

지능 로봇 제어를 위한 BDI 기반의 계획 실행기와 작업 계획기의 연동

김하빈⁰ 김인철
 경기대학교 전자계산학과
 {talkable⁰, ickim}@kyonggi.ac.kr

Integrating BDI Executive with Task Planner to Control Intelligent Robots

Ha-bin Kim⁰ In-Cheol Kim
 Department of Information Science, Kyonggi University

요 약

본 논문에서는 지능 로봇의 효율적인 제어를 위해 BDI기반의 계획 실행기와 작업 계획기간의 연동방식을 제안한다. 이를 위해 본 논문에서는 먼저 작업 계획기와 실행기의 대표적인 유형들을 살펴본 다음, BDI 기반의 계획 실행기에 별도의 작업 계획기를 연동함으로써 얻을 수 있는 잇점과 연동방식 등을 차례대로 설명한다. 끝으로 가상환경을 이용한 데모시스템 구현을 통해 심부름 서비스 로봇 제어를 위한 두 서브시스템간의 연동 효과를 점검해본다.

I. 서론

자율 로봇에 대한 제어구조는 로봇이 작업 목표 달성을 위해 센서 입력에 어떻게 반응해야 하는지를 정의하는 것으로 볼 수 있다. 일반적으로 자율 로봇의 제어구조는 복잡도를 고려하여 반응 계층(reactive layer), 시퀀싱 계층(sequencing layer), 숙고 계층(deliberative layer) 등 대략 3 개의 계층으로 나누어 구성되며, 각 계층은 각기 다른 레벨의 제어문제를 다룬다. 이 중에서 특히 숙고 계층은 작업(task) 단위의 제어를 다루는 최상위 계층으로서, 로봇이 처한 현재 상황과 작업 목표를 고려하여 향후 수행할 작업 계획을 수립하고 계획된 바대로 작업이 성공적으로 실행되고 있는지 모니터링하며, 필요한 경우 새로운 상황에 맞도록 재계획(replan)하는 기능들을 수행한다. 따라서 많은 경우, 지능 로봇의 숙고 계층 제어부는 작업 관리자(task manager)로 불리며, 로봇이 실행할 작업 계획들을 생성하고 관리하는 총괄 지휘부 역할을 한다. 본 논문에서는 효과적인 계획 기반의 작업 관리자 구현을 위해, BDI 기반의 계획 실행기와 작업 계획기간의 연동방식을 설계하고 데모시스템을 구현해본다.¹

II. 작업 계획기와 실행기

일반적으로 로봇 작업 계획(plan)이란 시간과 자원에 관한 제약(constraint)을 만족하면서 주어진 작업 목표들을 달성할 수 있는 일련의 동작(action)들을 말한다. 그리고 추론을 통해 이러한 작업 계획을 수립해주는 소프트웨어 컴포넌트를 작업 계획기(planner)라 부른다. 또 사전에 미리 수립해놓은 계획을 실제 작업 환경에서 실행해주는 소프트웨어 컴포넌트를 작업 계획 실행기(executive)라 부른다. 따라서 계획기와 실행

기의 개념적 관계는 그림 1에서 보듯, 작업 계획기는 실행기에 입력으로 주어질 작업 계획을 출력으로 생성하는 역할을 한다.

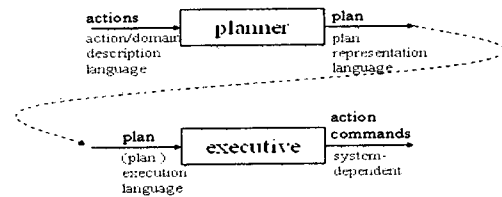


그림 1. 작업 계획기와 실행기

작업 계획기는 일반적으로 정적(static)이고 모든 정보가 접근 가능한(accessible) 완전한 환경을 가정함으로써 계획을 수립하지만, 계획이 실행되는 실제 환경은 대부분 동적(dynamic)이고 노이즈(noise)와 불확실성(uncertainty)이 존재하는 불완전한 특성을 지닌다. 따라서 일반적으로 실행기에 주어지는 계획은 실행 환경을 고려하여 계획기에 의해 생성된 계획보다 더 풍부한 제어구조를 가지고 있다. 즉, 실행기에 주어지는 실행언어(execution language)에는 동작들의 순차구조 외에 조건부 분기와 반복구조 등 다양한 실행 제어 구조들이 더 포함되어 있다. 또 실행기는 계획에 포함된 동작들을 차례대로 실행해주는 일 외에도 작업에 대한 계층적 분해(hierarchical decomposition)와 병행 동작들(concurrent actions)에 대한 조정, 자원 관리, 상태 모니터링, 장애 진단과 복구 등의 기능도 수행해야 한다.

작업 계획기들을 유형별로 나누어 보면 다음과 같다 [1]. 먼저 별도의 자동화된 작업 계획기를 가지고 있지 않은 소위 실행만 가능한 시스템(execution-only system)들이 있다.

¹ 이 연구는 산업자원부 지원으로 수행하는 인간기능 생활지원 지능로봇 기술개발사업의 일환으로 수행되었습니다.

이들은 실행할 계획들이 모두 프로그래머들에 의해 수작업으로 작성되어 입력되는 것으로 가정한다. 또 다른 유형으로는 별도의 외부 작업 계획기와 결합된 실행 시스템(execution systems coupled with an external planner)들이다. 마지막 유형으로는 자체적으로 내부 작업 계획기를 내장하고 있는 실행 시스템(execution systems with internal planner)들이다. 그 동안 지능 로봇 분야나 NASA 우주선 제어분야에 적용되었던 대표적인 작업 계획 실행기들은 대부분 자체적으로 내부 작업 계획기를 포함하고 있거나 혹은 별도의 외부 작업 계획기와 연동 기능을 제공하고 있다.

III. BDI 기반의 실행기와 작업 계획기의 연동

지능형 에이전트 연구분야에서 개발된 BDI(Belief-Desire-Intention) 구조는 동적 실행 환경을 고려한 대표적인 에이전트 내부구조로서, 다양한 지능 로봇 개발에 적용되어 왔다[2, 3]. BDI 구조는 그림 2 와 같이, 달성하고자 하는 목표(goal)들과 특정 목표들을 위한 일반화된 계획(plan)들의 집합, 그리고 현재 실행 환경에 대한 모델(belief)들을 바탕으로 매 순간 에이전트가 실행해야 할 동작(intention)들을 결정하고 수행하는 구조이다. BDI 구조의 큰 특징 중의 하나는 작업 환경의 실시간성을 고려하여 로봇에게 새로운 작업 목표가 주어질 때마다 매 순간 작업 계획기를 이용하여 계획을 새로이 생성하지 않고, 대신 미리 작성된 일반적인 계획(general plan, recipe)들을 저장하고 있다가 이것들을 선택적으로 적용하여 빠른 속도로 동작들을 결정하고 실행한다는 것이다.

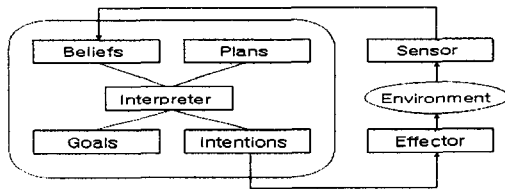


그림 2. BDI 구조

따라서 큰 분류범주로 보면, BDI 구조는 로봇 제어를 위한 계획 실행기(executive)의 역할을 수행한다. 하지만 BDI 구조가 작업 목표를 달성해가는 과정은 단순히 이미 주어진 하나의 계획에 따라 일련의 동작들을 실행해가는 과정이기 보다, 목표 달성에 적합한 계획을 실행상황에 따라 동적으로 선택하고 그 계획에 따라 다시 세부적인 부속 목표(subgoal)들로 분해를 계속해가면서 구체적인 동작들이 결정될 때마다 실행하는 과정이다. 따라서 BDI 구조는 실행환경에 따라 동적으로 구체적인 계획을 생성하고 실행한다는 의미에서 반응형 계획기(reactive planner)로도 볼 수 있다. 이러한 BDI 구조를 지능 로봇 제어를 위한 작업 계획 실행기로 이용하는 경우, 전통적인 작업 계획기는 BDI 실행기가 실행할 작업 계획들을 미리 작성하여 공급하는데 이용되거나 혹은 BDI 실행기가 실행 중에 필요한 새로운 계획을 추가 생성하기 위해 호출하는 주문형 계획기(On-Demand Planner)로 이용될 수 있다.

전자의 방식은 로봇 실행 환경이 비교적 정적인 경우에 효과가 높으며, 미리 작업 계획기를 통해 구체적인 작업 계획을 생성해줌으로써 실행시 BDI 구조의 불필요한 추론과정을 줄일 수 있다. 후자의 방식은 BDI 구조에 내장되어 있는 기존 계획들만으로는 적절히 대처할 수 없는 실행상황이 발생할 때 큰 도움이 된다. 그리고 이와 같은 주문형 계획기 방식은 로봇 작업 제어에 필요한 대부분의 계획들은 미리 내장하고 있으며, 필요한 경우에만 작업 계획기를 호출하여 신규 계획을 생성함으로써 실시간(real-time) 제약이 있는 동적 실행환경에 유리

한 방식이다.

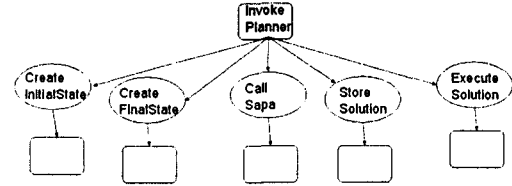


그림 3. BDI 계획의 세부 구성

본 논문에서는 후자의 방식으로 결합되는 BDI 기반의 실행기와 계획기의 연동구조를 제시한다. 먼저 주문형 계획기에 미리 만들어진 계획으로 해결할 수 없는 문제가 발생하면 주문형 계획기 호출을 위한 BDI 계획이 수행된다. 이 BDI 계획에는 그림 3 과 같이 주문형 계획기에 입력으로 넘겨줄 초기 상태(initial state) 묘사와 최종 목표상태(final state) 묘사를 각각 생성하는 작업들과 실제 외부 작업 계획기(예컨대, SAPA 시스템)를 호출하여 해 계획(solution plan)을 생성하고 실행하는 작업들을 차례대로 포함시킨다. 이렇게 함으로써, BDI 실행 중에 해당 목표를 위한 적절한 기존 BDI 계획들이 존재하지 않거나 이들의 실행이 모두 실패로 끝나는 상황을 만나면, 자동적으로 새롭게 추가된 계획기 호출용 계획이 선택되어 실행될 수 있다.

IV. 구현 시스템

이 절에서는 앞서 제안한 BDI 기반의 작업 계획 실행기와 작업 계획기간의 연동방식을 실제로 구현하여 가상의 데모 환경에서 실행해본다. 먼저 BDI 기반의 작업 계획 실행기로는 Java 로 구현된 JAM 시스템[4]을 이용하고, 작업 계획기로는 역시 Java 로 구현된 휴우리스틱 탐색 계획기인 SAPA 시스템[5]을 이용하여 두 시스템을 연동하였다. 구현된 데모시스템의 전체적인 구조는 그림 4 와 같다.

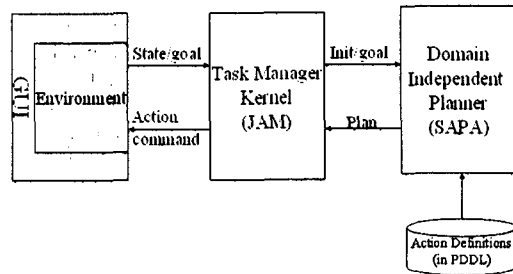


그림 4. 데모시스템의 구조

중앙에 위치한 작업 관리자 커널인 JAM 과 우측에 위치한 계획기 SAPA 간의 연동을 위해 앞서 설명한대로, JAM 내부에 SAPA 를 호출하는 JAM 계획을 새롭게 추가하였고, 이 계획안에서 계획 수립에 필요한 초기 상태와 목표 상태를 계산하여 SAPA 의 입력으로 넘겨주고 실제 SAPA 를 호출한다. 그 결과 새로운 JAM 계획이 구해지면 이어서 이것을 실행하는 과정이 이 계획 안에서 이루어지게 된다.

데모를 위한 환경과 사용자 인터페이스부는 그림 7 과 같이 심부름 서비스를 위한 가상의 로봇 작업 환경을 이용하였다. 데모 환경에 적합한 로봇의 기본 동작들을 표현언어인 PDDL(Planning Domain Definition Language)로 정의하여 계획기인 SAPA 에 사전영역지식으로 공급된다. 그림 5 는 PDDL 로

정의된 로봇 동작의 예로서, PutIntoRefrigerator 의 정의를 보여주고 있다.

```
(define
  (domain robot)
  (:requirements :typing)
  (:types
    Robot Commander Hand MovableObject FixedObject Location - Object
    Cup Drink - MovableObject
    Refrigerator Sofa Table DiningTable TV Bed - FixedObject
    Position Surface - Location
  )
  (:predicates
    (On ?x - Object ?y - Object)
    (OnTable ?x - MovableObject)
    (OnFloor ?x - FixedObject)
    (OnWall ?x - Object)
    (At ?x - Object ?y - Location)
    (Near ?x - Object ?y - Object)
    (In ?x - Drink ?y - Refrigerator)
    (InHand ?x - MovableObject ?y - Hand)
    (EmptyHand ?x - Hand)
    (OpenedRefrigerator)
    (ClosedRefrigerator)
  )
  (:action PutIntoRefrigerator
    :parameters (?Pos - Location ?Obj - MovableObject ?Hand - Hand)
    :precondition (and (At Refrigerator ?Pos) (At Robot ?Pos) (At ?Obj ?Pos)
      (OpenedRefrigerator) (InHand ?Obj ?Hand))
    :effect (and (In ?Obj Refrigerator) (EmptyHand ?Hand) (not (InHand ?Obj ?Hand))))
)
```

그림 5. PDDL 로 정의된 로봇 동작의 예

목표가 발생하면 JAM 은 미리 작성된 일반적인 계획을 통하여 이를 해결하려고 시도하게 되는데, 그림 6 은 계획 실행기인 JAM 에서 작업 계획기인 SAPA 에 의해 계획이 생성되고 수행되는 과정에서 사용되는 세 개의 JAM 계획을 보이고 있다.

JAM 에 미리 작성된 계획에서 적합한 행위를 찾지 못하게 되면 계획을 통한 문제 해결을 시도하게 되며 첫 번째로 그림 6 의 첫 번째 JAM 계획이 수행 된다. 여기에서 JAM 계획은 요구되는 목표를 검사하고 목표가 존재하면 계획기에 계획을 요구하는 부속 목표를 세우고, 생성된 계획은 JAM 의 지식 영역에 기록되고 미리 작성된 생성된 계획을 수행하는 두 번째 JAM 계획이 수행된다. 이 JAM 계획에서는 계획기인 SAPA 에 목표 도달을 위한 계획을 요청하게 된다. 이 때, JAM 의 현재 형태의 목표를 가질 수 있는 JAM 과 PDDL 은 목표 기술 형태의 차이를 보이게 되는데 이는 JAM 의 월드 모델을 대응해 주는 별도의 함수로 일치시켜주게 된다.

```
Plan:
{
  GOAL:
  ACHIEVE acceptCommand;
  BODY:
  RETRIEVE command $CMD;
  OR
  (
    TEST (= $CMD "NULL");
    ACHIEVE planning;
  )
  (
    TEST (= $CMD "NULL");
    FAIL;
  )
);
-----
Plan:
{
  GOAL:
  ACHIEVE planning;
  BODY:
  RETRIEVE environment $Env;
  RETRIEVE command $Cmd;
  EXECUTE MakePlan $Env $Cmd $Result;
  UPDATE (planResult) (planResult $Result);
  UPDATE (command) (command "NULL");
  ACHIEVE runPlan;
}
-----
Plan:
{
  GOAL:
  ACHIEVE runPlan;
  BODY:
  RETRIEVE planResult $Result;
  RETRIEVE environment $Env;
  EXECUTE RobotAction $Env $Result;
  UPDATE (planResult) (planResult "NULL");
}
)
```

그림 6. 계획 생성 및 수행 과정의 JAM 계획

SAPA 에 의해 계획이 성공적으로 작성될 경우 생성된 계획은 JAM 의 지식 영역에 기록되고 계획 수행을 위한 부속 목표가 추가되며 이에 따라 세 번째 JAM 계획이 수행된다. 이 계획에서 JAM 은 계획기에 의해 생성된 계획에 따라 순차적으로 행동 명령(action command)들을 내려 보냄으로써 가상 환경내의 로봇을 움직이게 한다.

이 세가지 JAM 계획에서 각각의 단계에서 환경의 변화나 정보의 부족 등의 이유로 계획의 생성 또는 수행이 실패할 경우 JAM 은 새로이 목표를 수행하기 위한 계획을 찾아 미리 작성된 JAM 계획을 수행하거나 계획기에 재계획을 요청한다.

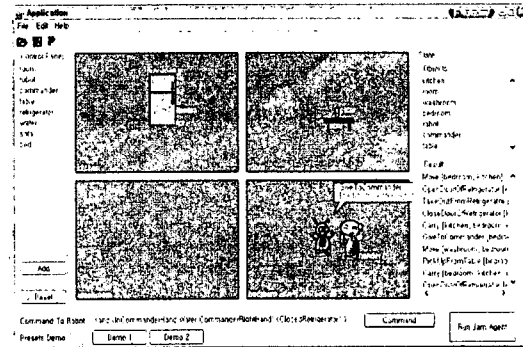


그림 7. 데모시스템의 사용자 화면

사용자는 그림 7 과 같은 데모시스템 사용자 화면을 이용하여 새로운 작업 환경과 초기 상태를 구성할 수 있다. 그리고 사용자 로봇의 새로운 작업 목표를 하나 입력하면, 자동적으로 JAM 내부의 월드모델과 목표집합에 각각 현재 상태와 목표조건이 표현된다. 사용자는 그림 7 의 화면 우측에서 새로 생성된 작업 계획을 텍스트(text) 형태로 확인 할 수 있고, 중앙의 그래픽 환경을 통해 로봇이 실제 동작하는 것을 볼 수 있다.

V. 결론

본 논문에서는 계획기반의 지능 로봇 제어를 가정하고, BDI 기반의 작업 계획 실행기와 주문형 작업 계획기를 어떻게 연동하는 것이 효과적인지를 알아보았다. BDI 구조 자체도 목표에 대한 계층적 분해과정을 통한 반응적 계획기능을 가지고 있으므로, 향후 연구에서는 BDI 구조를 전통 작업 계획기와 서로 다른 레벨의 작업 계획을 담당하도록 양자간을 연동하는 방식에 대해 연구해 보는 것도 가치가 있을 것으로 판단된다.

참고문헌

- [1] V. Verma, et al., "Survey of Command Execution Systems for NASA Spacecraft and Robots," *Proc. of the Workshop on Plan Execution, Int. Conf. Automated Planning and Scheduling*, 2005.
- [2] A. Rao and M. Georgeff, "BDI-agents: from Theory to Practice," *Proc. of the 1st Int. Conf. Multiagent Systems*, 1995.
- [3] J. Lee, M. Huber, E. Durfee, and P. Kenny, "UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications," *Proc. of the Conf. Intelligent Robotics in Field, Factory, Service, and Space*, pp. 842-849, 1994.
- [4] M. Huber, "JAM: A BDI-Theoretic Mobile Agent Architecture", *Proc. of the 3rd Int. Conf. Autonomous Agents*, pp. 236-243, 1999.
- [5] M. Do and S. Kambhampati, "Sapa: A Multi-objective Metric Temporal Planner," *Journal of Artificial Intelligence Research*, pp. 1-40, 2003.