

스크립트 기반 인간-로봇 상호작용에서의 사용자 개입 처리

User Intervention Handling in Script-Based Human-Robot Interaction

김현중, Hyunjoong Kim*, 윤완철, Wan Chul Yoon*, 김요찬, Yochan Kim*, 권혁태, Hyuk Tae Kwon*

*KAIST 산업공학과

요약 로봇의 활동 영역이 산업 현장에서 인간의 일상 생활 공간으로 확장됨에 따라서, 로봇은 사용자가 사용하는 성능 좋은 도구 개념에서 벗어나 일상 생활을 공유하며 살아가는 파트너로 생각되기 시작했다. 이에 따라서 미리 정해진 명령과 조작 방법을 통한 단순한 상호작용만이 일어나던 과거와 달리, 사용자는 마치 사람을 대하듯 자연스럽게 로봇과 상호작용하기를 기대하게 되었다. 사람들은 본래 context 를 활용하여 모호한 표현을 하더라도 상호작용이 가능하기 때문에, 사용자의 발화가 자유롭게 이루어지면 로봇이 처한 상황적 모호성은 가중될 수 밖에 없다. 이러한 문제에 직면하여, 스크립트 기반의 인간-로봇 상호작용을 통한 문제 해결 노력이 그 돌파구가 될 것으로 기대된다. 본 논문은 이러한 노력의 일환으로, 태스크 수행중인 로봇에 자유롭게 사용자가 개입하는 상황을 처리하는 방안을 제시하고자 한다. 사용자가 개입하는 상황을 어떻게 이해할 것이며, 로봇은 어떤 전략으로 이에 대처해야 하는지 분석한 뒤, 사용자 개입에 대한 구체적인 처리 프로세스를 제시한다. 그리고 제시된 프로세스를 구현해 봄으로써 그 효용성을 검증한다.

핵심어: Human-Robot Interaction, Task-script, User Intervention

1. 서론

로봇은 전통적으로 대량 운반, 용접, 조립과 같이 인간이 하기 힘들거나 단순히 반복되는 작업을 대신하는 도구의 역할을 수행하였다. 하지만 로봇 기술의 발전과 소비자들의 새로운 요구로 인해 로봇의 활동 영역이 인간의 일상 생활 공간으로 점차 확장되고 있다. 장난감 로봇, 투어 가이드 로봇, 그리고 집이나 병원에서의 케어 로봇 등과 같이, 단순한 도구에서 벗어나 사람과 함께 생활하고, 사람에게 정보를 제공하거나 즐거움을 주고, 어려운 일을 돕거나 사람을 보살피는 일을 하는 존재로 그 역할의 패러다임이 바뀌었다[1]. 이러한 로봇의 역할 변화는 로봇을 이용하는 사용자의 역할에도 변화를 일으켰다. 기존의 산업용 로봇과 같은 경우에는 사용자의 역할은 supervisor[2]로 한정되었다. 즉, 기본적으로 로봇은 스스로 행동하는 것이 아니라 사람이 조작하는 매우 정교한 도구로 생각되었기 때문에, 사용자는 로봇의 조작 방법에 대한 전문 지식을 갖추고 있는 전문가(specialist)이어야 했으며 로봇의 상태나 태스크의 진행 상황에 대해서도 사용자 스스로 지속적으로 모니터링해야만 했다. 하지만, 개인 서비스 로봇은 스스로 행동하는 로봇으로 그 자유도가 매우 높다. 이제 사용자가 로봇을 조작하는 것이 아니라, 로봇과 함께 공동의 일을 해결해 나가는 파트너의 역할을 담당하게 된 것이다 [3,4]. 또한 로봇에 대한 전문적인 지식을 갖추지 못한 일반적인 사용자

(novice user)로 그 대상이 확장되었다.

역할의 변화와 함께 로봇 연구에서 인간-로봇 상호작용(Human-Robot Interaction, 이하 HRI)에 대한 연구의 중요성이 더욱 부각되고 있다. 사용자가 supervisor 역할을 할 때에는 사용자와 로봇 사이에 미리 정의되어 있는 명령 방법을 통한 짧고 단순한 상호작용만이 발생하였다 [5]. 로봇은 사람과의 상호작용 능력 보다는 좀더 효율적이고 자동화된 행동을 취할 수 있는 능력을 갖춰야만 했다 [2]. 하지만 비전문적인 사용자들을 대상으로 개인 서비스를 제공하기 시작하면서 사용자와 로봇간의 상호작용은 더욱 복잡해지고 그 의미가 모호한 경우가 빈번하게 발생하게 되었다. 따라서 로봇의 태스크 수행 능력이 아무리 뛰어나도 상호작용 능력이 떨어지면 로봇이 사용자의 의도를 적절히 파악하지 못하기 때문에 사용자가 원하는 방향으로 태스크를 진행하지 못하고 엉뚱한 행동을 하게 될 것이다. 이에 따라 HRI의 연구가 로봇 연구의 중심 분야로 부각되기 시작했다. 사용자와 로봇간의 적절한 상호작용을 통해서 로봇의 작업 성능 향상은 물론이고 정서적 친밀감 제공과 같은 부가적인 유용성도 얻을 수 있으리라 기대되고 있다 [6].

특히, 사용자가 로봇의 태스크 진행에 자유롭게 개입할 수 있기 때문에 이에 대한 처리 방법의 연구가 필요하다. 본 연구에서는 로봇이 정보가 모호한 상황에서 스스로 판단하여 태스크를 수행할 때, 사용자가 마치 사람들 간의 협업

에서 하듯이 자유롭게 태스크에 개입하는 상황을 분석하고 사용자의 개입을 처리하기 위한 프로세스를 제안한다. 사용자는 로봇의 태스크 수행에 빈번하게 개입하여 부족한 정보를 채워주거나 로봇의 잘못된 판단을 바로잡게 될 것이다. 이 과정에서 로봇이 이미 수행한 행동에 대해서 그 결과를 되돌려야 하는 상황이 발생할 수 있다. 이런 행동을 본 연구에서는 ‘undo’라 명하고, 사용자 개입에 대한 로봇의 대처 방법의 하나로 고려한다.

사용자 개입을 처리하기 위해서, 먼저 사용자의 개입을 분석하여 그 대처 방법을 파악하고, 이때 고려해야 하는 정보 요소들을 정의한다. 또 실제 시스템을 구현함에 있어서 태스크 스크립트 기반의 HRI 시스템을 그 대상으로 하고, 시스템의 각 요소들의 역할을 규정한다. 그리고 제안된 시스템 안에서 사용자 개입을 처리하는 절차를 설계하고 시스템을 구현하여 검증하도록 한다.

2. 사용자 개입 분석

2.1 사용자 개입의 전제조건

‘사용자가 로봇의 태스크 수행에 개입하였다’라는 상황은 다음의 두 가지 사실을 전제로 하는 것이다.

① 사용자의 최초 명령의 모호함

사용자가 개입이 발생했다는 사실은 사용자의 최초 명령이 충분한 정보를 제공하지 못하여 로봇이 그 의도를 정확히 알지 못했음을 의미한다. “콜라로 줘” 라고 말을 하였다면 로봇의 음성인식 기능에 문제가 없는 한 로봇이 ‘콜라’를 집을 수 있으나, “시원한 음료 좀 줘.” 라고 하였다면 로봇이 사용자가 원하는 음료를 콜라가 아니라 사이다로 잘못 생각하여 행동할 수도 있는 것이다. 이를 바로잡기 위해 사용자가 개입하게 된다.

② 로봇의 모호한 상황 처리

사용자가 개입했다는 것은 또한 이미 로봇이 자율적으로 행동을 시작했다는 것을 의미한다. 로봇 스스로 정보가 모호한 상황을 해결하지 못한다면 정보가 모두 명확해질 때까지 행동을 취하지 못하고 사용자에게 추가 정보를 요구할 것이다. 이러한 방식으로 로봇이 행동한다면 그 효율성에 문제가 있을 수밖에 없기 때문에 로봇은 어느 정도 정보가 수집되었다면 그것을 바탕으로 사용자가 제시한 것이라 추론되는 목표(inferred goal)를 상정하고 이를 달성하기 위해 자율적으로 행동해야 한다. 이러한 모호성 해결을 위한 추론 및 행동 처리에 관한 연구는 이미 이루어진 바 있다[6].

2.2 사용자 개입의 대상 및 유형

사용자의 개입은 다양한 modality를 통해서 이루어질 수 있다. 크게 구분한다면 1) 사람의 발화를 통해서 2) 얼굴 표정 등의 감정을 통해서 3) 직접적인 행동을 통해서 이루어진다. 발화를 통한 사람과 로봇의 상호작용은 modality

의 기본이라 할 수 있으며, 표정 등을 통한 감정 인식[7,8], 사람의 행동 인식[7]을 통한 사람의 의도 파악에 대해서도 활발한 연구가 진행 중이다. 본 논문에서는 발화를 통한 사용자의 개입만을 다룬다.

사용자의 개입이 이루어졌을 때 그 의미를 파악하기 위해서는 그 의도가 로봇의 어떤 행동에 대한 것인지 알 수 있어야 한다. 이를 위해서는 로봇의 태스크 스크립트가 기반이 되어야 한다. 발화의 내용과 스크립트의 내용을 비교하여 정보를 얻어내는 것이다. 예를 들어 음료수 심부름을 수행한다고 하였을 때 로봇은 (음료 집기) - (컵 집기) - (사용자에게 전달) 의 기본적인 태스크 스크립트를 취할 수 있다. 이때 사용자의 개입이 발생한다면 그 내용이 과연 음료에 관한 것이냐 컵에 관한 것이냐에 따라서 로봇의 대처가 달라져야 하는 것이다.

사용자의 개입이 이루어지는 시점에 따라서도 로봇의 대처는 달라질 것이다. 같은 정보라 하더라도 그 해석과 대응이 시점에 따라서 달라지기 때문이다. 개입의 시점 또한 태스크 스크립트를 기반으로 하여 구분할 수 있다. 앞서 예로 들은 음료수 심부름 상황에서는 음료를 집었는가, 컵을 집었는가의 여부가 시점을 파악하는 기준이 될 것이다. 각 시점에서 사용자의 개입이 이루어졌을 때, 같은 내용의 발화라 하더라도 그것이 이미 이루어진 로봇의 행동에 대한 것인가 아니면 앞으로 수행하려는 행동에 대한 것인가에 따라서 로봇이 대처를 달리 하는 전략이 필요하다.

2.3 ‘undo’ 행동

사용자 개입은 그 개입의 유형과 시점에 따라서 대처해야 한다. 특히, 사용자 개입이 그 시점에서 이미 이루어진 로봇의 행동에 대한 것이라면 이미 행해진 행동의 결과를 되돌리는 과정이 필요하게 된다. 예를 들어 음료수 심부름의 경우 로봇이 사이다를 들고 있었는데 사용자가 콜라를 달라고 한다면 로봇은 ‘사이다를 집어 사용자에게 전달한다’ 라는 목표를 달성하는 것 이외에도 사이다를 원래 위치에 가져다 놓는 행동이 필요하게 되는 것이다. 본 논문에서는 이러한 행동을 “undo”라 명하였다. 사용자 개입에 로봇이 자연스럽게 대처하기 위해서는 로봇이 스스로 undo 행동의 필요성을 파악하고 수행하는 것이 가능해야 한다.

2.3.1 “undo” 행동의 목적

사용자 개입이 이루어졌을 때, 그 대상 행동이 이미 이루어진 경우에는 그 결과를 되돌리는 것이 자연스럽다. 이는 1) 앞으로의 태스크 수행에 기존 행동의 결과가 도움이 되지 않으며 2) 사용자가 해당 행동을 취하지 않을 것을 원했기 때문이다. 따라서 사용자가 별도로 “콜라는 제자리에 뒀”와 같이 발화를 통해서 undo 행동을 직접 지시하지 않았다 하더라도 undo행동을 취하는 것이 필요하다. 하지만 그 필요성이 사용자의 개입에 의해서 직접적으로 드러나지

않기 때문에, 로봇이 그 의도를 파악하는 것은 쉽지 않다. 바로 이러한 ‘직접적으로 드러나지 않은 사용자의 의도’를 파악하여, 명시적인 목표뿐만 아니라 사용자가 암묵적으로 제시하는 목표도 달성하는 것이 undo의 목적이라 하겠다.

2.3.2 ‘undo’ 행동의 방식

기본적으로 생각할 수 있는 undo 행동의 방식은 undo의 대상이 되는 행동의 backtracking이다. 즉, 콜라를 집은 행동이 undo 대상이라면 지금 들고 있는 콜라 캔을 다시 원래 위치에 놓는 것이 그 행동에 대한 undo가 될 것이다.

하지만 이러한 backtracking형태의 행동이 언제나 가능한 것은 아니다. 예를 들어 컵에 음료를 따르는 행동을 이미 취했을 때 이를 undo하기 위해서 backtracking을 한다면 컵에 있는 음료를 다시 원래 있던 용기에 담아야 하지만, 실제로 그런 행동이 불가능할 수 있다. 이 경우에는 음료를 버리거나 밀봉 처리해서 보관하는 것이 적절한 undo 방법이 된다.

따라서, 로봇이 undo를 적절한 방식으로 수행하기 위해서는 로봇이 수행하는 태스크의 분석이 선행되어야 한다. 특히, 정보가 모호한 상황에서도 스스로 판단하여 태스크를 수행하도록 설계된 로봇이라면, 그러한 정보의 모호성 때문에 자신의 행동을 되돌려야 하는 경우가 발생할 수 있다는 것을 인지하고 그 되돌리는 방법을 알아야 한다. 특히, 태스크 내의 어떤 요소로 인해서 backtracking을 통한 undo 행동이 불가능해 지는지가 undo 행동의 관건이 된다.

본 연구에서는 홈 서비스 로봇의 음료수 심부름 태스크를 대상으로 하여 사용자의 개입을 연구하였다. 음료수 심부름 태스크에서의 undo 대상 행동은 앞서 살펴본 바와 같이 음료 집기와 컵 집기가 된다. 사람이 음료와 컵을 집어서 전달해주는 상황을 생각해보자. 이를 다시 원위치 시키는 것이 불가능해지는 경우는 음료를 컵에 따랐을 때 입을 알 수 있다. 이때는 음료는 버리고 컵을 씻어 제자리에 두는 것이 컵에 따른 상황에서의 undo가 될 것이다. 이렇듯 backtracking을 통한 undo를 불가능하게 만드는 행동이 각 태스크 내에 존재한다.

3. 시스템 구성 요소 및 역할

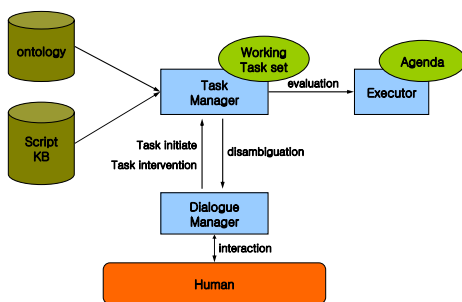


그림 1 system composition

본 연구는 홈 서비스 환경에서 음료수 심부름을 수행하는 로봇을 가정하여 사용자 개입의 문제를 다룬다. 이 로봇은 태스크 스크립트 기반으로 행동하며 사용자와의 원활한 상호작용이 가능함을 가정하고 있다. 이미 앞서 언급한 바와 같이 태스크 기반 HRI 연구가 진행되었고, 이를 처리하기 위한 cognitive architecture가 제안된 바 있다. 이 architecture 안에서, 사용자의 개입을 처리하기 위한 시스템의 요소들을 <그림 1>과 같이 제안하고, 각 요소들이 어떤 역할을 수행하는지 다음에서 살펴본다.

3.1 스크립트 지식베이스(Script Knowledge Base)

본 연구에서는 어떤 상황에서도 그 상황에 맞게 취할 수 있는 행동의 스크립트를 담고 있는 스크립트 지식 베이스가 존재하며 이를 충분히 활용할 수 있음을 가정한다. 따라서 이 지식 베이스의 구축 및 관리에 관한 논의는 하지 않겠지만, 그 안에 담겨있는 스크립트의 형태 및 내용은 규정하여 사용한다.

3.1.1 스크립트의 구조

스크립트란 도식적이고(schematic) 절차적인(procedural) 세부 사건(sub events)들로 구성되어 있는 구조화된(structured) 지식의 표현이다 [2]. 예를 들어 앞서 언급한 음료수 심부름과 같은 “물건 전달(fetch-and-carry)” 태스크는 <그림 2>와 같은 세부 사건들(sub-events)의 연결로써 이루어질 수 있다

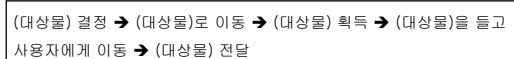


그림 2 fetch-and-carry task script

이때 주목해야 할 점은 대상물의 구체적인 언급이 없이 태스크 스크립트가 표현되고 있는 것이다. 즉, 태스크 스크립트는 태스크에 대한 일반적인 지식을 표현하고 있다. 이를 통해서 효율적으로 태스크 스크립트를 관리할 수 있다. 만약 각각의 대상물마다 해당되는 태스크 스크립트를 가지고 있어야 한다면 하나의 태스크에 대해서도 그 대상물의 수에 따라 무수히 많은, 그러나 담겨있는 절차적 지식은 같은 태스크 스크립트를 관리해야 한다.

하지만, 이처럼 대상물과 무관한 일반적인 절차적 지식 만으로는 항상 태스크를 성공적으로 수행할 수 없다. 따라서 대상물의 속성에 따라서 태스크 스크립트 내의 부분 계획(partial plan)들은 역동적으로 변화해야 한다. 즉, 일반적인 계획은 동일하게 유지 되더라도 부분 계획들이 대상물의 속성에 따라서 추가되거나 생략될 수 있다는 것이다. 예를 들어 같은 음료수 심부름에서도 식탁 위의 컵을 집을 때와 달리 냉장고 안의 콜라를 집을 때는 “(대상물) 획득” 이라는 부분 계획 전후에 “냉장고 열기”, “냉장고 닫기” 라는 부분 계획이 추가되어야 하는 것이다[9].

이는 일반적인 태스크 스크립트 내의 각 부분 계획들이 추상화된 수준(abstract level)에서 관리되고 있다고도 볼 수 있다. 집기를 하는 방법에 대한 다양한 절차적 지식들이 “(대상물) 집기” 라는 하나의 부분 계획으로 추상화 되어 있는 것이다. 본 연구의 태스크 스크립트 역시 이러한 추상화 계층(abstract hierarchy)을 가지고 있음을 가정한다.

3.1.2 상황의 구분

스크립트가 갖추어야 할 또 하나의 요소는 그 스크립트가 사용되는 상황에 대한 구분이다. 음료수 심부름이라는 동일한 태스크를 수행할 때에도 수행하는 상황에 따라서 로봇이 취하는 행동이 역동적으로 변화할 수 있어야 하기 때문이다. 특히, 사용자의 개입이 이루어졌을 때에는 로봇이 처한 상황이 일반적인 상황과 매우 다를 수 있고, 이를 적절히 반영한 태스크 스크립트가 요구된다.

이를 위해서는 먼저 ‘상황’을 어떤 요소에 따라 나눌 것인지 결정해야 한다. 상황은 여러 요소들의 조합으로써 정해진다. 현재 수행하고자 하는 태스크가 기본적인 요소가 될 것이며, 로봇과 사용자의 위치, 태스크 진행 정도, 취득 물체 등과 같은 상황 정보도 중요한 요소이다. 또한 날씨, 시간, 계절 등과 같은 환경 정보, 사용자의 나이, 성별, 개인적 선호도와 같은 사용자 모델도 상황을 규정짓는다고 고려해볼 수 있는 요소들이다. 중요한 것은, 여러 요소들이 모두 상황을 구분한다고 보기에 타당할 수는 있으나, 그것이 실제적으로 태스크 스크립트의 내용에 영향을 미치는, 즉 로봇이 행동을 달리 하게끔 만드는 요소인가를 생각해야 하는 것이다. 여기서 행동을 달리 한다 함은 앞서 추상화 수준에서의 변화를 필요로 하지 않는다는 것이다.

이러한 관점에서, 본 연구가 대상으로 하고 있는 음료수 심부름 태스크에 대해서 그 스크립트를 구분 짓는 상황의 요소로 1) 로봇의 현재 상태와 2) 사용자가 원하는 목표 상태를 선정하였다. 로봇의 현재 상태 중에서도 로봇의 위치는 태스크의 진행 내용에 영향을 미치지 않는 정보이므로 사용하지 않으며, 로봇이 현재 취득한 음료수 및 컵에 관한 정보를 그 대상으로 한다.

그런데, 로봇의 현재 상태 정보를 활용하기 위해서는 로봇이 최종적으로 달성해야 하는 상태가 어떤가를 알고 있어야 한다. 로봇이 현재 들고 있는 것이 사용자가 원하는 것인가를 판단할 수 있어야 그에 맞게 행동을 할 수 있는 것이다. 따라서, 상황을 구분하는 기준에 사용자가 원하는 상태 또한 포함시킨다. 본 연구에서는 로봇의 현재 상태 정보를 태스크 스크립트의 전제 조건(precondition), 사용자가 원하는 상태를 목표 상태(goal state)라 지칭하기로 한다. 이때 목표 상태는 사용자의 발화 내에서 명시될 수도 있지만 모호하게 표현될 수도 있다. 즉, 사용자가 “콜라 줘.” 라고 말할 수도 있지만 “시원한 것 좀 줘.” 라고 말할 수도 있다는 것이다. 따라서 로봇은 불확실성을 감안한 사용자의 의도 파악을 통해 목표 상태를 추정하기도 한다.

3.1.3 스크립트의 내용

주어진 상황에 따라서 적절한 태스크 스크립트들이 스크립트 지식 베이스 내에 존재한다고 하였다. 이때 스크립트들은 앞서 언급한 계층성(hierarchy)와 함께 다중성(multiple script)을 지니고 있다. 다중성이란 하나의 상황에서도 로봇이 취할 수 있는 스크립트는 다양하게 존재할 수 있다는 것이다. 사람이 같은 태스크를 수행하더라도 다양한 방법으로 행함을 염두에 둔 것이다.

3.2 정보 데이터 베이스 (information DB)

본 시스템에서 정보 데이터 베이스 (information DB)는 환경과 대상물 정보 및 사용자 모델(user model)에 대한 명세를 뜻한다. 로봇이 행동하기 위해서는 다양한 종류의 정보가 정보 데이터 베이스에서 수집, 관리되고 있어서 사용 가능해야 한다. 음료수 심부름 태스크를 수행하기 위해서 필요한 정보는 태스크 대상물의 속성, 사용자의 대상물에 대한 선호도 등이 있다.

태스크 대상물의 속성 정보는 태스크 수행 대상물을 결정하고, 태스크를 수행하기 위해서 스크립트 내의 필요 정보를 채우기 위해서 필요하다. 구체적으로 음료의 종류(탄산, 유제품 등), 온도, 위치 및 보관 상태, 보관 용기 그리고 컵의 위치 및 보관 상태가 그 대상이 된다.

사용자의 대상물에 대한 선호도 정보는 여러 가지 속성 정보들과 대상물 간의 상관 관계(correlation)를 제공한다. 시간대에 따라서 “음료수 좀 줘” 라는 발화의 대상물에 대한 선호도 (아침에는 커피를 마실 확률이 높다는 등과 같은)가 그 예가 될 것이다. 이 정보가 다양하고 정교해질수록 로봇의 판단 정확성이 높아진다. ATMS (Assumption-based Truth Maintenance System)와 D-S 증거이론 (Dempster-Shafer Theory of Evidence) [10]를 결합한 하이브리드 방식으로 각각의 정보를 갱신하고 관계를 파악하여 모호성을 해결하는 방법에 대한 연구가 이미 이루어진 바 있다 [6].

3.3 태스크 관리자 (Task Manager)

Task Manager는 태스크 수행에 관련된 일련의 과정들을 담당하는 핵심 모듈이다. 상황인지, 스크립트의 선택, 스크립트의 구체화, 상호작용 필요성 인지 및 요구와 같은 기능들을 수행한다.

먼저 사용자의 발화가 이루어지면 발화 내용 정보와 정보 데이터 베이스 내의 정보, 센서들로 감지한 상황 정보를 조합하여 상황을 파악한다. 파악된 상황에 맞게 스크립트 지식 베이스로부터 수행 가능한 태스크 스크립트 집합을 선출한다. 이를 Working Task Set이라 명한다. 이때 스크립트는 대상물이 없이 추상적이다. 따라서 먼저 그 대상물의 모호성을 해결해야 한다. 이를 위해서 로봇은 스스로 추론

을 하고, 만약 그 결과 어느 정도 모호성이 해결되었다면 자율적으로 태스크를 수행하고 그렇지 못하다면 모호성을 해결하고자 Dialogue Manager에 상호작용을 요청하는 것이다. 이때 상호작용을 요하는 사안들이 동시다발적으로 발생할 수 있기 때문에, 그 순서를 정하기 위해 긴급성 관리가 필요하다. 모호성이 해결되었으면 얻은 정보를 통해 태스크 스크립트의 내용을 구체화 한다. 이 과정에서 추상화 되어있던 스크립트 내의 부분 계획들이 구체화 되고, 그 대상물이 확정된다. 그리고 마지막으로 Working Task Set에 있는 다양한 스크립트 중에서 최종 의제(agenda)를 결정하기 위해서 스크립트들을 평가하는 과정이 이루어진다.

3.4 대화 관리자 (Dialogue Manager)

Dialogue Manager는 로봇과 사용자와의 대화를 자연스럽게 진행하는 역할을 한다. 즉, 사용자의 발화를 로봇이 이해할 수 있는 정보의 형태로 번역하고, 로봇의 산출물(output)을 사용자가 이해할 수 있는 문장으로 번역하는 기능이 그 기본이다[11]. 즉, 이는 사용자와 로봇간의 직접적인 인터페이스가 되는 모듈로써 모호성 해결, 문제 해결, 역할 분담과 같은 다양한 기능을 수행하는 기반이 된다.

상호 작용을 효율적으로 수행하기 위해서는 발화의 유형(query type)을 정의할 것이 요구된다[3]. 본 연구에서는 발화의 유형을 크게 호출, 질문, 정보 제공(대답 및 명령)으로 구분하였다. 호출은 상대방의 주의를 끌기 위해서, 즉 상호 작용을 시작하기 위해서 쓰인다. 질문은 그 대답 유형에 따라서 다시 yes-no question과 wh- question으로 구분한다. yes-no question은 “콜라 드릴까요?”와 같이 질문의 목적어가 명확하여 예, 아니오로 답할 수 있는 질문이며, wh- question은 “콜라, 사이다 중에서 어느 것으로 드릴까요?”와 같이 그 목적어가 모호하거나 여럿이어서 구체적인 대답을 원하는 질문이다. 이처럼 미리 질문의 유형을 구분하면 그 대답의 구조를 짐작하는 것이 가능하므로 사람의 발화를 이해하는 데 큰 도움이 된다.

4. 사용자 개입 처리 절차

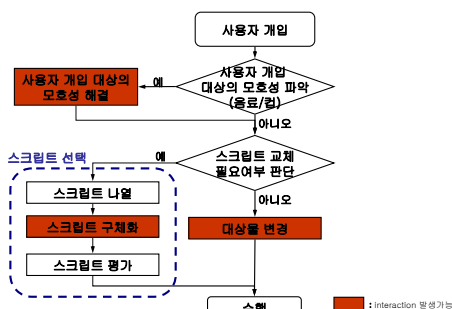


그림 3 사용자 개입 처리 절차

사용자 개입을 처리하기 위해서 문제를 해결하는 과정의 절차를 <그림 3>과 같이 제시한다.

4.1 사용자 개입 대상의 모호성 해결

사용자의 개입이 이루어질 때, 그 타이밍에 따라서 발화 대상이 모호할 수 있다. 로봇이 음료수 심부름을 하는 중에 사용자가 “그거 말고.” 라고 말한다면 이것이 ‘음료수 집기’에 대한 것인지, ‘컵 집기’에 대한 것인지 파악해야 하는 문제가 있다. 물론 “콜라 말고.”와 같이 대상을 명확히 지칭하는 개입에 대해서는 이러한 문제를 고민하지 않아도 된다.

이런 발화 대상 모호성은 그 개입 시점의 상황과 각 행동의 확신도를 활용하여 해결할 수 있다. 먼저, 개입 시점의 상황이 어떻게 영향을 미치는지 살펴보자. 발화 대상의 모호성은 ‘그것’과 같은 대명사를 포함하고 있을 때 발생한다. 그런데, 이러한 대명사 사용은 사용자와 로봇 사이에 일종의 공감대가 형성되어 있음을 전제로 한다. 이러한 공동 인지(joint cognition)가 일어나려면 발화의 대상이 로봇이 현재 들고 있는 대상이어서 사용자도 볼 수 있고 로봇도 알고 있거나, 로봇이 막 수행하려고 하는 동작이어서 사용자가 그 내용을 알고 있어야 한다. 로봇이 먼 미래에 수행하려는 동작에 대해서는 사용자가 그 내용이나 대상을 파악할 수 없기 때문에 대명사를 쓰지 못한다.

일반적인 음료수 심부름 태스크 스크립트를 (음료)로 이동 → (음료) 집기 → (컵)으로 이동 → (컵) 집기 → 사용자에게 전달 의 절차로 볼 때, 만약 ‘음료로 이동’, ‘음료 집기’ 행동을 수행하는 시점에서 사용자의 개입이 발생하면 그 대상은 음료라고 거의 확신할 수 있다. 이 시점까지는 사용자는 아직 로봇이 컵을 집으려는 것인지 아닌지 짐작하기 어려우며, 따라서 컵을 ‘그것’이라고 지칭할 수 없다. 로봇이 컵으로 이동하기 시작하면 이제 사용자는 로봇의 다음 행동이 컵을 집으려는 것임을 알 수 있기 때문에 개입 대상이 컵이 될 수도 있게 된다. 따라서 사용자 개입 대상이 컵일 확률이 증가하게 된다.

두 번째로 고려하는 요소는 로봇이 판단한 각 행동의 대상에 대한 확신도이다. 이는 로봇의 자율성에서부터 비롯된 요소라고 할 수 있다. 예를 들어 현재 사용자가 원하는 음료가 콜라라고 100% 확신할 수 없더라도 어느 일정 수준(threshold)만 넘어서는 확신을 준다면 이를 목표로 행동에 옮긴다는 것이다. 따라서 로봇의 각 행동의 대상에 대해서는 행동을 시작하면서 로봇이 판단한 확신도가 있을 것이다. 다른 요소를 고려하지 않는다면, 행동 대상 중 확신도가 작은 쪽에 대해서 사용자가 개입할 확률이 더 클 것이다. 따라서 각 행동 대상에 대한 확신도를 개입의 모호성 해결을 위한 정보로써 활용할 수 있다.

본 연구에서는 이러한 두 가지 정보를 조합하여 모호성을 해결하기 위해 D-S 증거이론을 활용하였다. D-S 이론은 확신의 정도를 참(True), 거짓(False) 외에 알 수 없음(Ignorant)을 고려하여 표현하는 방법과 각 증거를 결합하

여 그 확신의 정도를 얻어낸다는 특성, 그리고 현재 주어진 환경에 대해서 주관적 확신도가 고려된다는 점에서 고전 확률이론보다 모호성 해결에 적합하다.

4.2 스크립트 교체 필요 여부 판단

사용자 개입이 발생하였다는 것이 항상 undo 행동을 필요로 하는 것은 아니다. Undo 행동은 사용자의 개입이 1) 로봇이 이미 수행한 행동 및 그 결과에 대해서 이루어졌으며, 2) 개입의 의도가 로봇이 판단하고 있던 목표(goal)과 다를 경우에 필요하게 된다. 때에 따라서는 현재 수행 중이던 태스크 스크립트의 instance를 변경하는 등의 스크립트 수정을 통해서 사용자 개입을 처리하기도 하고, 때에 따라서는 사용자 개입에도 불구하고 현재 수행 중이던 태스크 스크립트를 그대로 이용하기도 한다. 사용자 개입에 대한 처리 유형의 결정은 다음의 두 가지 과정을 통해서 이루어진다.

① 구체화된 목표(concretized goal)와 로봇이 추론해낸 목표(inferred goal)의 비교

사용자의 개입은 로봇이 추론했던 목표를 명확히 하는 행동으로 이해 될 수 있다. 개입이 이루어지기 전에는 로봇이 주어진 정보를 바탕으로 추론해낸 목표(inferred goal) 목표 상태(goal state)로 하여 태스크를 수행하고 있었다. 하지만 개입을 통해 구체화된 목표(concretized goal)가 제시되면 로봇의 목표 상태는 이 구체화된 목표로 수정되어야 한다. 만약 기존 목표와 수정된 목표가 다르다면 이를 해결하기 위해서 스크립트를 수정하거나 undo를 포함한 새로운 스크립트로 교체하는 등 스크립트에 변화를 줄 필요가 있다. 하지만, 기존의 목표와 수정된 목표는 경우에 따라서 같을 수도 있다. 예컨대 로봇이 콜라를 집으려고 생각하고 이동 중에 있었는데 사용자가 “콜라로 줘.” 라고 한다면 로봇은 원래 하려던 대로 태스크를 계속 진행해나가면 되는 것이다.

② 구체화된 목표(concretized goal)와 로봇의 현재 상태(current state)의 비교

스크립트에 변화를 줘야 한다는 판단이 된 후에는 그 변화가 과연 undo를 필요로 하는 것인지 판단해야 한다. 이를 위해서는 구체화된 목표와 로봇의 현재 상태를 비교해야 한다. 사용자가 기존의 목표에 따라서 이미 사이다를 취득하였는데 사용자가 “콜라로 줘.” 라고 한다면 콜라를 집는 목표 달성을 위한 행동과 함께 이미 들고 있는 사이다의 undo 행동도 취해야 하는 것이다. 기존의 태스크 스크립트에는 undo 행동이 포함되어 있지 않기 때문에 이런 상황은 태스크 스크립트의 교체를 필요로 한다. 하지만 기존의 목표는 사이다였지만 아직 이를 취득하지 않았다면, 목표물을 사이다에서 콜라로 변경하기만 하고 기존의 태스크 스크립트를 그대로 이용할 수 있다. 이처럼 추론했던 목표와 구체화된 목표가 다른 경우에도 undo가 필요하지 않을 수 있기 때문에, undo 스크립트 필요 여부에 대한 최종적인 판단은 구체화된 목표와 로봇의 현재 상태 비교를 필요로 한다.

4.3 스크립트 나열

사용자의 개입이 태스크의 목표 상태(goal state)를 바꾸었고 이에 따라 Undo 행동을 수행해야 하는 상황으로 판단되었다면 로봇은 현재의 태스크 스크립트를 undo 행동을 포함한 새로운 태스크 스크립트로 교체해야 한다. 이미 3.1.2절에서 태스크 스크립트는 미리 정의된 상황에 따라서 구분을 짓고 상황에 맞게 선택하여 사용함을 가정하고, 이를 위해서 음료수 심부름 태스크에 대한 상황의 요소로 1) 로봇의 현재 상태와 2) 목표 상태를 선정하였다. 따라서 태스크 스크립트를 교체하게 될 때에도 스크립트의 선택은 로봇의 현재 상태와 목표 상태에 부합하는 태스크 스크립트의 검색을 통해 이루어진다

4.4 스크립트 구체화(Script Detailing) & 대상물 변경(Instance Replacement)

기본적으로 두 프로세스는 모두 스크립트 내의 정보 요소들을 확정하는 작업으로써 유사한 성격을 지닌다. 태스크 스크립트의 형태가 정해졌다면, 추상화 단계의 스크립트에 정보를 넣어 수행이 가능하도록 구체화하는 과정인 것이다. 음료수 심부름 태스크 예를 생각해보자. 로봇이 콜라를 집어야 하는데 콜라가 냉장고 안에 있는 상황이라면 ‘음료로 이동’ 행동의 목적지는 냉장고로 정하고 ‘음료 집기’ 행동은 ‘콜라 집기’로 구체화되며 콜라를 집기 위해서 ‘냉장고 열기’, ‘냉장고 닫기’ 행동이 ‘콜라 집기’ 행동 전후에 추가되어야 한다. 이처럼 태스크 스크립트를 구체화하는 과정이 ‘script detailing’ 과 ‘instance replacement’ 이다. 다만 새로운 스크립트로 교체한 경우에는 script detailing이 되며, 기존의 스크립트를 수정하는 경우에는 기존 정보를 교체하는 instance replacement가 되는 차이일 뿐이다.

4.5 스크립트 평가(Script Evaluation)

동일한 상황이라 하더라도 로봇이 취할 수 있는 태스크 스크립트는 하나로 결정되지 않을 것이며 대신 현재 상황에서 수행 가능한 working task set이 선택될 것임은 이미 언급한 바 있다. 이 working task set에 속한 태스크 스크립트들 중에서 가장 효율적인 스크립트를 선정하기 위해서 스크립트 평가를 해야 한다. 평가를 위한 척도(measure)로써 1)로봇이 태스크를 모두 수행하는 시간 2)사용자가 원하는 목적(ex. 음료 얻기)를 달성하는 시간의 두 가지를 선정하였다. 첫 번째 척도는 로봇이 태스크를 완수하여 로봇의 자유도가 높아졌고 따라서 새로운 태스크를 수행할 수 있는 상태가 되었다는 점에서 사용자에게 가치를 제공하며, 두 번째 척도는 현재 태스크에서 사용자가 실제 원했던 목적을 달성하였다는 자체로써 사용자에게 가치를 제공하기 때문에 두 가지 요소를 모두 고려할 때 효율적인 행동이 가능할 것이다. 이를 정리하면 <그림 4>와 같다.

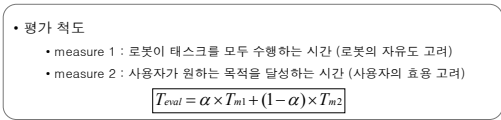


그림 4 태스크 스크립트 평가 척도

5. 시스템 구현

4장에서 제시한 절차로 HRI에서 사용자 개입 처리가 가능한지, 또 시스템이 실제 구현 가능한지를 검증하기 위해서 전산 모델을 구현하고 평가하였다.

5.1 시스템 구현 개요

구현 시뮬레이터는 <그림 5>에서 보는 것과 같이 이루어졌다. 시뮬레이터 상의 정보는 크게 다음과 같다.

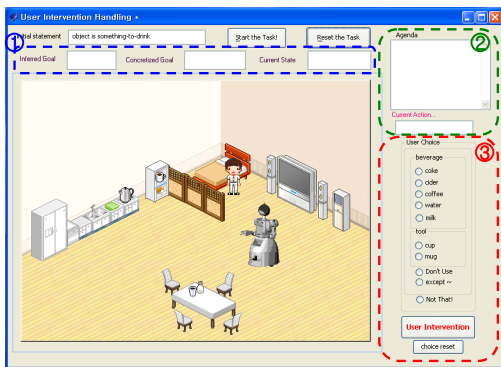


그림 5 시뮬레이터 기본 개요

① 로봇의 goal 및 state 정보: 4.1.2절에서 살펴보았듯이 사용자 개입 처리에서 가장 중요한 역할을 하는 ‘스크립트 교체여부 파악’을 위해서는 사용자 개입으로 구체화된 목표, 로봇이 추론했던 목표, 로봇의 현재 상태의 세가지 정보가 필요하다. 이를 보여줌으로써 로봇이 현재 어떻게 추론을 전개하고 있는지 살펴볼 수 있도록 하였다.

② 스크립트 정보: 추론에 의하여 최종적으로 로봇이 선택한 agenda와 agenda내에서 현재 진행중인 행동을 보여줌으로써 애니메이션으로 보여지는 로봇 행동에 대한 이해도를 더욱 높이고 그 적절성에 대한 판단을 돕도록 하였다.

③ 사용자 개입 정보: 사용자가 개입 내용을 결정하기 위해서 만들어진 창이다. 구현 과정에서 자연어 처리에 대한 어려움이 있으며 이는 본 연구의 주 대상이 아니기 때문에 이 창에서 정보들을 조합하여 다양한 사용자 개입을 가능케 했다.

5.2 스크립트 및 대상물 데이터베이스 표상

3장에서 다른 내용들을 기반으로 시뮬레이터에서 사용할 task script 및 object DB를 XML 문서 포맷으로 표상하였다.

먼저 object DB를 살펴보면 <그림 6>과 같이 object를 분류하되 hierarchy를 두어 그 속성간의 상속이 가능하고 또 확장에 용이하도록 하였다. Fetch-and-carry task를 대상으로 하였기에 대상물을 집는 방식의 결정을 위해 ‘accessible’ 속성을 부여하였으며 사용자의 선호도를 반영하기 위한 ‘temperature’ 속성, 컵 필요 여부 판단을 위한 ‘container’ 속성 등이 주요 속성이다.

```
<?xml version="1.0" ?>
- <object_HRI>
- <Class Object="Object">
  - <subClassOf>none</subClassOf>
  </Class>
- <Class Object="ob_foods">
  - <subClassOf>Object</subClassOf>
  <grp transfer="yes" add="yes" />
  </Class>
- <Class Object="ob_drink">
  - <subClassOf>ob_foods</subClassOf>
  <grp drinkable="yes" />
  </Class>
- <Class Object="ob_drink_water">
  - <subClassOf>ob_drink</subClassOf>
  <grp />
  </Class>
- <Class Object="ob_drink_water1">
  - <subClassOf>ob_drink_water</subClassOf>
  <grp location="110,264" repository="refrigerator" accessible="no" temperature="cold" instance="water1" container="bottle" />
  </Class>
- <Class Object="ob_drink_water2">
  - <subClassOf>ob_drink_water</subClassOf>
  <grp location="340,432" repository="Table" accessible="yes" temperature="medium" instance="water2" container="bottle" />
  </Class>
```

그림 6 object DB

Task script는 <그림 7>에서 보듯이 스크립트 내에 branch를 포함하고 있어 대상물의 속성에 따라 유연하게 스크립트를 적용할 수 있는 구조로 하였다. Branch를 구분하는 index를 object DB 내에 담긴 정보 등과 서로 비교하여 선택한다. 또한 스크립트 내의 각 행동은 그 target과 effect가 명세 되어 있어 현재 상황 파악 및 다음 행동 수행을 위한 정보 수집에 사용된다.

```
<?xml version="1.0" ?>
- <TaskModel>
- <class task="bring-ob_drink1" PreconditionGoal="none,none,ob_drink1,ob_cup1" scriptID="0101">
  - <scriptSegmt target="loc_robot" effect="loc_ob_drink1">MoveTo:ob_drink1</scriptSegmt>
  - <scriptSegmt>
    - <case index="accessible:no">
      - <scriptSegmt target="loc_ob_drink1" effect="open">Access:ob_drink1</scriptSegmt>
      - <scriptSegmt target="robot_state" effect="ob_drink1">PickUp:ob_drink1</scriptSegmt>
      - <scriptSegmt target="loc_ob_drink1" effect="closed">ArrangeAfterAccess:ob_drink1</scriptSegmt>
    </case>
    - <case index="accessible:yes">
      - <scriptSegmt target="robot_state" effect="ob_drink1">PickUp:ob_drink1</scriptSegmt>
    </case>
  </scriptSegmt>
  - <scriptSegmt target="loc_robot" effect="loc_ob_cup1">MoveTo:ob_cup1</scriptSegmt>
  - <scriptSegmt target="robot_state" effect="ob_cup1">PickUp:ob_cup1</scriptSegmt>
  - <scriptSegmt target="cup_state" effect="filled">Fill:ob_cup1</scriptSegmt>
  - <scriptSegmt target="loc_robot" effect="loc_user">MoveTo:user</scriptSegmt>
  - <scriptSegmt target="user_state" effect="ob_drink1,ob_cup1">Give:ob_drink1,ob_cup1</scriptSegmt>
  </class>
```

그림 7 task script

5.3 시뮬레이션 과정

먼저 ‘start the task’ 버튼을 누르면 ‘음료수를 줘’ 라는 명령에 대해서 로봇이 임의의 사용자 모델을 가정하고 추론을 통해 자율적으로 task를 수행한다. 로봇의 태스크 수행 도중에 사용자가 <그림 6>의 ③번 영역에 있는 라디오 버튼을 조합하여 개입을 하게 되고, 로봇은 이를 해석하여 적절한 대처를 보인다. 개입의 해석 과정에서 모호함이 있을 때에는 질문을 던지기도 하는데 크게 1)개입의 대상에 대해서 2)구체적인 대상물의 instance에 대해서 질문을 던진다. 질문을 던지는 상황이 <그림 8>에 표현되어 있다. 현재 우유와 컵을 집는 것을 목표로 하고 있던 로봇에게 ‘우유 말

고' 라고 개입을 하자 그럼 어떤 음료를 원하는지 질문하는 것이다. 이러한 상황은 애니메이션과 함께 inferred goal, concretized goal, current state 등의 정보와 agenda를 통해서 쉽게 이해할 수 있다.

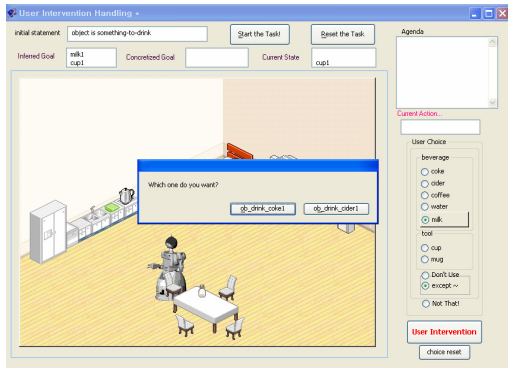


그림 8 로봇의 질문

현재 30여 개의 스크립트를 기반으로 경우에 따라 약 100여 종류 이상의 행동을 취할 수 있다. 또 행동이 어색하지 않고 사람과 유사한 방식으로 진행되었다.

6. 결론 및 추후 연구 방향

최근의 로봇은 그 영역을 산업 현장에서 인간의 생활 공간으로 넓히고 있다. 이는 인간이 로봇에게 취하는 상호작용 방식을 인간과 인간간의 자연스러운 상호작용과 더욱 유사해지게끔 하였고, 이에 따라서 로봇이 처한 정보의 모호성은 가중되었다. 로봇의 자율적 태스크 수행은 정보의 모호성으로 인해 사용자의 개입을 필연적으로 유발하며, 이에 대한 로봇의 적절한 대처가 요구된다.

본 연구에서는 스크립트 기반의 HRI에서 사용자의 개입을 분석하고, 특히 사용자의 개입이 이미 로봇이 수행한 행동의 결과를 되돌리는 과정을 'undo'라 명하였다. 사용자의 개입 내용과 시점에 따라서 로봇의 대처는 여러 가지 형태로 행해져야 함을 파악하였으며, 'undo' 행동이 필요하게 되는 경우에는 로봇이 수행하던 태스크 스크립트의 교체가 필요함을 보였다. 태스크 스크립트는 그 효율적 활용을 위하여 추상화 계층(abstract hierarchy)을 가지도록 하였고, 이를 운용하기 위한 시스템 구성 요소로 스크립트 지식베이스, 정보 데이터베이스, 태스크 관리자, 대화 관리자를 정하여 그 역할을 규정하였다. 마지막으로 사용자 개입을 처리하기 위한 프로세스를 제안하여 이를 바탕으로 시뮬레이터를 구현하여 검증하였다.

본 연구에서는 스크립트 지식 베이스의 구축에 관하여는 다루지 않고 그 존재를 가정하였다. 추후에 이들의 구축 및 관리에 관하여 연구가 필요할 것이다. 특히, 스크립트 지식 베이스는 스크립트 기반 HRI의 특성을 좌우할 수 있는 중요 요소이다. 스크립트의 형태에 대한 보다 심도 있는 연구

와 함께 스크립트의 자동 생성 및 지식 베이스 관리에 대하여 연구되어야 한다. 또한 본 연구에서는 태스크의 수행을 로봇이 담당하였다. 하지만, 인간과 로봇을 파트너 관계로 생각할 때, 이를 각각 독립적으로 생각할 것이 아니라 하나의 통합 인지체(joint cognitive system)으로 보고 스크립트의 활용을 연구하는 것이 필요하다.

참고문헌

- [1] A. Agah, "Human interactions with Intelligent systems: research taxonomy", Computers and Electrical Engineering, vol. 27, no. 1, pp. 71-107, 2001
- [2] J. Scholtz, "Human-robot Interactions: Creating Synergistic Cyberforces", In Proceedings of Hawaii International Conference on System Science, Jan. 2003
- [3] T. Fong, C. Thorpe, C. Baur, "Collaboration, Dialogue, and Human-Robot Interaction", In Proceedings of 10th International Symposium of Robotics Research, Lorne, Victoria, Australia, Nov. 2001
- [4] C. Breazeal, J. Gray, G. Hoffman, M. Berlin, "Socila Robots: Beyond Tools to Partners", in Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication, Kurashiki, Okayama, Japan, Sept 20-22, 2004
- [5] C. Breazeal, "Social interactions in HRI: The robot view", IEEE Trans. On Systems, Man, and Cybernetics-Part C: Applications and Reviews, vol. 34, no. 2, pp. 181-186, 2004
- [6] 윤영식, 김요찬, 윤완철, "홈서비스 작업 환경에서 인간-로봇 시스템의 모호성 해결을 위한 상호작용 모델", HCI 2005 학회, 2005
- [7] D. Kulic, E.A. Croft, "Estimating Intent for Human-Robot Interaction", in Proceedings of IEEE international conference on Advanced Robotics, 2003
- [8] J. A. Adams, P. Rani, N. Sarkar, "Mixed Initiative Interaction and Robotic Systems", AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems, Technical Report WS-04-10, 2004
- [9] 윤완철, 이강우, 윤영식, 양정연, "A Task Oriented Human-Robot Interaction", 인간기능 생활지원 지능로봇 기술개발 사업단 기술교류 Workshop, August. 2004
- [10] G. Shafer, "A Mathematical Theory of Evidence", Princeton University Press, 1976
- [11] R. C. Schank, R. Abelson, "Scripts, plans, goals and understanding", Hillsdale, NJ: Erlbaum, 1997