

이동 에이전트를 이용한 분산 전문가 시스템 개발환경의 설계 및 구현*

최 지영 박 승수
이화여자대학교 컴퓨터학과

Design and Implementation of a Distributed Expert System Developing Environment using Mobile Agent Technology

Ji Young Choi, Seung Soo Park
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

최근 네트워크 기술의 발달과 함께 분산환경에서의 대용량의 다양한 정보를 처리하기 위한 연구의 일환으로 이동 에이전트에 대한 관심이 높아지고 있다. 이동 에이전트 페러다임이란 기존의 메시지 전달 방식과는 달리 직접 작업을 수행할 수 있는 소프트웨어 객체를 전달하는 방식을 이용한 것이다. 본 논문에서는 이동 에이전트 페러다임을 적용하여 여러 호스트에서 독립적으로 개발한 기존의 전문가 시스템을 통합적으로 활용할 수 있도록 구성된 분산 전문가 시스템 개발 환경을 설계, 구현하였다. 이동 에이전트 페러다임은 소프트웨어 객체를 직접 전달함으로써 분산환경 구현시 문제가 되는 네트워크의 부하를 감소시켜 줄 수 있고 다수의 독립된 호스트에서 시스템의 개발을 손쉽게 할 수 있는 장점이 있다. 따라서 분산환경 하에서 보다 다양한 전문가 시스템을 개발할 수 있는 환경을 마련해 줄 수 있으리라 보고 보인다.

1. 서 론

최근 네트워크 기술의 발달은 분산 컴퓨팅 환경으로의 변화를 초래하였고 이를 지원하기 위한 다양한 기술의 발달을 가져왔다. 인공지능 분야에도 이러한 영향으로 인해 분산 인공지능 분야가 1970년대 이후 발달하기 시작하였다. 특히 성공적인 시스템으로서 언급되는 전문가 시스템도 이러한 컴퓨팅 환경의 변화에 힘입어 분산환경에서의 전문가 시스템에 대한 연구가 활발하게 진행되고 있다. 멀티 에이전트 시스템은 간단한 지식을 가진 다수의 에이전트들을 통합하여 다양한 형태의 정보를 동적으로 처리함으로써 대용량의 지식을 이용한 분산 전문가 시스템에 적용될 수 있다. 최근에 등장한 이동 에이전트 페러다임은 네트워크 부하를 감소시키기 위해 오브젝트 폐쇄이라는 새로운 개념을 도입한 것으로, 보다 빠르고 정확한 분산환경을 원하는 사용자들에게 적합한 해결책이라 할 수 있다.

본 논문에서는 분산 전문가 시스템에 이동 에이전트 페러다임을 적용하여 보다 효율적인 네트워크의 활용을 통해 대용량의 정보처리를 용이하게 할 수 있도록 설계한 새로운 모델을 제

시해 보았다. 제안된 시스템은 이동 에이전트를 직접 원하는 원격 호스트(remote host)로 이동시켜 독립적인 지식 베이스들을 필요한 것만 수집하여 이용할 수 있도록 구성된다. 다양한 형태의 호스트들에서 기존의 지식 베이스를 이용함으로써 개발자가 새로운 지식 베이스 재구성에 필요한 시간을 따로 소모하지 않도록 해결책을 제시해 보았다.

2. DIRM의 설계

2.1 이동 에이전트

이동 에이전트란 하나의 이형 네트워크에서 한 기계에서 다른 기계로 이동할 수 있는 프로그램이다. 이동 에이전트는 때와 장소를 선택하여 이동할 수 있고 원하는 시간에 실행하거나 프로그램 자체를 또 다른 기계로 옮겨서 실행토록 할 수도 있다 [3]. 이동 에이전트 기술은 그 기원이 분산 시스템 연구에서 발전되어 나온 것이다 [1].

기존의 의사소통 방법 등과 에이전트의 이동에 대한 비교는 다음 그림 1과 같다.

* 본 논문은 과학기술처가 지원하는 소프트웨어 기술 개발 사업(97-N-T-01-01-A-12)의 일환으로 수행되었음

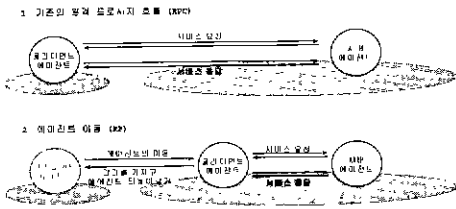


그림 1 기존 의사소통 방법과 에이전트의 이동

2.2 전체 시스템 구성

기존의 많은 다중 에이전트 시스템에서의 지식 추론이란 소규모의 에이전트의 지식을 바탕으로 하고 있다 본 연구에서는 보다 전문적인 지식이 요구되는 추론구조에 있어 각각의 영역 지식 에이전트가 가진 보다 전문적인 지식베이스 처리 능력을 활용할 목적으로 독립적인 추론 에이전트들을 두고 다중 에이전트 시스템과 같은 통합환경을 적용하여 분산된 추론구조를 구성하는 방안을 제시하고자 하였다.

DIRM(Distributed Reasoning-tool using Mobile-agent)은 분산된 네트워크 환경에서의 문제 해결을 위한 추론 시스템으로 이동 에이전트 기술을 기반으로 설계하였다. 두 에이전트간의 네트워크 성능이 그다지 좋지 않다면 지역 데이터 이동에 비해 비용과 시간의 소모가 많다 가령 원격 프로세서 호출방법을 사용하는 에이전트간에는 데이터를 옮기기 위해 여러개의 연산자들을 네트워크 상에서 주고 받아야 한다. 데이터 모니터링과 같은 여러날 계속되는 작업에서는 상주하는 에이전트들끼리 지속적으로 요구사항들을 보내는 것보다 하나의 이동 에이전트를 보내는 것이 비용과 자원의 효율성을 높여 줄 수 있다[5]

DIRM은 여러 호스트에 이동 에이전트 전용 프로토콜인 ATP(Agent Transfer Protocol)를 지원하는 서버와 전문가 시스템을 위한 지식 추론 서버를 두고, 호스트들 사이를 직접 이동하는 이동 에이전트, 각각의 호스트에 상주하는 추론 에이전트, 사용자 인터페이스 에이전트, 그리고 통합 에이전트의 네 가지 에이전트로 이루어진 분산 추론구조 시스템이다 사용자 인터페이스 에이전트는 사용자의 요구를 받아 통합 에이전트에게 보낸다 통합 에이전트는 계획 생성을 위한 사전추론 지식을 이용하여 사용자의 선호도 및 입장을 고려하여 계획을 세우고 각각의 영역 지식 베이스에 맞게 분해한다. 이동 에이전트는 통합 에이전트가 생성한 계획 및 메시지를 가진 전달 에이전트로, 정해진 호스트로 이동하여 추론 에이전트에게 메시지를 전달하고 활성화시킨다. 추론 에이전트는 각각의 호스트에 상주하는 에이전트로서, 이동 에이전트로부터 계획 및 메시지를 전달받아 추론과정을 수행하기 시작하고 수행이 끝나면 지식 추론 서버를 통한 추론 결과를 다시 이동 에이전트에게 전해준다 이동 에이전트는 수행

된 결과를 가지고 추론을 요구한 지식 호스트로 되돌아간다 DIRM의 전체적인 구성은 다음 그림 2와 같다.

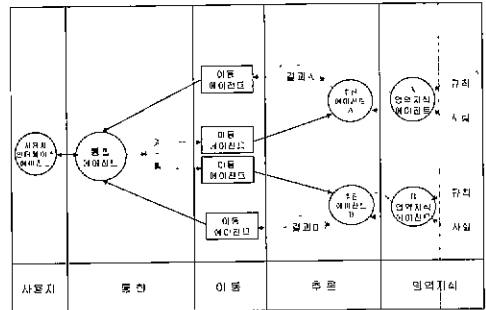


그림 3 DIRM의 전체적인 구성

3. DIRM의 구현

DIRM의 구현을 위해 시스템은 PC와 Sun SPARC10 두 가지를 모두 사용하였으며 Windows 95와 Sun Solans 2.5에서 동작한다. 자바 개발 도구인 JDK 1.1, 이동 에이전트 시스템 개발 도구인 Aglets Workbench, 다중 에이전트 추론을 위한 JMT(Java-based Moderator Templates), 전문가 시스템 개발 도구인 Jess4.0b3, 공용 데이터베이스 엔진으로 mSQL 1.0.16을 사용하였다. 이베지너리사에서 개발한 mSQL-JDBC_1.0a2를 사용하여 데이터베이스와 자바 프로그램을 연결하며 웹서버로는 NCSA Web server를 사용하였다

3.1 사용자 인터페이스와 통합 에이전트

DIRM은 이동 에이전트 서버를 이용하여 생성되는 Aglet으로 구현된 사용자 인터페이스 에이전트와 통합 에이전트를 가지고 있다 사용자 인터페이스 에이전트는 사용자들에게 학습하기 쉽고 편리한 그래픽 인터페이스 환경을 제공하며 통합 에이전트는 분산환경에서의 추론을 위해 여러개의 계획을 수립하고 이를 수행하기 위해 해당 호스트로 이동한 이동 에이전트를 생성한다.

사용자 인터페이스 에이전트를 비롯한 Aglet으로 구현된 대부분의 에이전트들은 서버 또는 다른 에이전트를 통해 생성된다 이것은 에이전트의 보안 유지에도 상당한 영향을 미친다 이동 에이전트 서버는 알 수 있는 코드인 경우에만 에이전트를 생성함으로써 신뢰성을 향상시키고 자신을 잘못된 에이전트로부터 보호한다

통합 에이전트는 사용자 인터페이스 에이전트로부터 세가지 입력정보를 받아 지식 추론 서버를 통해 원격 호스트들을 결정하기 위한 추론 실행을 요청한다. 추론의 결과는 어떤 방법으로 계획을 나눌 것이며 어떤 원격 호스트를 거치는 일련의 작업들을 정할지 결정하는데 반영된다 이렇게 결정된 일련의 작업들

수행하기 위해, 정해진 원격호스트로 이동할 이동 에이전트들이 다수 생성된다. 생성된 이동 에이전트들은 각각의 나눠진 계획 조각들을 가지고 정해진 원격 호스트로 출발한다. 재구성된 결과는 사용자 인터페이스 에이전트를 통해 사용자들에게 보여진다.

통합 에이전트에 의해 수립된 계획을 나누어 각 추론 에이전트에게 보내고 추론과정을 거쳐 얻은 결과들을 수집하는 과정은 그림 3과 같다

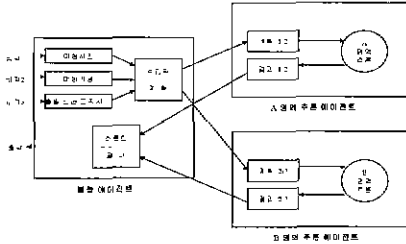


그림 4 통합 에이전트의 계획수립과 결과수집

3.2 이동 에이전트와 추론 에이전트

이동 에이전트는 계획의 나눠진 수에 따라 다수 생성된다. 하나의 이동 에이전트는 한 호스트가 처리할 하나의 계획 조각을 가지고 정해진 네트워크 상의 도착 원격 호스트로 직접 이동하게 된다. 현재 예에서 사용자 인터페이스 에이전트는 퍼스널 컴퓨터 상에서 실행되며 이동 에이전트 중 하나는 같은 퍼스널 컴퓨터의 다른 포트를 사용하는 서버로, 나머지 하나는 다른 워크스테이션의 서버로 이동하게 된다

추론 에이전트는 기존의 에이전트 개념과 크게 다르지 않으며, 정해진 서비스를 자율적으로 제공해주지만 여러 호스트를 옮겨 다니는 능력이 없다. 그러므로 이동 에이전트들이 적절히 이동하여 원격지 서버에서 규칙 기반 추론 서비스를 제공해 주는 추론 에이전트에게 사용자 정보를 전달하고 실행한 후 다시 추론 결과를 가지고 떠나는 원래의 서버로 되돌아가게 된다.

이동 에이전트는 결과를 통합 에이전트에게 전달한다. 통합 에이전트는 이동 에이전트들이 모아온 내용을 하나의 결과로 합쳐서 사용자에게 보여 준다

3.3 구현 예

구현된 시스템에서 입력되는 정보는 세가지이다. 사용자 인터페이스 에이전트는 사용자가 원하는 최대 여행시간과 최소 여행비용, 그리고 출발 또는 도착시간이 입력되면 통합 에이전트에게 적절한 계획 수립을 요청한다. 통합 에이전트는 추론 에이전트와 마찬가지로의 추론과정을 거쳐 계획을 나누고 각 호스트로 이동 에이전트를 이용하여 전달한다. 전달된 계획은 각각 상주하던 추론 에이전트에 의해 추론 과정을 거쳐 결과를 얻게 된다. 얻어진 결과는 다시 이동 에이전트에 의해 통합 에이전트에게

전달되고 통합된다. 예를 들어 최대 여행시간이 5시간이고 비용은 상관하지 않는 경우, 사용자 인터페이스에 의해 보여지는 추론 결과는 그림 4와 같다

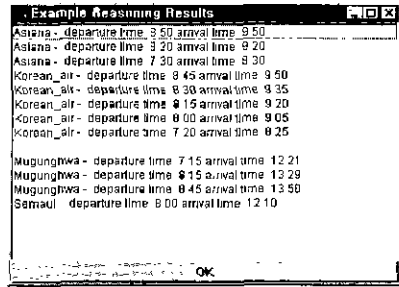


그림 4 추론 결과의 예

4. 결론

본 논문에서는 이동 에이전트 패러다임을 적용하여 보다 빠르고 효율적인 네트워크의 활용을 통해 대용량의 정보처리를 용이하게 할 수 있도록 재구성하고자 하였다. 이동 에이전트 기술은 기존의 원격 프로시저 호출과 같은 방법에서처럼 지속적으로 많은 데이터를 주고 받는 것이 아니라 프로그램 객체를 직접 이동시키기 때문에 성능이 그다지 좋지 않은 네트워크에서도 비용이나 자원을 보다 효율적으로 활용할 수 있다[5]

본 연구를 통해 얻을 수 있는 기대효과 및 연구의의는 다음과 같다 서로 다른 에이전트들이 산재해 있는 분산환경에서 전문적인 지식 베이스를 다양한 방법으로 추론에 응용하고 따라서 보다 신뢰성 있는 결과를 사용자에게 제공해 줄 수 있으며 이동 에이전트의 이용으로 인터넷상에서의 불필요한 메시지 패싱을 줄이고 능동적인 행동 능력을 갖춘 에이전트의 이용으로 인한 효율적인 네트워크 사용을 기대할 수 있다.

5. 참고 문헌

- [1] Cardelli, L., "A Language with Distributed Scope," In Computing Systems, 8(1):pp 27-59, 1995.
- [2] Ernest, J. Friedman-Hill, "Jess, the Java Expert System Shell", SAND98-8206 Unlimited Release, November 1997.
- [3] White, J. E. "Telescript Technology Mobile Agents," General Magic White Paper, 1996
- [4] Ernest, J. Friedman-Hill, "Jess, the Java Expert System Shell", SAND98-8206 Unlimited Release, November 1997.
- [5] "Mobile Agent Facility Specification", Joint Submission - Crystaliz, Inc., General Magic, Inc, GMD FOKUS, International Business Machines Corporation. OMG TC Document cf/xx-x-xx, June 2, 1997.