제25회 한국정보처리학회 춘계학술발표대회 논문집 제13권 제1호 (2006. 5)

계획 시스템에서 이정표상태 생성에 관한 연구

신재혁* 한현구
한국외국어대학 컴퓨터 및 정보통신공학과
{shin6060*, hghan}@hufs.ac.kr

Milestone States Creation in Planning Systems

Jaehyuk Shin* Hyungoo Han
School of Computer & Information Communications Engineering,
Hankuk University of Foreign Studies

요 약

계획기법 이란 목적을 달성하기 위해 일련의 행동 순서를 찾는 것이다. 계획 기법을 통해 수립된 많은 행동들을 로봇 등 에이전트가 실행할 때 예기치 않은 문제들이 발생하게 된다. 이러한 문제들을 실시간으로 해결하여 에이전트가 주어진 목적을 달성할 수 있도록 하여야 한다. 그러한 문제점을 해결하는 방법에는 반응 계획기법, 재 계획기법, 그리고 지역 복구기법 등이 있다. 본 논문에서는 재 계획기법과 지역 복구기법의 장점을 가지는 이정표상태 계획기법을 간단히 제안하고 이정표 상태의 생성 방법을 연구하였다. 이정표 상태들은 수립된 계획의 모의 실행을 통하여 생성하였으며 행동 실행에 관련된 조건들의 criticality value를 이용하는 방법과 계획에 나타나는 행동들의 빈도수를 이용하는 방법을 제안 하였다.

1. 서 론

1.1 연구배경 및 목적

계획이란 로봇 등 에이전트들이 주어진 임무를 달성할 수 있도록 사전에 수립된 에이전트들의 일련의 행동들이다. 에이전트로 하여금 주어진 문제를 해결하도록 하기 위하여 계획시스템은 계획을 수립하여 에이전트에게 전달한다. 계획 시스템은 주어진 문제의 초기상태와 성취하고자 하는 목적 상태를 가지고 탐색과 추론과정을 거쳐 계획을 수립한다.[1][7].

수립된 계획을 탑재한 에이전트는 일련의 행동들을 하나씩 수행하게 되는데, 수행하여야 할 행동의 선행조건들이 만족되면 현재의 상태에서 행동을 실행한다. 이 때 현 상태에서 행동을 실행하면 상태의 전이가 발생하게 된다. 실행된 행동을 중심으로 볼 때 그 행동의 선행조건들이 만족되어져 있던 직전의 상태에서 행동 실행의 효과가 반영된 직후의 상태로 전이되며 바로 다음 행동은 새로 생성된 상태를 직전의 상태로 하여 실행하게 된다. 이러한 방법으로 일련의 행동들을 아무런 문제 없이 모두 수행하면 원하는 목적상태를 달성하게 된다. 그러나 에이전트가 계획을 실행하는 도중에 에이전트의 결함, 오류동작, 또는 외부 이벤트, 도메인의 불확실성 등 여러 가지 요인으로 인하여 더 이상 에이전트가 행동을 실행할 수 없는 상황이 발생할 수 있으며 이러한 우발적인 상황을 본 논문에서는 "오류"라고 하였다[2].

에이전트가 이러한 오류를 접했을 때 이를 해결하기 위하여 모든 동작을 멈추고 장시간 탐색과 추론과정을 가진다면 매우 어려운 치명적인 상황에 처할 수도 있다. 예를 들어 8차선 대로의 횡단보도를 건너는 중에 오류가 발생했다면? 수술 도중이라면? 전쟁 중 적과 접전 중이라면? 그러므로 이러한 오류들을 효과적이고 효율적인 방법으로 해결할 수 있어야 한다. 기존의 오류 극복 방법들에는 수립된 계획을 모두 폐기하고 현재 상태에서부터 다시 계획을 수립하는 재계획 방법, 오류가 발생한 현재 상태의 일부분에 지엽적인 교정을 하는 지역 복구 방법이 있다. 또한 사전에 전체 계획을 수립하지 않고 현재상태에서 바로 다음 상태로 전이하기 위한 행동을 하나씩 추론하는 반응 계획 방법이 있는데 이는 모든 오류를 현재 상태의 조건들로 생각하여 다음 행동을 추론한다.

재계획 방법은 오류복구 알고리즘은 간단하나 현재 계획 진행 정도에 따라 사전에 수립된 많은 행동들을 버릴 수가

```
(1) pickup(x)
      Precondition & Delete list : Ontable(x)
                                 , Clear(x), Handempty
      Add formula : Holding(x)
```

[그림 1] STRIPS 행동 규칙.

있으므로 낭비가 있으며 재계획 시간도 많이 필요할 수 있다. 지역 복구 방법은 현재 수행 불가능한 행동의 선행조건들을 만족시키는 방법으로 기존의 행동들은 다시 사용할 수 있으나 선행조건을 만족시키기 위하여 새로이 행동들을 추론 및 탐색하여 추가해야 한다. 또한 추가된 행동들로 인한 기존의 행동들에 미치는 영향을 모두 검사하여 계획의 참을 유지해야 한다. 이를 truth maintenance라고 하며 알고리즘이 복잡하고 시간을 요구한다. 반응계획 방법은 행동을 실

시간에 하나씩 생성하여 실행하는 방법으로서 이론상으로는 모든 오류를 현재의 상태에 포함할 수 있으므로 가장 합리적인 오류극복 시스템이라고 할 수 있다. 그러나 목적상태에 도달하기 위한 탐색과 추론을 부분적으로 실행하여야 하므로 목적상태를 찾지 못하거나 매우 비효율적인 문제해결을 제시할 수도 있다.

본 논문에서 제안한 이정표 상태를 이용한 오류복구 방법은 계획의 사전 모의 실행에 의하여 생성되는 상태들 중에서 적당한 상태들을 선택하여 이정표상태로 정한다. 그리고 이러한 이정표상태들 이용하여 기존의 오류복구 방법의 장점을 모두 가지는 새로운 오류복구 방법을 제안하였으며 3장에서 더욱 상세히 설명한다. 본 논문에서는 이러한 이정표상태 생성을 위하여 행동 실행에 관련된 조건들의 criticality value를 이용하는 방법과 계획에 나타나는 행동들의 빈도수를 이용하는 방법을 제안 하였다.

2. 관련연구

2.1 계획수립시스템

계획 수립 시스템은 크게 비계층적 계획시스템과 계층적 계획시스템으로 분류할 수 있다. 비계층적 계획시스템은 목적은 부 목적 등으로 계층화 하지만 각 목적상태들을 달성하기 위하여 일련의 문제 풀이 행동들을 나열하여 계획을 수립한다[7]. 대표적인 계획시스템으로 STRIPS[3]가 있다. 본 논문에서 사용하는 행동들의 표현은 STRIPS 행동 표현방법을 사용한다. STRIPS는 행동을 Precondition, Delete list 그리고 Add formula로 표현한다. Precondition은 어떤 행동을 수행하기 위하여 필요한 선행조건들이며, Delete list는 그 행동이 수행된 후 삭제되는 조건들이다. Add formula는 행동 수행 후 추가되는 조건을 나타낸다. STRIPS의 행동 규칙을 [그림1]과 같이 간단한 block 세계를 예로 들어 설명한다.

[그림1]의 'Pickup' 이란 행동을 하기위해서는 선행조건으로 'Ontable(x), Clear(x), Handempty'가 필요하다. 그리고 선행조건은 행동이 실행된 후 현재 상태에서 삭제되므로 Delete list에 해당된다. 또한 행동이 실행된 후 'Holding(x)'는 다음 상태에 추가되는 조건이다. 이 시스템의 문제 해결을 위하여 초기상태와 목적상태의 차이점을 계산하고 그 차이점을 줄일 수 있는 행동들을 탐색해 나간다. 차이점을 줄일 수 있는 행동들이 많을 경우 모든 가능한 행동들에 대하여 탐색을 하므로 탐색공간이 커질 수 있다.

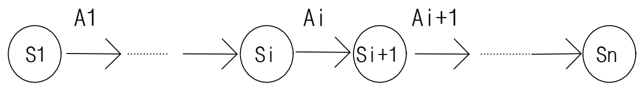
계층적 계획시스템은 목적의 부목적화를 통한 계층화외에 계획의 표현에 대한 계층화를 이룬다. 하나의 계획을 높은 층에서는 개략적이고 추상적인 계획을 수립하고, 낮은 층으로 갈 수록 조금씩 세분화된 계획을 수립해 나간다[7]. 대표적인 시스템으로는 ABSTRIPS[4], NOAH, MOLGEN 등이 있다. ABSTRIPS는 조건들을 criticality 값으로 분류하여 그 값이 높은 부류의 조건들부터 해결해 나가는 방법이며 현재 값보다 낮은 값을 가지는 조건들은 무시한다. 현재 단계에서 계획이 끝나면 다음 단계의 criticality value를 가지는 조건들을 해결해 나가는 방법으로서 criticality 값을 조건들에 부여하는 방법은 매우 주관적이며 도메인에 따라 다르고 그 값이 제대로 부여되지 않으면 계획을 수립할 수 없을 수도 있다.

2.2 오류복구방법

계획수립의 결과로 얻어지는 계획은 행동들의 나열이다. 에이전트는 행동들의 나열인 계획을 수행하며, 계획을 수행하는 도중에 예기치 않은 상황이 발생하면 에이전트가 더 이상 계획을 진행할 수가 없게 된다. 오류 발생 시점에서 시스템은 오류 복구를 하기 위한 적절한 방법을 선택해야 한다. 여러 가지 선택 기준 중에서 중요한 두 가지 요소는 응답시간과 효율성이다. 응답시간은 시스템이 오류 복구를 하는데 필요한 시간이고, 효율성이란 오류 복구에 의하여 새로이 생성되어 추가된 행동들의 수를 말한다. 오류를 복구하기 위해서 에이전트가 모든 행동들의 사전과 사후 상태에 관해 완전한 자료를 가지고 있으면 매우 용이하다. 그러나 오류복구를 위하여 이러한 모든 상태정보를 에이전트에 적재하는 것은 비효율적이다. 오류 복구 방법에는 대표적으로 재계획방법[5], 지역 복구방법, 그리고 반응 계획방법[6]이 있다.

3. 이정표상태계획시스템

계획을 모의 실행하면서 이정표상태를 생성하는 방법은 [그림2]와 같은 상태전이 과정을 거쳐야 한다. 행동들을 순서적으로 실행시키고 그 실행의 결과를 직전의 상태에 적용하여 다음 상태로 전이 시킨다. [그림2]에서 초기상태가 S1이며 Sn이 목적상태이다. 행동 Ai의 선행조건들은 상태 Si에 상태요소로 존재하며 그러한 선행조건들이 만족되면 행동 Ai가 실행되며 그 실행의 결과를 반영하여 상태 Si+1로 전이되고 행동 Ai+1이 다음으로 실행될 것이다. 이러한 상태들 중에서 이정표상태를 추출하여 저장하고 나머지 상태들은 버린다.



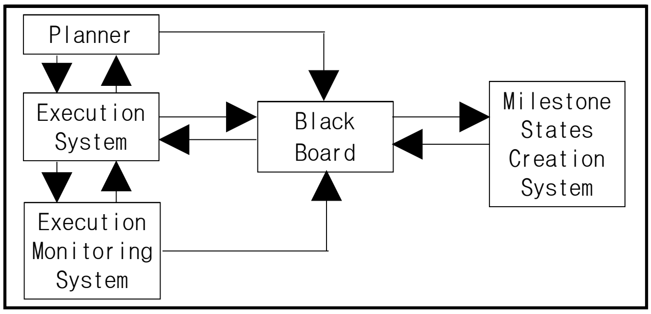
[그림2] 상태전이 과정.

3.1 이정표상태를이용한오류복구방법

본 논문에서 제안한 이정표상태 계획기법은 생성된 이정표 상태를 에이전트에 탑재시키고 오류가 발생하였을 때 현재 오류 발생 상태에서부터 목표 상태로의 전진 방향 중 첫번째 위치한 이정표상태까지만 계획을 부분 수립하여 오류를 처리한다. 이 방법은 현재 상태와 해당 이정표상태 사이의 행동들만을 버리고 나머지 행동들은 재사용할 수 있다. 또한 오류가 발생한 상태를 초기상태로 하고, 해당 이정표상태를 목적상태로 하여 기존의 계획자가 부분 계획을 수립하므로 오류처리를 간단하고 효율적으로 수행할 수 있다.

3.2 이정표상태계획시스템구조

[그림 3]과 같이 이정표상태 계획시스템은 총 5개의 모듈로 되어 있다. Planner는 주어진 초기상태와 목적상태를 가지고 계획을 수립하여 Black Board에 저장한다. Milestone States Creation System은 Black Board에 수록된 계획을 모의 실행하여 이정표상태를 생성하고 이를 Black Board에 저장한다. Black Board에는 계획, 초기상태, 목적상태 그리고 이정표상태가 저장되어 있으며 Execution System은 실제로 에이전트에게 명령을 내려 계획을 실행한다. Execution Monitoring System은 에이전트의 계획 실행을 감시하며 계획을 수행하는 도중 예기치 않은 상황의 발생으로 오류가 발생하면 이를 감지 및 분석하여 그 오류 상태를 Black Board를 기록한다. Execution System은 선행조건이 만족되어져 있지 않은 등 다음 행동을 실행할 수 없는 경우가 생기면 계획 실행을 멈추고 Planner에게 오류복구를 지시하며, Planner는 Black Board에 수록된 현재 오류상태를 초기상태로 하고, 다음 이정표상태를 새로운 목적상태로 하여 부분 계획을 수립하고 이를 Black Board에 기록한다. 그리고 Execution System에 오류복구 완료를 보고하며 Execution System은 새로이 수록된 부분 계획을 기존의 남은 계획 이전에 위치시켜서 계획을 재실행한다.



[그림3] 시스템 구조.

3.3 이정표상태생성방법

이정표상태 계획시스템에서 이정표상태를 생성하기 위하여 ABSTRIPS의 criticality value의 누적치를 이용하는 방법과 행동의 빈도수를 이용하는 방법을 제안하였다. Criticality value를 사용한 방법은 각 행동들의 선행조건들에 criticality value를 부여하여 이러한 모든 행동들에 대하여 선행조건들의 value들을 누적하고 이를 이정표상태의 생성에 이용하였다.

두 번째 방법인 연산자의 빈도수를 이용하는 방법은 완료된 계획에서 연산자들의 빈도수를 분석하여 이정표상태 생성에 적용하였다.

3.3.1 Criticality Value를 이용한 방법

계획을 수립할 때 필요한 모든 조건들에 대하여 criticality value를 ABSTRIPS와 같이 적절히 부여 하였다. 에이전트에 관련된 조건들은 낮은 값을 부여하고 에이전트 행동의 목표가 되는 개체에 관련된 조건들은 높은 값을 부여 하였다. 예를 들어 [그림1]에서 handempty는 에이전트에 관련된 조건이고 clear는 개체에 관련된 조건이다. 계획이 수립된 후 계획의 모든 행동들에 대하여 선행조건들을 추출하고, 그 선행조건들의 criticality value를 누적하여 해당 행동의 criticality value로 한다. 누적된 높은 criticality value 행동들의 선행조건들은 계획수립과 실행과정에서 중요한 부분이므로 높은 누적 criticality value를 가지는 행동들의 바로 전 상태를 이정표상태로 생성한다. 본 논문에서 사용한 실험 도메인에서는 가장 높은 누적치를 가지는 행동들의 직전 상태들을 이정표상태로 정하여도 무방하였다. 그러나 criticality value의 부여방법이 적절하지 않든가 주어진 문제의 성격 또는 계획의 길이에 따라서 생성되는 이정표상태의 수가 너무 많거나 적을 수 있다. 이정표상태가 너무 많으면 이러한 이정표상태를 에이전트가 모두 저장하고 있어야 하는 부담이 있고, 너무 적으면 오류복구 시 부분적인 재계획의 길이가 길어지므로 기존 계획의 많은 부분을 버리게 되는 낭비와 오류복구시간이 또한 부담이 된다. 이러한 문제를 해결하기 위하여 criticality value를 이용하여 생성된 이정표상태들의 수를 인위적으로 적절히 수정할 필요가 있다.

3.3.2 연산자빈도수를이용한방법

수립된 계획에는 반복되는 행동들이 많이 있다. 반복된 행동들의 빈도를 측정하고 그 중앙값을 구하여 이정표상태를 생성하는데 사용한다. 반복되는 행동들의 반복 횟수의 총합을 구하고 이의 중앙값의 빈도를 보이는 행동들의 직전의 모든 상태를 이정표상태로 결정한다. 빈도수의 중앙값을 택하는 이유는 빈도수가 많은 연산자들의 직전 상태들을 이정표상태로 결정하면, 이정표상태의 수가 너무 많아지고 또한 빈도수가 적은 행동들의 직전 상태들을 이정표상태로 결정할 경우에는 그 수가 너무 적어진다.

연산자의 빈도수를 이용한 방법은 계획시스템에서 인위적인 방법으로 연산자의 수를 분석하여 이정표상태를 생성하는 것으로 3.2.1절의 criticality value를 이용한 방법 보다는 이정표상태의 분포가 고르기 때문에 이정표상태가 많을 경우와 적을 경우의 발생을 줄일 수 있다. 하지만 행동들과 관련된 조건들의 중요도를 무시했기 때문에 큰 의미없는 행동의 선행조건을 가지는 상태를 이정표상태로 정할 수 있는 단점이 있다.

4. 실험

4.1 이정표상태생성실험

[표1]은 본 논문에서 제안한 두 가지 방법으로 이정표상태들을 생성한 후의 결과를 보이고 있다. 수립된 계획의 행동들의 수가 적을 때는 이정표상태의 수와 전체 상태 수와의 비율이 다소 안정적이지 못하다. 그러나 상대적으로 수립된 계획의 행동들의 수가 많을 때는 그 비율이 안정적이다. 비율이 안정적인 것은 수립된 계획의 행동의 수와 무관

하게 이정표상태의 수가 일정한 비율로 생성되어 만족스러운 경우이다. 그러나 전체적으로 보았을 때 criticality value를 사용한 방법보다는 행동의 빈도수를 이용한 방법이 아주 안정적이다. 그러나 이정표상태의 개수가 다소 많은 편이다. 3장에서 언급한 것과 마찬가지로 생성된 이정표상태의 수를 인위적으로 수정할 필요가 있을 수 있다. 실험에 의하면 대체적으로 이정표상태의 수의 비율은 전체 상태 수의 10%에서 20%정도이면 그 정도가 적당하게 나타나고 있다. 빈도수를 이용한 방법은 이정표상태를 생성하는데 있어 구하는 방법이 간단하고, 시간적 비용으로 볼 때 효율적 이지만 행동들의 계획에서 차지하는 비중이나 조건들의 criticality value를 무시함으로서 무의미한 이정표상태를 생성하게 되어 오류 복구 시 이러한 무의미한 이정표상태가 목적상태로 결정되어 필요 없는 부분계획이 수립될 수도 있다.

[표1] 실험 결과.

예제 문제	Number Of Action ①	Criticality Value 방법		연산자 빈도수 방법	
		개수 ②	비율 ③	개수 ④	비율 ⑤
PLAN 1	25	3	12.0%	4	16.0%
PLAN 2	26	2	7.7%	6	23.1%
PLAN 3	68	10	14.7%	16	23.5%
PLAN 4	106	16	15.1%	25	23.6%

① Number of Action : 수립된 계획의 총 행동 수.
② Criticality Value 방법에 의하여 생성된 이정표상태 수.
③, ⑤ 전체 상태 수에 대한 이정표상태의 비율.
④ 연산자 빈도수 방법 의하여 생성된 이정표상태 수.

5. 결론 및 향후 연구과제

많은 비용으로 수립된 계획을 에이전트가 실행할 때 예기치 않은 상황으로 더 이상 계획을 실행할 수 없는 경우가 발생한다. 이러한 상황을 극복하기 위하여 기존의 오류 복구시스템들의 장점들을 가지는 이정표상태를 이용한 오류 복구시스템을 간략하게 소개하고 이러한 이정표상태를 생성하는 방법을 제시하였다.

본 논문에서 제안한 두 가지 방법 모두 수립된 계획의 행동들의 수가 적을 때는 생성된 이정표상태의 수가 안정적이지 못했다. 이는 계획의 길이가 작을 때는 오류극복을 위한 복잡한 알고리즘이나 방법이 필요하지 않을 것이다. 간단한 문제 도메인에서는 오류가 발생하면 기존의 수립된 계획을 버리고 재계획을 수행하는 것이 더 효율적일 수 있다. 두 가지 방법 모두를 사용하여 만족한 비율을 얻었다고 하나 고려해야할 사항이 더 있다. 즉 이정표상태의 분포 상황이다. 이정표상태가 한곳으로 집중된다면 부분적으로 집중된 곳은 이정표상태의 수가 너무 많아서 부적절하고 집중되지 않은 곳은 그 수가 너무 적어서 문제가 된다. 향후 이정표상태의 개수뿐이 아니고 그 분포도 고려한 이정표상태 생성 방법을 연구해야 할 것이다.

참고문헌

[1] Malik Ghallab, Dana Nau, Paolo Traverso."Automated Planning Theory and Practice", pp 17-109, 2004.
[2] H. Han, "Error Recovery for Multiagent Planning Systems", Ph.D. Dissertation, Computer Science and Engineering, Auburrn University, Auburn, AL, December 1990.
[3] R.E Fikes, P. E. Hart, and N. J. Nilson, "Learning and Executing Generalized Robot Plans", Artificial Intelligence, Vol. 3, pp. 251-288, 1972.
[4] E. D. Sacerdoti, "Planning in a Hierarchy of Abstraction Space", Artificial Intelligence, Vol. 5 pp. 115-135, 1974.
[5] J.H. Muson, "Robot Planning, Execution, and Monitoring In An Uncertain Environment," Proc. of Int. Joint Conf. on Artificial Intelligence-71, London, England, pp. 338-349, 1971.
[6] P.S. Ow, S.F Smith, and A. Thiriez, "Reaction Plan Revision", Proc. of National Conf. on Artificial Intelligence, pp. 77-82, 1988.
[7] 도용태 "인공지능 개념 및 응용" (사이텍미디어) pp. 161-195, 2001.