실시간 환경에서 지능성과 반응성을 갖는 상황 기반 계획기 설계

박인숙° 이태경
동국대학교 전자계산학과

Design of State_based Planner with Intelligence and Reactivity in Real_Time Environment

InSook Park° TaeKyung Lee
Dept. of Computer Science, Dongguk University

요 약

정보 통신의 모든 분야에 걸쳐 활용될 수 있는 차세대 핵심 기술인 에이전트는 자율성, 지능성, 반응성, 협동성을 갖는 독립된 프로그램으로 지식과 추론 능력을 바탕으로 사용자의 작업을 대신 해준다.

본 논문은 복잡하고 실시간 환경에서 발생할 수 있는 상황에 있어 지능적인 추론과 즉각적인 반응이 가능한 혼합형 에이전트개념을 도입한 계획기를 설계한다. 계획기 구성을 위하여, 재난 발생시 즉각적인 반응을 하는 반응요소(reflexive component)와 계획라이브러리에 저장된 여러 계획들을 검색해 해결안을 찾는 인지요소(cognitive component)로 구성된다. 인지 요소에서 상황에 따라 저장된 계획을 찾고, 상황에 맞는 것을 추론하는 과정을 살펴본다. 혼합형 에이전트 개념을 도입한 계획기는 부분 순서화된 계획기로서 상황 기반 탐색(situation-based search)방법에 의하여 계획을 생성하도록 하였다.

1 서 론

재난관리시스템은 화재, 폭발, 붕괴, 교통사고, 화생방사고, 환경오염사고 등의 재난 발생 위험성을 제거하고 재난 발생시 피해의 수습과 복구를 위해 행하는 모든 활동을 제어하는 시스템이다[4]. 최근에는 재난관리시스템에 환경변화를 스스로 인지하고 그에 대응하는 행동을 취하며, 지식을 바탕으로 추론과 학습 기능이 있는 에이전트(Agent)를 도입한다.

본 논문에서는 지금까지 데이터베이스와 지리정보시스템 도구를 이용한 재난관리방법의 한계성을 벗어나, 에이전트의 성질 중 반응성과 지능성을 가진 혼합형 에이전트 기술을 도입했다. 이것은 재난 발생에 적시에 반응하고 입력된 재난 정보에 맞는 계획을 저장된 라이브러리에서 상황_기반 계획 생성 모듈을 이용해 적절한 계획을 탐색한다.[1, 3, 10].

논문 구성은 1장의 서론에 이어 2장에서는 에이전트 개념과 계획기 특징에 대해 알아본다. 3장에서는 재난관리를 위한 혼합형에이전트 시스템 설명하고 4장에서는 인지요소의 상황기반계획생성모듈의 동작과 재난관리를 위한 계획기에 대해 살펴본다. 끝으로 5장에서 상황기반 계획 생성 에이전트 설계에 대한 결론과 및 향후 연구방향을 제시한다.

2 에이전트 개념과 계획기 특징

에이전트는 이미 종래의 인공지능분야나 분산 문제해결 시스템에서 이들 시스템의 구성요소를 가리키는 표현으로 사용되어져 왔다. 에이전트는 몇 가지 특성으로 그것을 정의하기도 한다. 특히 자율성(autonomy), 지능성(intelligence), 반응성(reactivity), 그리고 협동성(cooperation)등을 들 수 있다[7].

에이전트의 종류는 적용분야나 목적에 따라 나눠지는데, 에이전트시스템에 지적 능력을 부여하는 방법에 따라서 지식 기반 에이전트(Knowledge Based Agent), 반응형 에이전트(Reactive Agent), 둘을 혼합한 형태의 혼합형 에이전트(Hybrid Agent)로 나뉜다[2].

첫째, 지식기반 에이전트는 필요한 지식과 놓여있는 환경에 대한 명시적인 표현과 지식 처리 기구를 갖추고 그것을 이용한 상황 인식과 추론 등으로 문제를 해결하는 에이전트의 총칭이다. 대부분의 에이전트가

여기에 해당된다.

둘째, 반응형 에이전트는 문제해결을 위한 명시적인 지식을 갖지 않고 외부환경의 변화에 대해서만 반응적으로 동작하는 에이전트를 말한다.

세째, 혼합형 에이전트는 위의 두 에이전트의 장점을 결합한 형태다. 그 예로서 하나는 반응적 요소(Reflexive Component)와 인지적 요소(Cognitive Component)를 내부에 동시에 가지고 있는 Phoenix[8]가 있다.

계획기는 초기상태에서 목적 상태를 얻기 위한 행동들의 일련의 생성과정이다. 인공지능에서 계획기는 동작하기 전에 동작의 전체 과정을 결정하는 것으로, 전체 순서화된 계획과 부분 순서화된 계획이 있다. 실시간적으로 입력된 상황에 대해 적절한 계획을 도출하기 위해서는 부분 순서화된 계획이 훨씬 더 효율적이다. 그 이유는 모든 가능한 계획 순서들을 탐색하는 것을 피할 수 있기 때문이다. 반면에 병렬적인 계획 수행에서 행동들 사이의 제약사항을 감지해야 될 뿐 아니라, 행동들의 제약사항들 사이의 관계도 신중하게 생각해야하기 때문이다. 이것은 부분적으로 순서화된 계획을 생성한다. 또한 이런 계획방법은 병렬적으로 수행되는 다중 에이전트 환경에 적합하다.

따라서 본 논문에서는 지능성과 반응성을 갖는 혼합형에이전트로 구성된 부분순서화된 계획을 생성하는 계획기를 구현하고자 한다.

3 혼합형 에이전트와 계획기

3.1 혼합형 에이전트의 구성 요소

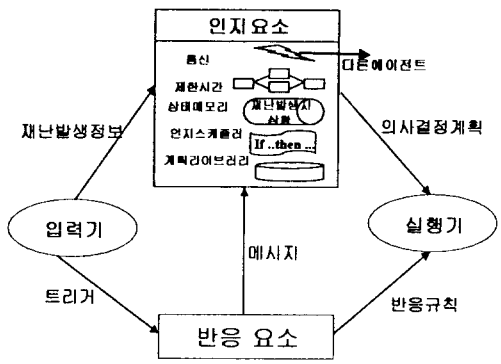
에이전트 특성들 중 한 가지 특성에 촛점을 둔 단일에이전트는 독립적으로 특정 응용분야에서 사용될 수 있다. 혼합형 에이전트는 단일 에이전트들이 갖는 장점은 부각시키고 약점은 최소화할 수 있도록 몇 가지의 에이전트 특성을 가진다. 혼합형 에이전트의 예로 지능성과 반응성을 결합해 하나의 에이전트로 구성한다면, 지식베이스를 이용해 행동을 추론하는 부분은 비교적 장기적인, 응답시간의 신속도가 중요하지 않은 목표에 대해 그 결과를 넘겨 줄 수 있고, 발생한 이벤트가 정확성보다 일단의 신속한 반응을 요구하는 경우에 빠른 응답을 제공할 수 있어야 한다.

혼합형 에이전트가 갖는 장점은 모듈들간의 정보공유가 용이하고 에이전트간의 통신부담이 적다는 것이다. 예를 들어 Soar 구조[2]는 계획과 실행이 같은 구조와 지식베이스를 공유하므로 계획과 실행간에 지식을 전달할 필요가 없다. 그리고, 여러 개의 모듈들의 계층으로 이루어져 있어 각각의 층은 에이전트에 필요한 서로 다른 기능을 구현하고 있다. 즉, 환경변화에 빠르게 반응하는 반응요소, 계획과 추론을 하고 제어규칙에 따라 필요한 의사결정 정보를 내는 인지요소로 각각 이루어진다[6, 10].

3.2 혼합형 에이전트에서의 계획생성과정

재난관리를 위한 지리정보시스템은 재해인식과 대처방안 그리고 재난 발생시 피해 최소화를 위한 의사결정 정보를 제공한다. 그러나 재난 발생시 재난발생 원인에 대한 지식의 불확실로 물리적 모델표현이 어렵고 모의실험 시간이 많이 걸리는 단점이 있다.

따라서, 본 논문에서는 동적인 환경에 적합한 혼합형 에이전트를 도입한 계획기를 모델링한다. 즉, 인지요소와 반응요소가 결합시킨 하나의 계획기를 만든다. 즉, 여기에서는 목표 지향적 추론을 통해 행동을 수행하게 하는 인지요소와 환경 변화에 대해 즉각적인 반응을 위한 반응요소의 혼합형 에이전트를 [그림 1] 같이 설계한다.



[그림 1] 시스템의 구조

입력기(sensor)와 실행기(effector)는 외부환경과 정보교환을 하고, 반응요소와 인지요소에 의해 제어된다.

입력기는 반응요소에 매 시간마다 트리거(Time generated trigger) 한다. 인지요소에는 재난 발생시에만 재난정보를 입력해 적합한 계획을 생성한다. 재난상황에 대해 처음부터 새로운 계획을 생성하고 수행하는 것이 아닌 상황에 따라 저장되어 있는 계획라이브러리에서 추론과 필터링을 통해 계획을 생성한다.

즉, 인지요소는 계획라이브러리(plan library), 인지스케줄러(cognitive scheduler), 상태 메모리(state_memory)와 제한시간(timeline) 이 네 가지로 구성되어 있다. 따라서 입력기에 의해 입력된 재해상황에 대해 구축되어있는 계획라이브러리를 이용해서 적절한 방재 계획을 검색한다. 그것을 제한 시간내에 위치시킨

다. 그리고 상태메모리에는 재난의 종류와 재난 발생지의 현재 상황 즉 풍속, 강우량 등의 기상학적 상황과 교통·통신 시설 등의 지리적 위치정보를 일시적으로 저장해서 계획라이브러리에서 적절한 계획을 검색하는데 도움을 준다. 계획라이브러리에서 선택된 일련의 계획 행동들은 인지스케줄러에 의해 우선 순위와 제한시간 그리고 상태메모리를 참조해 최종적으로 계획을 선택하고 실행기로 보낸다. 이렇게 인지요소가 계획을 검색하는 동안 반응요소는 항상 입력기에 의해 트리거되고 그 메시지를 인지요소에 넘겨 계획을 바꿔주는 융통성있는 행동을 진행한다.

인지요소에 의해 구축된 재난 관리를 위한 행동계획은 실행기로 입력되어져 일반인들에게 지시사항 등을 음성, 화상(위치정보), 문자와 경보음으로 알린다.

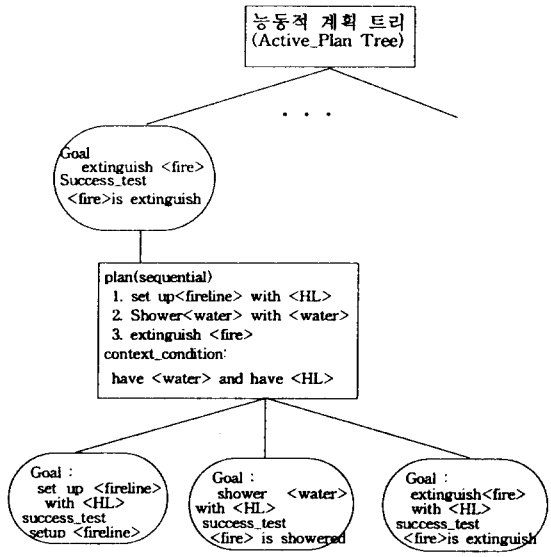
3.3 계획기 구성

고전적 인공지능 계획 문제를 표현하는 방법으로 STRIPS(STanford Research Institute Problem Solver) 방법이 오래도록 사용되어 왔다. 여기서는 계획 문제를 해결하기 위해서 문제가 술어논리로 표현된다.

이 논문에서는 목적 상태를 구성하는 술어사이의 간섭 관계를 파악하여 이를 바탕으로 부분목적 (목적상태를 이루는 전단계의 상태) 간의 선취관계를 파악하고 탐색과정에서 선취관계를 위반하는 선택을 가지치기하여 탐색의 효율성을 제고하는 방법을 제안한다. 지금까지 연구 발표된 계획기들도 거의 모두 지능 탐색을 사용했다. 다만 탐색공간을 이루는 노드(node)를 작업환경의 상태로 하느냐, 아니면 일련의 부분 동작(partial plan)으로 하느냐, 노드 사이의 관계를 기본동작으로 연관시키느냐 아니면 동작간의 간섭관계로 연관시키느냐 등 탐색공간의 구성요소에 따라 다양한 계획 시스템의 구현 방법이 발표되었다[11, 12, 13].

여기서는 탐색공간을 트리로 표현하고 각 노드는 계획으로 한다. 그리고 아크는 계획의 변화 상태를 나타내 도록한다. 따라서 각 아크는 추가적 단계 또는 순서의 추가에 의해 계획이 확대되고 각 단말은 해결된 계획 또는 dead_end이다. 그리고 중간 노드들은 끝나지 않고 아직 확대될 수 있는 계획이다. [그림 2]에서는 계획을 생성하는 과정을 간략하게 계획 트리로 나타낸 것이다. 여기서 능동적 계획트리(Active Plan Tree)는 목적과 계획의 계층으로 구성되었다. 각 계획 노드는 몇개의 목적 노드를 가지고 있고 목적 노드들은 계획을 위해 반드시 성공해야하는 하부 목적들이다. 이 때 도메인은 화재로 하여 예를 들었다.

화재 진화를 위한 계획기는 두 가지의 가능한 계획을 가진다. 헬기를 이용해 화재선(fireline)을 만들고 물을 이용해 불을 끈다.



[그림 2] 능동적 계획 트리의 예

만약에 화재선을 만들 필요가 없는 소형 화재의 경우에는 그냥 물을 이용해 불을 끄고, 유류에 의해 발생한 화재의 경우에는 물이 아닌 다른 분말 가루를 뿌린다. 이 예는 목적과 하부 목적을 자율적으로 획득하고 능동으로 계획 수행을 실세계에 상황에 맞게 바꾸는 것을 인지한다.

4 상황기반계획생성모듈과 재난관리에이전트

4.1 인지요소의 상황기반계획생성모듈의 동작

계획(Planning)은 주어진 조건하에서 목표를 달성하기 위한 행동을 설계하고 만일 새로운 정보가 생기면 계획되었던 일련의 과업들을 backtrack을 통해 변경시킬 수 있는 유연성을 가져야 한다. 계획은 행동들의 집합, 순서 제약 조건의 집합, 변수 바운딩 제약의 집합, 행동의 순서(causal link)의 집합으로 구성된다.

센서는 환경 변화에 대한 정보를 항상 반응요소에 트리거시키고 인지요소에는 재난이 발생했을 때 정보를 입력, 상황_기반 탐색(situation_based search)을 통해 제한 시간 내에 계획을 검색하고 필터링한다.

[그림 3]은 반응 요소에서 상황을 인식(percept)하고 행동(action)을 생성하는 과정을 나타냈다. 즉, 신속한 반응을 요구하는 재난발생시의 알고리즘이다. 우선 항상 반응 요소에는 트리거가 되므로 우선 순위가 높은 재난에 대해서는 조건-행동 규칙을 만들어 둔다.

```
function REFLEX-AGENT_WITH_STATE(percept)
        returns action
 static:state,현재 실세계 상태 설명
        rules, condition_action rules의 집합

 state←UPDATE_STATE(state,percept)
 rule←RULE_MATCH(state,rules)
 action←RULE_ACTION[rule]
 state←UPDATE_STATE(state, action)
 return action
```

[그림 3] 반응형 에이전트

그래서 현재 입력된 상황과 일치하는 규칙을 찾아 그 규칙과 일치하는 행동을 실행기로 보낸다.

```
function POP (initial. goal, operators)
        returns plan
 plan_MAKE_MINIMAL_PLAN(initial, goal)
 loop do
   if SOLUTION?(plan) then return plan
   Sneed, c←SELECT_SUBGOAL(plan)
   with timeline, cognitive scheduler,
        state memory information
 end
```

[그림 4] partial-order planning algorithm

[그림 4]은 계획 수행이 동시에 일어날 수도 있고 또는 정확한 순서가 정해지지 않은 것도 있으므로 POP(partial order planning)알고리즘 이용해 재난발생시 계획을 도출하는 것을 나타냈다.

즉, 센서에 의해 입력된 재난 정보가 비교적 응답시간의 신속성보다는 정확한 계획을 요구하는 경우에는 인지요소에 관한 계획이 발생한다.

최소부분계획에서 시작하여 Sneed(계획을 위해 하부목표를 선택하고 그것을 위한 선행조건필요)에의해 선행조건(c)을 획득하고 계획을 확장한다. 제한시간과 인지스케줄러, 상태메모리의 정보를 참고해서 계획을 찾는다. 또한 POP는 최소 제약의 원리(least commit-ment principle)를 사용하는데 결정은 미리 주어진 것이라해도 완성된 것이 아니므로 충분한 정보가 얻어질 때까지는 버릴 수 없는 것을 말한다.

4.2 계획 생성에 있어 제약사항과 재계획

실시간 환경에 적합한 계획기 개발에 있어 주요 고려사항은 외부 환경과의 실시간 통신문제, 외부에서의 요구가 짧은 시간에 집중될 때 우선순위를 고려한 규칙의 우선 처리문제, 시스템의 오류 발생시 대응책, 병렬추론 기법, 규칙 충돌해결책등 처리해야할 문제가 다양하다.

실시간 조건을 만족한다는 것은 재난 상황이 입력된 후 정해진 시간 내에 계획을 얻을 수 있어야 하고 또한 정확한 계획을 출력할 수 있어야 한다.

어떤 요구에 대한 추론이 제한 시간내에 종료되어야하고 하나의 요구에 대한 추론 도중에 새로운 추론 요구가 발생된다. 첫번째의 문제 해결을 위해서는 지식베이스의 구성과 추론엔진의 추론 기법으로 해결할 수 있고 두번째는 실시간 추론 스케줄러문제이다. 과부하를 관리하기위해서 모든 추론요구들을 스케줄링에 포함하고 사전에 취소하는 것이 없는 AE(All Eligible), 스케줄링이 시작되는 시점에서 제한시간을 넘지 않는 추론요구들만 스케줄링하고 제한시간을 넘어선 추론요구는 과부하로 간주해 취소하는 NT(Not Tardy), 제한시간내에 완료될 가능성이 있는 추론요구들만 서비스를 하는 FD(Feasible Deadline) 중에서 인지요소의 스케줄러는 FD를 이용해 과부하를 관리한다.

우선순위 할당기법은 LS(Least Slack)기법을 적용하는데, 이것은 여유시간(slack time)이 작을수록 높은 우선순위를 할당한다.

POP에서 필요한 기능은 결정을 내리기 위한 충분한 정보가 언제 얻어지는지를 알아야하고 정보가 불충분할 때, 부분문제에 대한 문제해결 활동을 일시 중단하는 기능이 요구된다. 또한 충분한 정보가 얻어졌을 때 부분 문제들을 이용해 중단되어 있던 활동을 재개하는 기능, 그리고 서로 다른 부분 문제들로 부터 얻어지는 정보를 결합하는 기능이 요구된다.

즉, 목표에 따라 계획을 추론했을 때 그 계획이 올바르지않을 경우에는 재계획이 요구된다. 여기서는 DFS(Depth First Search)를 이용해서 재계획을 세운다.

즉 한 노드는 다른 노드에 의해 추가 또는 삭제되는 선행조건을 가지 추가된 한 노드의 조건은 다른 노드에 의해 삭제된다.

[그림 5]는 계획이 잘못되었을 경우에 대한 재계획 알고리즘이다.

진행중인 계획 p와 완전한 계획 q에 대해 간단한 계획 알고리즘이 재계획된다. p의 첫번째 행동을 수행하기전에 p의 선행조건이 일치하는지를 체크한다. 아닐 경우 완전한 계획 q에 있어서 몇가지 상황(노드)을 선택하기 위해 CHOOSE_BEST_CONTINUATION을 호출한다. 즉, 계획 p'는 그 시점에서 q의 끝까지의 계획으로 현재 상태로 부터 가장 쉽게 획득된다. 새 계획이 p'의 선행조건으로 만족되면 그 때 그것을 수행한다.

```
function REPLANNING_AGENT(percept)
        returns an action
  static : PL, a planlibrary(includes action descriptions)
           p, an annotated plan, initially NoPlan
           q, an annotated plan, initially NoPlan
           G, a goal
  TELL(PL, MAKE_PERCEPT_SENTENCE(percept, t))
  current←STATE_DESCRIPTION(KB, t)
  if p = NoPlan then
    p←PLANNER(current, G, KB)
    q←p
    if p = NoPlan or p is empty then return NoOp
  if PRECONDITIONS(p) not currently true in KB then
    p'←CHOOSE_BEST_CONTINUATION(current, q)
    p←APPEND(PLANNER(current,
                  PRECONDITIONS(p'), KB), p')
    q←p
  action←FIRST(p)
  p←REST(p)
  return action
```

[그림 5] 재계획 알고리즘

4.3 시스템 흐름 설명

인지요소와 반응요소는 평행적으로 수행된다. 그러나 계획(plan, action)생성을 위해서는 거의 독립적인 방식을 가진 에이전트로 디자인했는데 이것은 동적으로 다른 시간 그리고 다른 크기로 이벤트가 발생하기 때문이다.

계획 라이브러리를 탐색하여 재난 해결을 위한 일련의 계획을 찾는다. 제한시간과 상태메모리의 두 조건을 만족하는 범위 내에서 인지스케줄러는 우선순위에 따라 계획을 검출한다.

계획과 순서, 제한시간을 초기화한다. 확장을 통해 계획을 다룬다.

재난 관리에 있어 인지요소에서는 실행기를위해 계획을 출력한다. 입력기로부터 재난 발생정보를 받는다. 반응요소의 트리거와 인지요소의 재난발생정보는 병렬적으로 수행된다. 결과로 계획을 출력한다.

센서로부터 재난 정보가 입력되었을 때 계획과 정보가 일치하면 참(true)값을 출력한다.

반응요소로의 트리거는 규칙을 탐색한다. 상황과 규칙이 일치하면 참(true)값을 출력하고 끝낸다. 이 때 실시간 작업이므로 제한 시간을 지켜준다.

계획 생성을 위해서 상태메모리에 입력, 저장된 정보와 인지스케줄러의 우선 순위를 참고해서 계획을 출력한다. 작업 상태에 따라 인지요소에서 무엇을 해야만 하는지 지능적인 감시를 통해 잘못된 계획에 대해서는 재계획을 한다.

```
  //계획의 초기화. 순서(ordering), 타임라인
initially {} = plan;

task treat_Disaster_via_expansion;
  nodes
     1 start,
     2 finish,
     3 action {treat_Disaster};
  orderings 1 ---> 3, 3 ---> 2;
  timeline 15:00 at 1; //1 : start
end_task;

task treat_Disaster_via_conditions;
  nodes
     sequential
        1 start,  //시작 단계
        2 finish  //끝 단계
     end_sequential;
  conditions
     achieve {cognitive component} = plan at 2;
                                          // 2 : finish
     time_line 15:00 at 1;  // 1 : start
end_task;

schema treat_Disaster;
  expands {treat_Disaster};  // 확장
  only_use_for_effects {cognitive Comp.} = plan;
  nodes
     sequential   // 순서
        1 action {get_Disaster_inform_from sensor},
        parallel  // 병렬 수행
           2 action {trigger_reflexive Comp.},
           3 action {data_to_cognitive Comp}
        end_parallel,
        4 action {treat_Disaster} // 행동 출력
     end_sequential;
end_schema;

;;; Primitives  // 센서로부터 재난 정보 입력
schema get_Disaster_inform_from sensor;
  expands {get_Disaster_information};
  effects {information} = true;     // pattern=value
  timeline every per 1 minute;
end_schema;

schema trigger_reflexive Comp.; //반응요소에 트리거
  expands {search_condition_rule}; //규칙 탐색
  effects {rule} = true;
  timeline duration 1 minutes .. 5 minutes; // 5분내
end_schema;

schema data_to_cognitive Comp.; //인지요소정보전달
  expands {search_plan_from plan library};
        // 계획 라이브러리에서 계획을 검색, 필터링
  effects {plan} = true; //검색된 계획이 옳음
  timeline duration 15 minutes .. 20 minutes;
end_schema;

;;; 재난발생지의 상황이 기록된 상태 메모리 참고
;;; 계획을 위해 인지스케줄러의 우선순위 참고
```

5. 결론 및 앞으로의 연구방향

본 논문에서는 재난 관리를 위한 의사결정에 혼합형 에이전트를 도입했다. 동적인 환경에서는 입력된 정보에 대해 추론을 통해 계획을 수립하는 지능적 행위와 환경 변화에 즉각적으로 반응 할 수 있는 반응 행위로 구성된 계획기는 재난 관리에서 계획과 실행이 밀접하게 결합된 복합 구조가 필요하다. 본 논문에서는

계획기반 탐색을 통해 계획 탐색을 알고리즘에 중점해 설명했다.

앞으로 연구되어야 할 분야는 분산환경에서 다른 에이전트와 협력하여 문제해결시 각 에이전트들 사이의 이형질성(heterogeneity)의 문제가 발생한다. 네트워크로 연결된 다른 지역에 대해서는 에이전트 통신언어(ACL : Agent Communication Language)를 이용하여 합의(agreement), 교섭(negotiation), 설득과 경합(competition)등의 프로세스를 통해 분산된 지역간의 재난 문제 해결 해야하는데 이를 위해 프로토콜과 언어등 에이전트의 표준화가 요구된다. 또한, 실시간 반응과 병렬적 수행에 있어 더 효율적인 알고리즘에 대해 연구한다. 또한 자료 표현에 있어 정보를 효과적으로 전달할 수 있도록 멀티미디어를 제공해 사용자가 이해하기 쉽게 하는 사용자 인터페이스에 대한 다양한 연구가 필요하다.

참고 문헌

[1] 성효현, "소중한 인명·재산 앗아가는 자연 재해 GIS 활용으로 예방할 수도 있다", 한국지리정보, vol.1, no.8, pp. 58-62, 1997

[2] 이강건, 이수원, "하이브리드 에이전트," 정보처리학회논문지, vol.4, no.5, pp. 42-54, 1997

[3] 이재규 외 5명, 전문가시스템 원리와 개발, 법영사, 1996

[4] http://www.assembly.go.kr/bill/b150267.htm 재난관리법 법률 개정안

[5] 최종욱 외 2명, 전문가시스템, 집문당, 1995

[6] K. Currie & A. Tate, "O-Plan:the open planning architecture," Artificial Intelligence, vol. 52, no. 1, pp. 49-86, 1991

[7] M. J. Wooldrige and N. Jennings, "Intelligent Agent," Lecture Note in Artiflcial Intelligence, Springer - Verlag, 1995

[8] P. R. Cohen, et al., "Trial by Fire : Requirement for Agents in Complex Environment," AI Magazine, vol. 10, no. 3, pp. 33-48, 1989

[9] Robert Neches, et al., "Enabling Technology for Knowledge Sharing", AI Magazine, vol. 12, no. 4, pp. 37-56, 1991

[10] S. Russell & P. Norbig, "Artificial Intelligence-A modern approach", Prentice hall, 1995

[11] A. Bryan Loyall. et al., "HAP A Reactive, Adaptive Architecture for Agents", Technical Report, CMU-CS-91-147.

[12] Steven Minton, et al., " Total-Order and Partial-Order Planning:A Comparative Analysis", JAIR, vol. 2, pp. 227-262, 1994

[13] Sukhan Lee and Kyusik Chung, "A Parallel Architecture for AI Nonlinear Planning", Applications of Learning and Planning Methods, World Scientific Publicshing Co., pp, 183-203