

지능적인 웹서비스를 위한 강건한 계획 생성과 동적 실행 방법

Robust Plan Generation and Dynamic Execution for Intelligent Web Service

황경순, 이승희, 이건명

충북대학교 전기전자컴퓨터공학부, 첨단정보기술연구소(AITrc)
hks@aicore.cbnu.ac.kr, shlee@aicore.cbnu.ac.kr, kmlee@cbnu.ac.kr

요 약

웹 서비스와 같은 분산된 환경에서, 특정 서비스를 수행하기 위해서는 원격의 컴퓨터나 사이트상에서 다중 에이전트들의 협업을 통해 이루어진다. 이때 서비스는 여러 에이전트들의 복잡한 행위들에 의해 구성된다. 또한 지능적인 서비스를 위해서는 에이전트들의 상태정보, 목적정보, 그리고 계획정보 등을 이용한다. 특히 계획정보는 에이전트들이 일련의 행위들로 구성된다. 하지만 계획수립을 위한, 기존 연구들 대부분은 정적으로 기술된 서비스 명세와 초기상태 정보를 이용하여 특정 목표를 만족시키는 단일 계획 생성 방법을 연구해왔다. 따라서 계획수립이 실행 도중에 기대하지 않은 네트워크 장애나 방해 등으로 서비스 수행을 실패하는 경우, 그 계획은 무효가 되고 다시 계획을 생성 해야만 한다. 그러나 다시 계획을 생성하기 위해서는 많은 시간을 소비하게 될 뿐만 아니라 태스크 중복이 불가피하므로 매우 비효율적이다. 이 논문에서는 강건한 계획수립과 그 계획을 실행하기 위한 효과적인 방법을 제안한다. 즉, 계획수립의 재생성을 피하기 위한 방법으로 단일 계획수립 대신에 실행 가능한 다중 계획들로 표현된 강건한 계획을 생성하는 것이다. 강건한 계획의 행위들이 실행되는 동안, 각 단계마다 실행 가능한 행위를 선택한 후, 그 행위를 실행한다. 그러나 선택된 행위가 실행결과를 낼 수 없을 경우, 대체 가능한 서브 계획 경로를 선택하여 실행한다. 강건한 계획을 표현하기 위해 페트리 넷 기반의 방법을 제안한다. 강건한 계획 생성 방법에서는 이용 가능한 모든 계획들을 입력으로 사용한다. 그 계획수립 방법은 HTN 계획수립기로 잘 알려진 JSHOP2 계획수립기 내에 구현하였다. 계획 실행 방법으로는 주어진 강건한 계획에 대하여 행위들이 직접 실행할 수 있도록 한다.

Key Words : 웹서비스(web service), 강건한 계획(robust plan), 계획수립(planning), JSHOP2

1). 서 론

정보(information)와 세부계획(logistics) 인프라가 발전함에 따라, 정보 처리 태스크들이나 제조업 태스크들이 네트워크상이나 나라에 걸쳐서 수행되고 있다. 이러한 복잡한 태스크들은 여러 분산된 참여자들에 의해 처리될 수 있는 다중 서브태스크들로 구성된다. 계획수립기(planner)는 주어진 태스크를 성취할 수 있는 일련의 행위(action)들로 구성되도록 개발되어 왔다[6]. 웹 서비스와 같은 분산된 환경에서, 계획된 행위들은 원격 사이트에서 실행될 수 있으므로 네트워크 장애나 공격과 같은 다양한 외부적인 방해에 대한 취약점을 가지고 있다[5, 6]. 계획이 실행(execution)되는 동안, 그 계획의 행위들 중 하나라도 실행 결과를 낼 수 없다면 더 이상 그 계획은 진행될 수 없게 된다. 이런 상황에서, 주어진 태스크를 성취하기 위하여 새로운 계획을 다시 만들어야 한다. 하지만, 일반적으로 계획수립(planning) 태스크가 계획들을 생성하는데 많은 시

간이 걸리게 되므로, 계획 실행이 실패할 때마다 새 계획을 다시 만드는 것은 너무 비효율적이다.

이러한 문제를 다루기 위하여, 인터리브 계획수립 접근방법은 부분-계획수립이나 부분-계획실행을 교대로 수행하도록 제안되었다. 하지만 인터리브 접근방법이 모든 계획수립 문제에 적용할 수 없고, 부분적인 계획수립이 적절한 시간 내에 실행할 수 있어야 하므로 매우 제한적이다[1].

이 논문에서는 자체에 다중 서브-계획 실행 경로들을 포함한 하나의 강건한 계획을 이용한 다음 현재 서브-계획 실행 경로가 진행하는 것을 실패했을 때, 대체 실행 경로를 선택할 수 있도록 하는 새로운 접근방법을 제안한다. 또한 페트리 넷 기반(Petri-net)모델에 다중 서브-계획 경로를 표현하기 위한 강건한 계획 표현방법과 강건한 계획을 만들기 위한 방법을 소개한다. 행위들의 실행이 실패하기 쉬운 분산 환경에서 어떻게 강건한 계획을 실행하는지를 기술한다. 제안된 계획 생성 방법은 계층적 태스크 네트워크 (HTN : Hierarchical Task Network) 계획수립기인 JSHOP2의 수정 버전에 구현했다.

본 연구는 첨단정보기술연구소(AITrc)를 통해서 과학재단 지원으로 수행된 것임.

이 논문의 구성은 다음과 같다. 2절에서 관련연구를 소개하고, 3절은 강건한 계획의 어떻게 표현되고 가를 소개하고 그것을 어떻게 생성하는 가를 설명한다. 4절은 강건한 계획 실행 방법에 대해 기술하고, 5절에서는 계획수립과 실행을 적용하기 위한 간단한 시나리오 소개한다. 6절에서는 결론 및 향후과제를 기술한다.

2. 관련연구

계획수립은 인공지능과 그 외에 다른 분야에서 수십 년에 걸쳐 연구되어 왔던 전통적인 문제이다. 계획수립은 그것을 실행하는데 시간이나 공간에 관련된 많은 자원들을 요구하는 매우 어려운 문제로 잘 알려져 있다. 대표적인 계획수립 접근방법으로는 상태-공간 계획수립, 계획-공간 계획수립, 그리고 HTN 계획수립기 등이 있다. 상태-공간 계획수립과 계획-공간 계획수립에서 계획수립 도메인은 실세계의 변화를 연산자(operator)들로 기술한다. 연산자들은 연산자 적용을 만족하는 전제조건과 삭제 리스트와 추가 리스트로 구성되는 결과로 정의된다. HTN 계획수립에서, 계획수립 도메인은 연산자와 메서드(method), 그리고 공리(axiom)로 기술한다 [1,3].

상태-공간 계획수립에 대해서는 그래프 계획 기반 방법과 SAT(satisfiability) 기반 방법이 널리 사용되고 있다[1,4,9]. 계획-공간 계획수립에서는 PGP(partial global planning) 기반 계획수립기들이 있다[1, 10]. SHOP(simple hierarchical ordered planner) 기반의 계획수립기는 HTN 계획수립으로 잘 알려져 있다. 이 계획수립기들은 주어진 태스크들을 수행할 수 있도록 일련의 행위들을 나열하여 계획들을 생성한다. 그러므로 일단 그 계획의 행위들 중 하나가 실패를 하면, 전체적인 계획이 무효가 되고 실패 원인이 되는 그 행위를 배제한 새로운 계획을 재생산해야 한다[1,3]. 인턴리브 계획수립은 계획수립과 실행을 교대로 수행하는 재-계획수립(replanning) 문제를 다루기 위해 제안되었다. 하지만 이것은 모든 계획수립 문제에 적용할 수 없으며, 부분적인 계획수립이 부가된 시간적 제약조건에서 이용 가능한 강한 제약을 가진다.

HTN 계획수립기는 가장 폭넓게 사용되는 계획수립들 중 하나이다. SHOP은 주어진 계획수립 문제에서 사용 가능한 모든 계획을 찾을 수 있다. SHOP의 소스코드(source code)는 공개되어 있으며, JSHOP2는 SHOP 계획수립기를 자바로 구현한 것이다[2].

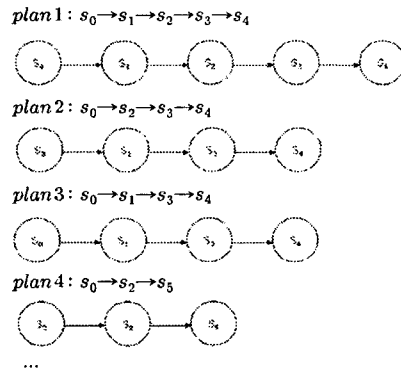
이 논문에서는, 인턴리브 계획수립에서 제공되지 않는 도메인에 대하여 계획 실행의 실패를 극복하기 위하여 강건한 계획을 사용할 수 있도록 제안한다. 제안한 강건한 계획 생성 방법은 JSHOP2 계획수립기에 구현하였다.

3. 강건한 계획 생성

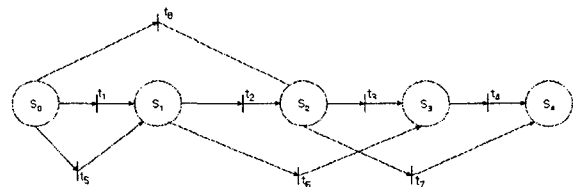
3.1 강건한 계획 표현

강건한 계획은 모든 이용 가능한 서브-계획 실행 경로를 나타낸 계획 표현(plan representation)으로

서 정의된다. 그러므로 하나의 강건한 계획으로부터, 다중 계획들이 하나의 태스크를 성취시킬 수 있도록 추출하는 것이다. 강건한 계획을 표현하기 위한, 페트리 넷 모델은 동시성(concurrency)과 동기적인 사건(synchronized event)을 표현할 수 있으며 가시적으로 표현이 가능하여 이해하기 편리하다는 장점이 있다. 강건한 계획은 행위에 대응되는 트랜지션(transition)과 상태에 대응되는 플레이스(place)로서 페트리 넷 모델 기반으로 표현한다. 대체 서브 계획 경로는 대응되는 플레이스로부터 분기된다. 한편 페트리-넷에서는 행위에 대응되는 트랜지션은 여러 조건을 만족해야 한다. 이러한 조건들에 대한 정보는 플레이스에 저장된다. 이때 어떤 플레이스는 한 트랜지션의 입력의 조건이 되기도 하고 어떤 플레이스의 출력이 되기도 한다. [그림 1]은 이용 가능한 서브-계획 리스트이다. [그림 2]는 페트리 넷 모델 기반의 표현된 강건한 계획이다.



[그림 1] 서브-계획 리스트



[그림 2] 페트리 넷 모델 기반의 표현된 강건한 계획

[그림 2]에서 다음과 같은 7개로 코딩된 강건한 계획을 보여준다. (t_1, t_2, t_3, t_4) , (t_8, t_3, t_4) , (t_8, t_7) , (t_5, t_2, t_3, t_4) , (t_1, t_6, t_4) , (t_5, t_6, t_4) , (t_5, t_2, t_7)

3.3 계획생성

강건한 계획을 생성하기 위하여, 기존 계획수립기에서 생성된 일련의 행위들로 모든 이용 가능한 계획들이 있다고 가정한다. 따라서 입력정보로는 이용 가능한 계획들의 집합과 초기 상태정보를 갖는다. 출력결과는 페트리-넷 기반의 강건한 계획 표현이 된다. [그림 3]은 강건한 계획수립 생성 알고리즘이다.

```

procedure Generate_Robust_Plan
input :
 $P = \{P_1, P_2, \dots, P_n\}$  : the set of available plans,
 $P_i = (a_{i1}, a_{i2}, \dots, a_{in})$  where  $a_{ij}$  is the  $j$ -th action in the plan  $P_i$ 
 $s_0$  = initial state
output :
 $PN: (PL, TR, FL, LB)$  :Petri net-based robust plan
representation
 $PL$  : the set of places
 $TR$  : the set of transitions
 $F$  : flows between places and transitions
 $LB$  : the action labels for transition
begin :
Crate a place  $PL$  for  $s_0$ 
 $PL := \emptyset, TR := \emptyset, FL := \emptyset, LB := \emptyset$ 
for  $P_i \in P$  do
 $S = s_0$ 
for  $a_{ij} \in P_i$  do
 $TL_{exist} := find\_transition\_with\_action(PL, a_{ij})$ 
if  $TL_{exist} \neq \emptyset$ 
 $PL := the\ out\_going\ place\ of\ TL_{exist} \in PL$ 
 $s := state\ of\ PL$ 
else
Createa transition  $T$  for action  $a_{ij}$ 
 $LB := LB \cup \{(T, a_{ij})\}$ 
 $TR := TR \cup \{T\}$ 
 $F := F \cup \{(PL, T)\}$ 
 $S_{new} := apply\_action\_state(a_{ij}, S)$ 
 $PL_{exist} := find\_place\_with\_the\_same\_state(PL, S_{new})$ 
if  $PL_{exist} \neq \emptyset$ 
 $F := F \cup \{(T, PL_{exist})\}$ 
 $PL := PL_{exist}$ 
else
create a place  $PL_{new}$  for  $S_{new}$ 
 $PL := PL \cup \{PL_{new}\}$ 
 $F := F \cup \{T, PL_{new}\}$ 
 $PL := PL_{new}$ 
end if
 $S := S_{new}$ 
end if
end for
end for
end
    
```

[그림 3] 강건한 계획수립 생성 알고리즘

4. 강건한 계획 실행

프로시저 *execute-robust_plan*은 강건한 계획이 어떻게 계획 실행 엔진에 의해 실행되는 가를 보여 준다. 이때, 입력은 강건한 계획에 대한 페트리-넷 기반의 강건한 계획 표현이 된다. [그림 4]는 강건한 계획 실행 알고리즘이다.

```

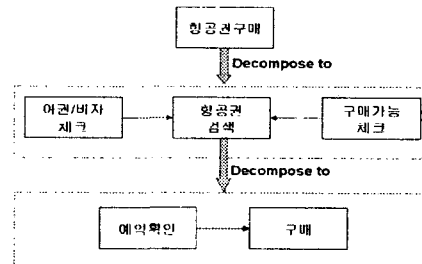
procedure Execute_Robust_Plan
input :
 $PN: (PL, TR, FL, LB)$  :Petri net-based robust plan
representation for a robust plan
begin :
Select a place  $PL$  with no incoming transition
do
Choose an unvisited outgoing transition  $TL$  of  $PL$ 
if there is no such transition  $TL$ 
Backtrack to the preceding place  $PL_{prev}$  on the
recent execution path
 $PL := PL_{prev}$ 
end
Execute the action  $a$  for the transition  $TL$ 
if  $a$  is executed successfully within time constraint
 $PL := the\ outgoing\ place\ of\ TL$ 
Bind the execution result to the place  $PL$ 
end if
while( $PL$  has no outgoing transition)
Return the result bound to  $PL$ , if exists.
end.
    
```

[그림 4] 강건한 계획 실행 알고리즘

5. 시나리오

계획수립과 실행을 위한 간단한 웹 서비스 시나리오를 다음과 같이 기술한다. 사용자는 웹 인터페이스를 통해 여행하고 싶은 목적지와 여행일정 등 여러 정보들을 입력한다. 예를 들어, “학회참석, 미국 뉴욕, 2007년 2월1부터 2007년 2월7일까지” 이 때, 여행을 위해 항공권 예약 서비스를 제공하고자 한다. 즉 항공권을 구매하기 위하여 목적지에 따라 여권/비자를 체크하고 여행일정에 맞춰 가격, 경유지 회수, 항공사, 등을 고려해 여러 여행사로부터 구매가 가능한 항공권을 체크한 다음 항공권 예약 및 항공권을 구매하고자 하는 계획수립을 한다.

먼저 HTN 계획수립에 의해 태스크 분해를 한다. [그림 5]와 같이 직접 실행할 수 있는 최소단위 태스크(primitive task)를 찾을 때 까지 계속 재귀적으로 태스크를 작은 서브태스크로 나눈다.



[그림 5] 태스크 분해(task decomposition)

계획수립 지식을 표현하기 위해서는 연산자, 메서드와 공리를 사용한다.

```

{ operator {userid ?uid}          ::head
  {uid ?uid}                      ::precondition
  {}                               ::delete list
  {(id-number id-number)}        ::add list
}
    
```

```
(:method (condition-test) ::head
  ((id-number ?id-number) (sdate ? sdate) ::precondition
   (rdate? rdate) (destination? destination)
   ((!(available-airport ? id-number) :: partially ordered list of subtasks
    (!confirm-sdate ?sdate) (!confirm-rdate ?rdate)
    (!confirm- destination ? destination)
   )
  )
)
```

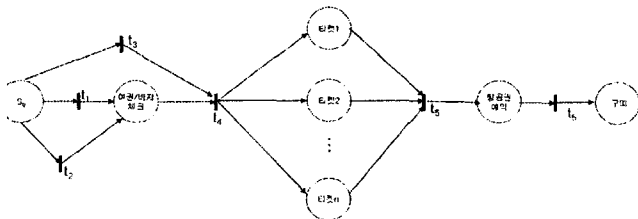
초기상태와 태스크 리스트 그리고 계획결과 리스트에 대한 기술은 다음과 같다.

```
initial state :
((id-number id70625402315)(destination? New York)
 (sdate 20070201)(rdate 20070207))

task list : ((getticket))

plan result list
plan1
((id-number ?1d5123456) (sdate ? 20070201)
 (rdate? 20070207) (destination? New York)
 (tra-agent?gaja) (price?152) (stop-num?2) (airport-com?kor)
 )
plan2
((id-number ?1d5123456) (sdate ? 20070201)
 (rdate? 20070207) (destination? New York)
 (tra-agent?hana) (price?180) (stop-num?1) (airport-com?kor)
 )
plan3
((id-number ?1d5123456) (sdate ? 20070201)
 (rdate? 20070207) (destination? New York)
 (tra-agent?hana) (price?140) (stop-num?3) (airport-com?jal)
 )
plan4
((id-number ?1d5123456) (sdate ? 20070202)
 (rdate? 20070207) (destination? New York)
 (tra-agent?web) (price?156) (stop-num?1) (airport-com?anal)
 )
...
```

위의 계획 결과 리스트와 여행을 위한 전제 조건 (초기 값)으로 여권과 비자 소지여부, 여행일정의 변경사항 등이 강건한 계획수립을 위한 입력 값이 된다. 이때 실시간 처리의 웹 서비스 환경에서, 항공권의 예약을 위해 단일 계획수립으로 작업을 수행하는 것은 매우 비효율적이다. 예를 들어, 여행사마다 일정량의 구매 가능한 항공권을 가지고 있다. 그러므로 사용자가 원하는 항공권을 검색하였더라도 예약 작업이 실시간으로 처리되기 때문에 적은 시간차이로도 예약을 실패하는 경우가 발생한다. 한편 [그림6]은 다중 계획 결과를 포함하는 페트리 넷 기반의 모델을 이용하여 강건한 계획수립 표현을 보여주고 있다.



[그림 6] 항공권 구매를 위한 웹 서비스 계획수립 예제

위 그림에서 플레이스(원:circle)에 해당되는 부분은 다음 작업을 수행할 수 있는 조건에 대한 정보를 저장하고 트랜지션(사각형: square)은 일반적으로 사건을 의미한다. 그러므로 티켓1 항공권 예약 작업을 실패하는 경우, 새로운 계획을 생성하는 것이 아니라 트랜지션 t5 수행하다 실패하게 되면 t4

로 백트래킹 하여 조건을 만족하는 플레이스를 선택 하여 실행한다. 따라서 계획수립의 재생성 없이 하나의 강건한 계획수립으로부터 태스크를 성취할 수 있다.

6. 결론 및 향후과제

강건한 계획은 다중 서브-계획 실행 경로들을 포함한 하나의 계획이다. 강건한 계획은 구성하는 행위들이 실패하기 쉬운 환경에서 주어진 태스크를 성취할 수 있도록 한다. 이것은 오프-라인으로 강건한 계획을 생성하고 실행 조건과 독립적으로 실행 경로를 동적으로 결정하는 방법으로 태스크를 수행할 수 있도록 하는 장점이 있다. 이 논문에서는 강건한 계획 기반의 페트리 넷 모델과 강건한 계획을 생성하기 위한 방법을 제안했다. 또한 동적 환경에서 강건한 계획을 실행하기 위한 방법을 소개했다. 제안된 강건한 계획 생성방법은 잘 알려진 HTN 계획수립기 JSHOP2를 수정해서 구현하였다.

향후 과제로는 시멘틱 웹 서비스 도메인으로부터 에이전트와 서비스에 대한 정보를 온톨로지 기술하고 이로부터 자동으로 계획수립을 생성할 수 있는 연구가 필요하다.

참 고 문 헌

- [1] M. Challab, D. Nau, P. Traverso. *Automated Planning : Theory and Practice*. Morgann Kaufmann.(2004)
- [2] JSHOP2. [HTTP://www.cs.umd.edu/projects/shop/description.html](http://www.cs.umd.edu/projects/shop/description.html)
- [3] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila. SHOP : Simple Hierarchical Ordered Planner. *Proc. of IJCAT-99*. 968-973 (1999)
- [4] A. Blum, M. Furst. Fast Planning Through Planning Graph Analysis. *Artificial Intelligence*. 90. 281-300(1997)
- [5] N. Gibbins, S. Harris, N.Shadbolt. Agent-based Semantic Web Service. *J. of Web Semantics*. 1. 141-154 (2004)
- [6] E. Sirin, B. Parsia. Planning for Semantic Web Service. *Proc. of the 3rd International Semantic Web Conference*. (2004)
- [7] K. Sycara, M. Paulucci, J. Soudry, N. Srinivasan. Dynamic Discovery and Coordination of Agent-based Semantic Web Service. *IEEE internet computing*. May-June. 66-73 (2004)
- [8] M. Vukovic, P. Robinson. SHOP2 and TLPlan for Proactive Service Composition. *Proc. of UK-Russia Workshop on Proactive Computing*. (2005)
- [9] M. Prasad, A. Biere, A. Gupta. A Survey of Recent Advances in SAT-based Formal Verification, Intl. *Journal on Software Tools for Technology Transfer (STTT)*, vol. 7, number 2, Springer (2005)
- [10] Mathijs de Weerd and Adriaan ter Mors and Cees Witteveen . Multi-agent Planning: An introduction to planning and coordination. In *Handouts of the European Agent Summer School*, pp. 1-32. (2005)