

영역-독립적인 인공지능 계획기를 이용한 시간 추론 서비스의 구현

(Implementation of temporal reasoning services using a domain-independent AI planner)

김 현 식*, 박 찬 영*, 김 인 철*

(Hyun-Sik Kim, Chan-Young Park, In-Cheol Kim)

요 약 가정용 서비스 로봇들은 사용자들에게 다양한 시간 추론 서비스들을 제공할 수 있어야 한다. 본 논문에서는 영역-독립적인 인공지능 계획기를 이용하여 이러한 시간 추론 서비스를 효과적으로 구현하는 방법을 제안한다. 영역-독립적인 인공지능 계획기를 이용하여 시간 추론 서비스를 개발하기 위해서는 다양한 실세계 시간 제약들을 어떻게 하나의 계획 영역 정의 언어로 표현하느냐 하는 지식 공학적 문제뿐만 아니라, 인공지능 계획기와 서비스 소비자 사이의 인터페이스를 어떻게 구현하여야 하느냐 하는 시스템 설계 문제를 함께 해결하여야 한다. 본 논문에서는 가정용 서비스 로봇이 고려해야 하는 하나의 예제 시나리오와 전형적인 시간 제약들을 제시하고, 이들을 표준 계획 영역 정의 언어로 표현하는 방법을 제시한다. 또한 본 논문에서는 새로운 시간 추론 서비스를 효율적으로 생성하고 제공하기 위해, 인공지능 계획기를 기초로 하나의 서비스 에이전트를 구현하는 방법에 대해서도 설명한다.

핵심주제어 : 시간 추론 계획, 서비스 에이전트, 가정용 로봇

Abstract Household service robots should be able to provide their users with a variety of temporal reasoning services. In this paper, we propose an effective way of developing such temporal reasoning services using a domain-independent AI planner. Developing temporal reasoning services with a domain-independent AI planner, we have to address both the knowledge engineering problem of how to represent various real-world temporal constraints in a planning domain definition language, and the system design problem of how to realize the interface between the AI planner and the service consumer. In this paper, we introduce an example scenario and a set of typical temporal constraints for a household service robot, and then present how to represent them in the standard planning domain definition language. We also explain how to implement a service agent based on an AI planner in order to develop and provide new services efficiently.

Key Words : Temporal planning, Service agent, Household robot

1. 서 론

사용자 일정관리와 가사도우미 등 다양한 편의

서비스를 제공하는 가정용 서비스 로봇에게는 사용자의 일상 활동뿐만 아니라 이를 지원하기 위한 로봇 자신의 작업을 준비하기 위해 효율적인 시간 추론 계획 기능이 필요하다[1, 2, 9]. 가정용 서비스 로봇을 위한 시간 추론 계획의 중요한 문제 중

† 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음.

* 경기대학교 컴퓨터과학과

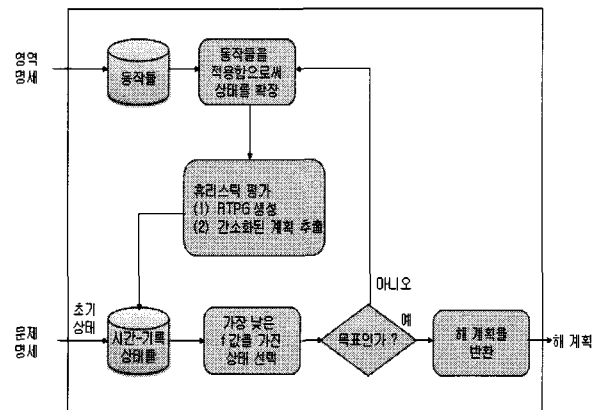
의 하나는 가정환경에서 자주 등장하는 전형적인 시간 제약들을 어떻게 계획 영역 정의 언어로 표현하느냐 하는 것이다. 특히 PDDL(Planning Domain Definition Language)[4, 5, 6]과 같은 표준 계획 영역 정의 언어에 기초하여 개발된 영역-독립적 인공지능 계획기를 시간 추론 서비스에 이용하는 경우, 이러한 계획 지식 공학 문제는 매우 중요한 설계요소가 된다. PDDL은 영역-독립적인 인공지능 자동계획문제를 표현하기 위해 개발된 언어로서, 계획 영역(domain)과 계획 문제(problem)를 정의하는데 사용된다. PDDL의 영역 정의부에는 계획생성을 위한 환경모델과 동작모델이 기술되고, 문제 정의부에는 실제 계획에 연관된 사물들(object)과 이들의 초기 상태, 그리고 목표 상태 등이 기술된다. PDDL은 STRIPS 형태의 초기 버전 외에도 다양한 기능이 추가된 여러 형태의 확장 버전들이 존재한다. 특히 PDDL 2.2[6]은 수치값(numeric value)을 갖는 상태변수 정의 기능과 각 동작의 소요시간(duration time), 시간 전조건(temporal preconditions), 시간 효과(temporal effects) 정의 기능, 유도 서술자(derived predicate) 정의 기능과 시간대별 초기화 사실(timed initial literal) 정의 기능 등이 추가된 확장 버전이다.

본 논문에서는 가정용 서비스 로봇이 고려해야 하는 하나의 예제 시나리오와 전형적인 시간 제약들을 제시하고, 이들을 표준 계획 영역 정의 언어로 표현하는 방법을 설명한다. 또한 본 논문에서는 로봇 지능 구조의 확장성과 지능 모듈의 재사용성을 위해 이러한 시간 추론 계획기를 중심으로 하나의 시간 추론 서비스 에이전트를 구현하는 방법에 대해서도 설명한다. 끝으로 이러한 시간 추론 서비스를 다른 에이전트들의 서비스와 유기적으로 결합함으로써 새로운 시간 추론 응용 서비스를 쉽게 제공할 수 있음을 보인다.

2. 관련 연구

본 논문에서는 시간 추론 서비스의 구현을 위해 선행연구를 통해 본 연구실에서 개발한 시간 추론 계획기인 TPlan[10]을 이용한다. TPlan은 각 동작의 소요시간(duration time)과 시간 조건들(timed predicates), 그리고 목표달성 한계시간

(deadline)을 포함한 시간 계획 문제를 풀 수 있도록 설계되었다. TPlan은 그림 1과 같이 상태 공간 탐색(state-space search)과 간소화된 시간 계획 그래프(Relaxed Temporal Planning Graph, RTPG)를 이용한다. 시간 계획 그래프는 동작실행과 시간흐름에 따른 상태변화를 추적해 볼 수 있는 확장된 계획 그래프이다[7]. TPlan은 이러한 시간 계획 그래프를 이용함으로써, 다양한 시간 제약 조건을 만족하는 최적의 시간 계획을 생성할 수 있다. TPlan에서는 표준 계획 영역 정의 언어인 PDDL (Planning Domain Definition Language) 2.2 [6]를 기초로 시간 계획 문제를 표현한다.



<그림 1> TPlan의 계획수립과정

PDDL 2.2는 수치 변수(numeric variable) 정의 기능, 각 동작의 소요시간(duration time) 정의 기능, 시간 전조건(temporal preconditions)과 시간 효과(temporal effects) 정의 기능, 유도 서술자(derived predicate) 정의 기능, 시간대별 초기화 사실(timed initial literal) 정의 기능 등 시간 추론 계획 문제(temporal planning problem)를 표현하는 중요한 기능을 포함하고 있다. PDDL 2.2에서 지원하는 유도 서술자는 일반적으로 동작들의 실행 결과에 따라 참, 거짓 값이 결정되는 기존 서술자들과는 달리, 미리 정의된 특정 규칙들에 의해서만 참, 거짓 값이 결정되는 서술자를 의미한다. 예컨대, 블록쌓기 계획문제에서 서술자 above는 표 1과 같은 규칙에 의해 참, 거짓 값이 결정되는 대표적인 유도 서술자로 볼 수 있다.

<표 1> 유도 서술자 예

```

if on(x,y) OR (exists z: on(x,z) AND
above(z,y))
then above(x,y)
    
```

한편, 시간대별 초기화 사실은 동작의 결과가 아니라 외적 요인에 의한 환경 변화를 나타내는 수단이다. 이 기능은 향후 실행될 동작들과는 무관하게, 계획 단계에서 미리 특정 시간대에 참이 되어야 할 사실들을 정의함으로써 외적 요인에 의한 사건발생(event)을 표현할 수 있는 것을 의미한다. 표 2의 예는 서로 다른 시간대인 9시와 20시에 서술자 shop-open의 참, 거짓값을 미리 설정함으로써 상점이 그 시간에 열고 닫히는 사건을 표현하고 있다.

<표 2> 시간대별 초기화 예

```

(:init
  (at 9 (shop-open))
  (at 20 (not (shop-open)))
)
    
```

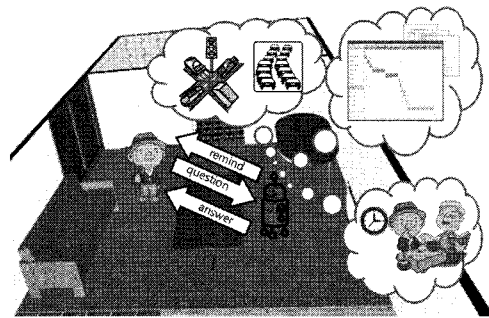
TPlan은 현재 PDDL2.2 Level3 수준의 입력 표현언어와 계획 기능을 제공하고 있다. 이와 같은 TPlan의 높은 시간 제약 표현력과 효과적인 시간 추론 기능에도 불구하고, 가정용 서비스 로봇 영역에서 요구되는 다양한 시간 제약 표현과 추론 서비스를 구현하기 위해서는 아직도 부족한 점이 많다.

3. 예제 시나리오

이 절에서는 가정용 서비스 로봇에게 요구되는 전형적인 시간 추론 서비스의 예를 설명한다. 그림 2와 같이, 이 예에서 로봇은 사용자 일정 정보를 기초로 사용자의 구체적인 향후 활동을 계획하고 이것을 적절한 시간에 사용자에게 알려주는 역할(remind)을 수행한다고 가정한다. 또한, 로봇은 사용자 일정과 연관된 다양한 질의(question)에 대해 스스로 추론을 통해 적절한 답을 찾아내고 응답(answer)하는 서비스도 제공하는 것으로 가정한다. 또, 사용자에게 새로운 약속이 발생하거나 혹은 계획된 활동 일정이 취소

되면 로봇은 이와 같은 변경사항을 즉시 반영하여 새롭게 사용자의 향후 일정과 자신의 작업을 계획하는 것으로 가정한다.

이러한 가정용 서비스 로봇에게 요구되는 시간 추론 서비스는 단순히 사용자 일정만을 고려하는 것이 아니라 또 다른 사용자의 일정이나 로봇에게 주어진 업무일정 등을 고려해야 한다. 또한 로봇에게 주어진 업무 목표가 같더라도 사용자 일정에 따라 로봇이 그 업무를 달성하기 위한 과정이 달라질 수 있어야 한다.



<그림 2> 사용자 일정 계획 서비스

사용자의 일상적인 활동들로는 '아침식사', '점심식사', '저녁식사', '집안 청소', '은행 방문', '치과 치료', 'TV 시청' 등을 예로 들 수 있다. 또한

<표 3> 사용자 활동 일정의 시간 제약 조건의 예

일정	제약 조건
아침식사	08:00 ≤ 아침식사 ≤ 09:00 (duration : 30 min)
점심식사	12:30 ≤ 점심식사 ≤ 14:00 (duration : 60 min)
저녁식사	19:00 ≤ 저녁식사 ≤ 20:30 (duration : 60 min)
은행방문	09:30 ≤ 은행방문 ≤ 16:30 (duration : 30 min)
치과치료	09:00 ≤ 치과치료 ≤ 20:00 (duration : 60 min)
TV시청	20:25 ≤ 일일드라마 ≤ 20:25 + duration (duration : 35 min) 22:00 ≤ 드라마 ≤ 22:00 + duration (duration : 55 min)
이동	집-치과 사이 이동 (duration : 20 min) 집-은행 사이 이동 (duration : 30 min) 치과-은행 사이 이동 (duration : 15 min)

이러한 사용자의 활동은 ‘식사 중 심부름’, ‘사용자 일정 알림’, 그리고 ‘TV 프로그램 녹화’와 같은 로봇 작업(robot task)과 시간적으로 매우 밀접한 연관성을 가지고 있다. 이 예제 시나리오에 따른 사용자의 활동 일정과 로봇 작업에 대한 세부적인 시간 제약 조건들을 정리해보면 표 3, 표 4와 같다.

<표 4> 로봇 작업의 시간 제약 조건의 예

일정	제약 조건
물심부름	$\text{아침식사}_{\text{start_time}} \leq \text{물심부름} \leq \text{아침식사}_{\text{end_time}}$
일정알림	$\text{물심부름}_{\text{end_time}} \leq \text{일정알림} \leq \text{아침식사}_{\text{end_time}}$ (duration : 10 min)
TV 프로그램 녹화	$20:25 \leq \text{일일드라마 녹화} \leq 20:25 + \text{duration}$ (duration : 35 min) $22:00 \leq \text{드라마 녹화} \leq 22:00 + \text{duration}$ (duration : 55 min)

로봇이 이와 같은 시간 제약 조건들을 만족하는 사용자 활동 계획과 자신의 작업 계획을 수립하기 위해서는, 단순히 사용자 활동과 자신의 단위 작업들 간의 논리적 인과관계(causal relationship)만을 분석하는 것으로는 충분치 못하다. 각 작업 및 활동의 소요시간, 특정 활동의 시작시간 및 종료시간 등에 관한 다양한 시간 제약 조건을 분석하는 시간적 추론이 계획수립 과정에 반드시 포함되어야 한다. 또, 계획 결과도 전통적인 인공지능 계획에서처럼 단순히 작업 및 활동들의 순차적인 실행순서로만 표현되는 것이 아니라, 각 작업의 서로 다른 시작시간과 소요시간, 그리고 상호 병행 수행 가능성 등이 함께 표현될 수 있어야 한다.

4. 시간 추론 계획문제의 표현

시간 추론 계획기인 TPlan을 이용해 앞서 소개한 가정용 서비스 로봇 영역에서 요구되는 시간 추론 서비스를 제공하기 위해서는 표준 계획 영역 정의 언어인 PDDL2.2로 주어진 문제의 시간 제약 정보를 표현할 수 있어야 한다. 이를 위하여 시간 제약 유형을 정리하고, 이러한 유형의 시간 제약을 계획 영역 정의 언어인 PDDL로 표현하는 방법에 대하여 설명한다. 대표적인 시간 제약 유형들로는 크게 (1) 동작의 소요시간

(duration time)과 (2) 동작의 최소 및 최대 시작 시간(earliest/latest start time), 그리고 (3) 동작의 최소 및 최대 종료시간(earliest/latest end time) 등이 있다.

4.1 동작의 소요시간

하나의 동작이 실행을 시작해서 완료될 때까지 소요되는 예상시간을 일반적으로 그 동작의 소요시간(duration time)이라고 부른다. 사용자의 활동이나 서비스 로봇의 동작은 어떤 경우에도 일정한 소요시간이 요구되는 경우와 다른 소요자원(resource)들에 비례 혹은 반비례해서 소요시간이 결정되는 경우, 이렇게 크게 두 가지 경우로 나눌 수 있다. PDDL에서는 전자의 경우 동작의 소요시간을 하나의 숫자 상수(numeric constant)로 표현하고, 후자의 경우 하나의 함수(function)를 이용하여 표현할 수 있도록 허용하고 있다.

표 5는 앞서 소개한 표 3, 표 4의 예제 시나리오에서 주어졌던 시간 제약들 중 고정된 소요시간을 가지는 한 동작을 표현한 예이다. 하지만 이와 같은 표현법은 각 동작의 유형별로 동일한 소요시간을 가지는 경우에는 매우 효과적인 표현법이나 그렇지 못한 경우 단지 소요시간 때문에 같은 유형의 동작들을 각기 독립적으로 정의해야 할 필요가 있다. 즉, 비록 유사한 동작들이라 하더라도 동작의 주체나 대상, 장소 등에 따라 각 동작의 소요시간이 조금씩 달라질 수 있다면, 이들을 하나의 동작 스키마(action schema) 혹은 연산자(operator)로 표현하지 못하고 각기 다른 개별 동작(action)으로 표현해주어야 한다는 문제점이 있다. 이런 경우, 술어논리(predicate logic)와 동작 파라미터(parameter)가 가지는 표현의

<표 5> 수치 상수를 이용한 고정 소요시간 표현 예

```
(:durative-action user_move_from_home_to_bank
:parameters (?U - user)
:duration (= ?duration 30)
:condition (and
(at start (user_in_home ?U))
(over all (accessible_from_home_to_bank)))
:effect (and
(at end (user_in_bank ?U))
(at end (not (user_in_home ?U)))
)
)
```

간결성과 함축성을 제대로 이용하지 못하고 사실상 명제논리(propositional logic) 수준으로 각 동작을 표현함으로써 사용자가 직접 매우 많은 수의 유사한 동작들을 모두 기술해야하는 불편함이 있다.

이러한 문제를 해결할 수 있는 한 가지 방법으로서 동작의 소요시간을 하나의 함수나 함수식으로 표현하는 방법이 있다. 함수는 일종의 수치변수(numeric variable)로서, 이것은 참/거짓 값을 가지는 술어 논식만으로도 상태를 표현하던 종래의 상태 표현 방식에서 벗어나 상태 변화에 따라 다양한 값을 가질 수 있는 상태 변수를 상태 표현에 도입한 결과이다. 변량(fluent)로도 불리는 함수는 특히 동작에 따른 소요시간이나 자원의 양을 표현하기 위한 목적으로 사용되며, 따라서 현재는 수치값만을 가질 수 있는 것으로 PDDL에서 제한하고 있다. 이러한 함수를 이용하여 동작의 소요시간을 표현하면 몇 가지 장점을 얻을 수 있다. 먼저, 동작과 연관된 다양한 자원들- 예컨대, 거리, 속도, 연료 및 배터리의 소모량 등 - 이 해당 동작의 소요시간에 미치는 영향을 하나의 함수나 함수식으로 기술할 수 있으므로, 가변적인 동작 소요시간을 보다 세밀하고 정확하게 표현할 수 있다. 그뿐만 아니라 함수가 가지는 표현의 함축성을 이용함으로써 조금씩 다른 소요시간을 가지는 개별 동작들을 독립적으로 정의하기보다는 이들을 하나의 동작 스키마나 연산자로 묶어서 표현할 수 있다.

표 6은 앞서 소개한 표 3, 표 4의 소요시간 제약들을 함수식을 포함한 하나의 연산자(operator)로 표현한 예이다. 이 예에서 사용자 이동 동작의 소요시간을 이동해야 하는 서로 다른 위치들 간의 거리(distance ?L1 ?L2)를 일정한 이동 속도로 나누는 함수식으로 표현하고 있다. 그리고 서로 다른 위치들 간의 거리는 계획 문제(problem)의 초기 상태(initial state) 정의부에 표 7과 같이 함수 distance에 대한 서로 다른 초기치 설정문들을 묶으로써 표현 가능하다. 이렇게 함으로써 소요시간이 서로 다른 6개의 개별 이동 동작들을 모두 따로 정의해야 하는 불필요한 노력을 줄일 수 있다.

<표 6> 함수를 이용한 가변 소요시간 표현 예

```
(:durative-action user_move
:parameters (?U - user ?L1 - location ?L2 - location)
:duration (= ?duration (/ (distance ?L1 ?L2) 10))
:condition (and
              (at start (user_in ?U ?L1))
              (over all (accessible ?L1 ?L2))
            )
:effect (and
          (at end (user_in ?U ?L2))
          (at end (not (user_in ?U ?L1)))
        )
)
```

<표 7> 함수 초기치 설정 예

```
(= (distance home dental_office1) 200)
(= (distance dental_office1 home) 200)
(= (distance home bank1) 300)
(= (distance bank1 home) 300)
(= (distance dental_office1 bank1) 150)
(= (distance bank1 dental_office1) 150)
```

4.2 동작의 최소 시작시간

사용자의 활동이나 서비스 로봇의 동작을 시작할 수 있는 가장 이른 시간을 그 동작의 최소 시작시간(Earliest Start Time, EST)이라 부르며, 이러한 최소 시작시간 제약을 효과적으로 표현할 수 있어야 다양한 시간 추론이 가능하다. 동작 a의 최소 시작시간 EST(a)가 t인 경우, 즉 $EST(a) = t$, 이것은 동작 a의 실제 실행 시작시간은 $t+a$ ($a \geq 0$)임을 의미한다. 시간 추론 계획인 TPlan의 표준 입력언어인 PDDL 2.2에서는 특정 동작의 최소 시작시간 제약을 그 동작의 시간 전조건(temporal preconditions)과 시간대별 초기화 사실(timed initial literals)을 이용하여 표현 가능하다.

표 8은 표 3, 표 4의 예제 시나리오의 일정에 대한 제약 조건들 중에서 ‘아침식사’ 동작의 최소 시작시간에 대한 제약을 시간 전조건과 시간대별 초기화 사실을 이용하여 표현한 예를 보여주고 있다. ‘아침식사’ 동작을 위한 전조건들 중 식사준비 완료사건을 (at start (prepared_breakfast_for ?U ?R))와 같이 시간 전조건 형태로 표현한 다음, 계획문제의 초기상태 정의부에 (at 480 (prepared_breakfast_for Tom Dining_Room))와 같이 시간대별 초기화 사실을 이용하여 특정 시

간에 해당 전조건을 만족하도록 선언함으로써, 실제 ‘아침식사’ 동작은 그 시간이후 해당 전조건을 포함한 모든 전조건들이 만족되는 어떤 시간에야 실행 가능해진다. 따라서 특정 동작의 최소 시작시간을 계획문제의 정의부에 표현하기 위한 방법을 요약하면, 그 동작의 실행 전조건에서 요구하는 사건을 최소 시작시간에 맞추어 만족시킬 수 있는 시간대별 초기화 사실을 선언하면 된다.

<표 8> 동작의 최소 시작시간 표현 예

```
(:durative-action have_breakfast
:parameters (?U - user ?R - room)
:duration (= ?duration (have_breakfast_duration_time))
:condition (and
  (at start (prepared_breakfast_for ?U ?R))
  (at start (hungry ?U))
  (over all (user_in ?U ?R))
)
:effect (and
  (at start (having_breakfast ?U ?R))
  (at end (not (having_breakfast ?U ?R)))
  (at end (done_breakfast ?U))
  (at end (not (hungry ?U)))
  (at end (not (prepared_breakfast_for ?U ?R)))
)
...
(at 480 (prepared_breakfast_for Tom Dining_Room))
...)
```

4.3 동작의 최대 종료시간

사용자의 활동이나 서비스 로봇의 동작을 종료할 수 있는 가장 늦은 시간을 그 동작의 최대 종료시간(Latest End Time, LET) 제약이라 부른다. 동작 a의 최대 종료시간 LET(a)가 t인 경우, 즉 LET(a) = t, 이것은 동작 a의 실제 실행 종료시간은 시간 t-a (a ≥ 0)임을 의미한다. 시간 추론 계획기인 TPlan의 표준 입력언어에서는 특정 동작의 최대 종료시간 제약을 그 동작의 시간 효과(temporal effects)와 시간제한 목표(goals with deadline)를 이용하여 표현 가능하다.

표 8과 표 9는 표 3과 표 4의 예제 제약 조건들 중에서 ‘아침식사’ 동작의 최대 종료시간 제약을 나타내기 위한 시간 효과 (at end (done_breakfast ?U))와 시간제한 목표 (done_breakfast Tom) - 540를 보여주고 있다. 여기서 (done_breakfast ?U)는 아침식사 완료상태를 나타내는 ‘아침식사’ 동작의 주된 효과중 하나이다.

특정 동작의 최대 종료시간을 계획문제의 정의부에 표현하기 위한 방법을 요약하면, 그 동작의 주된 실행 효과를 계획문제의 목표에 포함시키되, 그 목표의 시간제한을 그 동작의 최대 완료시간에 맞추어 설정하면 된다.

<표 9> 동작의 최대 종료시간 표현 예

```
(:goal
  (and
    (done_breakfast Tom) - 540
    (done_lunch Tom) - 840
    (done_dinner Tom) - 1230
    ...
  )
)
```

4.4 동작의 최대 시작시간

사용자의 활동이나 서비스 로봇의 동작을 시작할 수 있는 가장 늦은 시간을 그 동작의 최대 시작시간(Latest Start Time, LST) 제약이라 부른다. 동작 a의 최대 종료시간 LST(a)가 t인 경우, 즉 LST(a) = t, 이것은 동작 a의 실제 실행 시작시간은 시간 t-a (a ≥ 0)임을 의미한다. TPlan의 표준 입력언어에서 특정 동작의 최대 시작시간(LST) 제약은 앞서 소개한 방법에 따라 그 동작의 최대 종료시간(Latest End Time, LET)과 그 동작의 소요시간(duration time)을 함께 제약함으로써 간접적으로 표현 가능하다. 일반적으로 동작 a의 최대 종료시간 LET(a)과 소요시간 Duration(a)이 함께 주어지면, (식 1)의 관계식에 의해 그 동작의 최대 시작시간 LST(a)는 저절로 결정되기 때문이다.

$$LST(a) = LET(a) - Duration(a) \quad (\text{식1})$$

예컨대, ‘아침식사’ 동작과 관련하여 표 8과 표 9에서처럼 동작의 최대 종료시간을 540으로 정해 주고, 소요시간을 (= (have_breakfast_duration_time) 30)과 같이 30으로 정해 주면, 이와 같은 시간제약들을 만족하기 위해서는 늦어도 시간 510 이전에는 ‘아침식사’ 동작의 수행을 시작해야 한다는 최대 시작시간 제약을 암시하게 된다.

4.5 동작의 최소 종료시간

사용자의 활동이나 서비스 로봇의 동작을 종료할 수 있는 가장 이른 시간을 그 동작의 최소 종료시간(Earliest End Time, EET) 제약이라 부른다. 동작 a의 최소 종료시간 EET(a)가 t인 경우, 즉 $EET(a) = t$, 이것은 동작 a의 실제 실행 종료시간은 시간 $t+a$ ($a \geq 0$)임을 의미한다. TPlan의 표준 입력언어에서 특정 동작의 최소 종료시간(EET) 제약은 앞서 소개한 방법에 따라 그 동작의 최소 시작시간(Earliest Start Time, EST)과 그 동작의 소요시간(duration time)을 함께 제약함으로써 간접적으로 표현 가능하다. 일반적으로 동작 a의 최소 시작시간 EST(a)과 소요시간 Duration(a)이 함께 주어지면, (식 2)의 관계식에 의해 그 동작의 최소 종료시간 EET(a)는 저절로 결정되기 때문이다.

$$EET(a) = EST(a) + Duration(a) \quad (\text{식2})$$

예컨대, ‘아침식사’ 동작과 관련하여 표 8에서 처럼 시간대별 초기화 사실을 이용해 이 동작의 최소 시작시간을 480으로 정해주고 이 동작의 소요시간을 30으로 정해주면, 이와 같은 시간제약들을 만족하기 위해서는 빨라야 시간 510 이후에 ‘아침식사’ 동작의 실행이 완료된다는 최소 종료시간 제약을 암시하게 된다.

4.6 동작의 특정 시작시간

사용자의 특정 활동이나 서비스 로봇의 특정 동작이 정확히 특정 시간에 맞추어 실행 시작하여야 하는 경우, 이것을 동작의 특정 시작시간(Start Time, ST) 제약이라 부른다. 동작 a의 특정 시작시간 ST(a)가 t인 경우, 즉 $ST(a) = t$, 이것은 동작 a의 실제 실행이 정확히 시간 t에 시작됨을 의미한다. TPlan의 표준 입력언어에서 특정 동작의 특정 시작시간(ST) 제약은 앞서 소개한 방법에 따라 그 동작의 최소 시작시간(Earliest Start Time, EST), 소요시간(duration time), 그리고 최대 종료시간(Latest End Time, LET) 이 세 가지를 모두 제약함으로써 간접적으로 표현 가능하다. 방법은 다음과 같다. 먼저 동작의 최소 시작시간 EST(a)을 ST(a)로 설정해준다. 즉, $EST(a) = ST(a) = t$ 로 설정해준다.

이어서 그 동작의 소요시간 Duration(a)을 설정해준다. 마지막으로, (식 3)에 따라 이 동작의 최대 종료시간 LET(a)를 계산하고, 이 시간에 맞추어 최대 종료시간을 제약하면 된다. 결과적으로 (식 3)은 이 동작이 시간 $ST(a)=t$ 에 실행을 시작해서 추가적인 지연시간 없이 정확히 소요시간 Duration(a) 후에 실행을 완료하도록 동작의 최대 종료시간을 제약한다.

$$LET(a) = EST(a) + Duration(a) \quad (\text{식3})$$

예컨대, ‘아침식사’ 동작과 관련하여 이 동작의 실행을 특정 시간인 500에 정확히 시작해야 한다면, 시간대별 초기화 사실을 이용해 이 동작의 최소 시작시간을 500으로 정해주고 이 동작의 소요시간도 30으로 정해준 다음, (식 3)에 따라 이 동작의 최대 종료시간을 $500 + 30 = 530$ 으로 설정해주면 된다. 그러면 최대 종료시간인 530까지 이 동작을 완료하기 위해서는 최소 시작시간인 500 이후에 시작할 여유없이 500에 곧바로 실행을 시작하여야 한다는 것을 암시하게 된다.

5. 시간 추론 서비스 에이전트

이 절에서는 시간 추론 계획기 TPlan을 중심으로 하나의 시간 추론 서비스 에이전트를 구현하는 방법에 대해 설명한다. 서비스 에이전트(service agent)란 자신만의 독자적인 내부 모델(model)을 소유하고 있으면서, 다른 에이전트들의 요청에 응답하여 모델에 기초한 유용한 서비스(service)들을 제공할 수 있는 능동적 소프트웨어 단위를 의미한다. 일반적으로 이러한 단위 서비스 에이전트들의 다양한 결합과 이들 간의 상호작용을 이용함으로써 보다 큰 새로운 응용 서비스를 제공할 수 있다[3]. 로봇 지능 구조를 이와 같은 단위 서비스 에이전트들로 구성된 하나의 멀티 에이전트 시스템으로 구현하면, 로봇 지능 구조의 유연한 확장성과 적응성, 그리고 지능 모듈의 재사용성을 보다 향상시킬 수 있는 장점이 있다. 본 연구에서는 이와 같은 서비스 에이전트의 장점을 이용하기 위해, 시간 추론 계획기 TPlan을 추론엔진으로 삼고 사용자와 로봇 활동

일정을 내부 모델로 삼아 여러 가지 시간 추론 서비스들을 제공하는 하나의 시간 추론 서비스 에이전트를 구현하였다.

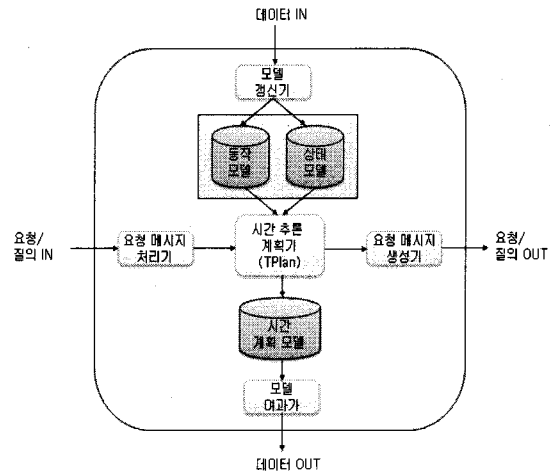
하나의 서비스 에이전트를 구현하기 위해서는 공통의 서비스 에이전트 개발 프레임워크에 따라 에이전트의 내부 모델과 입출력 메시지를 구성하여야 한다. 서비스 에이전트 개발 프레임워크에서 제공하는 서비스 에이전트의 공통 인터페이스는 Data IN, Data OUT, Request/Query IN, Request/Query OUT으로 구성된다. Data IN은 서비스 에이전트로 입력 가능한 데이터 메시지들을 의미하고, 반면에 Data OUT은 서비스 에이전트가 다른 에이전트들에게 제공 가능한 출력용 데이터 메시지들을 의미한다. 한편, Request/Query IN은 구현하려는 서비스 에이전트가 받아들일 수 있는 서비스 요청 및 질의 메시지들을 의미하며, Request/Query OUT은 거꾸로 서비스 에이전트가 다른 에이전트들에게 보낼 수 있는 서비스 요청 및 질의 메시지들을 나타낸다. 따라서 하나의 서비스 에이전트를 구현하기 위해서는 해당 에이전트의 공통 인터페이스인 Data IN, OUT, Request/Query IN, OUT을 먼저 정의하여야 한다.

본 연구에서 개발된 시간 추론 서비스 에이전트는 동작 모델(Action Model), 상태 모델(State Model), 시간 계획 모델(Temporal Plan Model) 등의 내부 모델과 요청 메시지 처리기(Request Processor), 요청 메시지 생성기(Request Generator), 시간 추론 계획기(Temporal Planner), 모델 갱신기(Model Updater), 모델 여과기((Model Filter) 등의 처리기들로 구성된다. 그림 3은 시간 추론 서비스 에이전트의 내부 구조를 나타낸다. 각 구성 요소의 역할은 다음과 같다.

- 동작 모델(Action Model)과 상태 모델(State Model)

시간 추론 서비스 에이전트는 사용자의 각 활동과 이를 지원하기 위한 로봇 스스로의 단위 작업 동작에 대한 동작 모델을 내부에 유지한다. 또한, 현재의 날씨 정보, 교통 정보 등의 환경 정보와 이미 예정된 활동이나 작업들의 일정 정보를 포함하는 상태 모델도 유지한다. 표 3과 표 4와 같은 다양한 시간 제약 조건들(temporal

constraints)은 앞서 소개한 방식대로 이 두 모델의 결합으로 표현 가능하다.



<그림 3> 시간 추론 서비스 에이전트의 내부 구조

이러한 동작 모델과 상태 모델은 그 자체만으로도 서비스로 제공 가능한 정보가 될 수 있으나, 주로는 에이전트내의 다른 구성요소인 시간 추론 계획기에 입력을 제공하여 또 다른 유용한 내부 모델인 시간 계획 모델을 새로이 생성하거나 갱신하는데 이용된다.

- 요청 메시지 처리기(Request Processor)와 요청 메시지 생성기(Request Generator)

다른 에이전트가 시간 추론 서비스 에이전트에게 서비스 요청 메시지를 보내오면(Request/Query IN), 요청 메시지 처리기는 이 메시지의 내용을 분석하고 요청된 서비스를 수행하기 위한 일련의 내부 처리를 시작시키는 역할을 수행한다. 시간 계획 모델을 포함한 현재 내부 모델의 정보로도 요청된 서비스를 충분히 완성할 수 있을 경우에는, 내부 모델 여과기(Model Filter)의 도움을 받아 필요한 정보만을 추출한 다음, 응답 메시지에 담아 서비스를 요청한 에이전트에게 보내준다(Data OUT). 만약 요청된 서비스를 완성하기 위해서는 새로운 시간 계획 모델의 갱신이 필요하다면, 새로운 목표 제시와 함께 시간 추론 계획기를 호출한다. 이때 시간 추론 계획기에 입력을 제공할 현재의 상태 모델과 동작 모델에 대한 갱신도 함께 필요하다면, 요청 메시지 생성기를 통해 모델 갱신에 필요한 정보를 제공해 줄 수 있

는 다른 에이전트에게 서비스 요청 메시지를 보낼 수도 있다(Request/Query OUT).

- 시간 추론 계획기(Temporal Planner)

시간 추론 계획기는 시간 추론 서비스 에이전트의 핵심 엔진 역할을 수행하는 구성요소로서, 앞서 소개한 TPlan을 이용한다. 시간 추론 계획기가 호출되면, 다양한 시간 제약들이 포함된 동작 모델과 상태 모델을 기초로 목표에 도달 가능한 최적의 시간 계획을 수립하고, 이것으로 현재의 시간 계획 모델을 갱신한다. 그리고 이때 시간 추론 계획기에 주어지는 목표는 요청되는 서비스의 유형에 따라 내부적으로 달리 구성된다.

- 시간 계획 모델(Temporal Plan Model)

시간 계획 모델은 동작 모델과 상태 모델에 포함된 현재의 시간 제약을 모두 만족할 수 있는 사용자 활동과 로봇 작업 동작에 관한 하나의 시간 계획을 담고 있다. 시간 계획 모델은 시간 추론 계획기 호출에 의해 새로 갱신되기 전까지는 현재의 시간 계획을 유지하며, 모델 여과기는 이러한 시간 계획 모델에 기초해 다른 에이전트로부터의 질의에 답할 정보를 추출한다.

- 모델 갱신기(Model Updater)와 모델 여과기(Model Filter)

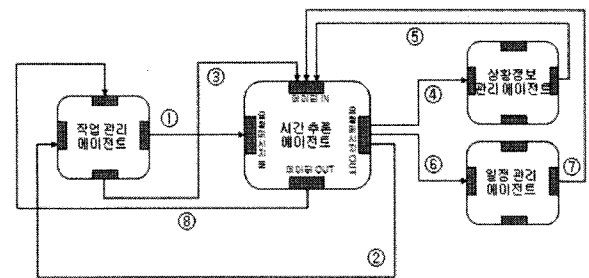
모델 갱신기는 시간 추론 서비스 에이전트가 다른 에이전트에게 보낸 요청이나 질의에 대한 결과로 받은 새로운 상태 정보나 추가 동작(Data IN)을 현재의 상태 모델과 동작 모델에 반영하는 역할을 수행한다. 모델 여과기는 다른 에이전트가 보내온 요청과 질의에 적합한 정보를 내부 모델로부터 추출하여 응답 메시지에 담아 보내주는 역할을 한다(Data OUT).

현재 구현된 시간 추론 서비스 에이전트는 특정한 날의 사용자의 하루 활동 일정이나 로봇의 작업 일정을 구해주는 서비스, 특정 활동이나 작업이 적어도 어느 시간 이전 혹은 이후에 이루어져야 하는지를 알려주는 서비스, 특정 활동이나 작업 이전 혹은 이후에 이루어져야 할 작업들은 무엇이 있는지를 알려주는 서비스 등을 제공하고 있다. 하지만 앞서 소개한 개방적 방식으로 설계된 시간 추론 서비스 에이전트는 이들 현재의 서

비스에 국한되지 않고, 응용 목적과 필요에 따라 보다 더 다양한 새로운 시간 추론 서비스들을 쉽게 구현하여 제공할 수 있다.

6. 응용 서비스

본 연구에서는 앞서 소개한 시간 추론 서비스 에이전트를 다른 에이전트들과 유기적으로 결합함으로써 다양한 시간 추론 응용 서비스를 효과적으로 제공할 수 있음을 보이기 위해, 하나의 응용 서비스를 구현하였다. 이 응용 서비스는 사용자의 현재 상황정보와 일정정보를 참고하여 향후 수행해야 할 활동계획을 구하고 이를 사용자에게 알려져 주는 서비스이다. 이 응용 서비스는 그림 4와 같이, 시간 추론 서비스를 제공하는 시간 추론 에이전트(Temporal Reasoner)와 실시간 상황정보를 제공하는 상황정보 관리 에이전트(Context Manager), 사용자의 일정정보를 제공해주는 일정 관리 에이전트(Schedule Manager), 그리고 사용자 인터페이스(UI)를 이용하여 사용자에게 향후 활동계획을 알려주는 작업 관리 에이전트(Task Manager) 등의 상호작용으로 구현된다.



<그림 4> 응용 서비스를 위한 에이전트간의 상호작용

단위 서비스 에이전트간의 상호작용을 세부적으로 살펴보면 다음과 같다.

① 지금부터(12시) 저녁식사 전까지 User1이 해야 할 일들이 무엇인지를 알기 위해 작업 관리 에이전트(Task Manager)가 시간 추론 에이전트(Temporal Reasoner)에게 표 10과 같은 요청 메시지를 보낸다.

<표 10> 시간 추론 에이전트에 대한 계획 요청 메시지

```
(get_user_actions
  (domain (name user_scheduling))
  (user (name User1))
  (from_time (time 12:00))
  (before_action (name have_dinner))
)
```

② 시간 추론 에이전트(Temporal Reasoner)는 요청에 대한 응답을 위해 내부 계획모델을 새로 갱신할 필요가 있으며, 시간 추론 계획에 필요한 동작모델을 작업 관리 에이전트(Task Manager)에게 표 11과 같이 요청한다.

<표 11> 내부 계획모델 갱신을 위한 요청 메시지

```
(get_action_models
  (domain (name user_scheduling))
)
```

③ 시간 추론 에이전트(Temporal Reasoner)의 요청에 대해 작업 관리 에이전트(Task Manager)는 사용자 일정 계획과 연관된 동작모델을 표 12와 같은 형식으로 응답 메시지에 담아 보내준다.

<표 12> 내부 계획모델 요청에 대한 응답 메시지

```
(action_models
  (action
    (name user_move)
    (parameters (?U - user ?L1 - location
      ?L2 - location))
    (duration (/ (distance ?L1 ?L2) 10))
    (precondition (and
      (at start (user_in ?U ?L1)) ... )
    (effect (and
      (at end (user_in ?U ?L2)) ... )
    )
  )
  (action ...
  )
)
```

④ 시간 추론 에이전트(Temporal Reasoner)는 시간 추론 계획에 필요한 상황정보를 상황정보 관리 에이전트(Context Manager)에게 표 13과 같이 요청한다.

<표 13> 상황정보 요청 메시지

```
(get_current_context
  (domain (name user_scheduling))
  (user (name User1))
  (robot (ID R1))
  (door (ID BedroomDoor)(belong_to Home))
)
```

⑤ 시간 추론 에이전트(Temporal Reasoner)의 요청에 대해 상황정보 관리 에이전트(Context Manager)는 사용자 User1과 서비스 로봇 R1에 관련된 현재의 상황정보를 표 14와 같은 형식으로 응답 메시지에 담아 보내준다.

<표 14> 상황정보 요청에 대한 응답 메시지

```
(current_context
  (user (name User1)(location Living_Room)
    (sit_on Sofa1)(watching_TV TV1) ...)
  (robot (ID R1)(location Living_Room)
    (charging_on Station1) ...)
  (door (ID BedroomDoor)(belong_to Home)
    (openness Open) ...)
)
```

⑥ 시간 추론 에이전트(Temporal Reasoner)는 시간 추론 계획에 필요한 사용자 User1의 일정정보를 일정정보 관리 에이전트(Schedule Manager)에게 표 15와 같이 요청한다.

<표 15> 일정정보 요청 메시지

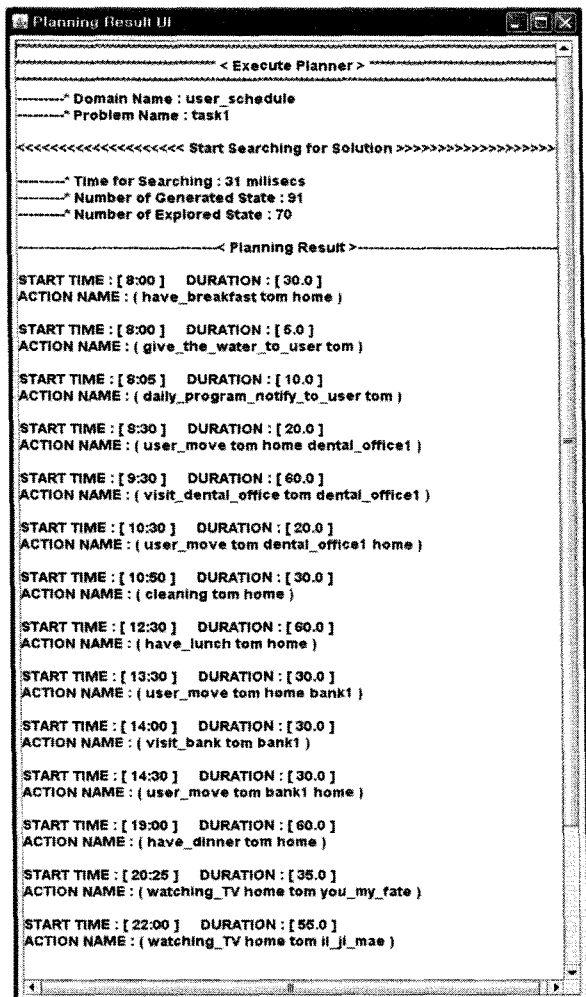
```
(get_user_schedule
  (domain (name user_scheduling))
  (user (name User1))
  (when (date Today))
)
```

⑦ 시간 추론 에이전트(Temporal Reasoner)의 요청에 대해 일정정보 관리 에이전트(Schedule Manager)는 사용자 User1의 하루 일정정보를 표 16과 같은 형식으로 응답 메시지에 담아 보내준다.

<표 16> 일정정보 요청에 대한 응답 메시지

```
(user_schedule
  (action (name have_breakfast)
    (parameters (User1 Home))
    (from_time 8:00)
    (to_time 9:00) (duration 30) )
  (action
    ...
  )
)
```

⑧ 시간 추론 에이전트(Temporal Reasoner)는 다른 에이전트들로부터 제공받은 정보를 기초로 다양한 시간 제약을 포함한 하나의 시간 추론 계획문제를 구성한다. 그리고 시간 추론 계획기 TPlan를 이용하여 새롭게 시간 추론 계획을 생성하고 내부 모델을 갱신한다. 그림 5는 시간 추론 에이전트가 생성한 시간 추론 계획의 한 예를 보여주고 있다. 시간 추론 계획은 순서화된 동작들(actions)과 이들 각각의 시작시간(start time)과 소요시간(duration)들로 표현된다.



<그림 5> 시간 추론 계획의 예

시간 추론 에이전트(Temporal Reasoner)는 이와 같은 자신의 시간 계획 모델을 이용하여 작업 관리 에이전트(Task Manager)의 요청에 대한 응답으로 표 17과 같은 결과 메시지를 보내준다.

<표 17> 계획 요청에 대한 응답 메시지

```
(user_actions
  (action
    (name have_lunch)
    (parameters (User1 Home))
    (start_time 12:30)
    (duration 30.0)
  )
  (action
    (name user_move)
    (parameters (User1 Home Bank1))
    (start_time 13:30)
    (duration 30.0)
  )
  ...
)
```

작업 관리 에이전트(Task Manager)는 시간 추론 에이전트(Temporal Reasoner)의 응답을 기초로 별도의 사용자 인터페이스를 통해 사용자에게 향후 활동계획을 알려줌으로써 최종적인 응용 서비스를 완성한다.

지금까지 가정용 서비스 로봇이 제공 가능한 전형적인 하나의 응용 서비스 개발에 대해 설명하였다. 이와 같은 응용 서비스의 개발을 통해 시간 추론 계획기인 TPlan과 이를 이용한 시간 추론 서비스 에이전트의 유용성을 확인할 수 있었다. 또한, 시간 추론 서비스 에이전트는 서로 공개된 모델과 서비스 명세를 기초로 다른 서비스 에이전트들과 유기적으로 결합함으로써, 보다 다양한 시간 추론 응용 서비스들을 쉽게 조합해낼 수 있을 것으로 판단된다.

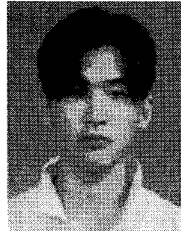
7. 결론

본 논문에서는 일반 가정환경에서 가정용 서비스 로봇에게 요구되는 전형적인 시간 추론 서비스와 시간 제약들을 살펴보고, 이들을 인공지능 표준 계획 영역 정의 언어로 표현하는 방법을 제시하였다. 특히 본 논문에서는 표준 언어인 PDDL 2.2의 표현 한계를 극복하고 각 동작의 최소/최대 시작 시간, 최소/최대 종료시간 등을 표현할 수 있는 방법을 제시하였다. 또 본 논문에서는 로봇 지능 구조의 확장성과 지능 모듈의 재사용성을 위해 영역-독립적인 인공지능 계획기를 중심으로 하나의 시간 추론 서비스 에이전트를 구현하는 방법을 소개하였다. 또한, 이 에이전트의 서비스를 다른 에이

전트의 서비스들과 결합함으로써 새로운 응용 서비스를 쉽게 제공할 수 있음을 예시하였다. 계획하고 있는 향후 연구로는 보다 다양한 시간 제약들을 직접 표현할 수 있도록 입력언어를 확장하는 일과 좀더 효율적인 시간 추론이 가능하도록 계획 알고리즘을 개선하는 일 등이 있다.

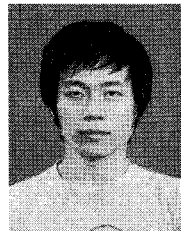
참 고 문 헌

- [1] M. E. Pollack, et al, "Autominder: An Intelligent Cognitive Orthotic System for People with Memory Impairment," Robotics and Autonomous Systems, vol. 44, pp. 273 - 282, 2003.
- [2] K. Myers, P. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe, "An Intelligent Personal Assistant for Task and Time Management," AI Magazine, 2007.
- [3] A. J. Cheyer, D. L. Martin, "The Open Agent Architecture", Journal of Autonomous Agents and Multi-Agent Systems, vol. 4, pp. 143-148, 2001.
- [4] D. McDermott. "PDDL - the Planning Domain Definition Language", AIPS'98 IPC Committee, 1998.
- [5] M. Fox, D. Long. "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains", Journal of AI research, vol. 20, pp.61-124, 2003.
- [6] S. Edelkamp, J. Hoffmann, "PDDL2.2: The Language for the Classical Part of IPC-4 — Extended Abstract —", IPC-4 Booklet, pp.2-6, 2004.
- [7] M. B. Do, S. Kambhampati, "SAPA: A Scalable Multi-Objective Metric Temporal Planner", Journal of AI Research, vol. 20, pp.155-194, 2003.
- [8] M. Fox, D. Long, "Modelling Mixed Discrete - Continuous Domains for Planning", Journal of AI research, vol. 27, pp.235-297 2006.
- [9] Y. Chen, C. Hsu, B. Wah. "Temporal Planning Using Subgoal Partitioning and Resolution in SGPlan". Journal of AI Research, vol. 26, pp.323-369 2006.
- [10] 김현식, 김인철, "시간 추론 작업 계획기", 21C 프론티어사업 지능로봇사업단 제 11차 기술교류워크숍 논문집, 2009.



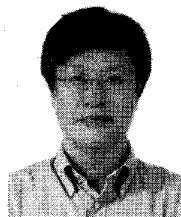
김 현 식 (Hyun-Sik Kim)

- 학생회원
- 2001년 2월 : 경기대학교 전자계산학과 (이학사)
- 2004년 2월 : 경기대학교 일반대학원 전자계산학과 (이학석사)
- 2005년 3월 ~ 현재 : 경기대학교 일반대학원 전자계산학과 박사과정
- 관심분야 : 자동계획, 시맨틱 웹 서비스, 에이전트



박 찬 영 (Chan-Young Park)

- 학생회원
- 2009년 2월 : 경기대학교 정보과학부 전자계산학전공 (이학사)
- 2009년 3월 ~ 현재 : 경기대학교 일반대학원 컴퓨터과학과 석사과정
- 관심분야 : 지능로봇, 자동계획, 기계학습, 에이전트



김 인 철 (In-Cheol Kim)

- 정회원
- 1985년 2월 : 서울대학교 수학과 (이학사)
- 1987년 2월 : 서울대학교 전산과학과 (이학석사)
- 1995년 2월 : 서울대학교 전산과학과 (이학박사)
- 1996년 3월 ~ 현재 : 경기대학교 자연과학대학 컴퓨터과학과 교수
- 관심분야 : 자동계획, 기계학습, 지능로봇, 에이전트

논문접수일 : 2009년 6월 12일
 논문수정일 : 2009년 7월 28일
 게재확정일 : 2009년 9월 20일