홍 기 형 교수지도

석사학위 청구논문

로봇에서의 멀티모달 결합을 위한 표준 인지언어의 설계

2007

성신여자대학교 대학원

전산학과

민 경 현

로봇에서의 멀티모달 결합을 위한 표준 인지언어의 설계

홍 기 형 교수지도

이 논문을 석사학위논문으로 제출함

2007년 2월

성신여자대학교 대학원
전산학과
민 경 현

논문 개요

멀티모달 상호작용 기술과 관련하여, 여러 채널을 통한 입력 정보의 인지 및 통합 결과를 동일한 형식으로 기술할 수 있는 표준 멀티모달 입력 기술 언어를 설계하였다.

본 논문에서는 로봇이 가진 다양한 입력 모달리티를 이용하여 사용자의 의도를 정확하게 파악하기 위한 멀티모달 인터페이스 개발을 위하여, 로봇의 다양한 입력 장치나 인지 모듈에서의 개별적인 인지 결과 기술 양식과 각 입력 정보를 통합한 멀티모달 인지 결과를 XML 기반으로 통일하여 사람과 로봇의 상호작용을 위한 멀티모달 입력 정보 기술언어인 ROMMA의 설계에 대하여 기술한다.

ROMMA는 RObot MultiModal Annatioan markup language의 약어로 W3C의 멀티모달 인터랙션 그룹에서 멀티모달 입력 정보 기술언어의 표준안으로 제정된 EMMA를 기본으로 로봇 인터페이스에 맞게 재구성한 것으로, 새로운 모달리티 및 로봇의 취할 수 있는 작업 영역 추가와 관련하여 확장성을 고려하여 설계하였으며 본 논문에서는 작업 영역과 관련하여 심부름을, 모달리티와 관련하여 감정 모달리티에 대한 확장 엘리먼트를 정의하였다. 또한 다양한 모달리티의 인지 결과 기술 양식을 통일하여 음성 인식, 제스처 인식, 감정 인식 등의 각 인지 모듈의 연구 개발이 멀티모달 통합 연구와 상호 독립적으로 이루어 질 수 있도록 한다.

목 차

표 목차

그림 목차

I. 서론

사람과 로봇간의 상호작용 기술은 새로운 도전을 제시하는 분야로, 사람과 사람 사이의 의사소통과 같이 보다 자연스럽고 편리한 상호작용이 이루어질 수 있는 방법을 추구하고 있다. 이를 위하여 주된 모달리티인 음성뿐만 아니라 제스처, 시선 등과 같은 여러 모달리티의 사용이 가능한 멀티모달 인터페이스가 제공되어야 한다.

현재 로봇의 멀티모달 인터페이스에서 모달리티별로 인지 결과 기술 방법의 상이함으로 인해 개별 모달리티의 대체나 수정이 발생하면 전체 로봇 시스템의 재구성이 요구되는 상황이다.

본 논문에서는 개별 모달리티의 인지 결과에 대한 통일된 기술을 위해 ROMMA(RObot MultiModal Annatation markup language)라는 마크업 언어를 설계하였다. ROMMA는 기존의 웹 인터페이스를 위하여 W3C에서 XML 기반으로 EMMA[1]를 로봇 인터페이스에 맞게 재정의 한 것이다.

ROMMA의 구성요소는 크게 기본 엘리먼트와 확장 엘리먼트로 구성된다. 기본 엘리먼트는 모달리티로부터의 인지 정보 및 통합 결과를 담기 위한 구조를 정의하며, 기본 엘리먼트의 하위 구조는 모달리티로부터 입력된 내용을 기술하기 위한 확장 엘리먼트로 구성된다. 확장 엘리먼트는 로봇이 수용 가능한 모달리티별 또는 작업 영역별 관련된 정보를 기술하기 위한 엘리먼

트로 새로운 모달리티 및 작업 영역의 추가와 관련하여 확장성이 용이하도록 설계하였다. 현재는 작업 영역과 관련하여 심부름 영역을, 모달리티와 관련하여 감정 및 포인팅 제스처 영역을 정의하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로 멀티모달 인터페이스에 관한 기존 연구에 대하여 기술하며, III장에서는 ROMMA의 개념 소개 및 설계 시 고려 사항, 그리고 기본 구조에 대해 설명한다. IV장에서는 작업 영역 및 모달리티와 관련하여 확장한 ROMMA의 확장 엘리먼트를 정의하며 V장에서는 ROMMA 기반으로 로봇의 인지모듈 결합 방법을 제시한다. 마지막으로 VI장에서는 결론 및 향후 연구에 관하여 기술한다.

II. 관련 연구

2.1 멀티모달 인터페이스

멀티모달 인터페이스(Multimodal Interface)는 사람과 시스템간의 상호작용에 있어서 다양한 모달리티를 사용하여 대화하는 형식으로, 1980년대 MIT 대학의 Bolt[2]는 기존의 직접 조작(direct manipulation)의 방식인 WIMP (Windows Icons Mouse Pull-down menus) 인터페이스로부터 음성과 제스처를 동시에 인식하는 새로운 형태의 인터페이스 패러다임을 제시하였다.

모달리티(modality)란 사람과 시스템 사이의 의사 전달을 위한 채널로, 의사를 다른 개체에 전달할 때 사용할 수 있는 출력 모달리티와 다른 개체의 의사가 전달되는 입력 모달리티로 구분할 수 있다. 입력 모달리티는 사람의 경우 음성, 제스처, 시선, 머리 및 몸의 움직임 등을 들 수 있으며 출력 모달리티는 말소리, 제스처를 들 수 있다. 컴퓨터나 휴대 단말의 경우에는 키보드, 마우스, 터치 스크린, 마이크(음성), 카메라(비전)는 입력 모달리티에 해당하며, 디스플레이, 스피커, 햅틱(Haptic)장비 등이 출력 모달리티라 할 수 있다[3].

사용자는 다양한 입력 채널을 통해 두 가지 이상의 모달리티를 동시에 사용할 수 있으며, 시스템은 여러 모달리티로부터 입력된 정보를 결합하여 사

용자의 의도를 파악하며, 그 결과를 음성, 영상 출력과 같은 출력 모달리티를 통해 제시하는 상호 작용이 이뤄진다.

2.1.1 기존 연구

멀티모달 인터페이스에 대한 연구로 사용자가 다양한 상황에서 사용 가능한 여러 종류의 모달리티를 제공하고 단일 모달리티만을 사용한 경우와 비교하는 연구가 진행되고 있다.

COHEN[3]은 지도 기반의 환경에서 군사 계획을 가상으로 수행하는데 있어 직접 조작을 통한 그래픽 사용자 인터페이스(Graphic User Interface)인 ExInit과 멀티모달 인터페이스인 Quick set을 비교 연구하였다. ExInit은 마우스와 키보드를 사용하며 Quick set은 음성과 스타일러스 펜을 사용하여 지도상에서의 부대나 제어 수단을 생성하고 위치시키는데 소요되는 시간을 분석하였다. 멀티모달 상호 작용을 수행한 Quick set의 경우는 GUI보다 수행 속도에 있어서 3.5배 정도의 증가를 보였으며, 에러를 처리하는 시간의 경우도 4.3배 정도 빠르다 라는 사실을 발견하였다. 이것은 현재의 비전이나 음성과 같은 인식 기반 기술만으로는 인식률이 높지 않기 때문에 여러 모달리티의 인지 결과를 통합함으로 의미 결합을 통해 모호성을 해결하며, 음성과 펜이 상호 보완적으로 사용자의 의도를 기술하게 하여 효율적이라는 결과를 보였다.

또한 멀티모달(음성/펜)을 이용한 상호작용과 음성만을 이용한 상호작용을 비교한 연구를 보면, 사용자부터 인한 오류율 및 모듈 오인식률는 35%와

30%로 감소되었으며[4][5][6], 사용자는 자신이 처한 상황에 따라 사용 가능한 모달리티를 선택하여 사용자에게 자유로운 의사소통이 가능케 함으로 사용자의 선호도가 매우 높은 것으로 나타났다[7].

멀티모달 인터페이스의 여러 가지의 이점이 부각되어 차세대 인터페이스로서 관심이 높아지면서 모바일 디바이스에서는 여러 모달리티를 활용한 서비스에 대한 연구가 진행되고 있다. SpeechPad는 음성과 키패드를 통하여 음성-문자간 변환이 가능케 하여 문자 입력을 지원하며, SmartKom Mobile은 멀티모달 대화 시스템 기반으로 음성과 펜 제스처를 이용하여 지도 검색 및 위치 인식과 같은 지도 관련 서비스를 가능케 하여 쉽고 편리한 인터페이스를 제공하고 있다.

W3C(World Wide Web Consortium)의 멀티모달 인터랙션 워킹 그룹에서는 인터넷 상에서의 멀티모달 상호작용 통한 웹 기반의 서비스를 가능하도록 EMMA, Ink Markup language[7] 등과 같은 표준안을 개발하고 있다. 본 논문에서 참조한 EMMA에 대해 2.2절에서 자세히 기술하도록 한다.

2.1.2 로봇 인터페이스

로봇 인터페이스에서는 로봇의 기계적인 구동에 초점에서 다양한 모달리티의 사용을 지원하여 로봇과 사람간의 상호작용이 사람과 사람간의 의사소통과 같은 자연스러운 멀티모달 인터페이스를 위한 연구가 진행되고 있다[8][9][10][11].

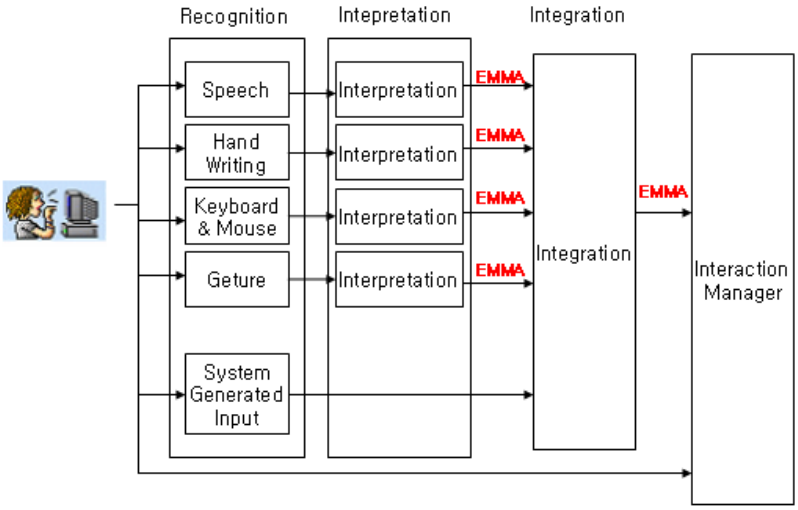
이와 같은 멀티모달 인터페이스 개발하는데 있어서, 개별 모달리티의 인지 결과를 표현하기 위해 기존의 로봇 인터페이스에서는 개발자가 마련한 API 수준에서의 인지 결과 표현 및 통합되어 개별 모달리티의 API가 변경되거나 다른 모듈로 대체되면 로봇 시스템 전체의 재구성이 필요하게 된다.

따라서 로봇과 사람 사이의 멀티모달 상호작용을 위하여 개별 모달리티와 멀티모달 통합 모듈 및 로봇 구동 모듈의 상호 독립성을 보장하여 모달리티의 확장 및 개선이 해당 모달리티나 모듈에 국한되도록 하는 방법론이 필요하며, 그 방법론으로 본 논문에서는 EMMA를 로봇 영역에 맞게 확장한 ROMMA를 설계하였다.

2.2 EMMA

EMMA(Extensible MultiModal Annotation markup language)는 사용자로부터 음성, 필기체 및 키보드 등의 다양한 입력을 인지하여 의미를 해석하여 시스템 간에 사용될 수 있는 입력 표준 언어이며, W3C의 멀티모달 인터랙션 그룹에서 2005년 9월에 초안(Working Draft)을 발표하였다.

멀티모달 입력 컴포넌트는 [그림 II-1]과 같이 인식, 해석 및 통합 모듈로 구성된다. 인식 모듈에서는 사용자로부터 신호를 인지하여 텍스트로 표현하고 그것은 해석 모듈을 통해 의미 있게 해석하여 그 결과를 EMMA로 변환한다. 통합(Integration) 모듈에서 여러 종류의 입력을 통합한 결과를 EMMA로 표현하여 인터랙션 매니저(Interaction Manager)로 전송한다.



[그림 II-1] 멀티모달 입력 컴포넌트

EMMA는 특정 어플리케이션에서의 사용자 입력/인지 해석 결과를 기술하기 위한 XML 형태의 마크업 언어이며 인식 결과에 대한 신뢰 값, 타임스탬프 및 입력 모달리티 매체 등과 같은 입력에 대한 메타 정보를 기술할 수 있다. [그림 II-2]에 나타난 EMMA 문서는 “origin”에 대한 음성 발화가 “Boston”과 “Austin”으로 두 가지의 경우로 해석되었으며 해석된 각 결과에 대해 신뢰 값을 기술하였다.

```
<emma:emma version="1.0" xmlns:emma ="http://www.w3.org/2003/04/ ">
 <emma:one-of id="r1" emma :start="1087995961542" emma :end="1087995963542">
   <emma :interpretation id="int1" emma :confidence="0.75">
     <origin>Boston</origin>
     <destination>Denver</destination>
     <date>03112003</date>
   </ emma : interpretation>
   <emma : interpretation id="int2" emma :confidence="0.68">
         <origin>Austin</origin>
         <destination>Denver</destination>
         <date>03112003</date>
   </emma : interpretation>
  </emma : one-of>
</emma :emma >
```

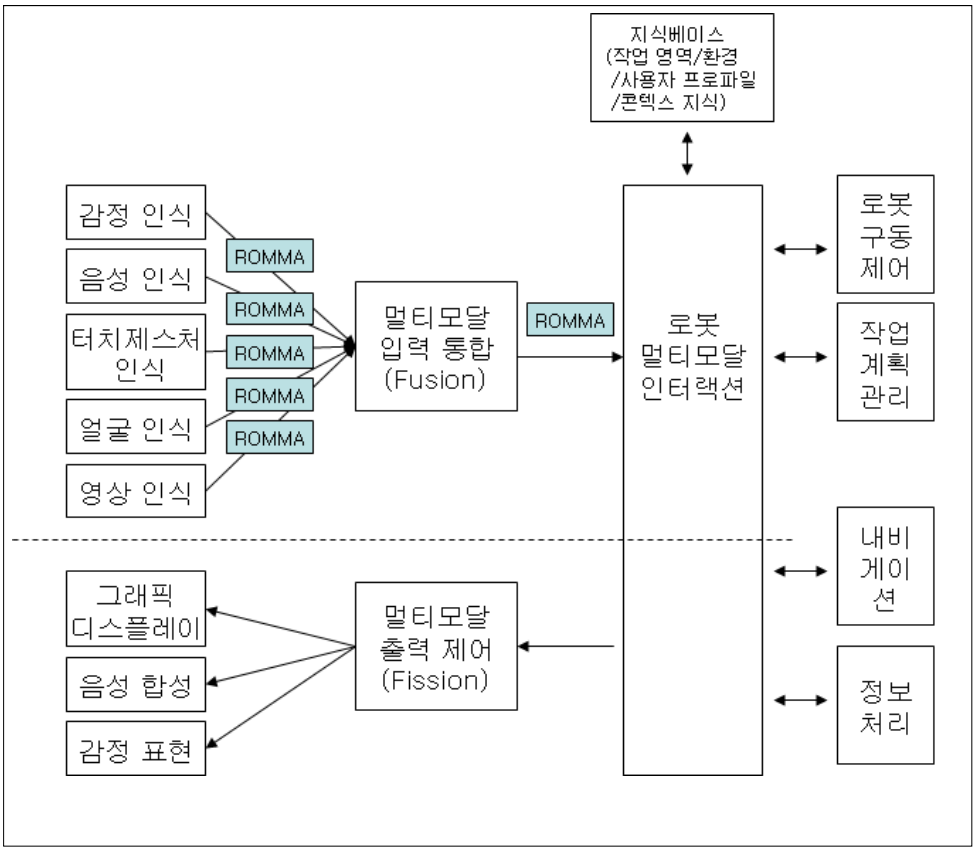
[그림 II-2] EMMA를 정의한 예시

III. ROMMA 소개 및 구조

3.1 소개 및 개념

사람과 로봇간의 상호작용이 이뤄지는 시스템의 일반적인 구조는 [그림 III-1]과 같다. 사람은 감정, 음성, 터치제스처, 얼굴 모습, 영상과 같은 멀티모달리티를 사용하여 로봇에게 명령을 내린다. 해당 인식 모듈에서 각 모달리티의 입력을 인지하고, 그 결과를 멀티모달 입력 통합 모듈에서 통합하여 로봇 멀티모달 인터랙션 모듈로 전송한다. 인터랙션 모듈에서는 작업 영역, 환경, 사용자 프로파일, 콘텍스와 같은 지식 베이스를 적용하여 사용자의 의도를 파악하여 로봇에게 주어진 임무를 수행할 수 있도록 로봇 자체를 움직이거나 일을 실행하는 등, 실제 응용 서비스를 제공하기 위한 관련 모듈에 명령을 내린다. 또한 일을 하는 과정 중에 또는 종료 시에 사용자와 의사전달을 위해 인터랙션 모듈에서는 멀티모달 출력 제어 모듈을 통해 음성, 감정, 터치제스처로 표현하여 사용자에게 그 결과를 전달한다.

기존의 로봇 인터페이스에서는 각 인식 모듈의 인지 결과 표현이 상이하기 때문에 인식 모듈마다 독립적인 인터페이스를 가져야 하므로, 본 논문에서는 입력 모듈에서의 통일된 방식을 위한 ROMMA를 제시하였다.



[그림 III-1] 멀티모달 로봇 상호작용 시스템 구조

ROMMA(RObot MultiModal Annotation markup language)는 W3C(World Wide Web Consortium)의 멀티모달 인터랙션 그룹에서 웹 인터페이스를 위한 멀티모달 입력정보 기술언어의 표준안으로 제정된 EMMA를 기본으로 인간-로봇 상호작용을 위한 로봇 인터페이스에 맞게 재구성한 것이다. 로봇은 감정, 음성 또는 터치제스처 등과 같은 사용자의 입력을 해

당 인식 모듈로부터 인지하면 인지된 정보를 의미 있게 해석하여 ROMMA로 기술한다. 그러면 멀티모달 입력 통합 모듈을 통해 관련 있는 여러 입력 정보를 결합하여 그 결과 또한 ROMMA 문서로 기술하여 로봇 멀티모달 인터랙션 모듈로 전달한다.

ROMMA는 로봇의 입력 컴포넌트에서 그 결과를 표현하기 위한 멀티모달 입력 정보 기술언어이다. 로봇의 다양한 모달리티를 수용할 수 있으며, 각 모달리티의 인지 결과의 표현 양식을 통일하고, 로봇 인터랙션 모듈이나 통합 모듈 사이의 상호 데이터 교환에 대한 통일된 양식을 제공하여 다음을 가능케 한다.

첫째, 멀티모달 통합 방법의 연구의 편의를 도모한다. 즉, 개별 모달리티와 멀티모달 결합 모듈이 소결합됨으로써 개별 모달리티 관련 기술의 완성과 관련 없이 다양한 모달리티 결합에 관한 연구가 가능하다.

둘째, 개별 모달리티 관련 기술 개발이 다른 모달리티 기술 개발 및 통합 기술 개발과 모두 독립적으로 이루어 질 수 있도록 한다. 이러한 로봇과 사람과의 상호작용이 이뤄지는 각 모듈 사이의 독립성 확보는 각 모듈 개발의 용이성뿐 아니라 교체의 편리성을 제공한다.

셋째, XML을 기반으로 하여, 기존 구조의 재사용과 새로운 모달리티의 추가나 작업 영역의 추가에 대비한 확장성과 하향 호환성을 지원한다.

넷째, 의미 해석에서 필요한 작업 영역 및 환경 관련 지식 정보의 활용이 용이하다.

3.2 설계 고려 사항

ROMMA의 설계에 있어서 고려한 사항은 다음과 같다.

첫째, 단순하면서도 다양한 모달리티의 인지 결과를 명시할 수 있는 구조를 가져야 한다.

둘째, 예상되는 로봇의 모달리티를 모두 수용할 수 있도록 설계하고, 추후 예상되는 새로운 로봇 모달리티를 수정 없이 수용할 수 있는 확장성을 가져야 한다.

셋째, 각 입력 모달리티의 하위 인지결과뿐 아니라 상위 인지 결과 역시 기술할 수 있는 구조를 가져야 한다. 하위 수준의 입력 정보의 표현 (x-y 좌표, 음성 신호), 하위 수준의 입력정보에 대한 인식 결과 (특정 의미를 가진 아이콘이나 버튼, 음성 인식 단어), 인식 결과에 대한 의미 해석 (TV 프로그램 예약 등 작업 영역 및 상황 기반 인지 결과)의 모든 과정에서 사용할 수 있도록 설계해야 한다.

넷째, 작업 영역의 수행 상황이나 환경 정보를 이용한 의미기반 해석이 가능하도록 다양한 작업 영역 및 환경 정보 구조를 수용할 수 있어야 한다.

3.3. 기본 구조

ROMMA는 새로운 입력 모달리티의 추가 용이 및 확장 가능한 XML 기반의 언어로서 멀티모달 처리를 위한 다수의 인지 결과의 해석을 위하여 각 모듈의 개별 인식 결과를 기술하며 모듈간의 데이터 교환의 단위로서 하나의 XML 문서로 표현된다.

ROMMA의 구성요소는 크게 기본 엘리먼트와 확장 엘리먼트로 구성된다. 기본 엘리먼트는 ROMMA의 구조 및 모든 모달리티와 작업 영역에서 공통으로 사용되는 기본 구조를 기술하는 엘리먼트이며, 확장 엘리먼트는 각 개별 모달리티로나 작업 영역에서의 인지결과 해석 및 통합 결과를 담기 위한 특수한 엘리먼트이다.

확장 엘리먼트는 지식 베이스의 작업 영역 지식(ontology)과 연계되어 실시간으로 변화하는 환경을 기술할 수 있으며, 로봇이 수용 가능한 모달리티 또는 작업 영역에 맞게 확장 정의가 가능하다.

ROMMA 문서는 XML 문서가 가져야 하는 기본 구조와 구문 규칙을 따르며 문서의 시작은 루트 엘리먼트인 <romma>로 시작하고 </romma>로 끝난다.

3.1.1 ROMMA 기본 엘리먼트

<romma>는 ROMMA의 버전 정보를 기술하며 하위 구조로 다음의 4가지 기본 엘리먼트를 포함하며, XML의 DTD 정의 방법에 따라 표현하면 다음과 같다.

romma (interpretation | choice | seq | group)+

interpretation, choice, seq, group은 ROMMA의 기본 엘리먼트 단위로, 모달리티의 입력 정보 또는 멀티모달 통합 결과를 표현하며 인식 시스템에 따라 한번 이상의 반복된 형태로 정의 가능하며, interpretation을 제외한 나머지 기본 엘리먼트는 상호 포함적(중첩) 관계의 형태로 나타날 수 있다.

ROMMA의 4 가지 기본 엘리먼트에 대한 정의 내역은 다음과 같다.

§ <interpretation> : 각 모듈의 개별 인식 결과 또는 통합 결과인 하나의 입력/인지 결과를 기술하기 위한 엘리먼트이며, 각 interpretation은 확장 엘리먼트를 하위 구조로 갖는다. 확장 엘리먼트는 모달리티별로 그 특성을 반영하여 정의할 수 있으며, 로봇의 작업 영역, 상황 및 환경 지식을 반영하는 엘리먼트를 필요에 따라 확장 정의하여 사용할 수 있는 엘리먼트이다. [그림 III-2]는"이거 좀 여기로 갖다놔"라고 음성과 함께, 두 개의 포인팅 제스처가 인지된 결과를 기본 엘리먼트와 확장 엘리먼트로 정의하여 표현하였다.

```
<romma ver="0.5">
  <group id="s1" start="1001000" end="10011000">
   <interpretation id="a1" function="spoken command" tokens="이거 좀 여기로 갖다놔">
       <request>
          <action id="5" actionType="put">갖다놔</action>
          <object id="1010" type="deictic" value="this">이거</object>
          <location type="deictic" value="here">여기로</location>
       </request>
    </interpretation>

    <seq id="s1" mode="pointer" function="guesture">
       <interpretation id="a2">
          <point startTime="1001010" endTime="1001020">220,165</point>
       </interpretation>
       <interpretation id="a3">
          <point startTime="1001035" endTime="1001040">25,155</point>
       </interpretation>
    </seq>
  </group>
</romma>
```

[그림 III-2] ROMMA의 기본 엘리먼트 및 확장 엘리먼트

§ <choice> : 단일 입력에 대한 다수의 가능한 해석 결과 표현하기 위한 엘리먼트이며, 두 개 이상의 상호 배타적인 <interpretation>을 하위 구조로 가진다. 또한 인식 결과에 대해 플랫폼마다 제시한 판단 기준에 따라 best-first순으로 문서상에서 <interpretation>을 정의한다. [그림 III-3]에 정의된 <choice>는 id가 "a1"과 "a2"인 2 개의 <interpretation>을 가지며, 입력/인지 결과가 이들 중에서 하나라는 뜻이다.

§ <seq>: 시간적으로 연속된 다수의 입력에 대한 해석 결과를 표현하는 엘리먼트이다. [그림 III-2]에서 <seq>의 하위에 나타난 id가"a2"와 "a3"인 <interpretation>는 사용자로부터 입력 받은 터치 제스처의 좌표를 나타내며 입력받은 순서대로 문서상에 정의된다.

§ <group> : 다수의 입력 모달리티의 결과를 동시에 기술하기 위한 구조이다. [그림 III-2]는 음성 인식 결과인 <interpretation id="a1"..>과 포인팅 제스처의 결과인 <seq id="s1"..>이 동시에 입력되어 특정 시간 내에 관련된 입력들을 그룹화한 예이다. 일반적으로 이와 같이 정의된 <group>은 통합 모듈을 거치면 하나의 <interpretation>으로 변환된다.

```
<romma ver="0.5">
  <choice  id="s1" start="1001000" end="1001100"   medium="acoustic"
                    device="microphone"  function="spoken command">
    <interpretation id="a1">
        <request>
            <action id="27" actionType="get">가져와</action>
            <object id="1010" type="beverage" value="coke">콜라</object>
        </request>
    </interpretation>

    <interpretation id="a2">
        <request>
            <action id="27" actionType="get">가져와</action>
            <object id="1010" type="movable" value="cup">컵</object>
        </request>
    </interpretation>
  </choice>
</romma>
```

[그림 III-3] ROMMA의 기본 엘리먼트 <choice>

3.3.2 ROMMA 기본 에트리뷰트

ROMMA의 기본 엘리먼트에 정의할 수 있는 기본 에트리뷰트는 다음과 같으며 세부적인 특성 값을 정의한다.

§ id : ROMMA 문서 내의 기본 엘리먼트를 위한 식별자로 필수 기본 에트리뷰트이며. 추후 다른 엘리먼트에서 참조를 하기 위한 것이다.

§ start, end : 입력/인지의 시작 시각과 완료 시각을 시스템상의 절대 시간으로 정의하며 단위는 밀리세컨드(millisecond)이다. [그림 III-3]의 <choice>는 시스템 시간을 1001000 밀리세컨드에서 시작하여 1001100 밀리세컨드 사이의 결과임을 나타낸다.

§ grammar-ref : 음성 인식 관련 인지 모듈에서 사용을 위한 인식 문법과 같은 부가 정보를 표시한다.

§ tokens : 의미 해석을 수행하기 전의 입력을 그대로 전달하기 위한 값으로 [그림 III-2]를 참고하면 tokens는 사용자가"이거 여기로 갖다놔"라고 발화한 일련의 입력 값이다.

§ signal : 음성 또는 비디오와 같이 기본 입력이 문자로 표현되지 않는 신호 스트림일 경우, 멀티모달 결합 등을 위하여 이러한 스트림이 필요할 경우가 있다. 이러한 기본 신호 입력을 파일로 만들어 전달하기 위한 값이며, 파일 이름을 지정한다.

§ verbal : 문자로 표현될 수 있는 가능 여부를 나타내는 것으로 가능한 경우는 true, 그렇지 않은 경우는 false이며 기본 값은 true이다.

§ medium, device, mode, function : 기본 엘리먼트가 나타내는 정보의 입력 모달리티가 무엇인지를 식별하기 위한 값이다. 현재 고려하고 있는 모달리티는 [표 III-1]과 같다.

- medium 및 device : medium은 사용자가 이용한 수단을 나타내는 개념으로 음성(voice), 터치(tactile), 시각(visual)으로 구분되며 device는 각 모달리티의 입력 시에 사용된 장치를 나타낸다. medium의 터치는 펜, 마우스, 키보드와 터치 스크린 등과 같은 장치를 통한 입력을 의미하며 현재 터치 스크린을 통한 터치의 경우만을 고려한다.
- mode : 각 medium 내에서의 지정한 장치를 이용한 의사소통의 여러 방법을 구분하여 정의한 특성 값이다.

Medium	Device	Mode	Function
Acoustic	Microphone	Speech	Sound stream
			Spoken command
			Dictation
			Speaker recognition
			Emotion
Tactile	Touch screen	Pointer	Gesture
			Hand-written command
		GUI	Icon
			Button
Visual	Video camera	Video	Movie
			Gesture
			Face recognition
			Audio-visual recognition
			Emotion

[표 IV-3] <modifier> 엘리먼트의 type 및 value

IV. 영역별 ROMMA 확장

ROMMA의 확장 엘리먼트는 앞서 제시한 [그림 III-2], [그림 III-3]에서와 같이 기본 엘리먼트의 하위 구조로 나타난다. 확장 엘리먼트는 현재 로봇의 수행 가능한 작업 종류에 따라 각 작업 영역에 적합한 정보 구조를 XML 엘리먼트로 확장 정의하여 사용할 수 있으며, 또한 로봇이 가지고 있는 모달리티에 따라 필요한 엘리먼트를 정의하여 사용할 수 있다.

본 논문에서는 확장 엘리먼트가 로봇의 작업 영역과 모달리티의 확장에서 어떻게 정의될 수 있는 지를 보여 주기 위하여, 작업 영역과 관련하여 심부름 영역을, 모달리티와 관련하여 감정 모달리티에 대하여 정의한 확장 엘리먼트를 설명한다. 다른 작업영역과 모달리티의 확장에서도 유사한 방법으로 쉽게 확장 엘리먼트를 정의하여 사용할 수 있다.

4.1 심부름 영역 확장 엘리먼트

작업 영역과 관련하여 확장 정의된 심부름 영역은 집안에서 로봇에게 심부름을 명령할 때의 발화 형태에 따라 다음의 <request>, <question>, <answer>, <information> 4가지 유형으로 심부름 영역 확장 엘리먼트를 분류하였다. 각 확장 엘리먼트는 다시 실제 사용자로부터 어떤 내용이 입력되었는지 정보를 상세히 기술하기 위한 확장-하위 엘리먼트로 구성된다.

심부름 영역 확장 엘리먼트에 대한 정의는 다음과 같다.

§ <request> : 로봇에게 지시하여 로봇이 물리적인 행위를 취하도록 요구하는 발화를 나타내며 로봇에게 요구하는 행위를 get, put, open, close, on, off, go, stop, turn, resume로 분류한다. [그림 IV-1]은"냉장고 안에 빨간 콜라 빨리 여기 옆에 가져와"라고 발화함으로서 사용자가 요청한 객체를 지시한 위치에 가져오도록 하는 행위를 요구한 경우로 request를 상위 엘리먼트로 정의하였다. 다음과 같은 확장-하위 엘리먼트로 구성되며 각 확장-하위 엘리먼트에 대해서는 4.1.1절에서 상세히 기술한다.

<action>, <object>, <location>(<source>,<destination>),<modifier>

<action>은 필수 요구 사항이며 <modifier>는 <action>, <object> 및 위치 관련 엘리먼트인 <location>,<source>,<destination>에 대해 상세적인 의미를 나타내며 각각에 대해 한 번 이상 정의할 수 있다.

```
<request>
  <action id="2" actionType="get">가져와</action>
  <modifier targetClass="action" targetId="2" type="speed" value="fast">빨리
   </modifier>
  <object id="1010" type="beverage" value="coke">콜라</object>
  <modifier targetClass="object" targetId="1010" type="color" value="red">빨간
   </modifier>
  <source id="2001">냉장고</source>
  <modifier targetClass="source" targetId="2001" value="in">안</modifier>
  <destination id="2005" type="deictic“ value="here">여기</destination>
  <modifier targetClass="destination" targetId="2005" value="side">앞</modifier>
</request>
```

[그림 IV-1] 심부름 영역 확장 엘리먼트 <request>

§ <question> : 현재 로봇이 알고 있는 사실에 대해 질의하는 질문하는 형태의 발화에 대해 정의한다. 기본 에트리뷰트인 type은 yes_no, what, where 3가지 형태를 가지며 예문은 다음과 같다.

yes_no :‘예’,‘아니오’로 대답할 수 있는 형태의 질문 형

예) 주스 있니?, 다른 거는 없니?, 콜라 말고 커피는 있니?

what :‘무엇’에 해당되는 의문을 갖는 형태의 질문 형

예) 어떤 음료수가 있지?, 이거 뭐지?

where : 위치를 묻는 형태의 질문 형

예) 콜라 어디 있지?, 어디에 불이 켜져 있니?

하위 구조는 다음과 같은 확장-하위 엘리먼트로 구성되며 predicate를 제외

한 확장-하위 엘리먼트의 구성은request와 동일하다.

<predicate>, <object>, <location>, <modifier>

[그림 IV-2]는 사용자가“테이블 위에 주스 있니?”와 같이 “yes-no”의 type의 형태로 발화한 경우를 정의한 것이다.

```
<question type="yes-no">
  <predicate  id="1" predType="is_there">있니</predicate>
  <object id="1010" type="beverage" value="juice">주스</object>
  <location id="3" type="positional" value="table">테이블</location>
  <modifier targetClass="location" targetId="3" value="on">위</modifier>
</question>
```

[그림 IV-2] 심부름 영역 확장 엘리먼트 <question>

§ <answer> :‘예’나 ‘아니요’로 응답한 평문 형태의 발화를 정의한다. value라는 에트리뷰트를 통해 yes인지 no인지를 나타낸다.

§ <information> : yes-no형태의 질의를 제외한 사람의 상태나 오브젝트의 위치 등에 대한 정보를 요구하는 질의 형태의 발화를 나타낸다.

하위 구조로 정의할 수 있는 확장-하위 엘리먼트는 question 엘리먼트의 경우와 동일하다.

4.1.1 심부름 영역 확장-하위 엘리먼트

심부름 영역 확장 엘리먼트의 하위 구조로 나타나는 확장-하위 엘리먼트

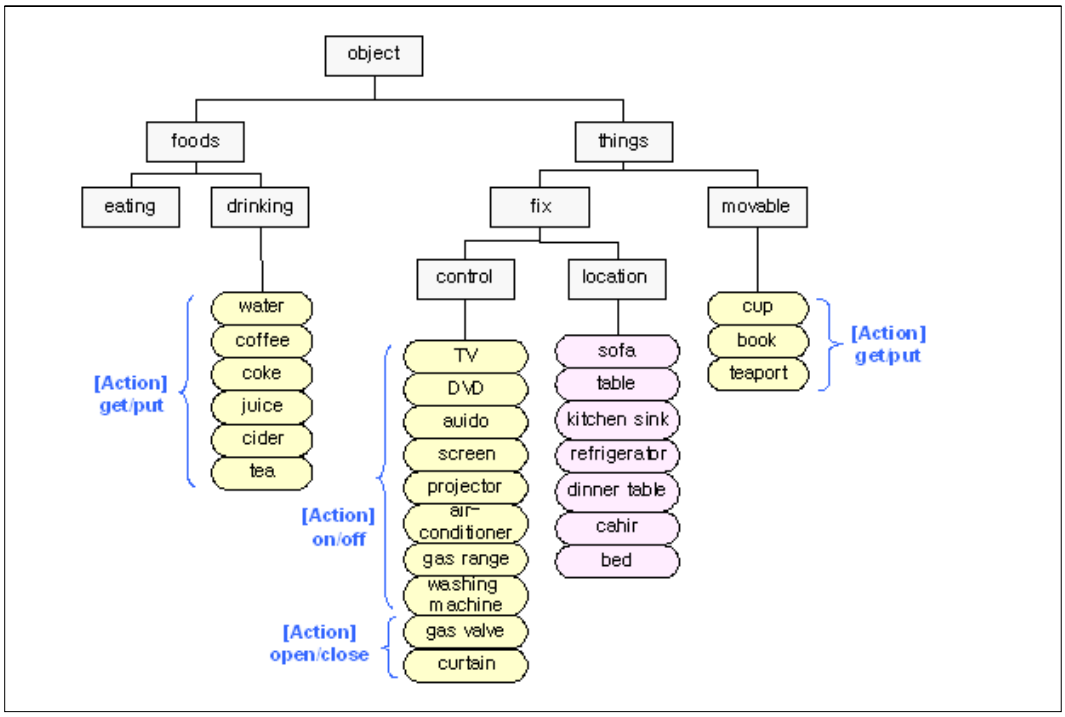
는 다음과 같다.

§ <object> : 행위의 목적이 되는 대상을 의미하며, 음성을 사용한 경우의 컨텐츠는 실제 오브젝트명이 되며 그 외 모달리티의 경우는 충분한 의미가 해석된 상태라면 컨텐츠는 생략이 가능하다.

현재의 심부름 영역에서 고려하는 오브젝트는 데모 공간에서 제안된 몇 가지로 경우로 오브젝트를 결정지어, 특성에 따라 [그림 Ⅳ-3]과 같이 오브젝트의 작업 영역 지식의 계층화(ontology hierarchy)를 도식화하였다. 각 개념의 범주에 포함되는 오브젝트를 각 개념의 하단에 나열하고, 오브젝트에 대해 로봇이 취할 수 있는 행동의 여부 및 종류를 나타냈다.

오브젝트는 가장 상위 개념으로서‘식품’(food)과‘물건’(thing)으로 구분된다.‘식품’은 다시‘먹는 것’(eating)과‘마시는 것’(drinking)으로 개념을 구분하며 실제 오브젝트인 coke, water, coffee, juice, cider, tea가 마시는 것에 포함된다.

‘물건’은 동적인 관점에서‘고정’(fix)와‘이동’(movable)으로 구분하며‘고정’은 다시 로봇의 행동을 통해 제어가 가능한 경우인‘제어적’(control)과 다만 위치적인 개념을 갖고 있는 오브젝트를 나타내는‘위치적’(location)인 개념으로 구분된다.



[그림 IV-3] <object> 엘리먼트의 작업 영역 지식 계층화

<object> 엘리먼트는 다음과 같은 에트리뷰트를 포함한다.

- startTime, endTime : 해당 오브젝트명를 발화 또는 포인팅한 시점을 나타내는 값으로 startTime은 시작 시간, endTime는 완료 시간을 시스템상의 절대적 시간으로 기술한다.
- id : ROMMA 문서 내에서 각 엘리먼트의 식별자로서의 값을 나타낸다.

- realId : 현 로봇 상황에서의 작업 영역 내에서 정의된 오브젝트의 id를 의미한다.
- type, value : 오브젝트의 작업 영역에서의 명칭 및 지시어를 사용하여 발화한 경우를 [표 IV-1]과 같이 분류하여 type을 지정하였다. value는 각 type에 해당하는 오브젝트의 작업 영역 내에서의 명칭이며 일종의 canonical value로서 사용자가'코크', '콜라' 등과 같이 'coke'를 지칭하는 어떠한 오브젝트 명으로 발화하더라도 value의 값은 모두'coke'로 표시된다.

type	value
beverage	coke, water, coffee, juice, cider, tea
deictic	this
controllable	tv, dvd, audio, screen, projector, airconditioner, gasrange, washingmachine, gasvalve, curtain
positional	sofa, table, sink, refrigerator, dinnertable, chair, bed
movable	cup, book, teaport

[표 IV-1] <object>엘리먼트의 type 및 value

§ <source>, <destination>, <location> : 목적물의 위치와 관련하여 행위가 발생하는 공간상에서의 위치를 나타내며 위치의 출처나 목적지가 명확한

경우는<source>나 <destination>으로, 그렇지 않은 경우는 <location>으로 위치를 정의한다.

컨텐츠는 상위 개념인 거실(living), 안방(mainroom), 부엌(kitchen)이거나 <object>중에서 위치가 반영구 또는 영구적으로 고정되어 있는 오브젝트명이 된다. 즉, 오브젝트의 이동 가능 여부에 따라 이동이 절대 불가능한 경우의 오브젝트를 '영구적', 이동은 가능하나 로봇이 옮기기 힘든 오브젝트를'반영구적'이라 하며 <object>의 작업 영역 지식의 계층화에서 "controllable"과"positional"개념에 속한 오브젝트가 이에 해당된다. [그림 IV-1]을 보면 사용자가 "냉장고 안에 빨간 콜라 빨리 여기 옆에 가져와" 라고 발화한 경우로, "냉장고"는 대상이 되는 "콜라"의 위치를 나타내므로 source로 정의하며 "여기"라고 지시한 위치는 "콜라"에 대한 "가져와"라는 행위의 목적지를 나타내므로 destination으로 정의하였다.

위치 관련 확장-하위 엘리먼트에 정의 가능한 에트리뷰트는 다음과 같다.

- startTime, endTime, id, realId : 위에서 언급한 <object>의 각 에트리뷰트 내역과 동일하다.
- type, value : [표 IV-2]를 참고하면 상위 개념인 room과 지시어를 사용하여 위치를 지칭하는 경우로 type을 정의하며, value는 <object>에서 정의된 value의 내역과 동일하다. 또한 컨텐츠에서 언급한 사항과 같이, object에서 정의한 "controllable"과 "positional"의 type 및 그에 속한 value는 object이면서 동시에 location에도 해당된다.

[그림 IV-1]을 참고하면 사용자는 위치에 해당하는 오브젝트명을 발화하는 대신에 “여기”라는 지시어를 발화하면서 그에 해당하는 위치를 가리킨 경우이다. 이러한 경우, destination의 type은 “deictic”이 되며 value는 “here”, 컨텐츠는 “여기”로 정의되었다.

type	value
room	living, mainromm, kitchen
deictic	here

[표 IV-2] <location>엘리먼트의 type 및 value

§ <action> : 주어진 작업을 수행하기 위해 로봇에게 지시된 행위를 나타내며 컨텐츠는 발화 문장 내에서 품사가 술어동사가 이에 해당된다. 예를 들어 [그림 IV-1]을 참고하면 “냉장고 안에 빨간 콜라 빨리 여기 옆에 가져와”라고 발화한 경우로 “가져와”가 action의 컨텐츠에 해당된다.

action에 정의 가능한 에트리뷰트는 다음과 같다.

- actionType : 로봇이 취할 수 있는 가능한 행위의 종류를 나타내며 현 상황에서 고려된 행위로는 “get”(가져오기), “put”(갖다 놓기), “open”(열기), “close”(닫기), “on”(끄기), “off”(켜기), “go”(진행하기), “stop”(중단하기), “turn”, “resume”(재개하기)가 있으며 향후 포괄적인 작업 수행을 위한 다양한 종류의 행위가 정의되어야 한다.

- startTime, endTime, id : <object>의 각 에트리뷰트 내역과 동일하다.

§ <modifier> : 행위, 행위의 대상, 위치와 관련하여 각 개별의 의미를 구체화시키는 역할을 하며 이에 정의 가능한 에트리뷰트는 다음과 같다.

- targetClass : 의미적으로 어떤 엘리먼트에 대한 수식어인지 적용될 대상이 되는 엘리먼트를 나타내는 속성 값으로서 <action>, <object>, <location>(<source>, <destination>) 이들 중에서 하나를 나타낸다.
- targetId : targetClass에서 참조한 그 엘리먼트의 id 값을 갖는다.
- type, value : 각 엘리먼트의 수식어에 해당되는 부분으로 [표 IV-3]을 참고하면 엘리먼트에 적용할 수 있는 속성을 분류하여 type을 정의하며 value는 각 type에서 정의할 수 있는 값이다. object의 type이 name인 경우의 value는 오브젝트의 실제 이름, 즉 제조명이 된다.
- startTime, endTime, id : <object>의 각 에트리뷰트 내역과 동일하다.

Element	type	value
object	name	*Each proper noun*
	color	red, yellow, green, blue. ..
	temperature	hot, cold, lukewarm
location	position	on, in, bottom, side
action	speed	fast, slow
	degree	much, little
	volumn	up, down

[표 IV-3] <modifier> 엘리먼트의 type 및 value

§ <predicate> : <question>과 <information>의 발화 문장에서 술어에 해당하는 부분으로 다음의 에트리뷰트를 기술한다.

- predType : 발화 종류를 대표하는 이름으로 상위 엘리먼트가 무엇인지에 따라 다르게 정의한다.

4.2 감정 영역 확장 엘리먼트

모달리티와 관련하여 정의된 감정 관련 확장 엘리먼트는 시각 및 음성 분석 기술을 통하여 사람의 얼굴 표정과 음성 속에 담긴 사용자의 감정 상태를 정의하여 표현하도록 한다. 이것은 감정 인식을 위한 모달리티로부터 입력된 정보를 융합하여 현재 사용자의 감정 상태를 유추하며, 확장 엘리먼트를 정의하여 4개의 기본적인 감정 즉, 평상심(Neutrality), 기쁨(Joy), 슬픔(Sad), 화남(Anger)으로 감정 상태를 표현한다.

감정 영역 확장 엘리먼트는 상위 엘리먼트로 <emotion>을 선언하며 하위 구조는 사용자의 감정 상태 표현을 위한 state_value 엘리먼트를 선언한다. state_value는 [그림 IV-4]를 보면 사용자의 감정에 따라 neutrality, joy, sad, anger로 정의한 엘리먼트로 confidence를 애트리뷰트를 선언하여 감정 상태의 신뢰도를 나타내며 0부터 1사이의 decimal의 값을 가진다.

이러한 특정 시점에서의 감정뿐만 아니라 <emotion>의 timing 애트리뷰트를 정의하여 감정의 전이 상태를 나타내도록 한다. [그림 IV-4]를 참고하면 음성을 통해 파악된 감정을 나타내며 timing 에트리뷰트의 before와 current를 통해 슬픔과 화남의 비율이 50:50이였던 이전의 감정이 현재는 화남으로 감정이 전이된 상태를 나타내고 있다.

```
<romma ver="0.5">
  <choice  id="c1" start="1001000" end="10011000" medium="acoustic"
                device="microphone"  function="emotion">
    <interpretation id="a1">
      <emotion timing="before">
                <anger confidence="0.5"/>
                <sadness confidence="0.5"/>
      </emotion>
    </interpretation>

    <interpretation id="a2">
      <emotion timing="before">
                <anger confidence="1.0"/>
      </emotion>
    </interpretation>
  </choice>
</romma>
```

[그림 IV-4] 감정 영역 확장 엘리먼트 <emotion>

V. ROMMA 기반 로봇 멀티모달 입력 통합

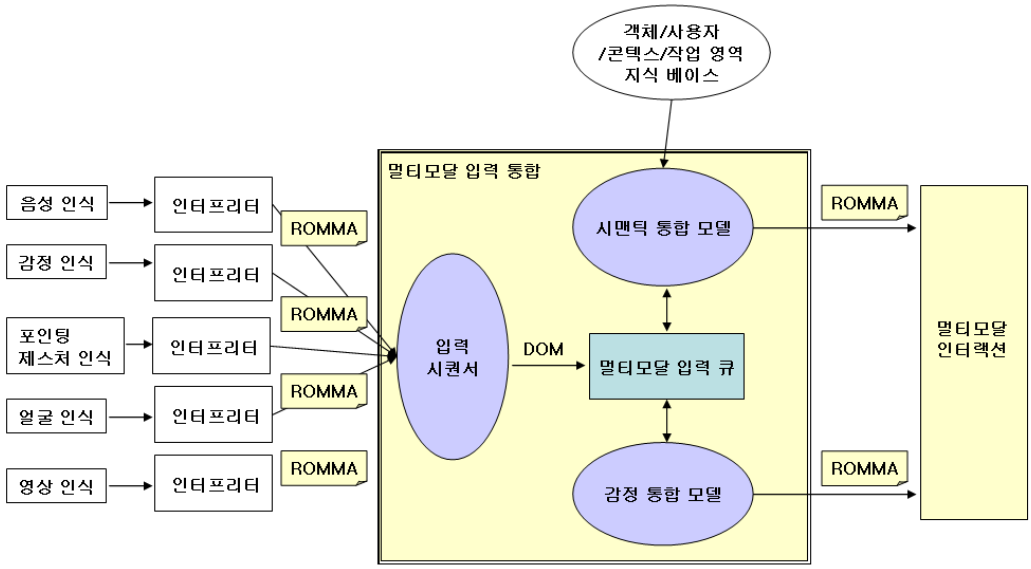
본 장에서는 다양한 상호작용 채널을 통해 들어오는 정보를 ROMMA 기반으로 통합하는 멀티모달 통합을 위한 멀티모달 입력 통합 모듈을 통해 사용자의 멀티모달 상호작용에서 다양한 로봇 인지 모달리티의 결과를 통합하는 과정을 설명한다.

5.1 멀티모달 입력 컴포넌트

멀티모달 통합을 위한 입력 컴포넌트의 구성은 사용자의 입력을 인지하는 인식 모듈과 그것을 의미 있게 해석하여 ROMMA로 기술을 위한 해석 모듈, 여러 입력을 통합하여 그 결과를 ROMMA로 기술하여 인터랙션 모듈로 전송하는 멀티모달 입력 통합 모듈로 구성되며 [그림 V-1]과 같다.

멀티모달 입력 통합 모듈은 다시 입력 시퀀서(Input Sequencer), 시맨틱 통합 모델(Semantic Integration Model), 감정 통합 모델(Emotional Integration Model)과 같은 구성 요소로 이뤄진다. 각 인식 모듈로 부터 입력받은 데이터는 해석 모듈을 거쳐 ROMMA로 작성되며 이 문서를 입력 시퀀서로 전송되면, 입력 시퀀서는 ROMMA 형태의 모달리티별 입력 결과

를 시간상 입력 순서에 따라 정렬하여 큐(queue)에 저장하도록 제어한다. 큐에 저장된 ROMMA는 시간 축 상에서의 선행관계에 따른 연관성을 분석하여 관련있는 ROMMA 문서들로 선택되고 이 문서들은 통합 모델을 통해 현재 사용자와 로봇이 처한 환경 정보를 가지고 의미적인 멀티모달 통합 과정을 거치게 된다.



[그림 V-1] 멀티모달 입력 컴포넌트 구성도

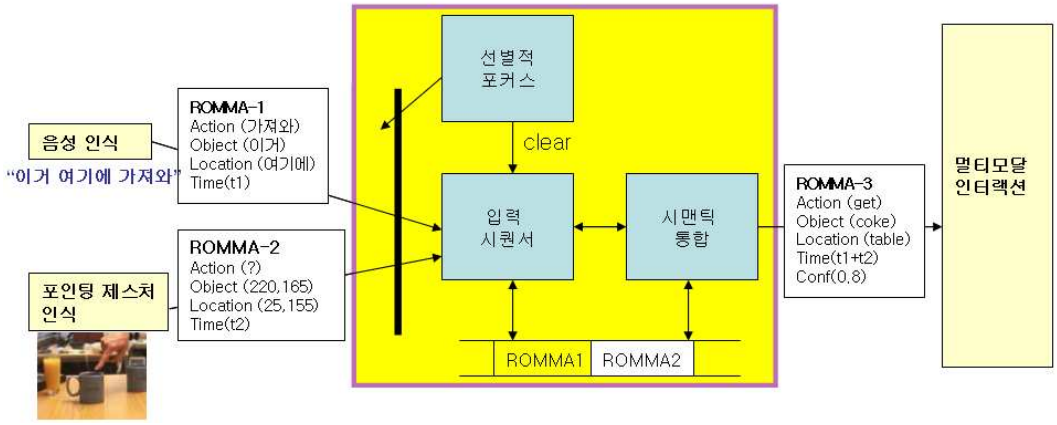
통합 모델은 크게 시맨틱 통합 모델과 감성 통합 모델로 구분되어 지며, 시맨틱 통합 모델은 문자화할 수 있는 정보를, 감정 통합 모델은 감정을 콘텍스트, 현재의 작업 영역, 사용자 및 객체 환경에 대한 지식 기반으로 의미적 멀티모달 통합을 진행하고 그 결과를 ROMMA 형태의 문서로 작성하여

멀티모달 인터랙션 모듈로 전송한다.

이와 같이 멀티모달 입력 통합을 통해 불완전하고 부분적인 정보의 결합에 필요한 의미해석을 부여하고 모달리티 입력의 결합을 통한 인식 성능 향상을 제고하도록 한다.

5.2 ROMMA 기반의 멀티모달 통합

음성과 터치제스처를 함께 사용하여 ROMMA로 표현한 멀티모달 통합의 경우를 예를 들어 설명하면 [그림 V-2]와 같다.



[그림 V-2] 멀티모달 통합 시나리오

로봇의 모달리티는 음성과 터치제스처이며 사용자는 “이거[터치제스처1] 여기에[터치제스처2] 갖다놔”와 같이 발화하면서 동시에“이거”와“여기에”에 해당하는 특정 객체를 포인팅하여 로봇에게 심부름 명령을 내린 경우이다.

사용자로부터의 음성과 터치 제스처는 각각의 인식 모듈에서 인지되고 각 인지 결과가 ROMMA로 표현되어 멀티모달 입력 통합 모듈로 전송되면, 멀

티모달 입력 통합 모듈에서는 사용자의 음성 발화 구간을 기준으로 연관되는 터치 제스처의 발생 시점을 분석하여 해당 ROMMA 문서들을 통합하게 된다.

[그림 V-3]은“이거 여기에 갖다놔”라고 사용자의 발화 내역을 기술한 것이며 [그림 V-4]는 발화와 동시에 매개 인터페이스로 부터 입력된 두 개의 포인팅 정보를 기술한 것으로, [그림 V-5]는 음성 발화 시작 시점과 종료 시점 사이에 터치제스처가 연관되어 발생된 것으로 보고 두 ROMMA 문서, [그림 V-3]과 [그림 V-4]를 통합한 결과이다.

```
<romma ver="0.5">
 <interpretation id="a1" start="1001001" end="1001045" function="spoken command">
        <request>
            <action id="5" actionType="put">가져와</action>
            <object id="1010" type="deictic" value="this">이거</object>
            <location type="deictic" value="here">여기로</location>
        </request>
 </interpretation>
</romma>
```

[그림 V-3] 음성 인식 결과

```
<romma ver="0.5">
 <seq id="s1" mode="pointer" function="gesture">
    <interpretation id="a2">
          <point startTime="1001010" endTime="1001020">220,165</point>
     </interpretation>
     <interpretation id="a3">
          <point startTime="1001035" endTime="1001040">25,155</point>
     </interpretation>
 </seq>
</romma>
```

[그림 V-4] 터치 제스처의 인식 결과

또한 음성 구간의 -2.3/+2.7(초)에 시작한 터치제스처는 해당 음성 발화 단어와 관련된 터치제스처로 간주하여 인식 단어가 "이거"와 같이 지시어인 경우에는 "이거"에 해당하여 인식된 터치 제스처의 결과로 대치하여 통합 결과를 기술한다[12].

[그림 V-3]에서 객체를 나타내는 "이거"에 해당되는 터치 제스처는 [그림 V-4]에서 interpretation의 id가 2인 경우의 좌표가 되며, 또한 위치를 나타내는 "여기로"에 해당되는 터치 제스처는 id가 3인 경우의 좌표가 이에 해당된다.

그 결과, 통합 모듈을 거치면서 음성 발화 구간에 연관된 터치 제스처의 각 좌표가 "콜라"와 "테이블"이라는 터치 제스처의 인식 결과를 음성 인식 결과와 통합한다. [그림 V-5]는 "이거"로 인식된 객체는 "콜라"로, "여기에"로 인식된 위치를 나타내는 객체로 "테이블"로 대치되어 표시되었다.

```
<romma ver="0.5">
 <interpretation id="a1" start="1001001" end="1001045" function="spoken command">
    <request>
       <action id="5" actionType="put">가져와</action>
       <object id="1010" type="deictic" value="this">이거</object>
       <location type="deictic" value="here">여기로</location>
    </request>
 </interpretation>
</romma>
```

[그림 V-5] 멀티모달 통합 결과

VI. 결론 및 향후 과제

본 논문에서는 개별 모달리티의 단위 인지 모듈과 멀티모달 통합 모듈, 그리고 멀티모달 인지결과를 사용하게 될 로봇 인터랙션 모듈이나 통합 모듈사이의 상호 데이터 교환을 위한 통일된 양식을 제공하는 XML기반의 멀티모달 입력정보 기술 언어인 ROMMA를 설계하고 통합된 결과를 제시하였다.

ROMMA는 사람과 로봇간의 상호작용에 있어서 각 모듈 사이의 독립성을 보장하여, 각 모듈 개발의 용이성뿐 아니라 교체의 편리성을 제공하고, 멀티모달 통합 방법 연구의 편의를 도모하도록 설계하였으며 앞으로 통합 관련 연구 과제가 진행될 때 현재의 제한점을 극복하고 적용되어야 할 고려 사항 및 향후 발전 방향을 제언하면 다음과 같다.

첫째, 모달리티 입력/인지 결과의 의미 해석에서 필요한 작업 영역 및 환경 관련 지식 정보의 활용이 보다 용이하도록 로봇 지식 베이스와 멀티모달 통합 모듈 사이의 인터페이스가 체계적으로 설계되어야 할 필요가 있다.

둘째, 멀티모달 통합을 위한 모달리티 결합 방법론을 보다 체계적으로 정립하여 각 모달리티의 결과를 의미 있게 결합하는 알고리즘 개발을 지속적으로 수행하여야 한다.

셋째, 각 모듈간의 모달리티의 인지 결과 또는 통합 결과를 전송하기 위

한 통신 규약(communication protocol)의 정립이 필요하다.

넷째, 특정 도메인에서의 오브젝트에 대한 정형화되고 명시적인 작업 영역 지식의 구축을 통해 명세로서의 지식 공유의 역할뿐만 아니라 의미 추론 규칙을 통한 통합을 제시하여 보다 정확한 사용자의 의도가 인지가 가능하도록 연구가 필요하다.

본 논문에서는 입력 컴포넌트에 대한 멀티모달 입력 정보 기술 언어로 ROMMA를 제시하였으며 추후에 멀티모달 출력과 관련한 양식의 통일 방법에 대한 연구가 진행되어야 할 것이다.

참고문헌

[1] Wu Chou, Deborah A. Dahl, G. McCobb, and D. Raggett, “EMMA: Extensible MultiModal Annotaion markup language”, W3C, http://www.w3.org/TR/2005/WD-emma-20050916/, 2005.

[2] R. Bolt, “Put-That-There : Voice and Gesutre at the graphic interface”, Proceedings of the SIGGRAPH, pp.262-270, 1980.

[3] 홍기형, “음성기반 멀티모달 인터페이스 표준”, 말소리, 제 51호, pp.117-135, 2004.

[4] P. Cohen, D. McGee and J. Clow, “The Efficiency of Multimodal Interaction for a Map-based Task”, Proceedings of the Applied Natural Language Processing(ANLP), pp.331-338, 2000.

[5] S. Oviatt, “Multimodal Interactive Maps: Designing for human performance”, Human Computer Interaction, pp.93-129, 2000.

[6] B. Suhm, B. Myers, and A. Waibel, “Model-based and empirical evaluation of multimodal interactive error correction”, Proceedings of the ACM CHI 99 Human Factors in Computing Systems Conference, pp.584-591, 1999.

[7] S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T.

Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro, “Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions”, In Human Computer Interaction, pp. 263--322, 2000.

[8] S. Oviat, A. DeAngelli, and K. Kuhn, “Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction”, Human factors in computing systems, pp.415-422, 1996.

[9] Yi-Min Chee, K. Franke, M. Froumentin,S. Madhvanath, Jose-Antonio Magaña, G. Russell, G. Seni, C. Tremblay, .Stephen M. Watt, and L. Yaeger, “Ink Markup Language (InkML)”, W3C, http://www.w3.org/TR/2006/WD-InkML-20061023/, 2006.

[10] N.O. Bernsen, L.Dybkjær, “Is Speech the Right Thing for your Application?”, Proceedings of International Conference on Spoken Language Proceedings(ICSLP)'98, pp.3209-3212, 1998.

[11] J. Jacko, A.Sears, “The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Application”, Lawrence Erlbaum, 2003.

[12] 김지영, 홍기형 “심부름 영역에서의 음성과 터치제스처의 통합 인식”, The 4th Technical Workshop, Center for Intelligent Robotics, 2005.

ABSTRACT

A XML-based Specification Language For Robot Multimodal Fusion

Min, Kyung Hyun
Department of Computer Science
Graduate School
Sungshin Women's University

We proposed ROMMA (RObot MultiModal Annotation markup language), a XML based modality input description language for human-robot multimodal interaction. The independency between various robot input modalities such as speech, touch gesture, emotion, face recognition and posture, and multimodal integration is very important in developing robot systems. ROMMA provides a unified way to describe the recognition results of different input modalities and the multimodal integration result. By adopting ROMMA, the modification, enhancement, and substitution of an input modality module does not require any change of the other

recognition modules and the integration module.

ROMMA is developed from EMMA, which is recommended by W3C as a multimodal input annotation language for multimodal web interfaces, by extending to robot domain. ROMMA consists of two parts: the basic elements that define the ROMMA structure and the common recognition features of different input modalities, and the extended elements that define the specific features to each input modality and the specific task domain. ROMMA can easily extend for newly introduced modalities and task domains. In this paper, we explain how to extend ROMMA for an emotion recognition and the errand task.