

통합 서비스 로봇을 위한 MS 로보틱스 스튜디오 기반 대화 인터페이스 플랫폼 구축

Development of Conversational Interface Platform Based on MS Robotics Studio for Integrated Service Robots

임 성 수¹, 조 성 배[†]

Sungsoo Lim¹ · Sung-Bae Cho[†]

Abstract The development of IT technology makes the functions and services of robots be integrated, and thus the robots become more intelligent and useful. As sophisticated usage of robots has evolved, direct communication by human language is necessary to increase the efficiency of their usage. In this paper, we propose a conversational interface platform for integrated service robots using MS Robotics Studio. The proposed platform consists of three types of components: a conversation manager to control the flows of the integrated service robots, a user interface to interact with users, and multiple service robots to perform actions or services. For a test-bed of the proposed platform, we build a schedule manager system and confirm the usability through SUS subject test by comparing the schedule manager system with MS Outlook.

Keywords: Conversational interface, Integrated service robots, MS Robotics Studio

1. 서론

휴머노이드 로봇에 대한 관심이 증대됨에 따라서 서비스 로봇에 다양한 기능과 서비스들이 통합되고 있다. 과거의 서비스 로봇과 같은 자율로봇은 보통 사용자의 직접적인 조작에 의해 작동되었다^[1]. 따라서 기존의 로봇 연구는 사람과 로봇 간의 상호작용보다는 로봇의 자율성에 더 많은 관심이 집중되어 있었다. 그러나 서비스 로봇이 사람의 일상생활 속으로 들어옴에 따라서 사람과 보다 자연스럽게 상호작용하는 방법이 요구되고 있다. 로봇이 가정이나 사무실에서 배달, 알림, 안내 등의 다양한 서비스를 제공하기 위해서는 정확한 서비스 수행뿐만 아니라 유연한 의사소통 기능이 필요하다^[2].

이를 위해 자연언어를 통한 대화가 사람과 로봇의 상호작용에서 효과적인 방법으로 많은 관심을 받고 있다^[3]. 자연언어는 인간이 의사소통 수단으로 사용하는 것으로 정

보 교환 및 사용자 의도 파악에 효과적이며, 유연성, 명료성, 표현력 면에서 뛰어난 장점을 가지고 있다. 이러한 자연언어는 인간과 컴퓨터의 상호작용(HCI) 분야에서 이미 활발한 연구가 진행 중이며, 인간과 로봇의 상호작용(HRI)에서도 큰 비중을 차지하고 있다^[4].

본 논문에서는 다양한 기능을 수행하는 통합 서비스 로봇에서 손쉬운 대화 인터페이스 제공을 위한 플랫폼을 개발한다. 제안하는 대화 인터페이스 플랫폼은 로봇의 다양한 기능 및 서비스를 효과적으로 통합하기 위해서 MS Robotics Studio를 이용하여 개발하며, 다른 서비스 상황에서 손쉽게 대화를 모델링하고 통합할 수 있도록 AIML (Artificial Intelligence Markup Language) 기반 스크립트 언어를 이용한 대화 인터페이스를 적용한다.

2. 관련연구

2.1 마이크로소프트 로보틱스 스튜디오

마이크로소프트 로보틱스 스튜디오(MSRS; Microsoft Robotics Studio)는 로봇 제어 및 시뮬레이션을 위한 윈도우 기반 개발 환경으로, 최근 로봇분야에 있어서 이를 이

Received : January 5th, 2009 Accepted: February 12th, 2009

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2009-(C1090-0902-0046)).

† 교신저자: 연세대학교 컴퓨터과학과 정교수

¹ 연세대학교 컴퓨터과학과 박사과정

용한 다양한 연구들이 있었다. Tick은 자율 모바일 연구용 로봇(Autonomous mobile research robots)을 위한 프로그래밍 개발 툴로서 MSRS의 가능성을 보였고^[5], Rango 등은 MSRS의 응용 어플리케이션으로 6개의 사륜 로봇을 구현하였다^[6]. 각 로봇은 서로 정보를 교환하면서 지도를 구성하고, 미리 정해진 목적지를 찾는다. Tsai 등은 MSRS 환경에서 협력적 SOA (Service-oriented architecture) 시뮬레이션 프레임워크를 제안했다^[7].

개발 툴로서 MSRS는 로봇과 센서기반 응용 어플리케이션을 쉽게 구현할 수 있으며, 서비스의 이식성과 재사용성을 제공하기 위해 서비스 기반의 런타임 아키텍처를 제공한다. 그리고 분산처리 및 비동기식 프로그램 환경을 제공하므로, 로봇 시스템을 구성하는 다양한 서비스 모듈 개발에 있어서 코드의 안정성과 신뢰성을 높이고 재사용성을 향상시킨다.

서비스 모듈 간의 통신은 분산화된 소프트웨어 서비스 프로토콜(Decentralized software service protocol; DSSP)을 통해서 이루어지며^[8], MSRS 내에서 개발되는 서비스 코드들은 분산화된 소프트웨어 서비스(DSS)를 기반으로 개발된다. DSS 기반의 개발을 통해서, 서비스 모듈의 단순성과 상호호환성을 유지할 수 있으며, 동시에 대량으로 발생하는 구조화된 메시지 처리와 이벤트 통지 등을 제공하여 모듈간 연동에 있어서 유연성을 제공한다.

2.2 서비스 로봇의 상호작용

Hüttenrauch 등에 의하면, 82%의 사용자가 여러 의사소통 방법 중 자연언어를 통한 의사소통을 선호하는 것으로 나타났다^[9]. 최근 다양한 대화 모델이 로봇 분야에 적용되고 있으나, 대부분의 경우 자연스러운 대화를 처리하기 보다는 간단한 명령만을 인식하고 있다^[2].

로봇 분야에서 자연스러운 대화 처리를 위한 연구도 진행되었다. Skubic 등은 환경의 공간적 관계를 고려한 정교한 대화처리 시스템을 제안하여 대화를 통한 공간 모델링으로 서비스 로봇의 성능을 향상시켰다^[10]. Lemon 등은 모바일 로봇의 협력적 행동을 위한 멀티 모달 대화 시스템을 제안했다^[11]. 그러나 이러한 대화 인터페이스는 제공하고자 하는 서비스에 의존적으로 구성이 되어, 서비스 도메인의 확장, 변경에 대해 대화 모델링 뿐만아니라 대화 인터페이스의 수정도 필요하다.

본 논문에서는 다양한 기능과 서비스를 제공하는 통합 서비스 로봇을 위한 대화 인터페이스를 제안하여, 통합 서비스 로봇의 대화 인터페이스 구축에 드는 비용을 줄인다.

3. 통합 서비스 로봇의 대화 인터페이스 플랫폼

3.1 대화 인터페이스 플랫폼

다양한 기능과 서비스를 제공하는 통합 서비스 로봇을 개발하기 위해서는 각 기능과 서비스를 수행하는 모듈들이 유기적으로 연결되어 동작해야 한다. 본 논문에서는 이를 위해서 그림 1과 같은 대화 인터페이스 플랫폼을 구축한다.

제안하는 대화 인터페이스 플랫폼은 대화관리, 사용자 인터페이스, 그리고 서비스 로봇의 세 가지 컴포넌트로 구성된다. 대화관리 컴포넌트는 각 서비스 로봇 컴포넌트에 존재하는 대화 데이터베이스를 입력받아 다양한 서비스를 위한 대화를 수행하며, 대화 흐름 관리를 통해서 통합 서비스 로봇을 구성하는 각 서비스 로봇 컴포넌트의 제어 흐름을 관리한다. 사용자는 사용자 인터페이스 컴포넌트를 통해서 통합 서비스 로봇과 정보를 주고받으며, 대화관리 컴포넌트를 통해서 인식된 사용자의 의도는 행동 수행을 위한 서비스 로봇 컴포넌트에 전달되어 필요한 서비스가 수행된다.

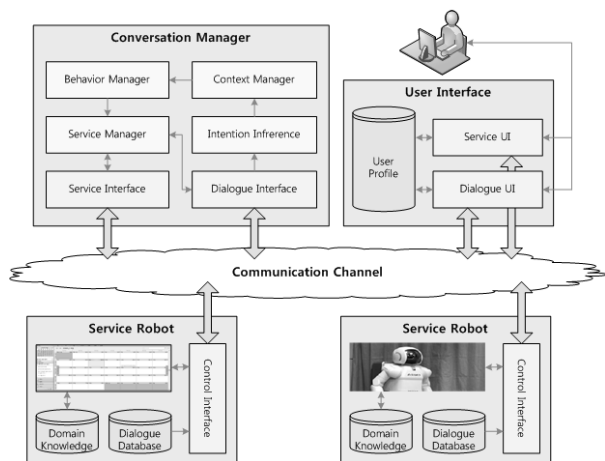


그림 1. 제안하는 대화 인터페이스 플랫폼

3.1.1 사용자 인터페이스 컴포넌트

사용자 인터페이스(UI) 컴포넌트에는 두 종류의 UI가 존재한다. 대화 UI는 사용자로부터 질의문을 받아 대화관리 컴포넌트로 해당 질의문을 전달하고, 질의문 수행 결과를 수신하여 사용자에게 전달한다. 만일 통합 서비스 로봇에서 음성인식이나 음성합성 모듈을 설치한다면 사용자 인터페이스 컴포넌트에 추가된다. 서비스 UI는 서비스 로봇 컴포넌트와 통신을 하는 것으로, 이를 통해 서비스의 수행결과를 받고 사용자 피드백을 전달한다.

3.1.2 대화관리 컴포넌트

대화관리 컴포넌트는 통합 서비스 로봇의 제어흐름 관리의 중심에 있다. 사용자의 질의문이 들어오면 대화 인터페이스 모듈이 해당 질의문을 받고, 의도 추론 모듈은 패턴매칭, 베이지안 네트워크를 이용한 주제 추론, 그리고 오토마타 기반 문형 분석 과정을 거쳐서 사용자의 의도를 분석한다^[12,13]. 상황 관리 모듈은 의도 추론 모듈에서 분석된 사용자의 의도와 과거 대화 기록을 바탕으로 대화 데이터베이스에서 현재 상황에 가장 적합한 대화 스크립트를 선택하고, 행동 관리 모듈에서는 선택된 스크립트에 기술되어 있는 행동 중에서 현재 상황에 적합한 행동을 선택하고, 선택된 행동은 서비스 관리 모듈을 통해서 행동 수행에 적합한 서비스 로봇 컴포넌트를 찾아서 명령을 전달한다.

본 논문에서 적용된 대화관리 컴포넌트는 사용자와의 대화를 수행에 필요한 정보와 외부 서비스 모듈을 동작시키기 위한 정보를 AIML 형식의 스크립트 언어에 표현하여 대화처리를 진행하므로, 도메인에 독립적인 대화 인터페이스를 제공할 수 있다^[14].

3.1.3 서비스 로봇 컴포넌트

제안하는 플랫폼을 기반으로한 통합 서비스 로봇의 실질적인 서비스는 서비스 로봇 컴포넌트를 통해서 제공된다. 서비스 로봇 컴포넌트는 소프트웨어 또는 하드웨어 로봇과 컨트롤 인터페이스, 그리고 지식기반으로 구성된다.

각 서비스 로봇 컴포넌트는 대화관리 컴포넌트의 제어를 받고 해당 서비스 도메인에 맞는 대화를 처리하기 위한 대화 스크립트들을 저장하는 대화 데이터베이스를 갖는다. 그리고 컨트롤 인터페이스 모듈은 대화관리 컴포넌트와 실제 서비스를 수행하는 소프트웨어 또는 하드웨어 로봇 간의 정보교환 및 사용자 인터페이스 컴포넌트와의 정보교환을 담당한다.

3.2 MSRS를 이용한 플랫폼 구축

소프트웨어나 하드웨어 로봇을 구축함에 있어서 부딪히는 주요 문제점 중 하나는 각 모듈이나 컴포넌트 간의 통합이다. 각 모듈이나 컴포넌트가 오프라인으로 정보를 주고받는 경우에는 객체지향 프로그래밍 언어를 활용하여 통합할 수 있지만, 온라인으로 정보를 주고받는 경우에는 네트워크 프로그래밍 작업이 필요하다. 하지만 많은 개발자들은 네트워크 프로그래밍에 충분한 경험을 가지고 있지 않기 때문에 네트워크 프로그래밍을 통한 시스템 통합은 개발자에게 큰 부담을 준다.

MSRS는 쓰레드(thread)나 락(lock), 그리고 세마포어(semaphore) 등과 같은 메시지 동기화를 위한 기법들을 사용하

지 않고, 컴포넌트 간의 메시지를 전달 할 수 있도록 설계되었다. 또한 기존의 웹기반 아키텍처의 서비스 지향적 어플리케이션 모델을 지원하므로, 이를 통해서 개발자는 모듈이나 컴포넌트 간의 통합을 손쉽게 구현할 수 있다.

3.2.1 DSS 모듈 구축

MSRS를 이용하여 제안하는 대화 인터페이스 플랫폼을 구축하기 위해서, 본 논문에서는 사용자 인터페이스, 대화관리, 서비스 로봇의 세 컴포넌트를 위한 세 종류의 DSS 모듈(UI DSS, CMDSS, SRDSS)을 개발한다.

DSS 모듈은 C#, C++, VB 등의 다양한 언어를 통해서 만들 수 있지만, 아직은 새로운 개발환경이어서 C#으로 만들어진 샘플이 대부분이다. 본 논문에서는 .NET 환경이 제공하는 공통 언어 실행환경(CLR)을 이용하여 DSS 모듈의 기본적인 기능을 C#으로 구현하고, 기존에 만들어진 C++로 구현된 각 컴포넌트 라이브러리를 .NET의 Managed C++로 래핑(Wrapping)하여 DSS 모듈에서 사용한다.

DSS 모듈에는 외부 모듈과의 통신을 위해서 자신의 상태를 모니터링할 수 있도록 내부 상태를 표현하고, 외부 모듈과의 데이터 송수신을 위한 포트를 구성해야 한다. 이를 위해 본 논문에서는 표 1과 같이 각 DSS 모듈에 다섯 가지의 내부 상태를 정의한다. UIDSS와 CMDSS, SRDSS를 위한 통신 포트로 각각 *UIRequest*, *CAInput*, *RequestResult* 클래스를 정의한다.

CMDSS의 내부 상태 값 중, *Query*는 CMDSS가 사용자의 질의문을 받은 상황을 나타내고, *Request*는 서비스 로봇에 특정 행동 수행을 요청한 상황을 나타낸다. 그리고 *Result*는 서비스 로봇으로 부터의 결과를 받은 상황을 나타내고, 마지막으로 *Answer*는 사용자 질의문을 수행한 결과 답변을 사용자에게 전달한 상황을 의미한다.

UIDSS 내부 상태 값의 앞부분은 CMDSS와의 연관된 상황을 나타내고, 뒷부분은 SRDSS와의 연관된 상황을 나타낸다. 앞부분이 *Query*인 경우는 사용자로부터 질의문을 받아서 CMDSS에 전달된 상황이고, *Answer*는 질의문의 수행 결과를 수행결과 사용자에게 전달할 답변 문장을 받은 상황을 나타낸다. 뒷부분이 *UI*인 경우는 SRDSS로부터 요

표 1. 세 종류의 DSS 모듈이 갖는 내부 상태

내부 상태	UIDSS	CMDSS	SRDSS
0	<i>Init</i>	<i>Init</i>	<i>Init</i>
1	<i>Query/Null</i>	<i>Query</i>	<i>Request/Null</i>
2	<i>Query/UI</i>	<i>Request</i>	<i>Request/UI</i>
3	<i>Answer/Null</i>	<i>Result</i>	<i>Result/Null</i>
4	<i>Answer/UI</i>	<i>Answer</i>	<i>Result/UI</i>

칭이 들어온 상황이며, *Null*은 요청을 모두 수행한 상황을 의미한다.

SRDSS의 내부 상태 값은 CMDSS와 같이 두 부분으로 구성된다. 앞부분이 *Request*인 경우는 CMDSS로부터 서비스 요청을 받은 상황이고, *Result*인 경우는 서비스 수행 결과를 다시 CMDSS로 전달한 상황을 의미한다. 뒷부분이 UI인 경우는 서비스 수행 결과, 필요한 UI의 변화를 UIDSS로 전달한 상황을 나타내고, *Null*은 UIDSS에서 필요한 수행을 끝낸 상황을 나타낸다.

DSS 모듈의 통신 포트는 기본적으로 문자열을 기준으로 정보를 송수신하도록 설계한다. *UIRequest*는 UIDSS로 UI 변경 요청 메시지를 전달하고, 수행 결과를 SRDSS로 전달하는 역할을 하고, *CAInput*은 CMDSS로 사용자 질의문을 전달하고, UIDSS로 질의문 수행 결과 답변문을 UIDSS로 전달한다. 그리고 *RequestResult*는 SRDSS로 서비스 요청을 전달하고, 서비스 수행 결과를 CMDSS로 전달한다.

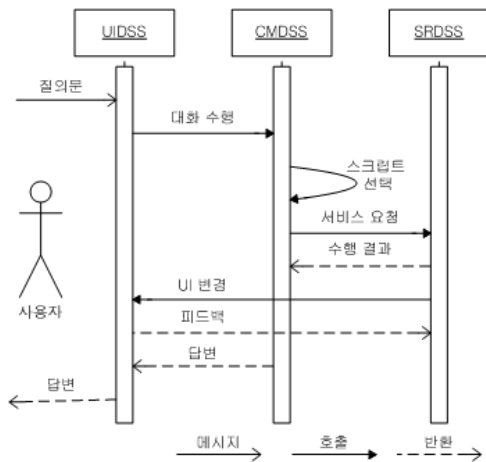


그림 2. DSS 모듈간 메시지 흐름

3.2.2 DSS 모듈간 통신 및 상태 변화

그림 2는 DSS 모듈 간의 메시지 흐름을 보여준다. 대화 인터페이스 플랫폼을 구성하는 각 컴포넌트와, 컴포넌트들 간의 통합을 위해서 이를 포함하는 DSS 모듈 간의 메시지 전달과정 및 DSS 모듈의 내부 상태 변화는 다음과 같다.

- 1) 각 DSS의 초기 내부 상태를 *Init*으로 설정한다.
- 2) 사용자가 질의문을 입력하면 UIDSS는 *CAInput*을 통해서 CMDSS로 해당 질의문을 전송하고 자신의 내부 상태의 앞부분을 *Query*로 변경한다.
- 3) *CAInput*을 통해서 들어온 사용자 질의문을 받으면, CMDSS는 자신의 상태를 *Query*로 변경하고, 대화관리

컴포넌트로 질의문을 전달한다. 대화관리 컴포넌트는 질의를 수행한 후 그 결과물을 다시 CMDSS로 전달하는데, 결과물은 특정 서비스를 위한 행동 혹은 질의문에 대한 답변이 된다.

- 4) 대화관리 컴포넌트가 서비스를 위한 행동을 요청하면, CMDSS는 자신의 내부 상태를 *Request*로 변경하고, 서비스 수행을 위한 SRDSS를 찾아 *RequestResult*를 통해서 필요한 행동을 요청한다.
- 5) 대화관리 컴포넌트가 사용자 질의문에 대한 답변을 전달하면, CMDSS는 자신의 내부 상태를 *Answer*로 변경하고, 해당 답변을 *CAInput*을 통해서 UIDSS로 전달한다.
- 6) CMDSS가 SRDSS로부터 서비스 수행결과를 받으면, CMDSS는 내부 상태를 *Result*로 변경하고 결과물을 대화관리 컴포넌트로 전달한다. 그러면 대화관리 컴포넌트는 다음에 수행해야 할 행동 혹은 답변을 CMDSS로 전달한다.
- 7) SRDSS가 *RequestResult*를 통해서 서비스 요청을 받으면, 자신의 내부 상태의 앞부분을 *Request*로 변경하고, 서비스 요청을 서비스 로봇 컴포넌트로 전달한다. 로봇은 요청을 수행하고 그 결과를 다시 SRDSS에게 전달한다.
- 8) SRDSS가 서비스 로봇 컴포넌트로부터 서비스 수행 결과를 받으면, 자신의 내부 상태의 앞부분을 *Result*로 변경하고, *RequestResult*를 통해서 그 결과를 CMDSS로 전달한다.
- 9) 서비스 로봇 컴포넌트의 서비스 수행 결과, 사용자 인터페이스의 수정이 필요하다면, SRDSS는 *UIRequest*를 통해서 UIDSS에 정보를 전송하고, 자신의 내부 상태의 뒷부분을 *UI*로 변경하고, UIDSS로부터의 응답을 기다린다.
- 10) UIDSS에서의 응답이 도착하면 SRDSS는 해당 메시지를 로봇에 전달하고, 자신의 내부 상태의 뒷부분을 *Null*로 변경한다.
- 11) UIDSS가 *UIRequest*를 통해서 데이터를 수신하면, 자신의 내부 상태의 뒷부분을 *UI*로 변경하고, 수신한 데이터를 사용자 인터페이스 컴포넌트로 전달한다.
- 12) 사용자 인터페이스 컴포넌트가 서비스 로봇 컴포넌트로부터의 데이터를 모두 처리하면, UIDSS는 자신의 내부 상태의 뒷부분을 *Null*로 변경하고, 사용자 피드백 등의 정보를 *UIRequest*를 통해서 SRDSS에 전달한다.
- 13) UIDSS가 *CAInput*를 통해서 CADSS로부터 사용자 질의문에 대한 답변을 수신하면, 자신의 내부 상태의 앞부분을 *Answer*로 변경하고 사용자 인터페이스 컴포넌트에 해당 답변을 전달한다.

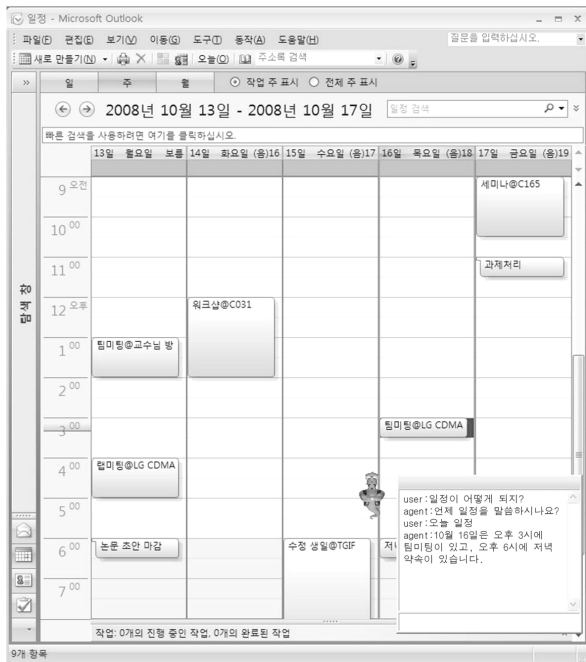


그림 3. 일정관리 시스템의 사용자 인터페이스

4. 실험 및 결과

본 논문에서는 제안하는 대화 인터페이스 플랫폼의 시험대로서 소프트웨어 로봇인 일정관리 시스템을 구축한다. 제안하는 대화 인터페이스 플랫폼이 미리 구축되어 있다고 가정 하면, 개발자는 대화 인터페이스를 제공하는 일정관리 시스템 구축을 위해서, 일정관리 서비스를 제공하는 소프트웨어 로봇 컴포넌트를 개발하고, 일정관리에 필요한 대화 스크립트를 통해 대화 데이터베이스를 설계하면 된다.

본 논문에서는 이를 위해서 MS Outlook의 일정관리 기능에 연동되는 소프트웨어 서비스 로봇을 개발하고, 소프트웨어 서비스 로봇과 다른 모듈과의 통신을 위한 SRDSS를 구현하였다. 그리고 대화관리 컴포넌트에서 일정관리에 필요한 대화를 수행할 수 있도록, 일정 추가, 일정 취소, 일정 수정, 일정 검색의 네 가지 주제에 대한 500여 개의 대화 스크립트를 생성하였다. 그림 3은 제안하는 대화 인터페이스 플랫폼 상에서 구현된 일정관리 시스템을 보여주며, 그림 4는 이 시스템에서의 대화처리 예를, 그리고 그림 5는 대화 수행에 따른 DSS 모듈간의 메시지 흐름의 단편을 보여준다.

제안하는 방법의 유용성을 보이기 위해 20~29세의 남녀 대학생과 대학원생 10명을 대상으로 MS Outlook 프로그램과 제안하는 방법에 대한 사용성 평가를 수행하였다. 사용성 평가는 표 2에서 보여주는 바와 같은 SUS(System

U: 안녕?
 A: 안녕하세요.
 U: 오늘 일정이 어떻게 되지?
 A: 오늘은 2개의 일정이 있습니다. 오전 9시에 세미나가 있고, 오전 11시에 과제처리가 있습니다.
 U: 오후 1시에 일정이 생겼는데 추가해줘.
 A: 어떤 일정인가요?
 U: 교수님과 미팅이야.
 A: 등록하였습니다.

그림 4. 대화수행 예

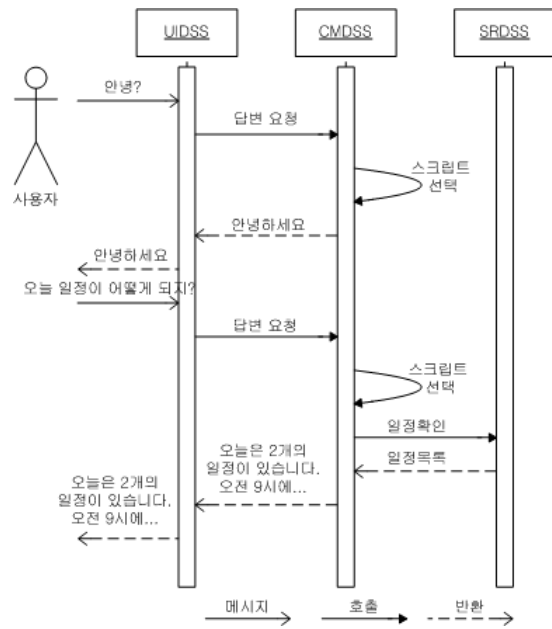


그림 4. 대화 수행에 따른 DSS 모듈간 메시지 흐름

표 2. SUS 항목

번호	문항
1	나는 이 시스템을 자주 사용할 것 같다.
2	시스템에 불필요하게 복잡한 부분이 있다.
3	시스템이 사용하기 쉽다고 생각한다.
4	이 시스템을 사용하기 위해서 전문가가 필요할 것 같다.
5	이 시스템은 다양한 기능이 조직적으로 잘 결합되어 있다.
6	이 시스템은 너무 불안정한 것 같다.
7	사람들이 이 시스템의 사용방법을 빨리 익힐 것이다.
8	이 시스템은 사용하기 귀찮은 부분이 있다.
9	나는 이 시스템을 사용했다는데 자부심이 생긴다.
10	이 시스템을 계속 사용하기 위해 많은 것을 배워야 한다.

Usability Scale)의 항목을 사용하여 진행하였다. 각 문항의 응답은 Likert 척도를 사용하여 강한 부정, 부정, 보통, 긍정, 강한 긍정의 1에서 5까지의 점수를 부여하도록 하였다.

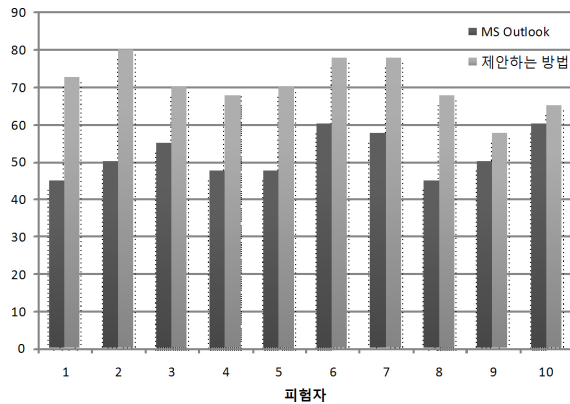


그림 4. SUS 사용성 평가 결과

홀수 문항은 평가 값에서 1을 빼고, 짝수 문항은 5에서 평가 값을 뺀 후 이를 모두 더하고, 여기에 2.5를 곱하여 0부터 100사이의 값을 갖는 점수로 환산한다^[5].

그림 4는 두 시스템에 대한 SUS 사용성 평가 결과를 보여준다. 제안하는 방법은 평균 70.3으로 MS Outlook(평균 52.5)에 비해서 더욱 높은 만족도를 보이는 것을 확인하였다.

5. 결론

서비스 로봇의 다양한 기능과 서비스들이 하나로 통합됨에 따라서, 인간과 로봇의 상호작용에 대한 관심이 증대되었다. 본 논문에서는 통합 서비스 로봇에서 인간이 의사소통의 수단으로 사용하는 자연언어를 통한 인터페이스를 손쉽게 적용할 수 있도록 도와주는 대화 인터페이스 플랫폼을 개발하였다. 제안하는 플랫폼은 대화관리 컴포넌트, 사용자 인터페이스 컴포넌트, 그리고 서비스 로봇 컴포넌트의 세 종류의 컴포넌트로 구성되며, 시스템 개발자들의 네트워크 프로그래밍에 대한 부담을 덜어주기 위해서 MSRS를 이용하여 개발되었다.

향후 연구로 제안하는 방법의 실효성을 검증을 위해 제안하는 대화 인터페이스 플랫폼 상에서 더욱 다양한 서비스들을 제공하는 서비스 로봇을 개발하고 피험자 표본의 수를 늘려 사용성 평가를 진행하고, 실제 통합 서비스 로봇에서의 대화 인터페이스 개발에 있어 제안하는 방법이 개발에 드는 시간과 노력을 얼마나 줄여줄 수 있는지에 대해 비교 실험을 수행할 필요가 있다.

참고문헌

- [1] C. Breazeal, "Social interactions in HRI: The robot view," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 181-186, 2004.
- [2] A. Agah, "Human interactions with intelligent systems: Research taxonomy," *Comput. Elect. Eng.*, vol. 27, no. 1, pp. 71-107, 2001.
- [3] A. Green and K. Eklundh, "Designing for learnability in human-robot communication," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 644-650, 2003.
- [4] H. Prendinger and M. Ishizuka, "Let's talk! Socially intelligent agents for language conversation training," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 31, no. 5, pp. 465-471, 2001.
- [5] J. Tick, "Convergence of programming development tools for autonomous mobile research robots," *Proc. of 4th Serbian-Hungarian Joint Symposium on Intelligent Systems*, pp. 375-382, 2006.
- [6] R. De Rango, and S. Nahavandi, "Simulating autonomous robot teams with Microsoft robotics studio," *SimTecT 2007 Simulation Conference and Exhibition*, pp. 1-6, 2007.
- [7] W. T. Tsai, Q. Huang, and X. Sun, "A collaborative service-oriented simulation framework with Microsoft robotic studio," *41st Annual Simulation Symposium*, pp.263-270, 2008.
- [8] H. Nielsen and G. Chrysanthakopoulos, "Decentralize Software Services Protocol - DSSP/1.0.," [Online] <http://download.microsoft.com/download/5/6/B/56B49917-65E8-494A-BB8C-3D49850DAAC1/DSSP.pdf>, 2007.
- [9] H. Hüttenrauch, A. Green, M. Norman, L. Oestreicher, and K. Eklundh, "Involving users in the design of a mobile office robot," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 113-124, 2004.
- [10] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 154-167, 2004.
- [11] O. Lemon, A. Gruenstein, and S. Peters, "Collaborative activities and multi-tasking in dialogue systems-Towards natural dialogue with robots," *Traitement Automatique des Langues (TAL)*, vol. 43, no. 2, pp. 131-154, 2002.
- [12] J.-H. Hong, S. Lim and S.-B. Cho, "Autonomous language development using speech-act template and

genetic programming for a conversational agent,”
IEEE Trans. on Evolutionary Computation, vol. 11,
no. 2, pp. 213-225, 2007.

- [13] K.-M. Kim, J.-H. Hong and S.-B. Cho, “A semantic Bayesian network approach to retrieving information with intelligent conversational agent,”
Information Processing & Management, vol. 43, no. 1, pp. 225-236, 2007.
- [14] S.-S. Lim and S.-B. Cho, “Domain independent script language for constructing mixed-initiative conversational agent,”
Proc. of The 33st KISS Fall Conference, vol. 33, no. 2, pp. 269-273, 2006.
- [15] J. Brooke, *SUS: A “quick and dirty” Usability Scale*,
In P. W. Jordan, B. Thomas, B. A. Weerdmeester & A. L. McClelland (eds.) *Usability Evaluation in INdustry*. London: Taylor and Francis, 1996.



임 성 수

2004 연세대학교 컴퓨터과학
과 졸업 (학사)
2006 연세대학교 컴퓨터과학
과 졸업 (석사)
2006~현재 연세대학교 컴퓨
터과학과 박사과정

관심분야 : Agent system, Dialogue system

E-mail : lss@sclab.yonsei.ac.kr



조 성 배

1998 연세대학교 전산과학과
졸업 (학사)
1990 한국과학기술원 전산학
과 졸업 (석사)
1993 한국과학기술원 전산학
과 졸업 (박사)

1993~1995 일본 ATR 인간정보통신연구소 객원 연
구원

1998 호주 Univ. of New South Wals 초청연구원

1995~현재 연세대학교 컴퓨터과학과 정교수

관심분야 : 신경망, 패턴인식, 지능정보처리

E-mail : sbcho@cs.yonsei.ac.kr