

JCBP : 사례 기반 계획 시스템

김인철
경기대학교 이과대학 컴퓨터학과
(kic@kyonggi.ac.kr)

김만수
경기대학교 이과대학 컴퓨터학과
(i3rew@kyonggi.ac.kr)

.....

사례 기반 계획 시스템은 과거의 유사한 사례 계획들을 이용함으로써 새로운 문제를 위한 계획을 효율적으로 생성할 수 있다. 하지만 대부분의 기존 사례 기반 계획 시스템들은 사례 검색 및 사례 일반화를 위한 제한적 기능들만을 제공할 뿐만 아니라, 계획 생성과정에 사용자의 참여를 허용하지 않는다. 본 논문에서 제안하는 JCBP 시스템은 효율적인 메모리 사용과 사례 검색을 위해 각 도메인의 동일한 작업 목표를 가진 사례들을 개별 사례 베이스로 그룹화하고, 이들에 대한 색인을 유지한다. 또 이 시스템은 문제모델로부터 자동으로 추출한 휴리스틱 지식을 사례 적응 단계에 이용하며, 목표 회귀를 통한 사례 일반화 기능도 제공한다. 또한 JCBP 시스템은 대화형 모드를 통한 혼합 주도 계획 생성 기능도 제공한다. 이와 같이 JCBP 시스템은 문제 해결을 위해 사용자의 기호와 지식을 이용함으로써 사용자의 요구를 더 잘 만족하는 해 계획을 생성할 수 있을 뿐 아니라 계획 생성의 복잡도도 줄일 수 있다.

.....

논문접수일 : 2008년 05월 논문수정일 : 2008년 11월 게재확정일 : 2008년 09월 교신저자 : 김인철

1. 서론

최근 들어 지능 로봇이나 무인 우주 탐사선의 작업제어, 컴퓨터 게임이나 가상현실에서 자율 캐릭터의 행동제어, 공장이나 사무실에서 워크플로우 프로세스의 자동생성 등 다양한 분야에 걸쳐 자동 계획 시스템의 응용이 증대되고 있다. 하지만 기존의 생성적 계획 시스템(generative planning system)은 문제 해결에 있어서 몇 가지 한계점을 지니고 있다(Ghallab, 2004). 우선 복잡도가 높은 다양한 실세계 계획 문제를 실시간으로 해결하기에는 아직도 생성적 계획 시스템들의 성능이 충분치 않다. 또한, 많은 기존의 생성적 계획 시스템들

은 매번 계획 문제가 발생할 때마다 처음부터 새로 계획을 수립해야 하기 때문에, 동일한 환경에서 빈번히 발생하는 유사한 계획 문제들에 효과적으로 대응하기 어렵다.

이러한 문제를 극복하기 위한 노력의 하나로 CHEF(Hammond, 1990), PRIAR(Kambhampati and Hendler 1992), PRODIGY/ANALOGY(Veloso et al., 1995), SPA(Hanks, 1995), PARIS(Bergmann, 1995), HICAP(Muñoz-Avila, 1999), DERSNLP(Ihrig, 1997), DIAL(Leake, 1997) 등과 같은 다양한 사례 기반 계획 시스템(case-based planning system)들이 제안되었다. 일반적으로 사례 기반 계획 시스템은 새로운 계획 문제의 해결을 위해

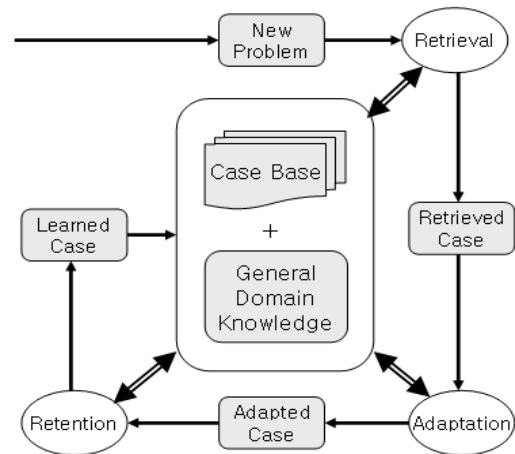
* 이 연구(논문)는 산업자원부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술 개발사업)의 일환으로 수행되었습니다.

유사한 과거의 사례 계획(case plan)들을 재사용한다. 사례 기반 계획 시스템은 생성적 계획 시스템과는 달리 매번 처음부터 계획 수립 전체 과정을 되풀이 하지 않아도 되기 때문에, 계획 수립에 드는 노력을 줄일 수 있고 실시간 요구에 빠르게 대응할 수 있다(Hammond, 1898). 하지만 기존의 사례 기반 계획 시스템은 다음과 같은 몇 가지 제한점들을 가지고 있다. 첫 번째, 몇몇 영역 의존적인 시스템들을 제외하고는 많은 영역 독립적인 사례 기반 계획 시스템들에서는 시스템의 성능에 큰 영향을 미치는 사례 분류와 색인 메커니즘의 효율성이 높지 않다. 두 번째, 인공지능 자동 계획 커뮤니티에서 개발된 많은 사례 기반 계획 시스템들은 관계형 데이터베이스 시스템에 대한 연동과 같은 대용량의 사례 베이스 저장을 위한 지원 기능이 미흡하다. 세 번째, 그동안 다양한 사례 일반화 방법들이 기존 시스템들에서 시도되어 왔으나 사례 계획의 적용 전 조건에 대한 일반화는 충분히 연구되지 못하였다. 마지막으로, 기존 시스템들의 가장 중요한 제한점은 사례 기반 계획 수립 과정에 사용자 참여 지원 기능이 없다는 점이다. 따라서 계획 수립 동안에 사용자의 선호도와 지식이 반영되지 못해 결과로 얻어진 해 계획들이 사용자의 요구를 충분히 만족하기 어렵다. 본 논문에서는 이러한 기존의 사례 기반 계획 시스템들이 가진 제한점들을 고려하여 계획 수립의 효율성을 높일 수 있는 새로운 사례 기반 계획 시스템인 JCBP 시스템의 설계와 구현에 대해 소개한다. 본 논문의 제 2장에서는 일반적인 사례 기반 계획의 순환구조와 기초 개념들을 소개하고, 제 3장에서는 JCBP 시스템의 전체 구조와 각 단계별 설계사항들을 설명한다. 제 4장에서는 사용자 참여를 지원하는 대화형 모드와 연관된 JCBP 시스템의 구현에 대해 설명하고, 제 5장에서는 JCBP 시스템의 성능 분석을

위한 실험 결과를 소개하며, 제 6장에서는 결론을 정리한다.

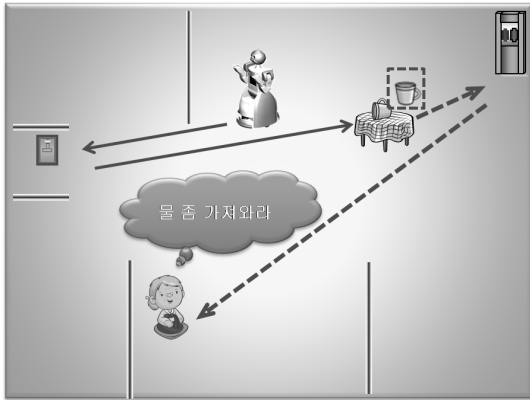
2. 사례 기반 계획

사례 기반 계획(case-based planning)은 자동 계획 수립을 위한 사례 기반 추론(case-based reasoning)(Kolodner, 1993)(Althoff, 1996)(Bergmann, 1998)의 한 응용으로 볼 수 있다. 따라서 사례 기반 계획은 일반적인 사례 기반 추론 방식에서와 같이 사례 검색, 사례 적용 및 재사용 그리고 사례 저장의 단계를 거치며 계획을 수립한다(Spalzzi, 2001). 사례 검색은 해결해야 할 새로운 계획 문제와 가장 유사한 과거의 사례 문제와 계획을 검색하는 단계이다. 사례 적용 및 재사용은 검색된 과거의 사례 계획을 새로운 계획 문제에 적합하게 수정한 뒤 이용하는 단계이다. 사례 저장은 해결된 새로운 계획 문제와 결과 계획을 향후 재사용하기 위해 일반화 등의 처리를 거친 다음 하나의 사례로 저장하는 단계이다.



<그림 1> 사례 기반 계획의 순환 구조

따라서 일반적인 사례 기반 계획 시스템들은 <그림 1>과 같은 사례 기반 계획 순환 구조(case-based planning cycle)를 따른다. <그림 1>의 순환 구조는 새로운 계획 문제가 주어질 때 그것을 해결하기 위해 사례 기반 계획시스템이 수행하는 3 가지 큰 실행 단계들로 구성된다. 첫 번째 단계는 과거 사례들의 집합인 사례 베이스로부터 새로운 계획 문제 해결을 위해 가장 유사한 과거의 사례를 검색하는 사례 검색(case retrieval) 단계이다. 순환구조의 다음 단계는 검색된 사례를 새로운 문제에 맞는 해 계획으로 수정, 변형하는 사례 적응(case adaptation) 단계이다. 순환구조의 마지막 단계는 앞의 두 단계를 통해 해결된 새로운 계획 문제와 해 계획을 새로운 사례로 저장하는 사례 저장(case retention) 단계이다. 일반적으로 새로운 사례로 저장하기 전에 보다 효율적인 사례 저장과 관리를 위해 사례 일반화 과정을 거친다.



<그림 2> 사례 기반 계획 문제의 예

다음은 사례 기반 계획 문제의 한 예를 소개한다. <그림 2>는 가정환경에서 인간과 함께 생활하며 가사 일을 도와주는 서비스 로봇의 한 작업계획의 예를 나타내고 있다. 이 예는 사용자가 서비

스 로봇에게 마실 물을 가져오기를 요구하자 로봇이 과거 사례 중 하나를 검색하여 이것을 확장함으로써 현재 상황에 맞는 작업계획을 수립하여 실행하는 것을 나타낸다. 과거 사례 중에는 실내가 밝을 때에 로봇이 탁자에 놓인 컵을 가지고 정수기의 물을 담아 사용자에게 성공적으로 가져다 준 경우가 있고, 현재의 상황은 그때와 달리 실내가 어두워 동일한 방식으로 서비스하기 위해서는 먼저 조명 스위치가 있는 곳으로 이동하여 조명 스위치를 켜 다음 탁자의 컵을 가지고 물을 담아 배달하는 것이 차이점이라고 가정한다. <그림 2>에서 점선은 과거의 사례 계획을 나타내며, (pickup robot tablearea cup table) → (carry robot cup tablearea purifierarea) → (fillWater robot cup purifier purifierarea) → (carry robot cup purifierarea userarea) → (give robot user cup userarea)과 같은 동작들의 시퀀스로 구성된다. 반면에, <그림 2>에서 실선은 사례 적응 단계를 통해 사례 계획에 추가된 동작부분들을 나타낸다. 따라서 실선 부분에 이어 점선 부분을 연결하면 새로운 계획 문제에 대한 해 계획(solution plan)이 된다. 해 계획은 (moveToCB robot circuitbreaker tablearea circuitbreakerarea) → (switchOn robot circuitbreaker circuitbreakerarea) → (automove robot circuitbreakerarea tablearea) → (pickup robot tablearea cup table) → (carry robot cup tablearea purifierarea) → (fillWater robot cup purifier purifierarea) → (carry robot cup purifierarea userarea) → (give robot user cup userarea)과 같은 동작들의 시퀀스로 구성된다.

본 논문에서 가정하는 사례 기반 계획 문제에 관한 용어와 개념들을 다음과 같이 정의할 수 있다.

- 계획 도메인(planning domain)

계획 도메인 D 는 (C, A) 로 구성되며, C 는 논리 조건(condition)들의 집합이고, A 는 동작들의 집합이다.

• **동작(action)**

동작 $a \in A$ 는 (C_a, E_a) 의 쌍으로 이루어지며, C_a 는 동작이 수행되기 위해 만족되어야 하는 전 조건(preconditions)들의 집합이고, E_a 는 그 동작의 효과(effect)를 나타내는 후 조건(postconditions)들의 집합이다.

본 논문에서는 자동 계획 수립을 위한 도메인 명세는 술어 논리 조건(predicate logic condition)들의 집합과 이들을 이용해 정의되는 동작들의 집합으로 주어진다고 가정한다. 그리고 각 동작들은 표준 계획영역기술언어인 PDDL(Planning Domain Description Language)(McDermott, 1998) 2.1에 따라 전 조건과 효과로 표현된다고 가정한다. 예컨대, <그림 2>의 예에서 한 동작인 automove는 술어 논리 조건들을 이용하여 아래와 같이 정의된다고 가정한다.

```
( : action automove
  : parameters (?R - robot ?R1 - room ?R2 - room)
  : precondition (and (robot_in_room ?R ?R1)
                     (not (= ?R1 ?R2))(connect ?R1 ?R2))
  : effect (and (not (robot_in_room ?R ?R1))
               (robot_in_room ?R ?R2))
)
```

• **계획문제(planning problem)**

계획 문제 P 는 (D, I, G) 로 표현하며 D 는 계획 도메인, I 는 초기 상태(initial state)를 나타내는 논리 조건들의 집합, G 는 목표 조건(goal conditions)들의 집합을 나타낸다. I 와 G 모두 도메인에서 정

의한 술어 논리 조건(predicate logic condition)들로 표현한다.

<그림 2>의 예에서, 새로운 한 계획 문제는 논리 조건들의 집합과 동작들의 정의를 포함하는 도메인 D 를 기초로, 초기 조건들의 집합인 $I = \{(on\ cup\ table), (emptywater\ cup), \dots, (dark)\}$ 와 목표 조건들의 집합인 $G = \{(has\ user\ cup)\}$ 로 정의된다.

• **사례 베이스(case base)**

사례 베이스 CB 는 집합 $\{z \mid z = (P, S)\}$ 로 구성되며, 이때 P 는 한 계획 문제, S 는 P 에 대한 하나의 해 계획(solution plan)을 나타낸다.

<그림 2>의 예에서, 과거의 사례들 중 하나인 z 는 과거에 푼 계획 문제인 $P' = (D, I', G')$ 와 해 계획인 $S' = ((pickup\ robot\ tablearea\ cup\ table), (carry\ robot\ cup\ tablearea\ purifierarea), (fill-Water\ robot\ cup\ purifier\ purifierarea), (carry\ robot\ cup\ purifierarea\ userarea), (give\ robot\ user\ cup\ userarea))$ 으로 구성된다. 이때, 과거 계획 문제의 초기 조건들의 집합과 목표 조건들의 집합은 각각 $I' = \{(on\ cup\ table), (emptywater\ cup), \dots, (bright)\}$ 와 $G' = \{(has\ user\ cup)\}$ 로 정의된다.

• **사례 기반 계획 문제(case-based planning problem)**

사례 기반 계획 문제 CP 는 (CB, P_{new}) 로 구성되며, 이때 CB 는 과거 사례들의 집합인 사례 베이스, P_{new} 는 새로운 계획 문제를 각각 나타낸다.

<그림 2>의 예에 표현된 사례 기반 계획 문제 CP 는 앞서 정의한 $z = ((D, I', G'), S')$ 와 같은 과거 사례들의 집합 CB 를 기초로, 새로운 계획 문

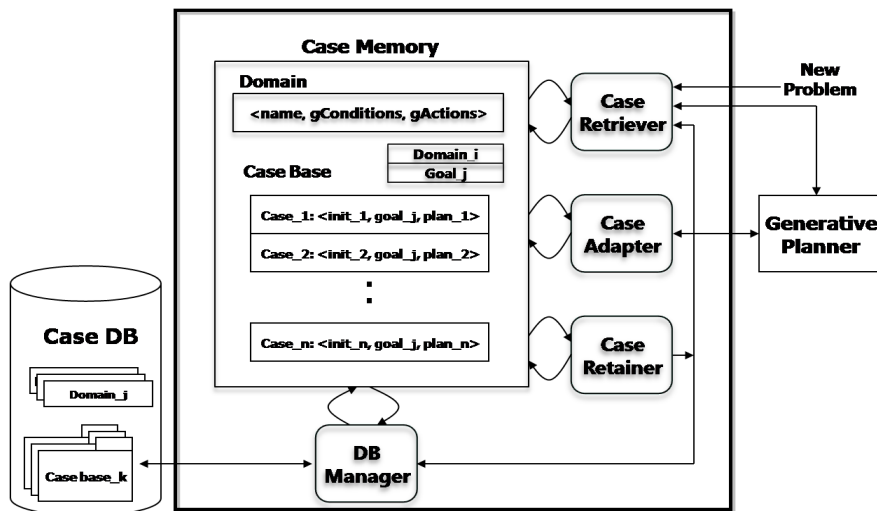
제인 $P = (D, I, G)$ 에 대한 해 계획을 생성하는 것으로 정의할 수 있다. 그리고 $S = ((\text{moveToCB robot circuitbreaker tablearea circuitbreakerarea}), (\text{switchOn robot circuitbreaker circuitbreakerarea}), (\text{automove robot circuitbreakerarea tablearea}), (\text{pickup robot tablearea cup table}), (\text{carry robot cup tablearea purifierarea}), (\text{fillWater robot cup purifier purifierarea}), (\text{carry robot cup purifierarea userarea}), (\text{give robot user cup userarea}))$ 는 이러한 사례 기반 계획 문제의 한 해 계획이 될 수 있다.

3. 시스템의 설계

3.1 시스템 구조

<그림 3>은 본 논문에서 제안하는 사례 기반 계획시스템인 JCBP의 전체 구조를 나타낸다. JCBP 시스템은 크게 여섯 개의 부분으로 구성된다. 이들은 사례 메모리(case memory), 사례 검색

기(case retriever), 사례 적응기(case adapter), 사례 저장기(case retainer), 데이터베이스 관리자(DB manager) 그리고 사례 데이터베이스(case DB) 등이다. 사례 메모리는 새로 주어진 계획 문제를 해결하는데 이용할 사례들의 집합, 즉 사례베이스의 일부를 주기억장치에 저장하는 역할을 한다. 각 사례는 사례 계획뿐만 아니라 계획 문제와 해당 도메인 정보도 함께 저장 관리되어야 검색과 재사용이 원활히 이루어 질 수 있다. 따라서 도메인과 작업 목표가 일치하는 사례들은 묶어 한꺼번에 사례 메모리에 저장하면서 이들이 공유하는 도메인 정보도 함께 사례 메모리에 보관하도록 설계하였다. 각 계획 도메인과 연관하여 해당 도메인의 이름(name)과 그 도메인을 표현하는데 쓰이는 논리 조건들의 집합(gConditions), 이들을 이용하여 정의한 동작들의 집합(gActions) 등의 도메인 정보를 사례 메모리에 저장한다. 또 사례 메모리에는 이와 같은 도메인 정보 외에, 현재 사례 메모리에 올라와 있는 사례들이 공유하는 공통의 작업 목표 조건들도 함께 저장한다. 그리고 사례



<그림 3> JCBP 시스템의 구조

메모리에 저장되는 각 사례들은 앞서 언급한 바와 같이 사례 문제(case problem), 사례 계획(case plan), 그리고 계획 길이(length)와 같은 유용한 속성(attribute)들과 속성값(value)으로 표현된다. 사례 메모리에 현재 올라와 있지 않은 많은 다른 사례들은 모두 외부의 관계형 데이터베이스에 유지 보관된다. 이와 같이 관계형 데이터베이스에 보관된 대규모 사례들의 집합을 사례 데이터베이스라 부른다. 한편, 데이터베이스 관리기는 사례 데이터베이스로부터 도메인과 작업 목표를 공유하는 특정 사례 집합을 선택하여 사례 메모리로 적재하거나 새로운 사례의 추가나 삭제 등으로 인해 변경된 사례 메모리내의 사례 집합을 데이터베이스에 저장 보관하는 역할을 담당한다.

JCBP 시스템의 사례 검색기는 새로운 계획문제가 주어지면 저장하고 있는 사례들 중에서 현재 문제에 가장 근사한 사례를 찾아내는 역할을 한다. 사례 검색을 위해서는 현재 문제와 과거 사례문제간의 유사도 척도(similarity measure)와 임계치(threshold)를 이용하며, 이들은 JCBP 시스템 사용자에게 의해 미리 주어진다. 유사도는 주로 현재 문제와 과거 사례 문제를 구성하는 초기 조건들과 목표 조건들의 일치도를 검사하여 결정된다. 사례 검색은 먼저 현재 사례 메모리에 올라와 있는 사례 집합을 대상으로 이루어진다. 사례 메모리내의 사례들은 모두 도메인과 목표 조건들이 일치하는 사례들이므로, 개별 사례의 유사도를 계산하기에 앞서 먼저 메모리내의 도메인 정보와 목표 조건들을 검사하여 현재 문제와 일치여부를 조사 해본다. 이와 같은 과정을 통해 만약 사례 메모리내의 사례들이 해결하고자 하는 현재의 계획 문제와는 상이한 도메인 정보 혹은 목표 조건들을 갖거나, 주어진 임계치 이상의 유사도를 갖는 사례를 발견할 수 없는 경우에는 데이터베이스 관리기에 요청

하여 사례 데이터베이스로부터 새로운 사례 집합들을 가져와 사례 메모리에 적재한 후 이들을 대상으로 검색 작업을 계속한다. 사례 검색기에 의해 현재 계획 문제에 충분히 유사한 사례가 발견되면, 이것을 사례 적응기에 제공하여 현재 계획 문제에 맞는 해 계획으로 변경한다. JCBP 시스템에서는 사례 적응을 위해 생성적 계획 알고리즘에 기초한 생성적 적응방식을 이용하고 있다. 사례 적응기에 의해 성공적으로 해 계획이 구해지면 그 결과를 계획 실행기나 사용자에게 제공할 뿐 아니라 사례 저장기를 통해 일반화 과정을 거친 다음, 새로운 사례로 사례 메모리와 사례 데이터베이스에 저장된다. 사례 검색이나 사례 적응에 실패하는 경우에는 생성적 계획기인 JPLAN을 호출하여 과거 사례의 도움 없이 처음부터 새롭게 해 계획을 수립한다.

3.2 사례 표현과 메모리 구성

사례 표현은 이전의 해결한 계획 문제를 메모리내에 표현하고 조직하는 것을 의미한다. 사례 기반 계획 시스템에서 사례가 어떻게 표현되느냐에 따라 시스템 전체의 효율성과 성능에 영향을 미친다. JCBP 시스템은 도메인(domain), 작업(task), 사례(case) 등으로 구성되는 삼 단계 개념 계층을 기초로 사례들을 표현하고 분류한다. 각 도메인은 계획 문제가 정의될 수 있는 하나의 계획 도메인을 나타내며, 앞서 정의한 바와 같이 해당 도메인의 각 상태를 표현하기 위한 논리 조건들의 집합과 상태를 변경할 수 있는 동작들의 집합으로 구성된다. 예컨대 (on cup table)은 하나의 논리 조건식을, (pick-up robot tablearea cup table)은 이러한 조건식에 기초해 정의되는 하나의 동작을 각각 나타낸다.

한편, 각 작업은 <초기 조건 집합, 목표 조건 집합>의 쌍으로 이루어진 하나의 계획 문제를 나타

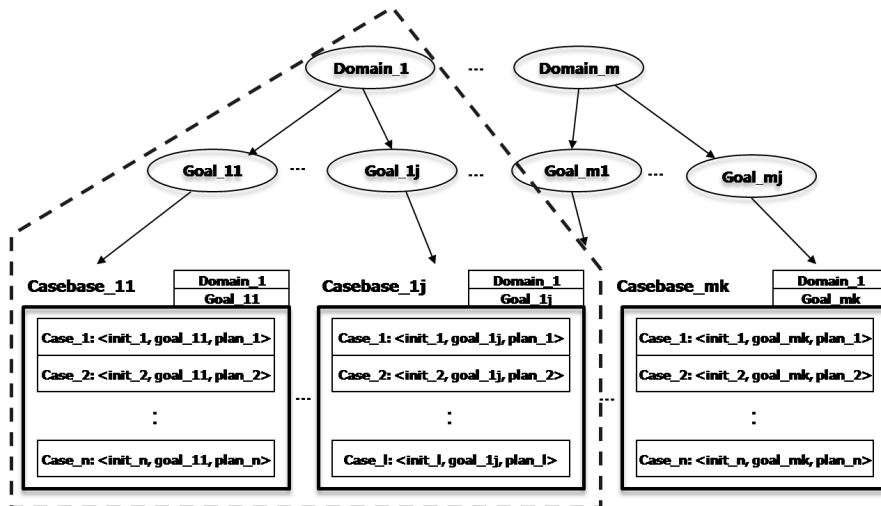
낸다. 각 사례는 하나의 계획 문제와 하나의 사례 계획으로 구성된다. 사례 계획은 초기 조건 들을 만족하는 작업의 한 초기 상태에서 시작하여 목표 조건들이 모두 만족되는 상태까지 도달 가능한 일련의 동작들로 표현된다. 일반적으로 동일한 계획 도메인에서 서로 다른 여러 작업들이 정의될 수 있고, 동일한 하나의 작업에 대해서도 서로 다른 다양한 사례 계획들이 존재할 수 있다. <그림 4>는 JCBP 시스템의 메모리 내 사례 표현과 분류를 나타낸 것이다. 사례들은 먼저 도메인에 따라서 최초 분류가 되며, 다음은 달성하고자 하는 작업 목표에 따라 분류가 된다. 이러한 분류에 따라 동일한 도메인과 작업 목표를 공유하는 사례들은 개별 사례 베이스로 그룹화 된다. 그리고 이러한 개별 사례 베이스는 새로운 문제 해결을 위해 사례 데이터베이스에서 메모리로 적재되거나 메모리에서 데이터베이스로 저장되는 사례 집합들의 최소 단위가 된다. 따라서 JCBP 시스템에서는 언제나 메모리 내에 현재의 도메인 정보와 이 도메인에 속한 하나의 개별 사례 베이스가 보관된다. 동일한

하나의 개별 사례 베이스에 속한 사례들은 주로 문제의 초기 조건들과 사례 계획이 서로 상이하다.

한편, 메모리 내에서 도메인 정보와 사례들을 표현할 때, 기호(symbol)로 정의된 논리 조건들과 동작들이 여러 번 반복 사용되어야 한다. 메모리 절약과 처리의 효율성을 위해 JCBP 시스템에서는 논리 조건들의 집합(gConditions)과 동작들의 집합(gActions)에 속한 논리 조건들과 동작들의 기호 원형은 각기 별도의 테이블에 저장 보관하는 대신 도메인 정보와 사례들을 표현할 때에는 이들에 대한 고유한 수치 식별자(numeric identifier)를 사용한다. 그리고 필요한 경우, 이 식별자를 색인 키(index key)로 삼아 논리 조건 테이블과 동작 테이블에서 기호 원형에 접근할 수 있도록 한다.

3.3 사례 검색

사례 검색(case retrieval)은 주어진 문제를 해결하기 위해서 기존의 사례들 중에서 현재 문제에 가장 유사한 사례를 선택하는 과정을 말한다. 사례



<그림 4> 사례 표현과 분류

베이스가 작업 목표 조건들에 따라 다수의 개별 사례 베이스로 분할 저장되기 때문에 사례 검색 과정은 새로 주어진 계획 문제와 도메인 정보 및 목표 조건들이 일치하는 개별 사례 베이스를 먼저 찾은 뒤, 그 개별 사례 베이스에 속한 각 사례별로 유사도(similarity)를 평가하는 방식으로 진행된다. 먼저 새로운 계획 문제가 주어지면, 이 문제가 정의된 도메인과 현재 메모리 내의 도메인 정보와 일치 여부를 검사한다. 만약 도메인이 일치한다면, 현재 메모리에 올라와 있는 개별 사례 베이스의 작업 목표 조건이 새로운 계획 문제의 작업 목표 조건과 일치하는지를 검사한다. 만약, 현재 메모리 내의 도메인 정보나 개별 사례 베이스의 작업 목표 조건이 새로운 문제와 일치하지 않으면, 사례 데이터베이스(case DB)로부터 일치하는 도메인 정보와 개별 사례 베이스를 찾아 메모리로 적재(load)한다. 새로운 개별 사례 베이스가 메모리로 적재될 때, 이전에 메모리에 보관 중이던 개별 사례 베이스는 메모리에 놓여 있는 동안 추가된 사례들을 데이터베이스에 반영하기 위해 해당 개별 사례 베이스를 사례 데이터베이스에 재 저장한다. 이와 같은 과정을 통해 일단 메모리에 도메인과 목표 조건들이 일치하는 개별 사례 베이스가 놓여 있음이 확인되면, 개별 사례 베이스 내의 각 사례 별로 초기 조건들을 중심으로 새로운 계획 문제와의 유사도를 계산한다. 본 시스템은 유사성 척도로 CMR(Condition Match Ratio)을 정의하고, 이것을 사례 검색에 사용한다. CMR은 식 (1)과 같이 새로운 계획 문제와 사례의 초기 조건들이 얼마나 일치하는지를 측정하여 유사도를 계산한다.

$$\text{Similarity}_{\text{CMR}}(P_{\text{new}}, P_{\text{case}}) = \frac{|I_{\text{new}} \cap I_{\text{case}}|}{|I_{\text{new}} \cup I_{\text{case}}|} \quad (1)$$

식 (1)의 분모는 새로운 계획 문제와 사례 문제의 초기 조건들의 합집합의 크기를 나타내고, 분자는 두 문제의 초기 조건들의 교집합의 크기를 나타낸다. 즉, CMR은 새로운 계획 문제와 사례 문제의 초기 조건들의 상대적인 일치도를 계산한다. 따라서 CMR은 단순히 두 문제 간에 일치하는 초기 조건들의 수가 많다고 유사도를 높게 평가하지 않고, 두 문제의 초기 조건 집합의 크기에 비해 서로 일치하는 조건들이 차지하는 비율이 클수록 유사도를 높게 평가한다. 이와 같은 유사도 계산은 이후에 진행될 사례 적응 단계에 소요될 실질적인 노력의 양을 미리 고려하여 이를 최소화할 수 있는 사례를 검색하도록 도와준다. 메모리내의 각 사례의 유사도가 계산되면 이들 중에서 유사도가 가장 높은 사례가 자동으로 선택된다. 하지만 JCBP 시스템에서는 사례 검색 단계에서도 사용자의 판단과 참여를 위해 사용자가 그래픽 사용자 인터페이스를 통해 슬라이드방식으로 임계치(threshold)를 설정할 수 있도록 하고, 설정된 임계치 이상의 유사도를 가지는 사례들만을 가려내어 제공하는 필터링 기능도 제공한다.

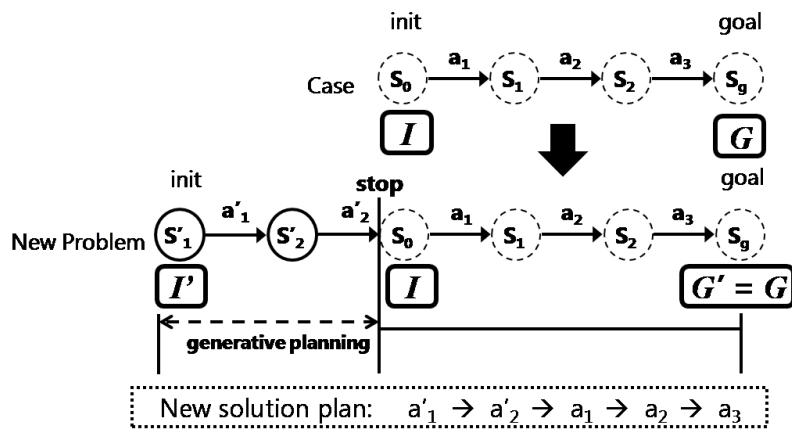
3.4 사례 적응

JCBP 시스템은 검색된 사례 계획을 새로운 계획 문제에 성공적으로 재사용하기 위해 사례 적응 과정을 수행한다. JCBP 시스템에서 채택하는 사례 적응 방법은 생성적 계획(generative planning)이다. 이 방법은 선택된 사례 계획을 새로운 계획 문제에 적합한 하나의 해 계획으로 확장하기 위해 생성적 계획 알고리즘을 이용하는 것이 큰 특징이다. 즉, 생성적 계획 방식의 사례 적응은 사례 문제의 초기 조건들과 새로운 계획 문제의 초기 조건들 사이의 차이를 극복하거나 또는 목표

조건들 사이의 차이를 극복하기 위해 이들을 또 다른 하나의 계획 문제로 정의한 뒤 이 문제에 대한 해를 상태 공간 탐색과 같은 생성적 계획 알고리즘으로 구하여 사례 계획에 덧붙임으로써 새로운 문제를 위한 완전한 하나의 해 계획을 구해내는 방법이다.

<그림 5>는 JCBP 시스템에서 채택하고 있는 사례 적용 방법을 나타내고 있다. 사례 문제 $P = (D, I, G)$ 와 사례 계획 $\langle a_1, a_2, a_3 \rangle$ 가 주어졌을 때, 이것을 새로운 문제 $P' = (D, I', G')$ 에 적용하는 과정은, P 의 초기 조건들의 집합인 I 와 P' 의 초기 조건들의 집합인 I' 간의 불일치성을 해소하는 과정으로 축소해 볼 수 있다. 그 이유는 앞서 설명한 바와 같이 사례 검색 단계를 통해 선택된 사례들은 이미 새로운 계획 문제와 모두 목표 조건들이 일치하기 때문이다. 따라서 JCBP시스템에서는 이 불일치성을 해소하기 위해 새로운 문제와 사례 문제의 초기 조건들부터 새로운 생성적 계획 문제 $P'' = (D, I', I)$ 를 정의한다. 이 문제의 초기 조건들은 새로운 계획 문제의 초기 조건들인 I' 과 일치하며, 이 문제의 목표 조건들은 사례 문제의 초기 조건들인 I 와 일치한다. 일단 계획 문제 P''

이 정의되고 나면, 본 연구그룹에서 선행 연구를 통해 개발한 휴리스틱 탐색 계획 알고리즘을 적용해 계획 문제 P'' 에 대한 해 계획 $\langle a'_1, a'_2 \rangle$ 을 구한다. 이와 같은 사례 적용을 위한 생성적 계획은 <그림 5>에서 점선으로 표시되어 있다. 사례 적용의 마지막 단계에서는 기존의 사례 계획 $\langle a_1, a_2, a_3 \rangle$ 에다 새로 구한 계획 문제 P'' 의 해인 $\langle a'_1, a'_2 \rangle$ 을 앞쪽에 추가함으로써, 새로운 계획 문제를 위한 하나의 완전한 해 계획을 $\langle a'_1, a'_2, a_1, a_2, a_3 \rangle$ 와 같이 완성한다. JCBP 시스템에서 사용하는 휴리스틱 탐색 계획 알고리즘은 대표적인 지역 탐색 알고리즘인 언덕 오르기(hill-climbing)를 개선한 EHC+ 알고리즘이다. 또한 이 계획 알고리즘에서는 탐색의 효율을 높이기 위해 문제의 명세로부터 자동으로 추출하는 휴리스틱 정보를 이용한다. 그리고 이 알고리즘에서는 휴리스틱 자동 추출을 위해 특수한 자료구조인 간략화한 계획 그래프(relaxed planning graph)를 이용한다. Graphplan 계획기[13]를 통해 처음 소개된 계획 그래프는 상태 공간상의 한 상태에서 목표 상태까지 도달성 휴리스틱(reachability heuristic)을 손쉽게 계산할 수 있는 자료구조이다[14].



<그림 5> 생성적 사례 적용 방법

3.5 사례 일반화와 저장

사례 저장은 새로운 계획 문제에 대한 해 계획을 또 다른 하나의 사례로 저장하는 단계이다. JCBP 사례 기반 계획 시스템은 새로운 사례를 저장하기 전에 사례 일반화(generalization)를 수행한다. 대표적인 일반화 방법으로는 미리 정의된 하나의 개념 계층(concept hierarchy)구조에 따라 사례 구성요소를 추상화(abstraction)하는 방법과 사례 표현에 등장하는 사물(object)들 대신 변수(variable)들로 파라미터화(parameterization)하는 방법 등이 있다. JCBP 시스템에서는 이들과는 달리 새로운 사례의 초기 조건들 중 필수 조건들은 남기고 나머지 불필요한 조건들을 제거함으로써 사례를 일반화한다. 사례의 필수 초기 조건들을 구하기 위해 JCBP 시스템에서는 목표 회귀(goal regression) 방법을 사용한다.

<표 1> 목표 회귀 알고리즘

```

/* Given the plan as a sequence of actions
   <a1, a2, ..., an> and the goal conditions G */

I = G
  For i = n to 1
    I = Cai ∪ {(I - Eai+) ∪ Eai}
  End
Return I
    
```

<표 1>는 JCBP 시스템에서 사례 일반화를 위해 사용하는 목표 회귀 알고리즘을 나타내고 있다. 이 알고리즘은 주어진 사례의 목표 조건들의 집합 G 에서 출발하여, 계획 $\langle a_1, a_2, \dots, a_n \rangle$ 에 포함된 각 동작의 효과를 차례대로 되돌려 놓음으로써 목표 조건들을 달성하기 위해 꼭 필요한 초기 조건들을 구한다. 먼저 목표 조건 집합 G 에서 계획의 마지막

동작 a_n 의 실행효과를 되돌려 놓기 위해, 긍정적 효과인 E_{ai}^+ 에 포함된 조건들은 모두 삭제하고 대신 부정적 효과인 E_{ai}^- 와 전 조건(precondition) C_{ai} 에 포함된 조건들은 모두 추가한다. 마지막 동작에서 시작하여 역방향으로 계획에 포함된 모든 동작에 대해 이와 같은 회귀 과정을 거친 후 마지막에 얻는 결과 집합인 I 를 필수 초기 조건들로 반환한다. 목표 회귀를 통해 사례의 필수 초기 조건들이 구해지면 이들만 남기고 나머지 초기 조건들은 모두 삭제하여 사례를 일반화한다. 이러한 사례 일반화는 각 사례의 적용 범위를 넓혀주고, 불필요한 초기 조건들의 차이로 인해 수많은 사례들을 각기 따로 저장해야하는 문제를 해결할 수 있고, 그 결과 사례 베이스를 저장하기 위해 필요한 전체 메모리의 양을 줄일 수 있다. 이로 인해 사례 검색의 효율성도 증대될 수 있다.

JCBP 시스템에서 대용량의 사례를 저장하기 위한 사례 데이터베이스는 하나의 관계형 데이터베이스(relational DB)로서 도메인 테이블, 작업 목표 테이블, 사례 테이블 등 총 3개의 테이블로 구성된다. <표 2><표 3><표 4>는 도메인과 작업 목표 그리고 사례 테이블에 대한 각각의 스키마(schema)를 나타낸다. 도메인 테이블에는 가용한 동작들에 대한 세부사항(gActions), 논리 조건과 식별자와의 매핑정보(ConditionMap), 동작과 식별자와의 매핑정보(ActionMap) 등이 주로 저장된다. 또, 작업 목표 테이블에는 작업이 정의되는 도메인 이름(DomainName)과 작업 목표 조건들(GoalConditions)들이 보관되며, 사례 테이블에는 각 사례의 초기 조건들(InitConditions), 목표 조건들(GoalConditions), 사례 계획(Plan), 계획의 길이(Length) 등이 저장된다.

4. 시스템의 구현

JCBP 시스템은 크게 자동 모드(auto mode)와 대화형 모드(interactive mode) 등 서로 다른 2가지 모드로 수행할 수 있도록 구현되었다. 자동 모드에서는 계획 실행기(plan executive)와 같은 외부 시스템으로부터 계획 문제와 함께 계획 생성 요청 메시지가 도착하면, JCBP 시스템은 별도의 사용자 개입없이 이미 설정되어 있는 다양한 기본 설정값에 따라 적절한 사례를 검색하고 적응과정을 거쳐 주어진 작업 계획 문제에 대한 해 계획을 도출한다.

<표 2> 도메인 테이블

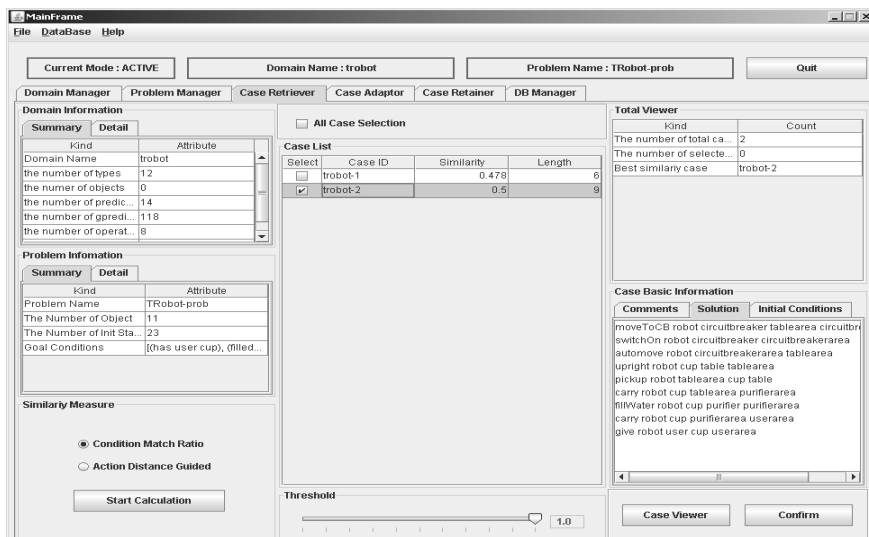
Field	Type	Key	Null
DomainName	varchar<100>	PRIMARY	no
ConditionMap	varchar<700>		no
ActionMap	varchar<700>		no
gActions	varchar<1000>		no

<표 3> 작업 목표 테이블

Field	Type	Key	Null
TaskName	varchar<100>	PRIMARY	no
GoalConditions	varchar<500>		no
DomainName	varchar<100>	FOREIGN	no

이 자동모드는 실시간으로 발생하는 비 동기 계획 요청에 효과적으로 대응하기 위해 마련되었다. 반면에, 대화형 모드에서는 외부 시스템에서 계획 생성 요청 메시지를 보내오거나 혹은 사용자가 직접 JCBP 시스템의 사용자 인터페이스를 통해 작업 계획 생성 명령을 내리면, 그래픽 사용자 인터페이스(GUI)를 기초로 사용자와 대화를 통해 계획 생성을 위한 사례 기반 계획 전체 과정을 진행해간다. 이 모드에서는 가능한 시각화한 형태로 사용자에게 계획 생성 진행과정과 선택 사항들을 보여주고 사용자가 직접 선택하게 함으로써 사용자 주도로 원하는 해 계획을 도출하도록 지원한다.

JCBP 시스템은 대화형 모드를 지원하기 위해



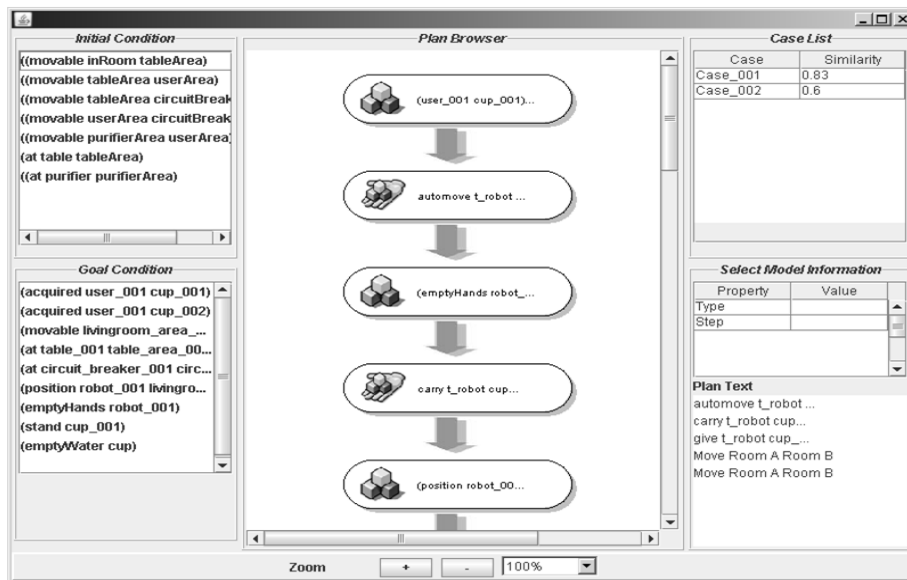
<그림 6> 사용자 대화형 모드를 지원하는 GUI

<그림 6>과 같은 GUI를 제공한다. JCBP의 GUI는 현재 진행되고 있는 시스템 내부의 상황을 사용자가 이해하기 쉽게 표현하고, 또한 사용자의 의도와 결정이 시스템에 잘 반영되도록 구현되었다. JCBP 시스템의 GUI는 사례 기반 계획의 순환 구조에 따라 각 단계별 사용자와의 상호작용을 지원하기 위해, <그림 6>처럼 Domain Manager, Problem Manager, Case Retriever, Case Adapter, Case Retainer, DB Manager 등 다수의 탭으로 구성된 탭 브라우저(tab browser) 방식을 택했다.

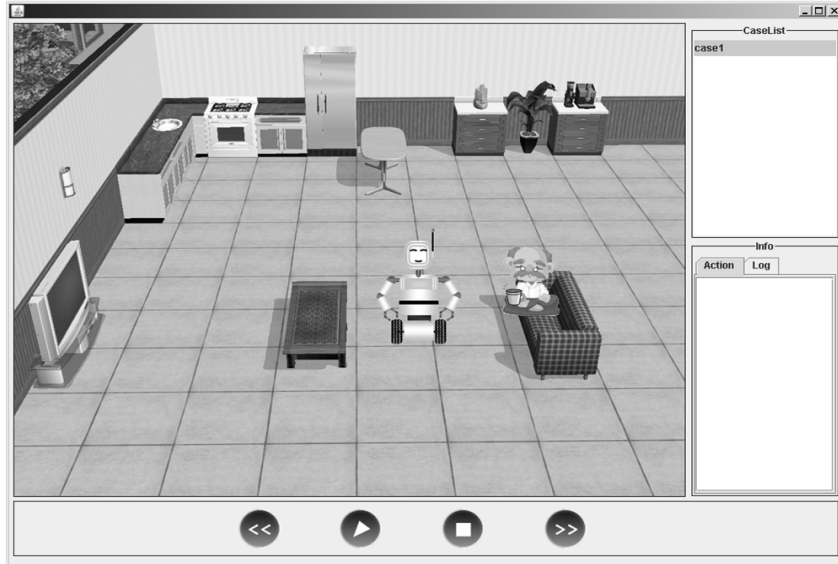
JCBP 시스템에는 사용자에게 사례 계획이나 새로 생성된 해 계획을 쉽게 이해할 수 있도록 시각화 해주는 도구인 계획 브라우저(plan browser)를 내장하고 있다. <그림 7>은 JCBP 시스템에 내장된 계획 브라우저의 한 실행화면을 보여주고 있다. 계획 브라우저의 화면 좌측에는 초기 조건과 목표 조건으로 구성된 계획 문제 명세를 보여주고, 화면 중앙에는 계획을 구성하는 동작들의 순서를 보여

주며, 화면 우측에는 기타 관련 정보를 보여준다. 또한 동작과 동작을 잇는 아크를 클릭하면 계획이 그 위치까지 실행되었을 때 나타날 중간 상태 정보를 보여주기도 한다. 이와 같은 계획 브라우저의 도움으로 사용자는 사례 문제와 사례 계획에 대한 정보를 쉽게 파악할 수 있고, 따라서 그것을 기초로 주어진 현재 문제를 어떻게 해결해 나갈지 결정을 내리는데 도움을 얻을 수 있다.

또한 JCBP 시스템은 사용자의 이해를 돕기 위해 앞서 설명한 계획 브라우저 외에 계획 시뮬레이터(plan simulator)도 제공하고 있다. <그림 8>은 계획 시뮬레이터의 한 실행화면을 보여 주고 있다. 계획 브라우저는 기호로 정의된 계획 정보를 도형을 이용해 시각화해주는 반면, 계획 시뮬레이터는 실세계 작업 공간과 작업 로봇을 2차원 그래픽 가상 환경위에 재구성해 놓고 이 가상 환경 안에서 계획에 따라 로봇 동작을 하나씩 실행시켜 그 결과를 확인할 수 있도록 해준다. 따라서 계획



<그림 7> 계획 브라우저의 실행화면



<그림 8> 계획 시뮬레이터의 실행화면

브라우저가 계획 정보 자체에 대한 시각화 도구라면, 계획 시뮬레이터는 계획 실행 효과를 시각화해주는 도구로 볼 수 있다. 이러한 기능을 가진 계획 시뮬레이터는 새로 구한 해 계획의 품질과 실행 결과를 평가할 때 유용하게 쓰일 수 있다.

따라서 계획 시뮬레이터가 주로 호출 되는 시점은 사례 적응 단계를 통해 해 계획이 구해진 이후가 되며, 실행을 위해 해 계획을 계획 실행기로 보내기 이전에 미리 가상 환경에서 해 계획을 평가하기 위해 불린다. 시뮬레이션 해본 결과, 해 계획이 사용자의 만족도에 미치지 못하면 사용자는 JCBP 시스템을 사례 적응 단계 혹은 사례 검색 단계 등으로 되돌려 사례 기반 계획 과정을 다시 시작할 수 있다.

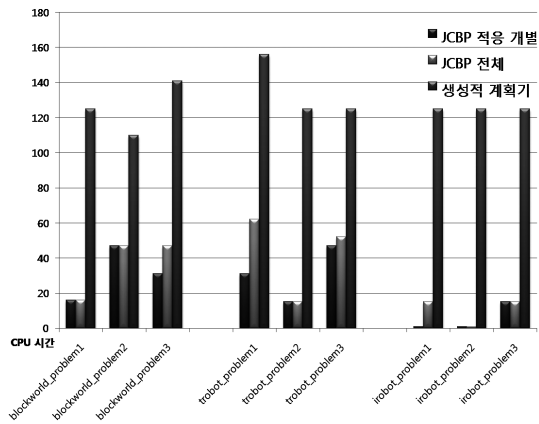
5. 실험

JCBP 시스템의 효율성과 계획의 품질을 평가하

기 위해 몇 가지 도메인과 계획 문제를 이용한 실험을 전개하였다. 사례 기반 계획 시스템인 JCBP의 효율성을 평가하기 위해서는 계획 문제를 해결하는데 소요된 시간을 측정하여 생성적 계획기 (generative planner)의 경우와 비교하였다. 또, 계획의 품질 면에서 JCBP를 평가하기 위해서는 해 계획의 길이를 측정하여 생성적 계획기와 비교하였다. 실험에 사용된 계획 도메인은 블록쌓기 (blockworld), T-Robot, I-Robot 등 서로 다른 3가지 도메인들이며, 각 도메인별로 임의로 생성한 10가지 계획 문제들을 풀어 사례 베이스를 구축한 뒤, 다시 각 도메인별로 난이도가 다른 3가지 계획 문제들을 생성하여 이들을 사례 베이스를 이용해 풀어 보았다.

<그림 9>의 그래프는 실험 결과 측정된 각 계획 문제별 총 소요 시간을 나타내고 있다. 그래프에는 각 계획 문제에 대해 사례 검색에서 적응 및 저장까지 JCBP 전 과정에 소요된 시간과 단지 사

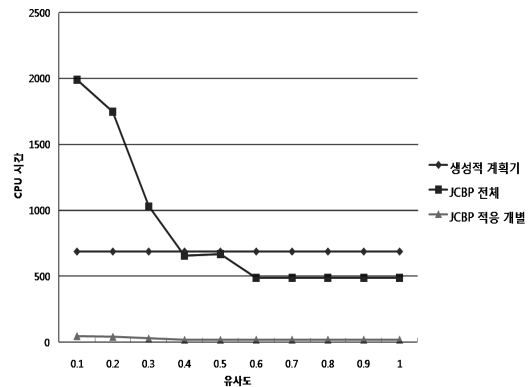
례 적응에 소요된 시간, 그리고 사례를 이용하지 않고 순수하게 생성적 계획기로 풀었을 때 소요된 시간 등을 비교하고 있다. 그래프의 첫 번째 막대는 JCBP의 적응 단계에 소요된 시간을 나타낸 것이며, 두 번째 막대는 JCBP 시스템의 전 과정에 소요된 시간을 나타낸 것이다. 또, 마지막 막대는 생성적 계획기를 사용하여 문제를 해결하였을 때 소요된 시간을 나타낸다.



<그림 9> 계획 문제 별 소요 시간

모든 계획 문제들에서 JCBP 시스템의 소요 시간이 생성적 계획기의 소요 시간보다 훨씬 짧다는 것을 확인할 수 있다. 이것은 새로운 계획 문제의 해결을 위해 유사한 사례들을 이용하는 사례 기반 계획 시스템인 JCBP가 매번 처음부터 새롭게 계획 생성과정을 거치는 생성적 계획기보다 매우 효율적임을 입증하는 것이다. 또한 JCBP 시스템의 사례 적응에 소요된 시간과 전 과정에 소요된 시간을 비교해보면 큰 차이가 나지 않았다는 것을 알 수 있다. 이것은 JCBP 시스템의 독특한 사례 분류와 저장, 그리고 검색 기능이 사례 기반 계획 전 과정에 매우 효과적이었음을 암시하는 것이다. 실험에 사용된 도메인 중에서는 I-Robot 도메인에

서 JCBP 시스템의 소요 시간과 생성적 계획기의 소요 시간간의 차이가 가장 크게 나타났다. 하지만 일반적으로 사례 기반 계획 시스템의 경우, 계획에 이용되는 사례의 유사도에 따라 해 계획 생성에 소요되는 시간에 큰 차이를 보일 수 있다. 따라서 본 연구에서는 블록쌓기 계획 문제들을 대상으로 계획 생성에 이용되는 사례의 유사도를 달리하면서 소요 시간을 측정하는 실험을 추가적으로 수행해 보았다.



<그림 10> 사례의 유사도에 따른 소요시간

<그림 10>은 블록쌓기(blockworld) 계획 문제들을 유사도 다른 사례들을 가지고 풀어 본 실험 결과를 나타내고 있다. 생성적 계획기는 사례를 이용하지 않기 때문에 동일한 소요 시간을 보여주고 있으나, JCBP 시스템은 이용하는 사례의 유사도에 따라 소요 시간에 큰 차이를 보인다. 새로운 계획 문제 해결에 이용하는 사례의 유사도가 높을수록 JCBP의 사례 적응과 사례 기반 계획 전 과정에 소요된 시간 모두가 크게 감소하는 것을 알 수 있다. 따라서 이러한 실험 결과를 통해 사례 검색 단계의 결과가 사례 기반 계획 전 과정의 성능에 미치는 영향을 다시 확인 할 수 있었다.

<표 5> 해 계획의 길이

Problem	JCBP	Optimal Solution
blockworld_1	5	3
blockworld_2	5	3
blockworld_3	7	5
trobot_1	7	6
trobot_2	7	6
trobot_3	11	8
irobot_1	3	3
irobot_2	4	2
irobot_3	5	3

<표 5>는 서로 다른 계획 문제들에 대해 JCBP 시스템이 구한 해 계획의 길이를 나타내고 있다. 최적 해의 길이와 비교를 해보면, JCBP 시스템은 대부분의 문제에 대해 최적 해를 구하지는 못한 것으로 나타났다. 이러한 현상은 부분적으로는 현재 JCBP 시스템의 사례 적응 방식에 기인하는 것으로 파악된다. 현재 JCBP 시스템은 생성적 사례 적응 과정을 통해 새로운 문제에 대한 해 계획으로 확장하는 동안 기존의 사례 계획 부분에 대한 변경은 허용하지 않고 다만 사례 계획의 앞쪽에 새로운 동작들을 추가하는 방식으로 진행된다. 따라서 주어진 새로운 계획 문제에 대한 최적 해를 얻기 위해서는 기존의 사례 계획 부분에 대한 삭제 및 변경이 반드시 요구하는 경우, JCBP 시스템은 이를 허용할 수 없기 때문에 최적 해를 얻기 어려운 것으로 파악된다. 하지만 대신 JCBP 시스템은 새로운 문제의 초기 조건과 사례 문제의 초기 조건으로 구성되는 부분적인 계획 문제를 최적 계획 생성 알고리즘으로 풀기 때문에, 대체적으로 JCBP 시스템의 해 길이가 최적의 해 길이에 비해 그다지 큰 차이가 나지 않은 좋은 결과를 보인 것

을 알 수 있다. 따라서 JCBP 시스템은 계획의 최적성이 엄격히 요구되지 않는 많은 실세계 계획 문제들에 대해서 높은 효율성과 성능을 보여줄 수 있다고 판단한다.

7. 결론

본 논문에서는 사례 기반 계획 생성 시스템인 JCBP 시스템의 설계와 구현에 대해 소개하였다. JCBP 시스템은 효율적인 메모리 사용과 사례 검색을 위해 도메인별로 동일한 작업 목표를 가진 사례들을 개별 사례 베이스로 그룹화하고, 이들에 대한 색인을 유지한다. 또 이 시스템은 휴리스틱 지식을 이용하는 효율적인 생성적 적응 방법을 사례 적응 단계에 적용하며, 목표 회귀를 통한 사례 일반화 기능도 제공한다. 또한 JCBP 시스템은 대화형 모드를 통해 사용자의 지식과 기호를 계획 생성 과정에 반영함으로써 사용자의 요구를 더 잘 만족하는 해 계획을 생성할 수 있을 뿐 아니라 계획 생성의 복잡도도 줄일 수 있다. 하지만 해 계획의 최적성을 보장하기 위해서는 기존의 사례 계획 부분도 새로운 계획 문제에 맞추어 삭제 및 변경을 허용하는 보다 유연한 사례 적응 알고리즘에 대한 향후 연구가 필요할 것으로 판단한다.

참고문헌

- Althoff, K.-D. and Aamodt, A., "Relating Case-based Problem Solving and Learning Methods to Task and Domain Characteristics : Towards an Analytic Framework", *AI Communications*, Vol.9, No.3(1996), 109~116.
- Bergmann, R. and K. D. Althoff, "Methodology for Building CBR Applications. In Lenz, M., Ba-

- rtsch-Sporl, B., Burkhard, H.D. and Wess, S. (eds.) Case-Based Reasoning Technology from Foundations to Applications”, *Lecture Notes in Artificial Intelligence*, Vol.1400 (1998), 299~331.
- Bermann, R. and W. Wilke, “Building and Refining Abstract Planning Cases by Change of Representation Language”, *Journal of Artificial Intelligence Research*, Vol.3(1995), 53~118.
- Corkill, D. D., “Collaborating Software : Blackboard and Multi-Agent Systems and the Future”, Proceedings of the International Lisp Conference, 2003.
- Ghallab, M., D. Nau, and P. Traverso, Automated Planning : Theory and Practice, Morgan Kaufmann, 2004.
- Hammond, K. J., “Explaining and repairing plans that fail”, *Artificial Intelligence*. Vol.45(1990), 173~228.
- Hammond, K. J., Case-Based Planning : Viewing Planning As A Memory Task, Boston, Academic Press, 1898.
- Hanks, S. and D. Weld, “A Domain-Independent Algorithm for Plan Adaptation”, *Journal of Artificial Intelligence Research*. Vol.2(1995), 319~360.
- Ihrig, L. H. and S. Kambhampati, “Storing and Indexing Plan Derivations Through Explanation-based Analysis of Retrieval Failures”, *Journal of Artificial Intelligence Research*, Vol.7(1997), 161~198.
- Kambhampati, S. and J. Hendler, “A Validation-structure-based Theory of Plan Modification and Reuse”, *Artificial Intelligence* vol.55 (1992), 193~258.
- Kolodner, J., Case-Based Reasoning, Morgan Kaufmann Publishers, 1993.
- Leake, D. and A. Kinley, “Integrating CBR components within a Case-Based Planner”, Proceedings of the AAAI-98 Workshop on Case-Based Reasoning Integrations, San-Mateo, CA, 1998.
- Leake, D., A. Kinley, and D. Wilson. “A Case Study of Case-Based CBR”, Proceedings of the Second International Conference on Case-Based Reasoning(ICCBBR-97), (1997), 371~382.
- McDermott, D., “PDDL-the Planning Domain Definition Language”, Technical Report, www.cs.yale.edu/homes/dvm, 1998.
- Muñoz-Avila, H., J. A. Hendler, and D. W. Aha, “Conversational Case-Based Planning”, *New Review of Applied Expert Systems*, Vol.5 (1999).
- Spalzzi L., “A Survey on Case-Based Planning”, *Artificial Intelligence Review*, Vol.16, No.1 (2001), 3~36.
- Veloso, M., J. Carbonell, A. Pérez, D. Borrajo, E. Fink, and J. Blythe, “Intergrating Planning and Learning : The PRODIGY architecture”, *Journal of Experimental and Theoretical Artificial Intelligence*. Vol.7, No.1(1995).

Abstract

JCBP : A Case-Based Planning System

Incheol Kim* · Mansoo Kim*

By using previous similar case plans, the case-based planning (CBP) systems can generate efficiently plans for new problems. However, most existing CBP systems show limited functionalities for case retrieval and case generalization. Moreover, they do not allow their users to participate in the process of plan generation. To support efficient memory use and case retrieval, the proposed case-based planning system, JCBP, groups the set of cases sharing the same goal in each domain into individual case bases and maintains indexes to these individual case bases. The system applies the heuristic knowledge automatically extracted from the problem model to the case adaptation phase. It provides a sort of case generalization through goal regression. Also JCBP can operate in an interactive mode to support a mixed-initiative planning. Since it considers and utilizes user's preference and knowledge for solving the given planning problems, it can generate solution plans satisfying more user's needs and reduce the complexity of plan generation.

Key Words : Case-Based Planning, Case Adaptation, Case Generalization, Interactive Plan Generation

* Dept. of Computer Science, Kyonggi University

저자 소개



김인철

1995년에 서울대학교에서 전산과학(인공지능) 전공으로 각각 석사학위와 박사학위를 취득하였다. 1989년부터 1995년까지는 경남대학교 전산통계학과에서 조교수로 근무하였으며, 1996년부터 현재까지 경기대학교 자연과학대학 컴퓨터과학과 교수로 재직 중이다. 관심분야는 자동계획 및 기계학습, 지능형 에이전트, 지능로봇 제어구조 등이다.



김만수

2008년에 경기대학교 컴퓨터과학과에서 각각 학사학위와 석사학위를 취득하였다. 2007년부터 산업자원부 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)과 경기도 지역 연구센터(GRRC)의 하나인 콘텐츠 융합 소프트웨어 연구개발 센터에 참여하고 있다. 관심분야는 지능로봇 작업계획, 사례기반 계획 및 학습 등이다.