

# 이동 에이전트의 Meta 아키텍처 개발을 위한 방안 및 에이전트의 이동성을 지원하는 다중 에이전트 시스템 아키텍처 개발

이계휘<sup>1)</sup> 김민성 박수용  
서강대학교 컴퓨터학과

(khlee<sup>1)</sup>, mskim)<sup>1)</sup>@selab.sogang.ac.kr, sypark@ccs.sogang.ac.kr

## Scheme for Development of Meta Architecture of Mobile Agent and Development of Multi-Agent System Architecture Support to Mobility of Agent

Kehwi Lee<sup>1)</sup>, Minsung Kim, Sooyong Park  
Dept. of Computer Science, Sogang University

### 요 약

에이전트는 보다 폭넓고 정확한 정보를 다른 도메인과 플랫폼의 다중 에이전트 시스템과의 협력을 통해 사용자에게 제공해야 하고, 다른 시스템의 유희 컴퓨팅 용량을 활용하는 진정한 의미의 분산환경을 제공해야 할 필요성이 대두되었다. 이러한 문제를 해결하기 여러 논문에서는 이동 에이전트를 제안하고 있다. 하지만 이동 에이전트를 보다 효율적으로 개발하기 위해서는 이에 대한 아키텍처가 여러 가지 도메인에서 사용될 수 있도록 제시되어야 한다. 따라서 본 논문에서는 이동 에이전트의 아키텍처를 재사용이 가능하도록 설계하기 위한 방안을 연구하고, 이동 에이전트를 다중 에이전트 시스템에 적용할 수 있는 방안에 대한 연구 결과로 에이전트의 이동성을 지원하는 다중 에이전트 시스템 아키텍처를 제안한다.

### 1. 서 론

분산된 환경에서 동적인 작업을 능동적으로 수행하는 소프트웨어를 지원하기 위하여, 분산 인공지능에서는 다중 에이전트 시스템(Multi-Agent System) 환경을 이용하여 에이전트들간의 상호 협력을 통해 문제를 해결하는 방법을 제시해 왔다. 다양한 종류의 분산 인공지능 문제들을 에이전트라는 추상적 단위의 능동적 개체와 이들간의 상호작용을 토대로 해결하려는 시도가 활발히 진행되고 있으며, 그 중에서도 분산된 환경에 적용될 수 있는 에이전트 시스템의 분석과 아키텍처 개발에 대한 연구가 진행되고 있다. 이러한 연구의 결과로 현재 다중 에이전트 시스템에 대한 아키텍처가 많이 제시되고 있다. 또한 그동안에는 이러한 기능을 수행하기 위해 다중 에이전트 시스템을 고정 에이전트들 사이의 상호작용을 중심으로 한 아키텍처에 대한 연구가 진행되어 왔고, 이들 다중 에이전트 시스템 아키텍처 대부분은 같은 도메인과 플랫폼 내에서 에이전트 관리자에 등록된 에이전트들 사이의 협력만을 고려하고 있다. 하지만 에이전트는 보다 폭넓고 정확한 정보를 다른 도메인과 플랫폼의 다중 에이전트 시스템과의 협력을 통해 사용자에게 제공해야 하고, 다른 시스템의 유희 컴퓨팅 용량을 활용하는 진정한 의미의 분산환경을 제공해야 할 필요성이 대두되었다. 또한 최근에 노트북과 PDA, 그리고 핸드폰 등과 같은 이동 컴퓨팅 장비의 증가와 발달로 인한 이동 컴퓨팅에 대한 수요가 급격히 증가하고, 중요시되고 있다. 이러한 문제의 해결책으로 많은 논문에서 에이전트의 한 종류인 이동 에이전트(Mobile Agent)를 제안하고 있다. 그 이유는 다음과 같다. 이동 에이전트는 시스템간의 지속적인 통신이 필요하지 않다. 또한 이동 에이전트는 서로 다른 시스템 사이를 이동해야 하기 때문에 플랫폼에 유연하다.[1][2][3]

하지만 현재의 다중 에이전트 시스템 아키텍처들에서는 이동 에이전트를 어떻게 적용하여 사용할 것인지에 관한 연구는 미미한 실정이며, 대부분의 연구가 같은 도메인과 플랫폼 상에서 에이전트 관리자에 등록된 에이전트 사이의 이동만을 다루고 있다. 또한 이동 에이전트 자체의 아키텍처에 대한 연구는 미

미한 수준이며, 또한 아키텍처 재사용 관점에서의 연구는 진행되지 않고 있다.

따라서 본 논문에서는 이동 에이전트의 아키텍처를 재사용이 가능하도록 설계하기 위한 방안을 연구하고, 이동 에이전트를 다중 에이전트 시스템에 적용할 수 있는 방안에 대한 연구 결과로 에이전트의 이동성을 지원하는 다중 에이전트 시스템 아키텍처를 제안한다.

### 2. 기본 개념 및 기존의 관련 연구

#### 2.1 기본 개념

에이전트의 성질은 기본적으로 자율성(Autonomy), 적응성(Adaptation), 협력성(Cooperation)등 세 가지로 나누어진다.[4] 이러한 기본적인 성질 외에도 이동성(Mobility)과 합리성(Reasoning) 등이 논의되어지고 있다. 이러한 성질들 가운데 에이전트사이에 협력과 가장 밀접한 관계를 가진 것은 이동성이다. 이동성에 대한 정의는 다음과 같다. 에이전트 이동성이란 에이전트가 자신의 내부 상태를 유지하면서 한 위치에서 다른 위치로 이동할 수 있는 능력을 의미한다. 현재 이러한 성질은 에이전트 관리 시스템(Agent Management System)에 등록된 에이전트들 사이에서만 가능하다.

이동 에이전트란, 자신의 판단에 따라 호스트와 호스트 사이를 이동해 다니면서 스스로 문제를 해결하는 소프트웨어 또는 프로그램이다. 즉, 사용자는 필요한 서비스를 제공받기 위해 적절한 서비스를 찾아내거나 별도의 로그인등의 절차를 거치지 않아도 된다. 이것은 사용자가 요구한 작업의 결과가 나올 때까지 어떤 과정을 거치는지 알 필요가 없다는 것을 의미한다.[5] 에이전트의 이동성을 지원하는 MAS란, 이동 에이전트가 자신을 생성한 에이전트 플랫폼으로부터 다른 에이전트 플랫폼으로 이동하기 위한 과정에서 이동을 위해 필요한 정보를 이동 에이전트에게 준비해 주는 역할을 하는 에이전트가 포함된 MAS를 의미한다.

#### 2.2 관련 연구

이동 에이전트 아키텍처에 관한 연구 중 가장 많이 참고했던

모델 2가지는 Malleable Services라는 논문에서 나오는 Mojave 구조와 Mobile Agent Architecture에서 나오는 레벨 순서로 정리된 아키텍처이다. 이 두 가지의 공통적인 모습은 이동 에이전트가 다중 에이전트 시스템으로부터 지원을 받아야 한다는 것이었다. 생성되고, 작업을 수행하고, 소멸되는 순간까지 모든 과정을 다중 에이전트 시스템으로부터 지원을 받는다. 이 두 논문에서 수용한 아이디어는 이동 에이전트가 고정 에이전트 (Stationary Agent)와 동일한 능력을 갖고 작업을 수행할 수 없기 때문에 다중 에이전트 시스템에서 이동 에이전트의 기능 중 일부를 지원해 주어야 한다는 것이다. 하지만 이 두 가지 모두 이동 에이전트가 이동시에는 다음 그림[1]에서처럼 중간 매개체를 거쳐야만 한다.[6][7]

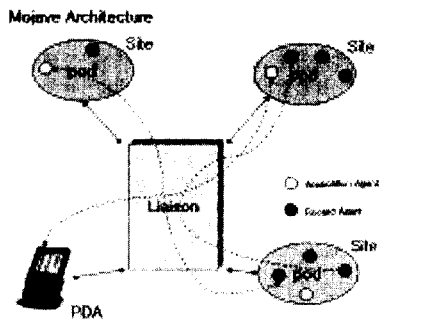


그림 1 Malleable Services

이런 경우 발생 가능한 문제점은 중간 매개체의 병목현상이다. 따라서 이러한 문제점을 해결해 주어야 한다. 또한 본 논문에서는 도메인 엔지니어링 기법을 활용하여 이동 에이전트의 Meta 아키텍처를 제안하고자 한다. Meta 아키텍처란 아키텍처 형태들 중 Adaptable 시스템에서 Reflection 분야에 나오는 것으로 Base 레벨 아키텍처와 짝을 이루어 작용을 한다. Base 레벨 아키텍처들에게 공통적으로 필요한 요소들을 가지고 있으면서, Base 레벨 아키텍처의 환경 변화에 따라 요청이 들어올 경우 필요한 서비스를 제공해주는 개념을 갖고 있다. 도메인 엔지니어링은 Product Line에서 이용되는 개념으로, 재사용 가능한 컴포넌트 제작이 목적이다. 도메인 분석을 통해 도메인의 Commonality와 Variation을 추출해 낼 수 있다.

**3. 이동 에이전트의 Meta 아키텍처 개발을 위한 방안**

이동 에이전트는 사용 목적에 따라 크게 세 가지로 나누어 질 수 있다. 그 분류는 다음과 같다.

첫째, 정보(Information) 이동 에이전트 즉, 단순 통신을 위해서 개발되는 경우이다. 이는 단순히 통신비용과 네트워크의 부하를 줄이고, 통신장비의 전력 소모를 줄이는 것을 목적으로 한다. 이는 이동 통신 장비들을 위한 것이다. 현재 이동 통신장비 즉, Mobile Phone, PDA, Wireless LAN등의 통신비용은 매우 비싸다. 또한 이들의 배터리 용량은 매우 한정적이다. 따라서 이러한 경우 클라이언트 에이전트와 서버 에이전트 사이에 통신을 위해 지속적으로 통신 포트를 열어둔다는 것은 매우 비경제적인 일이다. 이러한 문제점을 해결하기 위해서 이용되는 이동 에이전트를 정보 이동 에이전트라 한다.

둘째, 프로세스(Process) 이동 에이전트 즉, 프로세스의 이동을 위해서 개발되는 경우이다. 컴퓨팅 용량이 부족하거나 프로세스 수행 중에 필요한 기능이 제공되지 않는 경우 다른 컴퓨터로 프로세스를 이동시켜 작업을 수행하게 하는 것이다. 이는 시스템을 보다 효율적으로 사용하고, 구축할 수 있게 해주는

역할하며 완전한 분산환경을 제공하는데 기여한다.

셋째, 작업(Task) 이동 에이전트 즉, 이동 에이전트 자체를 이용하여 특정한 작업을 수행하기 위해서 개발되는 경우이다. 일반적으로 Web Robot이라 불리는 형태의 이동 에이전트로 검색 엔진의 기본 자료를 수집하는 역할을 수행하는 것을 예로 들 수 있다. 이는 이동 에이전트로부터 얻은 검색 결과를 미리 데이터베이스에 저장해 둬으로써 필요한 정보의 위치를 빠르게 제공하여 준다. 즉, 에이전트가 만들어지면 그 후에 별도의 관리가 필요 없이 스스로 작업을 완벽하게 수행하는 이동 에이전트를 의미한다.

그밖에도 이동 컴퓨팅 장비를 위해서 개발되는 경우가 있다. 이 경우의 에이전트 이동의 개념은 위에서 살펴본 세 가지 종류의 이동 에이전트와는 다르다. 이 경우의 이동 에이전트란 프로그램 또는 소프트웨어의 네트워크 상에서의 이동이 아닌 컴퓨터 장비의 실제 공간상의 이동을 의미한다. 이러한 이동 에이전트의 예로 이동전화로 이야기할 수 있다. 이동전화는 이동에 따라 기지국이 바뀌게 된다. 이러한 변화는 사용자는 전혀 알지 못한다. 이러한 종류의 에이전트의 역할은 컴퓨팅 환경의 변화에 따라 프로그램들이 그에 맞게 적응하도록 해주는 기능을 주로 담당한다. 현재 사용되고 있는 이동전화에서 이러한 역할을 하는 프로그램이 에이전트로 구현이 되었는지 여부는 알 수 없다. 그럼에도 본 논문에서 예로 든 이유는 에이전트의 개념에 가깝다고 보았기 때문이다. 하지만 이러한 종류의 이동 에이전트는 본 논문에서 이용하고 있는 이동 에이전트의 정의와는 다르다. 이것은 고정 에이전트의 개념에 속한다고 볼 수 있다. 따라서 이에 대한 아키텍처는 이야기하지 않을 것이다. 하지만 컴퓨팅 환경의 변화에 따른 적응성은 본 논문의 이동 에이전트의 아키텍처에서도 고려되어야 한다.

또한 앞에서 살펴보았듯이 첫 번째와 두 번째의 경우의 이동 에이전트는 MAS에서 이동 에이전트에 대한 관리를 해주어야 한다. 여기서 관리란 에이전트의 생성 및 파괴 그리고 이동 위치 등이 이동 에이전트 관리자에 의해서 관리되어진다. 세 번째의 경우는 이와 같은 관리가 필요 없는 특징을 가지고 있다. 또한 두 번째에서 소개하고 있는 이동 에이전트와 세 번째에서 소개하고 있는 에이전트는 프로세스가 이동한다는 점에서는 유사하지만, 두 번째 에이전트는 프로세스의 일부분만을 다른 MAS의 지원을 받아 문제를 해결하는 형태를 가진다. 하지만 세 번째 에이전트는 하나의 이동 에이전트 자체가 모든 작업을 처리한다는 점에서 차이가 있다고 하겠다.

이와 같이 이동 에이전트는 사용 목적에 따라 분류가 가능하다. 이는 이동 에이전트를 하나의 도메인으로 보고, 그에 따른 재사용 가능한 아키텍처의 설계가 가능하게 하는 요소이다. 따라서 이러한 분류에 따라 도메인 분석 기법을 통해 이동 에이전트의 Commonality와 각각의 이동 에이전트의 종류에 따른 Variation을 추출하고, 여기서 추출된 Commonality를 이용하여 이동 에이전트의 Meta 아키텍처를 설계하고, Variation을 이용하여 각각의 특성에 맞는 재사용 가능한 이동 에이전트 아키텍처를 설계한다. 이러한 도메인 분석 기법을 이용하여 설계한 이동 에이전트 아키텍처는 전체 시스템의 목적에 맞추어 아키텍처에 대한 선택적 사용을 가능하게 한다.

**4. 에이전트의 이동성을 지원하는 다중 에이전트 시스템 아키텍처 개발**

일반적인 다중 에이전트 시스템은 다음 그림[2]와 같은 구조를 갖는다. 이동 에이전트는 고정 에이전트와는 달리 에이전트의 모든 구성 요소들을 모두 가지고 있을 수 없다. 그 이유는 이동을 해야한다는 것 때문이다. 이동시 이동하는 소프트웨어 혹은 프로그램의 크기가 크면 아무리 네트워크의 속도가 빠르다 하더라도 별다른 효율을 갖지 못한다.

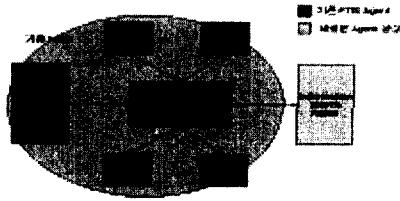


그림 2 다중 에이전트 시스템

또한 이동 에이전트의 강점은 이동 통신에서 나타난다. 이동 통신의 특성상 높은 통신비용과 낮은 네트워크의 속도를 가질 수밖에 없다. 이러한 이유로 이동 에이전트는 반드시 필요한 기능만을 가지고 이동을 해야한다. 따라서 다음 그림[3]과 같이 에이전트의 이동성을 지원하는 다중 에이전트 아키텍처가 필요하다.

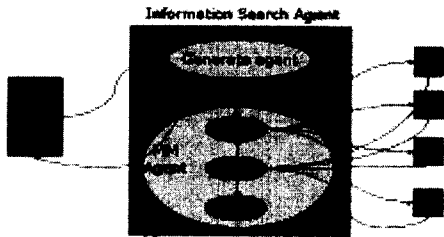


그림 3 에이전트의 이동성 지원

본 논문에서 제안하는 이동 에이전트 지원을 위한 에이전트의 명칭은 Information Search Agent이다. 이는 Generate Agent와 MM(Mobile agent Management) Agent로 나뉘어진다. Generate Agent의 역할은 다중 에이전트 시스템의 Agent Coordinator로부터 요청 받은 작업을 표준화된 통신 언어로 변환(ACL or SDL), MM Agent로 넘겨준다. 또한 요청한 작업을 마친 MM Agent로부터 받은 정보를 자신이 속한 에이전트 플랫폼에 맞도록 변환, Agent Coordinator로 넘겨주는 역할을 한다.

그리고 MM Agent의 아키텍처 설계시 주요 관점은 다음과 같다. Generate Agent로부터 요청 받은 작업을 파악하고 그것을 성취하기 위해, 이동 에이전트를 생성할지, 다중 에이전트 시스템간에 통신을 할지 결정하는 메커니즘이 고려되어야 한다. 이는 그 동안의 작업들로부터 축적한 지식을 바탕으로 결정을 하게 된다. 그 결과 에이전트가 이동하는 것으로 결정되면 언제, 어디로 이동을 할 것인지에 대해 판단하여야 한다. 또한 이동 에이전트로부터 제공받은 정보 중에서 최적의 것을 판단하는 메커니즘이 필요하다. 물론 이동 에이전트 자체에서도 제공받은 정보에 대한 판단을 하게 된다. 하지만 여기서는 사용자에게 정말 필요한 정보가 무엇인가를 판단해야 한다. 이러한 요소들을 바탕으로 MM Agent에는 다음과 같은 기능들도 필요하게 된다. 우선 요청 받은 작업을 수행할 수 있는 다중 에이전트 시스템들을 검색할 수 있는 기능을 갖고 있어야 한다. 이

는 이동 에이전트가 직접 검색하여 이동을 하게 하면 그에 소요되는 시간과 프로그램을 크기가 커지기 때문에 미리 정보를 주어야 한다. 그리고 최적의 정보를 제공받은 곳의 정보를 이동 에이전트로부터 받아 저장해 두었다가 다시 동일한 요청이 들어오면 활용할 수 있도록 하는 기능을 갖고 있어야 한다. 또한 이동 에이전트를 생성 및 파괴하는 기능 그리고 작업 현황을 체크하는 기능을 갖는다. 또한 다른 다중 에이전트 시스템에서 생성된 이동 에이전트에게 필요한 정보를 제공할 수 있는가에 대한 판단과 정보를 제공할지 여부를 판단해야 한다. 이러한 요소들을 갖고 있어야 에이전트의 이동성을 지원할 수 있게 된다.

5. 구현

출발전 여행자 정보 제공 시스템이라는 도메인에 앞에서 제시한 아키텍처를 적용하여 보았다. 아직 이동 에이전트에 대한 아키텍처는 제시하지 않았으나, 이동 에이전트를 Aglet을 이용하여 구현하였고, 전체적인 에이전트의 관리는 FIFA의 Jade 플랫폼을 이용하였다.

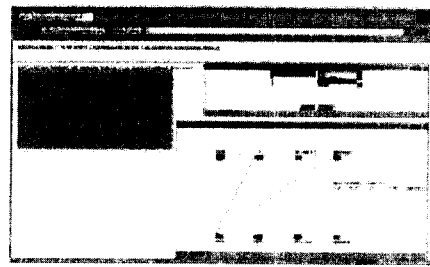


그림 4 출발전 여행자 정보시스템

6. 결론 및 향후 연구

도메인 엔지니어링 기법을 이동 에이전트 아키텍처 설계에 도입하여 재사용 가능한 이동 에이전트 Meta 아키텍처와 각각의 특징에 따른 아키텍처를 설계하는 방안을 제시하였다. 또한 에이전트의 이동성을 지원하기 위한 아키텍처를 제안하였다. 앞으로 앞에서 제시한 설계방안에 따라 이동 에이전트의 Meta 아키텍처와 각각의 이동 에이전트 아키텍처를 제안할 것이다.

7. 참고 문헌

- [1]David Kotz, Robert S. Gray, "Mobile Agent and he Future of Internet", 1999
- [2]G. Cabri, L. Leonardi, F. Zambonelli, "Mobile Agent Technology: Current Trends and Perspectives", 1999
- [3]이승연, "다중 에이전트 시스템 구축을 위한 아키텍처 개발 방법 및 이의 응용", 2000
- [4]S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, and R. Evans, "Software Agents: A review", 1997.
- [5]Hyacinth S. Nwana, "Software Agents: An Overview"
- [6]Venu Vasudevan, Sean Landis, "Malleable Services", Motorola Lab.
- [7]Jonathan Dale, "A Mobile Agent Architecture to Support Distributed Resource Information Management", 1997
- [8]Danny B. Lange, Mitsuru Oshima, "Programming and Developing Java Mobile Agents with Aglets", Addison Wesley
- [9]"Domain Engineering", [http://www.sei.cmu.edu/domain-engineering/domain\\_engineering.html](http://www.sei.cmu.edu/domain-engineering/domain_engineering.html)