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 에이전트의 빠르고 안정적인 기존의 메시지 전달 방식은 단지 직접 작업을 수행할 수 있는 소프트웨어 객체를 통합하는 방식을 이용한 것이다. 본 논문에서는 이동 에이전트 게리다임을 적용하여 여러 호스트를 독립적으로 개발한 기존의 전문가 시스템을 통합으로 활용할 수 있도록 구성하고 분산 전문가 시스템 개발 환경을 설계, 구현하였다. 이동 에이전트 게리다임은 소프트웨어 객체를 직접 통합함으로써 분산환경 구현시 문제가 되는 네트워크의 부하를 감소시킬 수 있고 다수의 독립된 호스트에서 시스템의 개발을 순서 있게 할 수 있는 장점이 있다. 따라서 분산환경 하에서 보다 다양한 전문가 시스템을 개발할 수 있는 전문가를 마련하여 줄 수 있으리라고 보인다.

1. 서론

최근 네트워크 기술의 발달은 분산 컴퓨팅 환경으로의 변화를 초래하였고 이를 지지하기 위한 다양한 기술의 발달을 가져왔다. 인공지능 분야가 이와 같은 분야가 1970년대 이후 발달하기 시작하였다. 특히 쌍공적인 시스템으로서 연결되는 기존의 시스템도 이러한 컴퓨터 환경의 변화에 힘입어 분산환경에서의 전문가 시스템에 대한 연구가 활발하게 진행되고 있다. 이로 에이전트 시스템은 간단한 지식을 가진 다수의 에이전트들을 통합하여 다양한 정보의 정보를 동적으로 처리함으로써 대용량의 전문가 시스템에 적용될 수 있다. 최근에 등장한 이동 에이전트 게리다임은 네트워크를 활용하여 정보를 전달하기 위해 각 기계에 연계된 새로운 개념을 도입한 것으로, 보다 빠르고 정확한 분산환경을 제공하는 사용자들에게 적합한 해결책이라 할 수 있다.

본 논문에서는 분산 전문가 시스템에 이동 에이전트 게리다임을 적용하여 보다 효율적인 네트워크의 활용을 통해 대용량의 정보의 처리를 용이하게 할 수 있도록 설계한 새로운 모델을 제시해 보았다. 제안된 시스템은 이동 에이전트를 직접 원하는 원격 호스트(remote host)로 이동시키 독립적인 지식 메이스를 필요한 것만 수집하여 이용할 수 있도록 구성된다. 다양한 경우의 호스트에서 기존의 적합 메이스를 이용함으로써 기발자가 새로운 지식 메이스를 구체화한 필요한 사건을 따로 도출하지 않도록 해결책을 제시해 보았다.

2. DIRM의 설계

2.1 이동 에이전트

이동 에이전트란 하나의 이동 네트워크에서 한 기계에서 다른 기계로 이동할 수 있는 프로그램이나 이동 에이전트는 매화 장소를 선택하여 적합할 수 있도록 기계를 실행하거나 프로그램 자체를 또 다른 기계로 옮겨서 실행하도록 할 수도 있다. 이동 에이전트 기술은 그 기계가 분산 시스템 연구에서 발전되어 나온 것이다[1]. 기존의 의료소통 방법 등과 에이전트의 이동에 대한 비교는 다음 그림 1과 같다.
3. DISMIC 운영 체계 구성

DIRMIC의 구성요소는 PC와 Sun SPARC10 두 가지를 모두 사용하였으며, Windows 98와 Sun Solars 25에서 동작한다. 자바 개발 도구인 JDK 1.1, 이동 에이전트 시스템 개발 도구인 Aglets Workbench, 다중 에이전트 시스템을 위한 JMT(Java-based Modeller Templates) 전문가 시스템 개발 도구인 Jess 4.03, 공동 데이터베이스를 위한 mSQL 1.0.16을 사용하였다. 이전의 인터페이스에서 개발한 mSQL-JDBC 1.0a2를 사용하여 데이터베이스와 자바 프로그램을 연결하며, 웹 서버로는 NCSA Web server를 사용하였다.

3.1 사용자 인터페이스와 통합 에이전트

DIRMIC은 이동 에이전트 서버를 이용하여 생성되는 Aglet으로 구성된 사용자 인터페이스와 통합 에이전트를 하고 있다. 사용자 인터페이스의 사용자에게 하위화된 정보를 제공하기 위해 사용자 인터페이스를 웹 서버로 통합하였으며, 사용자 인터페이스는 사용자의 요구를 받아 통합 에이전트가 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인터페이스가 전달받은 정보를 처리한 후, 사용자에게 제공한다. 사용자는 인тер
3.2 이동 에이전트와 주변 에이전트

이동 에이전트는 제한된 공간에서 다수 생성된 다. 하나의 이동 에이전트는 한 호스트가 처리할 하나의 제어 조각을 가지고있고 대부분 네트워크상의 모든 브로드 호스트로 직접 이동하게 된다. 이러한 제한에서 사용자 인터페이스 에이전트는 퍼스널 컴퓨터 상에서 실행되며 이동 에이전트 중 하나는 같은 Pernal 컴퓨터의 다른 턴도 포트를 사용하는 서버로, 나머지 하나는 다른 웹사이트의 서버로 이동하게 된다.

주변 에이전트는 기존의 에이전트 개념과 크게 다르지 않으며, 영향을 주는 서버의 자율적으로 제공해주지만 여기 호스트를 잡게나 다니는 데에 문제가 없다. 그러므로 이동 에이전트들은 직접 이동하여 전자적인 서버에서 유동적 기반 주변 서비스를 제공해 주는 주변 에이전트에게 사용자 정보를 전달하고 실행한 후 다시 주변 에이전트 결과를 가지고 미리 정해놓은 위치로 돌아와야한다.

이동 에이전트는 결과를 통한 에이전트에게 전달한다. 통합 에이전트는 이동 에이전트들이 원하는 내용을 하나의 결과로 합쳐서 사용자에게 보여준다.

4. 결론

본 논문에서는 이동 에이전트 패러다임을 적용하여 보다 빠르고 효율적인 네트워크의 활용을 통해 대용량의 정보처리를 용이하게 할 수 있도록 제안하고자 하였다. 이동 에이전트 기술은 기존의 전통 프로세서 효율과 같은 방법에서처럼 저용적이지 않은 데이터를 주고 받는 것인이 아니라 프로그램에 직접 이동시키기 때문에 성능이 그저 좋지 않은 네트워크에서도 보다 빠르고 신속한 결과를 사용자에게 제공해 줄 수 있으며 이동 에이전트의 이용으로 인터넷에서의 불필요한 채널의 사용을 가능하게 된다.

5. 참고 문헌