멀티 에이전트 시스템에서의 효율적인 작업 수행 방법

박정훈, 최중민
한양대학교 전자계산학과
경기도 안산시 사1동
E-mail : {jhpark, jmchoi}@cse.hanyang.ac.kr

Efficient Task Execution Methods in Multi-Agent Systems

Junghoon Park, Joongmin Choi
Dept. of Computer Science and Engineering, Hanyang University
Sa-1-dong, Ansan, Kyunggi-do, 425-791 Korea
E-mail : {jhpark, jmchoi}@cse.hanyang.ac.kr

Abstract

This paper proposes efficient methods that integrate and execute local plan rules of task agents in a multi-agent environment. In these methods, each agent's plan rules are represented in a network structure, and these networks are then collected by a single task agent to build a integrated domain network, which is exploited to achieve the goal. Agent problem solving by using the domain network enables a concurrent execution of plan rules that are sequential in nature.

1. 서 론

에이전트는 복잡한 동적 환경에 존재하면서 주변 환경을 감지하며 지능을 갖고 필요에 따라 다른 에이전트와 통신을 통하여 사용자의 일을 대신해 주는 자율적인 프로그램이라 할 수 있다. 대부분의 에이전트는 그들이 고안될 때 자신의 타스크를 완성하기 위한 연속적인 행동으로 정의되는 계획 규칙(plan rule)을 갖고 있으며 이에 기반하여 행동하게 된다. 일반적으로 하나의 독립된 에이전트는 간단한 작업을 처리하며 좀더 복잡하고 큰 작업을 처리하기 위해서는 에이전트들이 그룹화되어 멀티 에이전트(multi agent) 시스템을 구성하게 되는데 미리 정의된 각 에이전트의 계획 규칙을 그대로 사용하는 것은 시스템의 작업 처리의 효율성을 떨어뜨린다. 그러므로 효과적으로 각 에이전트의 계획 규칙을 통합하는 것이 필요하다. 또한, 각각의 에이전트는 도메인 전체에 대한 완벽한 지식을 갖고 있지 않기 때문에 멀티 에이전트 시스템에서는 공동의 문제를 해결하기 위하여 에이전트간 통신을 통한 정보 교환으로 상호 협력을 한다. 따라서 이런 정보를 내부의 지식 베이스(knowledge base)와 일관성(consistency)있게 유지하는 것이 필요하다.

본 논문에서는 동적인 환경에 존재하는 에이전트가 belief revision 메커니즘을 통하여 그들의 belief를 일관성있게 유지하는 것을 보이고 하나의 공통된 문제 해결을 위한 에이전트들의 계획을 네트워크 형태로 표현하고 이를 효과적으로 통합하여 수행하는 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보며 3장에서는 계획 규칙을 정의하고 각 에이전트의 계획 규칙을 신경망과 같은 네트워크 형태로 변환한 후 이를 통합하는 방법을 제시한다. 4장에서는 에이전트의 belief에 대하여 설명하고 5장에서는 앞서 제시한 내용을 해외 출장 시나리오에 적용시킨 예를 보인다. 마지막으로 6장에서 결론을 맺고 향후 연구 계획을 언급한다.

2. 관련 연구

Malheiro[1] 등은 멀티 에이전트 환경에서 동적으로 변화하는 에이전트들의 belief를 다루기 위해 기존의 ARCHON 시스템[2]에 ATMS(Assumption-based Truth Maintenance System)와 problem solver가 결합된 형태의 belief revision 시스템을 추가한 DiBeRT(Distributed Belief Revision Testbed)를 개발하였다. DiBeRT의 에이

전트는 자신과 에이전트 사회사이의 인터페이스 역할을 하는 cooperative layer와 belief revision 시스템으로 구현되며 특정 도메인의 전문지식을 갖고 있는 domain level 시스템의 두 구성 단위로 나뉘어진다. DiBeRT에서는 자신이 도출해서 스스로 유지하는 private belief와 적어도 하나의 다른 에이전트와 공유하는 shared belief가 있다. private belief는 에이전트내의 ATMS에 의해서 자동으로 수정되며 shared belief는 belief를 만든 에이전트에 의해서만 수정이 된다. DiBeRT에서 에이전트들은 통신을 통하여 private 및 shared belief의 일관성을 유지하면서 동작하게 된다. 본 논문에서는 이와 유사한 방식으로 belief를 유지하게 된다.

Jacowicz[3] 등은 belief revision과 nonmonotonic reasoning을 통한 계획 인식(plan recognition) 방법을 제안하였다. 이 방법은 계획 인식 지식 베이스의 동적인 변화를 지원하기 위해 AGM 모델[4]의 belief revision을 채택하고 있다. 확장(expansion), 축소(contraction), 개정(revision)등 belief revision의 기본적인 연산을 통하여 새로 관찰된 사실을 지식 베이스에 통합시킨다. 이는 지식 베이스가 불완전한 지식으로 구성되는 것을 가능하게 하며 이로부터 계획 라이브러리는 동적인 특성을 갖게 된다. 에이전트는 불완전한 지식 베이스에서 추론을 통하여 수행 가능한 행동을 계획 라이브러리에서 선택하게 된다.

3. 계획 규칙과 네트워크 표현

3.1 계획 규칙 정의

계획은 단순히 일련의 행동으로 볼 수 있다. 본 논문에서의 계획 규칙은 STRIPS(STanford Research Institute Problem Solver) 언어와 유사한 형태로 표현한다[5]. STRIPS 연산자는 기본적으로 에이전트가 취하는 행동을 나타내는 action descriptor, 연산의 수행 전에 갖추어져야할 조건을 나타내는 precondition, 연산 수행이후의 변화를 나타내는 effect등의 세 가지 요소로 구성된다. 본 논문에서 정의하는 계획 규칙의 일반적인 형태는 다음과 같다.

OP(ACTION : Function expr,
PRECOND : Predicate expr,
EFFECT : Function expr)

ACTION 부분의 *Function*은 수행해야 될 타스크로, *PRECOND* 부분의 *Predicate*에 사용된 변수는 타스크 수행에 필요한 파라미터로, *EFFECT*의 *Function*은 결과를 만들어내는 타스크로 볼 수 있다.

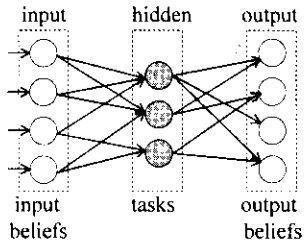

그림 1. 에이전트 수행의 네트워크 표현

3.2 계획 규칙의 네트워크 표현

멀티 에이전트 환경에서 에이전트는 크게 서비스를 요구하는 클라이언트 역할의 에이전트와 요구받은 서비스를 수행하는 서버 역할의 에이전트로 나눌 수 있다. 이 경우 클라이언트 에이전트는 서버 에이전트의 작업 수행에 필요한 정보 혹은 belief를 넘겨주고 결과로써 서버 에이전트로부터 새로운 정보나 belief등을 돌려 받게 된다. 이는 그림 1과 같이 신경망(neural network)과 같은 네트워크 형태로 표현될 수 있는데 서버측 에이전트로 넘겨지는 파라미터는 신경망의 input layer, 서버측 에이전트가 실행하는 타스크는 hidden layer, 서버측 에이전트가 만들어 내는 결과는 output layer에 대응된다고 볼 수 있다[표 1].

계획 규칙	실제 대응값	신 경 망
PRECOND	Input beliefs	Input layer
ACTION	Tasks	Hidden layer
EFFECT	Output beliefs	Output layer

표 1. 계획 규칙과 신경망의 대응 관계

이러한 계획 규칙과 신경망 구조의 대응 관계로부터 각 에이전트의 계획 규칙은 네트워크로 표현될 수 있다. 만약 두 정수 A, B가 존재하면 A와 B의 사칙연산을 계산하여 결과를 belief로 만드는 다음과 같은 계획 규칙이 있을 경우, 이는 그림 2와 같은 네트워크로 표현된다.

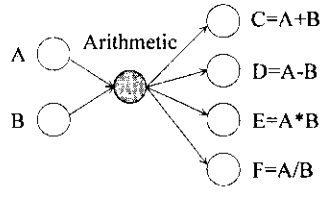

그림 2. Arithmetic 계획 규칙의 네트워크 변환

OP(ACTION:Arithmetic(A, B),
PRECOND:exist(A, B),
EFFECT:MakeBeliefs(C=A+B,D=A-B,E=A*B,F=A/B))

3.3 계획 규칙의 단일 네트워크로의 통합

앞 절의 방법으로 에이전트들의 모든 계획 규칙을 네트워크 형태로 표현한다면 특정 도메인에서의 멀티 에이전트 시스템은 그림 3과 같은 형태를 갖는다. 그림 3에서 도메인 계획 규칙의 네트워크 표현인 *D*와 각 타스트 에이전트의 계획 규칙을 네트워크로 표현한 *L1, L2, L3*를 통합하면 최종적으로 하나의 도메인 네트워크로 표현하는 것이 가능하다. 연산 'A⊗B'가 A와 B의 통합을 의미한다면 이는 다음과 같은 식으로 나타낼 수 있다.

$$Domain\ Network = D \otimes L1 \otimes L2 \otimes \ldots \otimes Ln$$

각 에이전트의 계획 규칙을 표현한 네트워크를 통합하여 도메인 네트워크로 만드는 것은 하나의 네트워크에서 도출된 belief 노드를 input belief 노드로 받아들일 수 있는 다른 모든 네트워크에 연결해 나가면 된다.

이처럼 도메인 문제 해결에 대한 계획 규칙들을 각각의 네트워크로 표현한 후 이를 통합하여 하나의 도메인 네트워크를 만들면 계획 규칙 대신에 구성된 네트워크로 도메인 문제를 해결 할 수 있다. 이를 통한 작업 진행은 미리 정의된 계획 규칙에 의한 순차적인 행동의 수행과 달리 일반적인 계획 규칙이 자연스럽게 병렬로 수행되는 것을 가능하게 하는 장점이 있다.

4. 에이전트의 belief 유지

에이전트들은 작업 진행중에 통신을 통하여 belief를 주고 받게된다. 이런 과정에서 belief revision의 기본적인 연산을 수행하면서 belief의 일관성을 유지해 나간다. 여기서 belief는 [1]에서 사용한 형태와 유사하게 표현되어진다.

<Proposition, Label, Scope, Agent, Status>

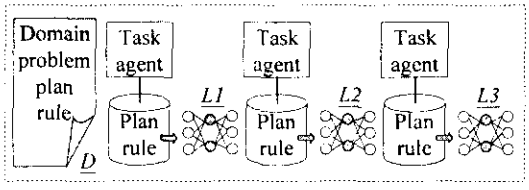

그림 3. 도메인 문제에 대한 시스템 표현

*Proposition*은 'field=value'의 형태를 가지며 *Label*은 proposition을 추론할 수 있는 가정들의 집합이고 *Scope*는 proposition의 공유 여부(shared, private)를 가리킨다. *Agent*는 proposition의 기원 에이전트를 나타내고 *Status*는 proposition의 믿음 상태(believed, unbelieved)를 나타낸다. belief revision의 수행에서 belief 확장은 새로운 belief의 추가를, 축소는 존재하는 'believed' 상태의 belief를 'unbelieved' 상태로 바꾸는 것을, 수정은 proposition에서 value값의 변경을 의미한다. 에이전트는 어떤 작업을 진행하기 위해서는 필요한 belief들이 존재하고 이것이 모두 'believed'로 인식되어야 한다.

5. 시나리오

이번 장에서는 해외 출장 시나리오에 대한 에이전트의 계획 규칙을 네트워크 형태로 만들고 이것을 하나의 도메인 네트워크로 통합하는 것을 보인다.

도메인은 해외 출장에 한정되며 해외 출장 시스템의 전체적인 구조는 그림 4와 같다고 하자. 시스템에서 사용자 인터페이스 에이전트는 사용자로부터 개인 정보를 입력받으며 해외 출장 도메인 계획 규칙을 실행한다고 가정한다. 사용자 인터페이스 에이전트의 계획 규칙은 다음과 같다(실제 파라미터는 'field=value'의 형태를 취한다).

① OP(ACTION:GetUserData(이름,주민,주소,...,연락처),
PRECOND:None,
EFFECT:MakeBeliefs(이름,주민,주소,...,연락처))
② OP(ACTION:MakeVisa(이름,주민,주소,...,여행목적),
PERCOND:exist(이름,주민,주소,...,여행목적),
EFFECT:None)
③ OP(ACTION:ReserTraf(이름,주민,주소,...,은행잔고),
PERCOND:exist(이름,주민,주소,...,은행잔고),
EFFECT:None)
④ OP(ACTION:ReserLodg(이름,목적지,...,은행잔고),
PERCOND:exist(이름,목적지,...,은행잔고),
EFFECT:None)

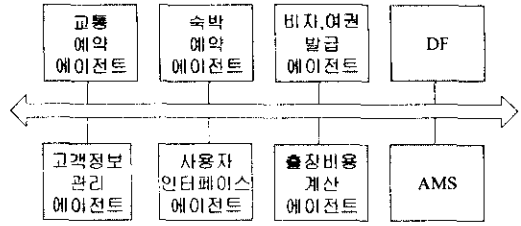

그림 4. 해외 출장 시스템

⑤ OP(ACTION:CompTotCost(여권발급수수료,...),
PERCOND:exist(여권발급수수료,...),
EFFECT:MakeBeliefs(총준비비용,...,은행잔고))

비자/여권 발급 에이전트의 계획 규칙은 다음과 같다.

① OP(ACTION:QueryToDip(이름,주번,...,여행목적),
PERCOND:exist(이름,주번,...,여행목적),
EFFECT:MakeBeliefs(여권번호,발급일,수수료))
② OP(ACTION:QueryToEmb(이름,주번,...,여행목적),
PERCOND:exist(이름,주번,...,여행목적),
EFFECT:MakeBeliefs(비자번호,발급일,수수료))

다른 에이전트들도 이와 같은 형태로 계획 규칙이 정의된다.

계획 규칙이 정의되면 이를 네트워크 형태로 바꾸면 된다. 우선 사용자 인터페이스 에이전트가 해외 출장 도메인 계획 규칙을 실행하기 때문에 앞서의 계획 규칙중에서 EFFECT가 'None'이 아닌 ①과 ⑤만 바꾼다. EFFECT가 'None'인 규칙은 ACTION에 있는 함수를 수행하는 타스크 에이전트의 계획 규칙으로 네트워크를 만들어준다. 그림 5는 이의 결과를 나타낸 것이다. 이렇게 얻어진 네트워크들을 3.3에서 언급한 방법으로 통합하면 그림 6처럼 하나의 도메인 네트워크를 얻게 된다. 이 네트워크로부터 여권신청, 비자신청, 교통예약, 숙박예약 등의 작업은 병렬적으로 수행된다.

6. 결론 및 향후 연구 과제

본 연구에서는 멀티 에이전트 시스템을 구성하는 각 에이전트의 계획 규칙을 네트워크로 표현하고 이들을 효율적으로 통합한 후, 작업을 수행하는 방법을 제안

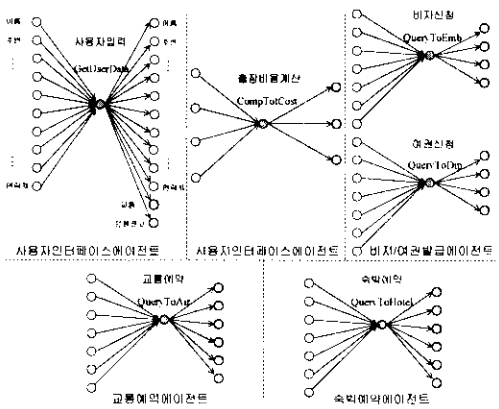

그림 5. 해외 출장 계획 규칙의 네트워크 변환 결과

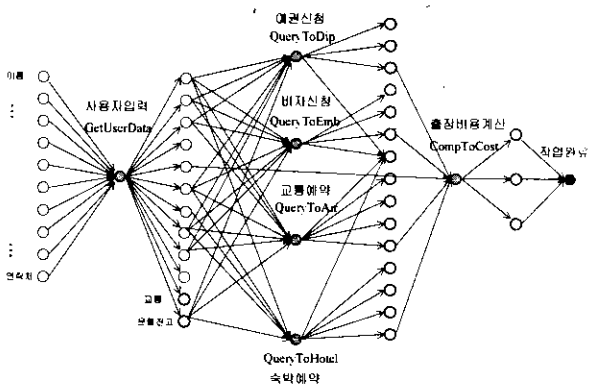

그림 6. 해외 출장의 최종 네트워크

하였다. 이렇게 구성된 네트워크를 통한 작업 수행은 추가적인 노력없이도 순차적인 계획 규칙 중 병렬 수행이 가능한 부분을 자동으로 병렬로 진행되도록 해주는 장점을 지니게 된다. 물론 네트워크를 구성하기 위한 추가적인 작업 시간이 소요되지만 이는 병렬 수행으로 단축되는 시간에 비해 짧기 때문에 크게 고려하지 않아도 된다고 보여진다.

향후 연구 계획은 구성된 네트워크를 바탕으로 추론할 수 있는 메커니즘의 정립 및 주변환경의 불확실성에 대처하기 위한 방법에 대하여 연구하고자 한다.

참고문헌

[1] Malheiro, B., Oliveira, E. "Consistency and Context Management in a Multi-Agent Belief Revision Tesbed," in Agent Theories, Architectures and Languages, Springer-Verlag, Lecture Notes in AI, Vol. 1037, January 1996.

[2] T. Wittig, N. R. Jennings and E. H. Mamdani, "ARCHON - A Framework for Intelligent Cooperation," IEE-BCS Journal of Intelligent Systems Engineering - Special Issue on Real-time Intelligent Systems in ESPRIT, 3 (3) 168-179, 1994.

[3] Pawel Jachowicz and Randy Goebel, "Describing plan recognition as non-monotonic reasoning and belief revision," In Proceedings of the 10th Australian Joint Conference on Artificial Intelligence, pages 236-245, Perth, Australia, November 1997.

[4] Peter Gardenfors, "Belief revision:An introduction," Pp. 1-20 in Belief Revision, ed. by P. Gardenfors, Cambridge University Press.

[5] Struart Russell and Peter Novig, "Artificial Intelligence: a modern approach," Prentice-Hall, 1997