

조정을 지원하는 다중 에이전트 시스템 아키텍처

이승연^U 박수용
서강대학교 컴퓨터학과 컴퓨터학과
sylee@selab.sogang.ac.kr sypark@ccs.sogang.ac.kr

The Architecture Supporting Agents' Coordination in Multi-Agent Systems

Seung-Yun Lee^U Soo-Yong Park
Dept. of Computer Science, Sogang University

요 약

실세계에서 발생하는 복잡한 문제들을 해결하기 위한 노력으로, 다중 에이전트 시스템(Multi-Agent System) 구축에 대한 관심이 높아지고 있다. 다양한 종류의 분산 인공지능 문제들을 에이전트라는 추상적 단위와 에이전트간의 상호작용을 토대로 해결하는 시스템을 개발하기 위하여, 본 연구에서는 다중 에이전트 지향의 소프트웨어를 개발함에 있어 중요한 요소인 조정(Coordination)을 지원하는 아키텍처를 제안한다. 문제영역을 분석하고, 다중 에이전트 시스템의 특성을 파악하여 시스템 요소들의 조정을 지원하는 아키텍처 공정을 제안한다. 또한, 이를 지능형 교통정보 시스템에 적용하여 본다.

1. 서론

인터넷의 발달과 웹의 사용이 증가됨에 따라, 실세계에서 발생하는 대부분의 문제들은 분산된 환경에서의 서비스 처리가 요구된다. 이러한 문제를 해결하기 위해 분산 인공지능에서는 다중 에이전트 시스템(Multi-Agent System) 환경을 이용하여 에이전트들간의 상호 협력을 통해 문제를 해결하는 방법을 연구해 왔으며, 다양한 종류의 분산 인공지능 문제들을 에이전트라는 추상적 단위와 에이전트간의 상호작용을 토대로 해결하려는 시도는 인터넷과 웹의 급속한 발전으로 그 연구가 활발히 진행되고 있다. 현재 다중 에이전트 시스템에 대한 연구는 분산인공지능에 국한되어 있는 것이 아니라, 고도의 분산되고 복합적인 시스템을 분석, 설계, 구현하는 실용 소프트웨어를 효과적으로 개발하려는 소프트웨어 개발 방법론으로 확대되고 있는데, 본 연구에서는 다중 에이전트 지향의 소프트웨어 개발의 핵심인 아키텍처 개발을 위한 공정을 제시하고자 한다. 다중 에이전트 시스템의 특성을 파악하고 그 특성들을 반영한 에이전트 조직의 아키텍처를 제안하고, 이를 지능형 교통 정보 시스템에 적용해 본다.

2. 관련연구

2.1 다중 에이전트 시스템

다중 에이전트 시스템은 에이전트 혼자만의 능력이나 지식으로는 해결하기 힘든 문제를 다른 에이전트들과 상호 협력하여 문제를 풀어나가는 시스템으로 각 에이전트의 목표와 문제 해결 상황의 차이에 따라 합의(agreement), 교섭(negotiation), 설득(persuasion), 경쟁(competition) 등의 프로세스를 통해 문제 해결을 추구하는 시스템이다[1]. 다중 에이전트 시스템은 독립적인 단일 에이전트로는 해결하기 힘든 복잡한 문제에 대한 서비스를 제공해주고, 기존 시스템과의 상호작용을 가능하게 하며, 속도, 안정성, 확장성이 용이하다.

이러한 장점은 다중 에이전트 시스템이 문제를 해결하는데 사용하는 조정, 교섭, 통신의 성질을 이용하므로 이뤄질 수 있다[2]. 개개의 에이전트들이 이루는 집단이 통신을 통하여 서로 화합하고 타협하여 서비스를 제공한다. 따라서, 다중의 에이전트들이 서로 긴밀하게 화합하기 위해서는 이들 사이의 조직적 구조와 이들의 작업과 자원 할당에 대한 적절한 기준이 필요하다.

2.2 소프트웨어 아키텍처

소프트웨어 시스템에서 아키텍처라 함은 환경을 고려한 시스템의 최상위 레벨의 추상적 개념으로, 소프트웨어 시스템 아키텍처는 시스템을 나타내는 주요요소와 인터페이스, 이들 사이의 상호작용을 구조화시킨 것이

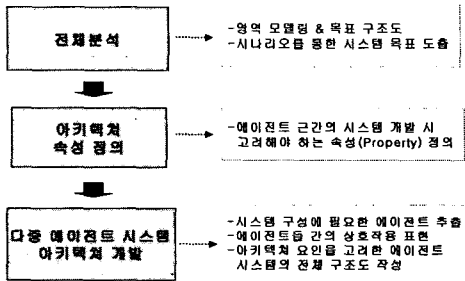
다[3]. 시스템을 개발하는데 있어서 아키텍처는 체계적인 개발을 유도하는 핵심요소이다. 따라서, 소프트웨어 아키텍처는 환경과 시스템간의 관계를 정립해주고, 그 환경 안에서 시스템이 행해야 하는 기능을 추상적으로, 또는 필요하다면 구체적으로 제시해준다.

그렇다면, 에이전트 근간의 소프트웨어를 개발함에 있어 적용되는 아키텍처는 어떠한 기준을 가지고 개발되어야 하는가? 다중 에이전트 시스템에서 문제해결을 위해 조정, 통신 등의 기능들을 어떻게 적용하는 아키텍처는 어떠한 모습을 가지고 있어야 하는가? 본 논문에서는 다중 에이전트 시스템을 개발하는데 핵심요소인 아키텍처를 개발하는데 있어 고려해야 하는 속성들에 대해 정의하고, 이에 맞는 아키텍처를 제안해 보도록 한다.

3. 다중 에이전트 시스템 아키텍처

3.1 전체과정

다중 에이전트 시스템을 개발하는데 핵심인 아키텍처는 [그림 1]과 같은 공정을 거쳐 개발된다.



[그림 1] 다중 에이전트 시스템의 아키텍처 개발과정

시스템 전체 영역에 대한 분석(Global Analysis)을 통하여, 개발하는 시스템이 어떤 기능을 수행하는지를 파악한 후, 시스템 개발에서 고려해야 하는 아키텍처 속성들을 정의한다. 이렇게 시스템의 전체 분석의 결과와 정의된 아키텍처 속성들을 바탕으로 실제 다중 에이전트 시스템의 아키텍처를 개발한다. 각 단계에 대해 살펴보면 다음과 같다.

3.2 전체분석

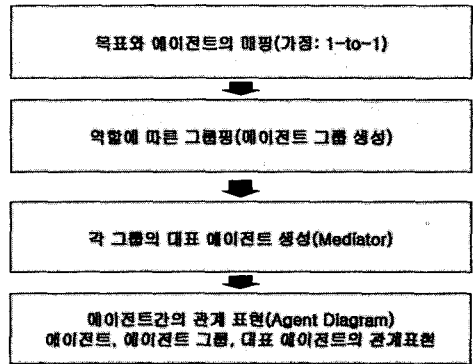
전체분석단계에서는 영역 모델링(Domain Modeling)을 통하여 개발하는 영역에 필수적인 기능들을 분석하고, 목표 계층도(Goal Hierarchy Diagram)를 이용하여 영역 모델링에서 정의된 서비스들이 어떻게 표현되는지, 시스템의 기능이 모두 표현되고 있는지 확인해본다. 시스템의 목표는 시나리오에 기초하여 시나리오와 목표의 집합을 구한 후, 정제과정을 통하여 시스템구축에 필수적인 목표들을 도출해낸다[4].

3.3 아키텍처 속성 정의

전체 분석을 통해 시스템을 이해한 후, 시스템 개발에서 고려해야 하는 아키텍처 속성들을 정의한다. 이러한 속성들은 개발하는 시스템 영역에 따라 그 중요도가 달라질 수 있는데, 분산된 환경에서 복잡한 문제를 효율적으로 처리하기 위한 다중 에이전트 시스템은 속도와 안정성, 확장성과 유연성의 장점을 가지고 있다. 이러한 아키텍처 속성들을 만족시키기 위하여 다중 에이전트 시스템은 특화된 기능을 수행하는 에이전트들간의 조정을 통하여 일을 수행하며, 각각의 에이전트는 자율적으로 변화하는 환경에서 작업을 처리할 수 있어야 한다. 본 연구는 조정을 고려한 아키텍처를 제안하도록 한다.

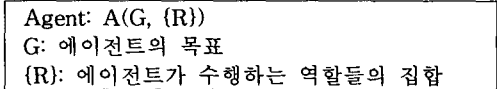
3.4 조정을 지원하는 시스템 아키텍처

다중 에이전트 시스템을 개발하는데 있어서 아키텍처는 시스템 구성 시 중요한 에이전트들을 정의하고 그들 사이의 관계를 표현한다. 에이전트들 사이의 조직적 구조와 이들의 작업과 자원 할당에 대한 기준을 제시해주는 시스템 아키텍처의 도출공정은 [그림 2]와 같다.



[그림 2] 조정을 지원하는 MAS 아키텍처 공정

전체 분석을 통하여 얻어진 목표 구조도와 목표 정제 과정을 통하여 시스템의 목표들을 정의한다. 목표와 에이전트를 일대일 관계로 매핑하여 필요한 에이전트를 도출하고 각 에이전트의 역할을 정의한다. 도출된 에이전트들은 다음과 같이 표기된다.



도출된 에이전트들을 분석하여 시스템이 수행하는 역할들에 에이전트가 가지는 역할들을 매핑시켜 시스템의 역할 각각에 관여하는 에이전트 그룹을 형성한다. 각각의 에이전트는 자신이 수행할 수 있는 역할을 정의하고 있으므로, 공통의 역할을 가지는 에이전트끼리 그룹을 구성할 수 있게 되는데 에이전트 그룹은 다음과 같이 표현될 수 있다.

AGroup: G(R, {A})
R: 에이전트 그룹이 수행하는 역할
{A}: 역할 R에 참여하는 에이전트들의 집합

이 때, 하나의 역할은 여러 에이전트가 모여 수행하고, 하나의 에이전트는 시스템의 여러 역할을 수행하는데 관여할 수 있다.

이렇게 구성된 에이전트 그룹에는 그룹을 대표하는 대표자가 존재하여, 현재 시스템이 수행하는 역할을 위임받아 소속 에이전트들이 협력을 통하여 수행할 수 있도록 관리한다. 대표자는 다음과 같이 표현된다.

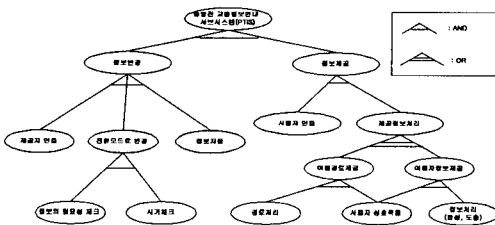
ARep: M(G)
G: 대표하고 있는 에이전트 그룹

위의 방법으로 다중 에이전트 시스템 개발에 필요한 주요요소들(에이전트, 에이전트 그룹, 대표자)을 정의한 후, 에이전트 다이어그램과 에이전트 상호작용 다이어그램을 이용하여 시스템 아키텍처를 나타낸다. 에이전트 다이어그램은 에이전트들 간의 연관관계를, 에이전트 상호작용 다이어그램은 에이전트, 에이전트 그룹, 대표자의 매핑관계를 설명한다.

4. ITS 도메인의 적용 예

3장에서 제시한 공정과 산출물을 ITS 구성 아키텍처 중 출발전 여행정보 안내 서비스시스템에 적용하면 다음과 같다.

▶ 전체분석을 통한 목표계층도



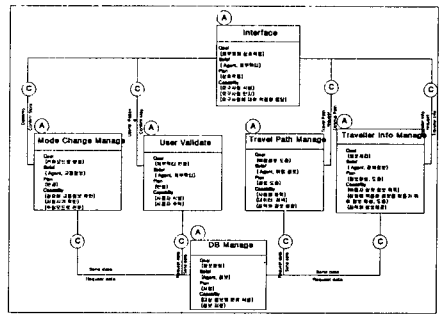
[그림 3] 예제에 적용한 목표계층도

▶ 시스템 개발의 필수요소

- **Agents - A(G, {R_i})**
 - Interface A: A(외부와 상호작용, R((여행정보제공), (여행정보제거), (정보변경)))
 - Actor Validate A: A(인증, R((여행정보제공), (여행정보제거), (정보변경)))
 - Travel Info Manage A: A(정보처리, R((여행정보제공)))
 - Travel Path Manage A: A(경로처리, R((여행정보제공)))
 - Mode Change Manage A: A(전차모드변경관리, R((정보변경)))
 - DB Manage A: A(DB관리, R((여행정보제공), (여행정보제거), (정보변경)))
- **Agent Groups - G(R, A)**
 - G(여행정보제공, (Interface, Actor Validate, Travel Info Manage, DB Manage))
 - G(여행정보처리, (Interface, Actor Validate, Travel Path Manage, DB Manage))
 - G(정보변경, (Interface, Actor Validate, Mode Change Manage, DB Manage))
- **Representatives - M(G)**
 - M(여행정보제공)
 - M(여행정보처리)
 - M(정보변경)

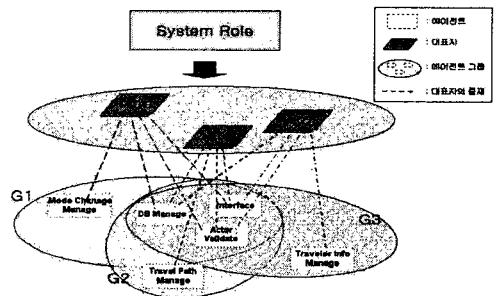
[그림 4] 시스템 개발의 필수요소

▶ 에이전트 다이어그램



[그림 5] 에이전트 다이어그램

▶ 에이전트 상호작용 다이어그램



[그림 6] 에이전트 상호작용 다이어그램

5. 결론

다중 에이전트 시스템을 개발하는데 핵심인 아키텍처를 개발하기 위하여, 에이전트들 사이의 조직적 구조와 이들의 작업과 자원 할당에 대한 기준을 제시해 주는 시스템 아키텍처를 제안하였다. 제안한 아키텍처에 표현된 에이전트간의 조정지원을 검증함과 더불어, 에이전트 내부의 자율성을 표현하는 방법을 연구해야 한다.

6. 참고문헌

- [1] OHare, G. and Jennings, N., Foundations of Distributed Artificial Intelligence, John Wiley & Sons, 1996.
- [2] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, and R. Evans. Software Agents: A review, 1997.
- [3] Ivar Jacobson, Grady Booch, & James Rumbaugh. The Unified Software Development Process, Addison Wesley, 1999.
- [4] Colette Rolland, Carine Souveyet, and Camille Ben Achour, Guiding goal modeling using scenarios, IEEE Transactions on Software Engineering, Vol. 24, No. 12, pp. 1055-1071, December, 1998