> <transition target="visual_unfilled"/></initial>

       <state id="visual_unfilled">
        <transition event="svg.dialog.onclick">
          <assign name="xcoord" expr="emma.formSVGTest.svgXCoord"/>
          <assign name="ycoord" expr="emma.formSVGTest.svgYCoord"/>
         <send sendid="5" target="SCXMLSession" targettype="mmi" event="timer.start" namelist="waiting_time"/>
          <target next="visual_filled"/>
        </transition>
       </state>
       <state id="visual_filled">
        <transition event="time">
             <assign name="visual_timer" expr="waiting_time"/>
              <target next="visual_unfilled"/>
        </transition>

        <transition cond="!In('aural_filled')">
         <send sendid="5" target="SCXMLSession" targettype="mmi" event="timer.stop" namelist=""/>
             <target next="visual_unfilled"/>
        </transition>
        <transition event="svg.dialog.onzoomin" target="visual_unfilled"/>
        <transition event="svg.dialog.onzoomout" target="visual_unfilled"/>
       </state>
   </state>

   <state id="aural">
    <initial><transition target="aural_unfilled"/></initial>

    <state id="aural_unfilled">
          <transition event="voicexml.dialog.step" cond="emma.command == 'zoomin'" target="aural_filled"/>
          <transition event="voicexml.dialog.step" cond="emma.command == 'zoomout'" target="aural_filled"/>
    </state>

    <state id="aural_filled">
     <transition cond="emma.command=='zoomin'">
      <send sendid="9" target="HTMLSession" targettype="html" event="svg.dialog.zoomin" namelist="objectid xcoord ycoord"/>
      <target next="aural_unfilled"/>
     </transition>

     <transition cond="emma.command=='zoomout'">
      <send sendid="9" target="HTMLSession" targettype="html" event="svg.dialog.zoomout" namelist="objectid xcoord ycoord"/>
      <target next="aural_unfilled"/>
     </transition>

    </state>
   </state>
  </parallel>
 </state>
</state>
```

<그림 7> 보조결합 멀티모달 입력을 처리하기 위한 SCXML 마크업

4.2.2. 멀티모달 출력 처리

멀티모달 출력은 모달리티 수만큼 병렬 상태를 초기화한 후 음성 합성기 또는 디스플레이 장치 등에 구동 메시지를 보낸다. 예컨대, 상태를 3개로 설정하면 결과 테이블은 디스플레이로, 배경음악은 스피커로, 안내음은 음성합성기로 동시에 출력된다. 이처럼 본 미들웨어는 멀티모달 동기화 조정기, 융합 엔진 등 별도의 하드웨어 장치를 필요로 하지 않고 SCXML 마크업 시나리오만으로 멀티모달 입출력 제어를 할 수 있다.

4.3. 상황인지 인터페이스

본 미들웨어는 동기식 또는 비동기식 메시징 방식으로 동적 환경에 적응하는 인터페이스를 탑재하였다. 동기식 방법으로는 DCI를 이용한다. 사용자와 단말 프로파일 DB의 저장소인 DCI는 디바이스 속성과 사용자 선호 데이터가 저장된 DB이다. PDA가 서버에 접속되면 디바이스 정보가 DCI에 로드되고 인터랙션 매니저로 하여금 동적으로 데이터에 접근할 수 있다. 디바이스 정보는 화면 크기, 배터리 레벨, 메모리, 네트워크 신호 세기 등이며 사용자 선호 정보는 지능 레벨, N-best 후보 수 등을 사용자가 직접 설정할 수 있다. 인터랙션 매니저는 DCI에 질의문을 보내 결과 값을 얻으며 이 값으로 응용 프로그램 서버에 이와 관련된 개인화된 프로파일 정보를 얻는 데 활용한다. 비동기식 방법은 외부 서비스 번들을 통해 미들웨어에서 해결하기 어려운 추론 등의 지식기반 서비스를 위해 만들어졌다. 온톨로지 기반의 콘텍스트 정보는 RDF 형식으로 전달돼 사용자에게 보이기 위한 정보로 변환된다.

우리는 서비스가 동작함을 보이기 위해 DCI와 외부 추론 엔진을 활용한 웹 응용 프로그램을 개발하였다. 첫 번째, DCI에 지능 단계를 두어 능숙한 사용자가 상호작용 중에 시스템의 빈번한 간섭을 회피하고, 초보 사용자는 시스템이 자주 관심을 갖도록 하는 설정이 가능케 하였다. 두 번째, 교통이나 날씨 등 예외 상황이 발생하면 예정 시각보다 일찍 알람이 울리는 지능형 알람 서비스이다. 외부 모듈과 연동하여 온톨로지에 의한 추론화된 지식을 활용하여 변경된 시각을 얻어온다. 따라서 본 미들웨어는 인터페이스를 통해 모바일 서비스를 한층 지능적으로 구성할 수 있게 되었다.

5. 실 험

본 논문에서는 두 가지 다른 실험을 통해 멀티모달 통신에 대한 평가와 능력을 검증하였다. 첫 번째, 본 미들웨어가 음성기반 멀티모달 통신을 수행하는 만큼 상시(free-running) 인식 모드로 동작하는 단말용 인식 프로그램을 개발하였다. 프로그램이 단말기 상에서 메모리에 상주하여 대기하다가 사용자가 발성 시 음성 발성 키워드를 감지하고 OOV(Out-Of-Vocabulary) 즉, 인식대상어휘가 아닌 경우 거부하는 기능을 갖고 있다.

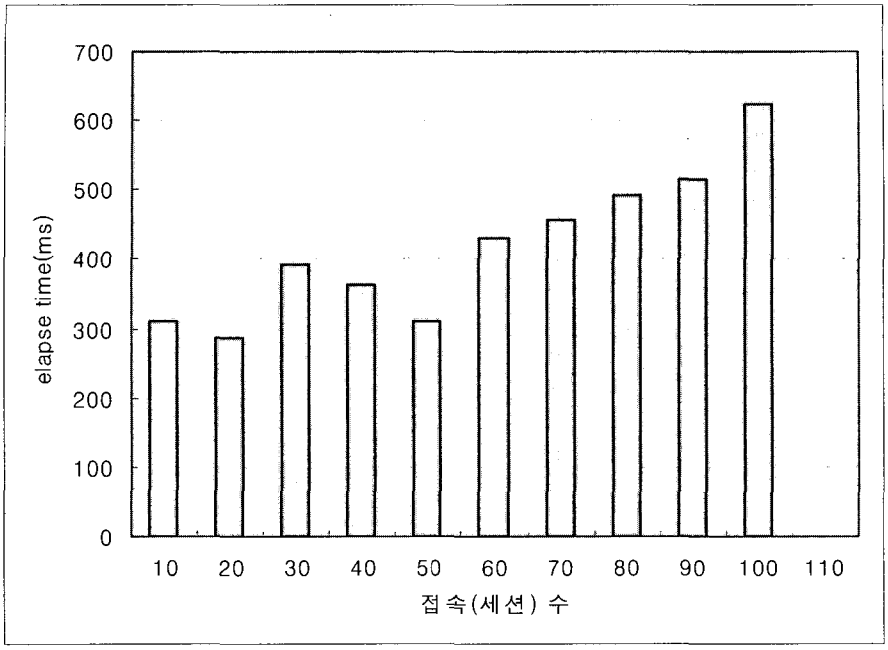
<표 1> 2가지 단말의 인식테스트

Test DB	Recognition rate (%)
RW6100 (982)	95.72
SPH4300 (945)	95.45
Noise (500)	62.40
Total (2,427)	88.75

우리는 삼성 SPH4300과 HP iPAQ-RW6100의 두 가지 PDA에 대한 실험을 수행하였다. 음성 코퍼스에서 대략 20,000개의 어휘를 수집하여 1:9 비율로 테스트와 훈련 DB로 나누었다. 인식 대상 어휘는 37개의 단어로 구성된 자동차 내비게이션용 명령어 셋으로 "확대", "축소", "위로", "아래로" 등 통상 차량용 내비게이션을 조작하는 2 단어 이내의 어구(phrase)들이다. <표 1>에서와 같이 테스트 코퍼스는 HP PDA의 982개 어휘, 삼성 PDA의 945개 어휘, 500개의 OOV로 구성되었다.

인식률은 모두 95%이상을 기록하였고 OOV 테스트에서도 62.4%의 거부율을 보였다. 시나리오는 음성, 비 음성 모드를 결합한 멀티모달 모드로 운용되어 멀티모달 입력의 경우 비 음성 모드는 지점 클릭 등의 펜 입력으로 구성돼 인식 성공률이 100%이다. 따라서 전체 보조결합 멀티모달 입력의 인식성공률은 음성 인식 성공률에 전적으로 의존한다.

두 번째 실험은 음성, 비 음성 환경에서 웹 접근의 메시징 패턴을 모델링하고 사용자 세션을 모의실험 하려는 목적으로 이루어졌다. 사용자 클라이언트가 서버에 접속 인증될 때마다 SCXML 세션을 생성하여 여러 개의 가상 사용자를 만든 다음 이들로 하여금 단순 또는 보조결합 멀티모달 입력을 위한 메시지를 동시에 발생시켜 평균 처리 시간을 체크하였다. 미들웨어는 사용자 수만큼 SCXML 세션을 생성하고 각각의 세션은 또 다시 VoiceXML과 XHTML 세션을 만들어 브라우저를 구동하게 된다. 우리는 간단한 SCXML 스크립트를 제작하여 여러 사용자에 대한 단순결합 또는 보조결합 입력 메시징을 통해 다수 사용자에 대한 미들웨어의 수행능력을 파악하였다.



<그림 8> 멀티 사용자 테스트

우선 미리 작성된 SCXML 스크립트를 읽어 n명의 사용자를 가정한 n개 세션을 동시에 활성화하였다. 이어 각 세션에서 동시에 단순결합 또는 보조결합 입력 메시징 처리를 모의 실험하여 1 사이클에 대한 평균 수행 시간을 기록하였다. 1 사이클이란 클라이언트에서 입력이 발생하여 서버에 통보되고 클라이언트가 확인 메시지를 받는 일련의 메시징 과정을 말한다. <그림 8>은 위 실험 결과로서 n을 증가시키면 평균 처리시간도 단조 증가함을 나타내었고 대략 n을 100전후로 늘릴 때까지 안정된 처리 능력을 보여 주었다. n이 110 이상일 때에는 시스템이 불안정하였다. 이는 서버 메모리와 인터랙션 매니저의 메시지 큐가 일시에 처리할 수 있는 범위를 넘어선 것으로 추정된다.

6. 결 론

본 논문에서는 분산형 미들웨어를 설계, 구현하였다. 멀티모달 실행 프레임워크는 전체 미들웨어의 근간이며 모달리티 컴포넌트는 사용자의 시스템 사이에서 상호작용할 수 있도록 마크업으로 동작한다. XHTML, VoiceXML, SVG 마크업은 이러한 상호작용을 위한 시나리오를 작성하는 데 이용된다. 대화 매니저는 복수의 모달리티 컴포넌트로부터 단순결합 또는 보조결합 입력을 받아 적절한 가공을 거

쳐 결과를 표시해 주거나 외부 모듈과 연동하여 추가적인 서비스 요청을 한다.

음성 기반 응용 프로그램의 테스트 결과는 첫째, 필드에서 강인한 인식성공률과 거부율을 나타냄에 따라 원활한 보조결합 또는 단순결합 멀티모달 인터페이스를 지원하였다. 따라서 멀티모달 응용 프로그램이 둘 이상의 모달리티에서 상호협력모델로서 효율적인 인터페이스로 운용될 수 있음을 보였다. 둘째, 여러 사용자가 동시에 접속하여 상호작용하는 서비스 중에도 적정 시간 이내에서 안정적으로 처리되었다. DCI와 외부의 지식 서버를 연동하여 응용 프로그램을 제작하면 개인화된 맞춤형 서비스와 환경변화에 적응력을 높인 상황인지 서비스도 가능하다.

다음 과제는 멀티모달 통신에서 사용자의 명시적(explicit) 입력뿐 아니라 GPS 위치 정보, 모션(gaze), 입술의 움직임 같은 센서 기반의 비명시적(implicit) 입력도 처리해야 한다. 미들웨어는 모달리티 컴포넌트를 추가하여 기존 모달리티와 결합하여 수행된다. 여기에는 다양한 멀티모달 통신 패턴에 대한 연구와 보조결합 입력을 효율적으로 통합 처리하는 방안이 모색되어야 한다. 특히 보조결합 입력 시 상호 대기 시간 임계값(threshold)은 사용자 별로 차이점을 보이는 것으로 행동 패턴에 관한 임상 실험 결과[5]가 보고되고 있다. 따라서 고정된 값을 사용하지 않고 사용자 습성에 적응하여 임계값을 기계적 학습으로 얻어오는 연구가 진행 중이다. 앞으로 온톨로지 등 지식 정보가 체계화되어 미들웨어가 환경 콘텍스트를 지금보다 다양하고 수월하게 수집하여 이용할 수 있다면 보다 지능적인 모바일 응용 프로그램 서비스가 가능해 질 것이다.

참 고 문 헌

[1] J. A. Larson, T. V. Raman, D. Raggett, "W3C Multimodal Interaction Framework", W3C, http://www.w3.org/TR/mmi-framework, May. 2003.

[2] R. J. Auburn, J. Barnett et al., "State Chart XML(SCXML): State Machine Notation Abstraction 1.0", W3C, http://www.w3.org/TR/WD-scxml-20050705/, Jul. 2005.

[3] K. Waters, R. Hosn et al., "Delivery Context Interfaces(DCI) Accessing Static and Dynamic Properties", W3C, http://www.w3.org/TR/2005/WD-DPF-20051111/, Nov. 2005.

[5] S. Oviatt, "Toward Adaptive Information Fusion in Mulimodal Systems", *ICMI '05* invited talk, pp. 3-5 & 9-19, Nov. 2005.

[6] Voice Extensible Markup Language(VoiceXML) Version 2.0, W3C, http://www.w3.org/TR/voicexml20/, Mar. 2004.

[7] XHTML+Voice(X+V) Profile 1.2, http://www.voicexml.org/specs/multimodal/x+v/ 12/spec.html, W3C, Mar. 2004.

[8] W. Chou, D. A. Dahl et al., "Extensible MultiModal Annotation markup language (EMMA)", W3C, http://www.w3.org/TR/ emma/, Sep. 2005.

[9] Composite Capability/Preference Profiles(CC/PP): Structure and Vocabularies 1.0, http://www.w3.org/TR/CCPP-struct-vocab/, W3C, Jan. 2003.

접수일자 : 2006년 8월 30일
게재결정 : 2006년 12월 20일

▶ 박성수(Seong-Soo Park)
주소: 137-792 서울특별시 서초구 우면동 17번지 KT
소속: KT 미래기술연구소 프로젝트 그룹
전화: 02) 526-6522
E-mail: soopark@kt.co.kr

▶ 박성찬(Sung-Chan Park) : 교신저자
주소: 137-792 서울특별시 서초구 우면동 17번지 KT
소속: KT 미래기술연구소 프로젝트 그룹
전화: 02) 526-6725
E-mail: rapport@kt.co.kr

▶ 안세열(Seyeol Ahn)
주소: 137-792 서울특별시 서초구 우면동 17번지 KT
소속: KT 미래기술연구소 HCI연구담당
전화: 02) 526-6732
E-mail: syahn@kt.co.kr

▶ 김원우(Won-woo Kim)
주소: 137-792 서울특별시 서초구 우면동 17번지 KT
소속: KT 미래기술연구소
전화: 02) 526-6163
E-mail: wwkim@kt.co.kr

▶ 구명완(Myoung-Wan Koo)
주소: 137-792 서울특별시 서초구 우면동 17번지 KT
소속: KT 미래기술연구소 HCI연구담당
전화: 02) 526-5090
E-mail: mwkoo@kt.co.kr