

모빌 에이전트간의 효과적인 공동작업 지원을 위한 통신관리자 설계

이대규⁰ 이승근 왕창중

인하대학교 전자계산공학과

{activer, sglee}@selab.cse.inha.ac.kr, cjiwangse@inha.ac.kr

Design of Communication Manager for effective cooperative work supporting between mobile agents

Dae-Kyu Lee⁰ Seung-Geun Lee Chang-Jong Wang
Dept. of Computer Science & Engineering, Inha University

요 약

에이전트간의 통신 방법에는 에이전트간 직접 통신을 하는 직접 통신 방법과 다른 매개체를 이용하여 통신하는 간접 통신 방법이 있다. 기존 연구에서의 직접 통신 방식은 에이전트의 위치를 계속하여 추적하고 관리하는 방식으로 동기 통신을 지원하고, 간접 통신 방법은 매개체를 두어서 그를 통한 비동기 통신방법을 지원한다. 하지만, 직접 통신의 경우, 에이전트의 추적과 관리에 따른 오버헤드와 공동 작업에 필요한 멀티캐스팅을 지원하기 어려운 문제점이 있어 이동 에이전트에는 부적합한 통신방법이고, 간접 통신의 경우 동기적 상호작용을 지원하기 어렵다.

따라서 본 논문에서는 기존의 간접 통신방법을 확장하여, 멀티캐스팅 기능과 동기 통신 방식을 지원하기 위하여 에이전트간의 통신 관리자를 설계한다. 이를 위해서 에이전트의 이벤트 리스너를 통신관리자에 등록을 시켜서 이벤트가 발생할 때마다 해당되는 에이전트에게 전달되도록 하였다. 또한 통신 관리자가 멀티캐스팅 기능을 지원하도록 하였다. 이렇게 함으로써, 공동 작업에서 필요한 신뢰성 있는 동기 통신 방식과 멀티캐스팅 기능을 지원함으로써 이동 에이전트 기반 CSCW 응용개발에 사용할 수 있다.

1. 서론

최근 들어서 이동 에이전트에 대한 연구가 활발히 진행되면서 이동 에이전트를 공동작업에 이용하려는 연구가 활발하다 [1]. 이동 에이전트를 공동 작업에 이용하는 이유는 하나의 커다란 작업을 혼자 하기엔 무리가 있어 여러 에이전트들에게 작업을 분배하여 일을 수행하게 함으로써 효율적인 작업이 되도록 하는 데 있다[2, 3].

공동작업을 하기 위해 사이트들을 옮겨 다니는 에이전트들 간에는 서로의 의견이나 작업 결과에 대한 협의가 필요하므로 서로간의 통신 방법이 필요하다. 에이전트간의 통신 방법에는 에이전트간 직접 통신을 하는 직접 통신 방법과 블랙보드(blackboard)와 같은 매개체를 이용하여 통신하는 간접 통신 방법이 있다[4].

기존 연구에서의 직접 통신 방식은 에이전트의 위치를 계속하여 추적하고 관리함으로써 동기 통신을 지원한다. 이 방식은 에이전트 서버와 에이전트간의 지속적인 통신이 필요하여 시스템에 부하를 가할 수 있어 이동 에이전트간의 통신방법으로는 적합하지 않다. 따라서 이동 에이전트간의 통신 방법으

로는 매개체를 두어서 통신하는 간접 통신방법이 적합하다. 하지만 대부분의 간접 통신 방법은 비동기 통신 방법만을 지원하는 단점이 있다. 또한 이 방식은 에이전트가 주기적으로 블랙보드를 검사해야 하므로 에이전트의 부담을 증가시킬 수 있다[1].

따라서 본 논문에서는 에이전트간의 효율적인 통신을 지원하기 위해서 기존의 간접 통신 방법인 블랙보드 방식을 확장한다. 제안한 방식을 위해, 본 논문은 각 사이트에 있는 블랙보드를 관리하는 통신관리자를 설계한다. 이는 기존 블랙보드 방식의 문제점인 에이전트가 주기적으로 블랙보드를 검사하지 않고, 에이전트가 통신관리자에 자신의 이벤트 리스너를 등록시켜 통신관리자를 확인하지 않고도 이벤트를 처리할 수 있게 한다. 이렇게 함으로써, 에이전트가 주기적으로 통신관리자를 검사해야 하는 부담을 줄였으며, 에이전트간의 동기적 멀티캐스트 방식도 지원하였다. 따라서 제안한 통신 관리자는 공동 작업에서 필요한 신뢰성 있는 동기 통신 방식과 멀티캐스팅 기능을 지원함으로써 이동 에이전트 기반 CSCW 응용개발에 사용할 수 있다.

2. 관련연구

이 장에서는 기존 이동 에이전트간의 통신 방식인 직접 통신 방식과 간접 통신 방식에 대해서 알아본다.

2.1 직접 통신

직접 통신은 에이전트간의 메시지 전달이 1:1로 직접 이루어지는 통신이다. 직접 통신의 장점은 에이전트간 통신을 할 때, 다른 에이전트에 영향을 받지 않고 작업을 수행할 수 있다는 것에 있다. 또한 동기 통신을 하여 서로간의 메시지 전달과 처리가 바로 이루어진다는 장점이 있다. 하지만 이동 에이전트 시스템에서의 동기 통신은 적합하지 못하다. 왜냐하면 에이전트간의 메시지를 주고 받으려면, 모든 에이전트의 위치를 추적하고 관리해야 하는데, 이는 시스템에 상당한 부하를 줄 수 있기 때문이다. 그러나, 이런 단점에도 불구하고 이동 에이전트를 이용한 공동 작업을 하는 경우에는 에이전트간의 직접 통신을 필요로 하기 때문에 반드시 요구되는 통신 방식이다. 그림 1은 에이전트간의 직접 통신 방식을 나타낸다.

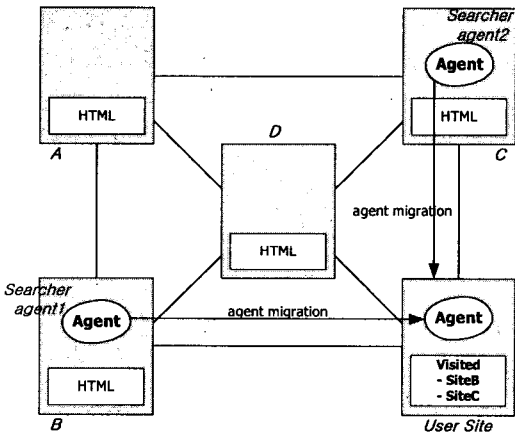


그림 1 직접 통신 방식

2.2 간접 통신

기존 이동 에이전트 시스템에서의 간접 통신은 에이전트간 직접 통신을 하는 것이 아니라 통신관리자와 같은 매개체를 이용해서 이루어진다. 통신을 원하는 에이전트들은 매개체에 메시지를 남기고 다른 에이전트가 그 메시지를 나중에 받는 방식이다. 이 방식은 전하고자 하는 메시지를 매개체에 남겨 놓고 자신의 작업을 계속 수행할 수 있는 장점이 있지만, 다른 에이전트의 작업 결과가 있어야만 다음 작업을 수행할 수 있는 에이전트의 경우에는 그 결과를 계속 기다려야 한다는 단점이 있다. 게다가 메시지가 반드시 전달된다는 보장이 없어 신뢰성이 떨어진다[4,5]. 그림 2는 간접 통신 방식인 블랙보드 기반 조정방식을 나타낸다.

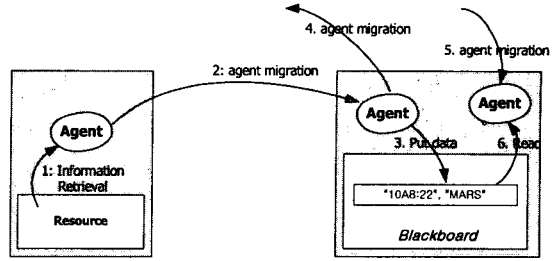


그림 2 간접 통신 방식

3. 이벤트 기반 에이전트간의 통신 방법

이 장에서는 동기적 멀티캐스트와 효율적인 에이전트간 통신을 위하여 이벤트를 기반으로 한 에이전트간의 통신을 제공하는 통신관리자를 설계한다.

3.1 통신관리자의 설계

제안한 방식인 확장된 블랙보드 방식은 에이전트간의 통신을 하기 위해서 각 사이트에 통신 관리자를 설계한다.

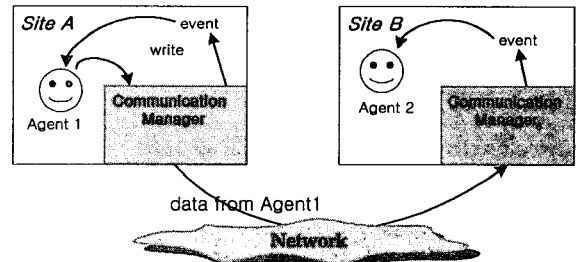


그림 3 통신 관리자의 설계

통신 관리자를 통한 에이전트간 통신은 그림 1과 같이 이루어진다. 통신 관리자는 효과적인 에이전트간 멀티캐스트를 지원하기 위해 스케줄러를 두며, 응답 받은 메시지들을 팩(pack)해 주는 팩커(packer)를 둔다. 또한 통신관리자는 에이전트들이 메시지를 남기고, 모든 메모리를 서로 공유함으로써 통신관리자 사이의 일관성을 유지한다.

에이전트가 사이트에 도착하면, 자신의 이벤트 리스너를 통신관리자에 등록시킨다. 그리고 에이전트가 통신관리자에 메시지를 남기면, 통신관리자는 이벤트를 발생시키고, 에이전트는 그 이벤트를 받게 된다. 에이전트가 처리한 결과는 에이전트에 의해 통신관리자에 남겨지고, 통신관리자는 그 결과를 처음 남긴 에이전트에게 다시 이벤트를 발생시켜 줌으로써 서로간의 통신이 이루어지도록 한다.

3.2 이벤트 등록

이벤트 기반 통신을 하기 위해서 에이전트들은 통신관리자에 자신의 이벤트 리스너를 등록시키는데, 이벤트 리스너는

에이전트가 통신관리자에 메시지를 남기면, 바로 이벤트를 발생시켜서 해당 에이전트와의 동기 통신을 가능하게 한다.

이벤트가 발생된 에이전트는 동기 통신인 경우, 하던 작업을 중지하고 이벤트를 처리하고 비동기 통신인 경우 하던 작업을 수행한 후에 처리하도록 한다.

에이전트가 통신관리자에 등록하는 이벤트 리스너는 통신관리자에 의해 관리된다. 이렇게 함으로써, 에이전트들은 통신관리자를 주기적으로 검사하는 것 없이 자신의 작업을 수행할 수 있을 것이다. 또한 에이전트들이 이동하였을 경우에도 각 통신관리자에 등록되어 있는 이벤트 리스너를 통해 메시지가 전달되기 때문에 에이전트들을 계속해서 추적하는 단점을 보완한다.

통신관리자에서 제공하는 메소드들은 그림 4 와 같다.

```

addListener(agent);
read(msg);
sync_write(source_agent, target_agent, msg);
async_write(target_agent, msg);
write(source_agent, result);
writeMultiple(source_agent, target_agent_list, msg)
    
```

그림 4 통신관리자에서 제공하는 메소드들

addListener 는 에이전트가 통신관리자에 이벤트 리스너를 등록시키는 메소드이고, read 메소드는 통신관리자에서 메시지를 가져오는 메시지이다. sync_write 메소드는 에이전트가 통신관리자에 메시지를 남길 때 사용하는 메소드로 동기 통신인 경우에 사용한다. 이 메소드를 사용하여 이벤트를 발생시키면, 하던 작업을 중단하고 처리를 해줘야 한다. 또한 async_write 메소드는 단지 다른 에이전트 메시지를 전달하는 경우에 사용한다. write 메소드는 결과값을 남길 때 사용하는 메소드이다.

동기적인 멀티캐스트를 지원하기 위해서 통신관리자는 writeMultiple 메소드를 제공한다. writeMultiple 메소드는 sync_write 메소드를 통하여 다른 여러 에이전트들에게 응답을 한 번에 요구하는 경우에 사용한다. 응답을 돌려 받는 경우에는 여러 에이전트로부터 온 메시지를 모아서 하나의 객체로 만든 후 이벤트를 발생시키게 된다. 그러면 처음에 요청한 에이전트가 이벤트를 받을 것이다.

그림 5 는 에이전트가 통신관리자를 통한 통신의 순서를 시퀀스 다이어그램으로 표현한 것이다.

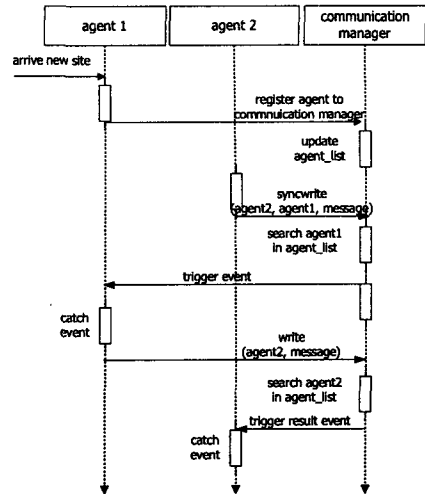


그림 5 에이전트간 통신 순서

4. 결론

본 논문에서는 기존의 블랙보드 방식을 확장하여 동기적 멀티캐스트 방식과 에이전트간의 효율적인 통신을 지원하는 통신방법을 제안하였다. 제안한 통신 방식은 이벤트를 기반으로 한 통신방법으로, 에이전트가 통신관리자에 이벤트 리스너를 등록시키게 하여 이벤트가 발생하면 에이전트가 그 이벤트를 받도록 하였다. 또한 에이전트간 통신할 때 필요한 메소드들을 통신관리자가 제공하게 하여 멀티캐스트를 지원하는 등 효율적인 통신이 이루어지도록 하였다. 이렇게 함으로써, 공동작업에서 필요한 신뢰성 있는 동기 통신 방식과 멀티캐스팅 기능을 지원하여 이동 에이전트 기반 CSCW 응용개발에 사용할 수 있었다.

참고 문헌

- [1] Hyacinth S. Nwana, "Software Agents : An Overview," Knowledge engineering Review, Vol 11, No3, 1996.
- [2] Neeran M. Karnik, Anand R. Tripathi, "Design Issues in Mobile Agent Programming Systems " IEEE Concurrency, Vol. 6, No. 3, 1998
- [3] C. Harrison, D. Chess, and A. Kershenbaum, "Mobile Agents: Are they a good idea?, " IBM Research Report, IBM T.J. Watson Research Center, 1995.
- [4] G. Cabri, L. Leonardi, F. Zambonelli, "Mobile-Agent Coordination Models for Internet Applications, " IEEE Computer Magazine, Vol. 33, No. 2, pp. 1241-1263.
- [5] J. Baumann, F. Hohl, N. Radouniklis, M. Straßer, K. Rothermel, Univ. Stuttgart, Germany, "Communication Concepts for Mobile Agent Systems, " First International Workshop on Mobile Agents 97, 1997.