

분산 실시간 다중 에이전트 시스템의 설계

김윤형^o 김태형
 한양대학교 컴퓨터공학과
 {yunhyung^o, tkim}@cse.hanyang.ac.kr

Designing a Distributed Real-Time Multi-Agent System

Yun-Hyung Kim^o Tae-Hyung Kim
 Dept. of Computer Science & Engineering, Hanyang University

요 약

에이전트 기반 시스템은 그동안 지능형 시스템의 구축을 위해 많이 사용되고 있었으며 최근 유비쿼터스 컴퓨팅 환경에서 context-aware computing을 통한 사물/사람/장소의 통합운영에 매우 중요한 소프트웨어 적 특성을 갖고 있다. 본 논문에서는 실시간 미들웨어 시스템의 기반위에 이러한 지능형 에이전트간 실시간 협업을 가능하게 하는 서비스 계층 설계를 제안한다. 이를 위해 기존에 많이 연구되었던 실시간 시스템과는 다른 연성(軟性) 실시간 통신 시스템 및 스케줄링 서비스를 아울러 설명함으로써 에이전트간 통신 및 스케줄링 방법을 제시한다.

1. 서 론

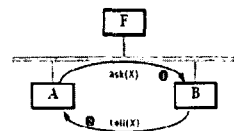
지능형 시스템의 구축을 위해 에이전트 기반 시스템이 많이 사용되고 있다. 자율적 작업 수행, 자가 학습 능력, 원하는 곳으로 이동하여 작업을 수행하는 이동성, 필요에 따라 그룹을 형성하여 그룹간의 상호작용 등의 특징을 가진 지능형 에이전트 기반 시스템은 시간, 장소, 장치에 구애받지 않고 통신이 가능한 유비쿼터스 환경에 적합하다. 하지만, 이제까지의 에이전트 기반 시스템의 연구는 유비쿼터스 환경에서 요구하는 실시간성이 배제되어 있었다. 최근 전자상거래환경에서 지능형 에이전트 컴퓨팅을 이용하는 응용이 연구보고[1] 되기는 하였지만 이것은 유비쿼터스 컴퓨팅 환경에 적합하지 않다. 다양한 기기, 다양한 역할의 사람, 다양한 장소에 지능적으로 적응 하면서 시간적인 제한을 갖는 상황은 컴퓨팅 환경의 변화 정도라는 측면에서 차이가 있기 때문이다. 본 논문에서는 다양한 환경에서 적응성과 실시간성을 제공하기 위해 일반 CORBA 구조를 실시간성을 포함하여 확장한 RT-CORBA를 근간으로하여 CORBAServices Specification에 따르는 실시간 에이전트 서비스 계층(Real-Time Agent Service Layer)을 설계한다. CORBA 구조는 분산환경에서 이동 객체를 지원하고 내장형 시스템을 포함한 소형기기에서부터 대형 서버까지 넓은 응용 스펙트럼을 갖고 있으므로 다양한 컴퓨팅 환경이 통합적으로 응용되는 유비쿼터스 컴퓨팅 환경에 적합하다고 판단하였다.

이를 위해, 실시간 에이전트간의 통신 방법과 이러한 통신을 원활하게 하기 위해 필요한 서비스를 제안하고 유비쿼터스 컴퓨팅 환경에 적합한 분산 실시간 다중 에이전트 시스템의 구조를 소개한다. 우리의 시스템은 이 기종간 실시간 에이전트의 통신을 위해서 Real-Time CORBA를 기반 시스템으로 사용하고 있으며, 실시간 에이전트의 통신을 지원하기 위해 RT-ASL을 CORBAServices Specification에 따라서 설계한다.

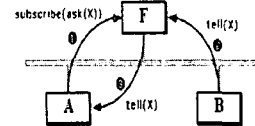
2. 관련연구

2.1 에이전트간 통신

에이전트간 통신은 아래 그림과 같이 두가지 방식으로 나눌 수 있다. 첫째, 그림1과 같이 에이전트 A가 에이전트 B에게 직접 필요한 서비스를 요청하고 이에 대한 응답을 받는 경우이다. 또는, 그림2와 같이 중재 기능을 수행하는 에이전트 F를 지정하여 에이전트간 통신을 수행하는 경우이다. 이 경우, 에이전트 A는 중재 에이전트에게 원하는 서비스 X를 가진 에이전트가 존재하는지를 묻는다. 에이전트 B는 자신이 제공하는 서비스 X가 이용가능한 순간 중재 에이전트에서 이 사실을 알렸으므로 중재 에이전트는 서비스 X의 제공자를 이미 알고 있다. 에이전트 A는 이 사실을 중재 에이전트로부터 제공받아서 에이전트 B와 직접 통신하여 원하는 서비스를 요청한다.



[그림1] 직접적인 통신



[그림2] 중재 에이전트를 이용한 통신

이러한 에이전트간 통신은 KQML[2] (Knowledge Query Manipulation Language)이라고 하는 에이전트 통신 언어를 통해 그 행동이 정의될 수 있다. KQML은 performative를 정의함으로써 해당하는 KQML이 무엇을 의미하는지를 표현하고

특 해준다. KQML 메시지에는 여러 개의 인자가 정의될 수 있다. 따라서 performative와 content 인자를 보면 해당하는 KQML 메시지의 의미를 알 수 있다. 예를 들면, performative 가 ask-one일 경우에는 무엇인가 하나를 묻는다는 것(예: 에이전트 A와 중재 에이전트간의 통신)을 의미하고, tell일 경우에는 중재 에이전트에게 말하려는 것(예: 에이전트 B와 중재 에이전트)임을 알 수 있다.

FIPA(The Foundation for Intelligent Physical Agents)에서는 이러한 에이전트 통신 언어의 표준화를 위한 FIPA-ACL을 제안하고 있다. [3] FIPA-ACL도 performatives를 이용하여 통신을 한다는 점은 KQML과 유사하지만 의미상으로 차이가 있다. 본 논문에서는 에이전트간 통신에 실시간 제약을 표현하기 용이하므로 KQML 방식의 에이전트 통신 언어를 기반으로 RT-KQML을 채택하였다.

2.2 에이전트 구조

본 절에서는 다중 에이전트 시스템을 지원하기 위한 에이전트 구조들을 살펴본다. OAA(Open Agent Architecture)[4]는 에이전트간 통신을 표현하기 위해 독자적인 언어인 ICL(Interagent Communication Language)을 사용하고 있으며 분산 이기종 환경을 지원하고 있다. JADE(Java Agent Development Framework)[6]는 Java 기반의 다중 에이전트 시스템이며 에이전트간 통신을 위해 Java RMI를 채택하였다. DECAF(Distributed Environment-Centered Agent Framework)[5]는 자바 기반의 다중 에이전트를 지원한다는 점에서 JADE와 유사하지만 통신방식으로 소켓통신을 이용하고 있다. 이 구조들은 통신방식을 고유의 것으로 사용하고 있으므로 다양한 컴퓨팅 환경을 포하기 위해 필수적인 표준 미들웨어를 사용하지 않고 있다는 문제가 있다. CORBA를 기반통신으로 사용하기 위한 시도는 KCobalt 시스템[7]에서 제시된 바 있다. KCobalt는 KQML과 CORBA를 기반으로 하는 에이전트 구조이지만 시스템 설계 내용이 구체적으로 보고된 바가 없다. 본 논문에서는 KCobalt에서 제시하는 개념을 기본으로 하여 실시간 분산 다중 에이전트 시스템을 설계한다.

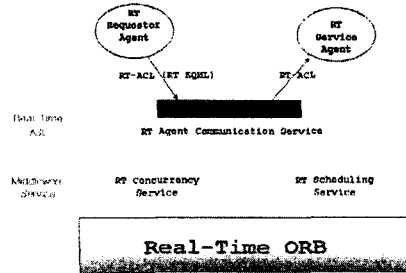
2.3 실시간 스케줄링 알고리즘

다중 에이전트 시스템 내에서 특정 서비스를 제공할 때 소요되는 시간은 각 서비스 에이전트에 따라 다르다. 또한, 제공된 서비스에 대한 품질도 같지 않다. 일반적으로, 서비스 수행 시간이 길면 더 나은 품질을 기대할 수 있을 것이다. 이와 같은 상식을 감안하여 Anytime algorithm들이 실시간 스케줄링 방법으로 제시된 바 있으며 이것을 특히 에이전트 스케줄링에 유용하므로 본 논문에서 제안하는 RT-ASL에 포함된다.

Design-to-time Scheduling[8] 알고리즘은 정해진 시간 내에서 상대적으로 우선순위가 높은 작업을 먼저 수행하도록 스케줄링을 하고 상대적으로 우선순위가 낮은 작업은 나중에 수행하도록 하는 방법이다. Design-to-Criteria[9] 알고리즘은 Design-to-time 알고리즘을 확장하여 제한시간, 우선순위, 실행시간 등의 실시간 요소 중에서 하나의 Criteria를 정해

Criteria를 기준으로 스케줄링을 하는 알고리즘이다.

3. 실시간 에이전트 서비스층 구조



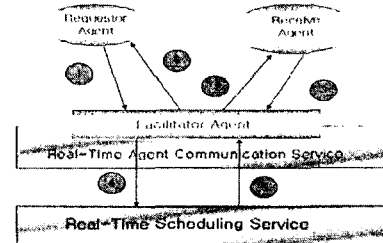
[그림4] 실시간 에이전트 서비스층 구조

우리가 제안하는 시스템 구조는 그림4와 같다. TAO와 같은 Real-Time ORB에서 제공되는 Real-Time CORBA Services 계층이 있고, 이 위에 분산 실시간 다중 에이전트 시스템을 위한 RT-ASL이 위치하고 있는 구조이다.

3.1 실시간 에이전트 통신 서비스

실시간 에이전트 서비스층은 실시간 에이전트 통신 서비스를 이용하여 에이전트간의 통신을 수행한다. 각각의 에이전트는 그림4에서와 같이 실시간 속성을 가진 실시간 에이전트이며 에이전트간 통신을 표현하는 RT-ACL로는 RT-KQML이 사용된다. RT-ASL의 중앙에 위치한 RT-Facilitator Agent는 Requestor Agent와 Service Agent 간의 중재자 역할을 수행하는 중재 에이전트이다.

3.2 실시간 스케줄링 서비스



[그림5] 실시간 스케줄링 서비스

실시간 스케줄링 서비스는 다음과 같이 제공될 수 있으며 그림5는 이러한 시나리오를 개략적으로 도시하고 있다.

- (1) Request Agent는 QoS 인자가 포함된 콘텐츠를 갖고 중재 에이전트에게 Service Agent 우선순위를 요청한다.
- (2),(3) 중재 에이전트는 Requestor Agent의 요구에 맞는 에이전트들을 Receive Agent들 중에서 찾는다.
- (4),(5) 중재 에이전트는 후보로 에이전트가 스케줄 가능함을 실시간 스케줄링 서비스에 요청한다. 만약, 스케줄 가능하다면 스케줄링 서비스에 적절한 우선순위를 반환한다. 그렇지 않으면 스케줄이 불가능함을 알린다.
- (6) 중재 에이전트는 Requestor Agent에게 원하는

Service Agent의 우선순위를 반환한다. 또는 Requestor Agent에게 해당하는 서비스 에이전트가 존재하지 않음을 알린다. Anytime algorithm, Design-to-Time scheduling, Design-to-Criteria scheduling 방법들의 실시간 스케줄링 서비스는 RT-ASL의 기본 서비스로 제공되므로 에이전트 프로그래밍에 사용할 수 있다.

4. 지능형 병원 시스템의 예

본 논문에서 제시한 실시간성이 포함된 분산 다중 에이전트 시스템은 병원 관리 시스템의 원활한 기능 보장 및 확장가능성을 제공하기 위해 매우 유용하다. 실제로 병원관리시스템은 상당한 정도의 context-aware computing의 지원을 요구하고 있다는 사실은 최근 [10]과 같은 연구에서 많이 보고되고 있다. 병원 관리 환경을 위해 사용자 에이전트, 관찰 에이전트 및 스케줄링 에이전트를 정의할 수 있다. 사용자 에이전트는 의사, 간호사 등과 같은 사람과 실질적으로 통신을 하도록 구성할 수 있다. 관찰 에이전트는 환자의 상태를 관찰하면서 환자의 상태 변화가 나타날 때 기록을 유지하며 스케줄링 에이전트에게 알리는 에이전트이다. 스케줄링 에이전트는 관찰 에이전트로부터 받은 정보에 대해 적절한 스케줄링을 하여 적합한 사용자 에이전트를 호출하는 에이전트이다. 에이전트 각각은 독립적으로 작업을 수행한다. 작업을 수행하는 도중에 다른 에이전트의 서비스를 필요로 할 경우에는 다른 에이전트를 호출하여 해당하는 서비스를 제공 받는다. 응급실의 경우에는 평소보다 다른 에이전트를 호출하는 작업이 빈번해진다. 또한, 시간제약에 대한 중요성도 커지게 된다. 얼마만큼 빨리 해당하는 서비스를 제공하고 있는 에이전트를 찾느냐에 따라 생사의 갈림길을 넘나드는 경우도 있기 때문이다. 그렇다고 무조건 시간에만 치중한 나머지 품질 낮은 혹은 요구에 맞지 않는 서비스를 제공하는 에이전트가 제공되는 경우도 결과는 마찬가지이다. 실시간성이 포함된 분산 다중 에이전트 시스템이 구축된다면 이와 같은 지능화된 병원 관리용 시스템의 구축이 용이하게 수행될 수 있다.

5. 결론

본 논문에서 우리는 실시간 멀티 에이전트 시스템을 위한 RT-ASL을 제안하였다. 실시간 에이전트 통신 서비스를 제공하기 위해 Real-Time CORBA를 기반 통신 구조로 채택하였으며 RT-CORBA Service Specification에 따라 실시간 스케줄링 서비스를 포함한 RT-ASL을 정의하였다. 향후 RT-ASL과 RT-IDL의 변환 방법, 연성 스케줄링 기능을 효율적으로 이용할 수 있는 중재 에이전트의 구조, 실제로 병원관리시스템과 같은 응용분야의 요구사항을 분석하여 에이전트간 실시간 협업에 필요한 실시간 파라미터들을 결정하기 위한 연구 등이 구체적으로 수행될 예정이다. 현재는 TAO 환경에서 다중 에이전트 관리를 위한 핵심적인 요소들을 구현하고 있는 중이다.

7. 참고문헌

- [1] Lisa Cingiser DiPippo, Victor Fay-Wolfe, Lekshmi Nair, Ethan Hodys and Oleg Uvarov. A Real-Time Multi-Agent System Architecture for E-Commerce Applications. *Autonomous Decentralized Systems*, 2001. Proceeding. 5th International Symposium on, 26-28 March 2001 Pages:357-364
- [2] <http://www.cs.umbc.edu/kqml/kqmlspec/spec.html>
- [3] <http://www.fipa.org/spec/fipa98.html>
- [4] David L. Martin, Adam J. Cheyer, Douglas B. Moran. The Open Agent Architecture: A Framework for Building Distributed Software System. *Applied Artificial Intelligence*. vol. 13, pp. 91-128. Jan-Mar 1999.
- [5] John Graham and Keith Decker. Towards a Distributed, Environment-Centered Agent Framework. In *Proceedings of the 1999 Intl. Workshop on Agent Theories, Architectures, and Languages [ATAL-99]*, Orlando, July 1999.
- [6] JADE Project. <http://jade.tilab.com>
- [7] D. Benech, T. Desprats. A KQML-CORBA based Architecture for Intelligent Agents Communication in Cooperative Service and Network Management. In *Proceedings of IFIP/IEEE International Conference on Management of Multimedia Networks and Services '97 July 8-10, 1997*.
- [8] Alan Garvey, Victor Lesser. Design-to-time Real-Time Scheduling. *IEEE Transactions on Systems, Man and Cybernetics - Special Issue on Planning, Scheduling and Control*. vol. 23, no. 6, 1993.
- [9] Thomas Wagner and Victor Lessor. Design-to-Criteria Scheduling: Real-Time Agent Control. *Proceedings of the 2000 AAAI Spring Symposium on Real-Time Systems*.
- [10] Applications of context-aware computing in hospital work: examples and design principles, *Proceedings of the ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004*