

Jini 기술 기반 P2P 옥션 시스템 모델

이금용**

**영산대학교 정보통신공학부

e-mail : office@java-tech.com

P2P Auction System Model Based on Jini Technology

Geum-Yong Lee**

**Dept. of Information and Communication, Young-San University

요 약

다수에 의한 다수에 대한 P2P 전자경매 시스템에 대한 모델과 Java 의 Jini 기술을 기반으로 하는 구축방안에 대하여 기술한다. Jini Service 형태로 네트워크에 공개되는 모든 전자경매 진행관련 정보에 대한 검색과 입찰 처리과정을 자동화 할 수 있을 뿐만 아니라, 네트워크 문제 발생에 대한 자동복원, 참가 이후 신규 공개되는 정보에 대한 비동기 자동 확인, 그리고 ACID 개념을 보장하는 트랜잭션 관리 등이 가능하다. 다양한 MIME 타입을 지원하는 컨텐츠와 EJB 를 활용할 경우 유선 및 무선 정보화장치를 통하여 모델에 참여할 수 있으며, 수동 접속된 특정 서버의 제공 정보 및 절차에 제한 받지 않는다.

1. 서론

통상적인 클라이언트 - 서버 시스템에서 가동되는 전자경매 시스템을 이용하기 위해서는, 멤버십을 보유한 전자경매 참가자가 중앙 컴퓨터 서버(= "경매서버")에 접속한 후 경매대금입금 및 사용자 신원사항 입력 등 필요한 조치를 취하여야 경매에 참가할 수 있다. 또한, 경매 참가자는 접속된 경매서버에 등록된 물품과 조건으로만 그 선택의 폭을 제한 받는다.

특히 지정된 경매서버에 접속할 필요가 없이, 네트워크에 등록된 모든 경매관련 정보에 자동 접근할 수 있고 전자 경매 알고리즘에 따라 지정된 조건의 물품의 입찰에 자동 참여할 수 있는 새로운 형태의 전자경매 시스템을 개발할 필요가 있다.

Java 의 Jini 기술을 이용하면 위와 같은 필요성을 만족시켜줄 수 있는 새로운 형태의 전자경매 시스템을 개발할 수 있다. Java 로 개발되기 때문에 모든 컴퓨터 운영체제에서 실행될 수 있으며, 유선 및 무선 정보화장치를 구별하지도 않는다.

주어진 네트워크에서 경매에 참여하고자 하는 각 개인은 필요한 정보를 편집한 다음 Jini Service 형태로 네트워크에 공개(publish)하며, 원하는 조건을 충족시키는 물품의 자동검색, 입찰참여 및 낙찰시의 필요조치, 경매결과 통보 등 전 경매과정은 자동으로 이루어지게 된다.

경매 최초 참가 이후에 네트워크에 공개되는 정보라 할 지라도 비동기 이벤트로서 경매 시스템에 등록된 모든 참가자에게 공지된다.

제 2 절에서는 Jini 기술에 대한 개요를 소개하고, 제 3 절에서는 본 논문에서 제시되는 P2P 경매시스템에 대한 모델을 제시한다. 제 4 절에서는 설계 내용을 제시한다.

2. P2P 를 위한 Jini 기술

P2P 는 네트워크 정보공유, 네트워크 커뮤니티 운영, 그리고 장치관리, 서비스 모니터링, 리소스 관리, 웹서비스 등의 어플리케이션 통합 등 다양한 노드간 직접 서비스를 제공할 수 있다. 전자상거래를 위한 다양한 서비스도 P2P 형식으로 속속 개발되고 있다.

P2P 아키텍처는 일반적으로 전송층 (TL; Transport Layer) 이상에서는 표준화되지 않았다고 말할 수 있다. TL 이상의 프로토콜은 개별 어플리케이션에서 정의할 수 있는 것이다.

TL 이상의 P2P 용 프로토콜로서 Gnutella, Freenet, Jxta 등을 사용할 수 있으나, 여기에서는 단순한 프로토콜 이상의 고급 서비스를 지원하고 Java 바이트코드에 의해 실행되는 Jini 기술에 대하여 간단하게 소개함으로써 본 논문에서 제안하는 P2P 전자경매 시스템 모델에 대한 이해를 돕고자 한다.

Jini 기술[1],[2][3]은 네트워크에 존재하는 다양한 서비스("Jini Services" - 사용자 정의된 데이터와 기능을 제공하는 자바 오브젝트)들의 네트워크 연합체 (Jini Federation)를 구성하고 운영하며, Federation에 대한 참여방법 및 서비스간 통신을 지원한다. 사용자 정의 Jini 서비스는 Jini API에서 정의되는 "Service Interface"를 구현하여 정의된다. 사용자 정의 서비스를 지원하기 위한 시스템 서비스를 요약하면 다음과 같다.

2.1 Discovery Service

Jini Federation에 존재하는 레지스트리, 즉 룩업서비스 (Lookup Service, LUS)를 검색하고, LUS에 등록된 사용자 정의 서비스를 식별하는 시스템 서비스이다. 지정된 URL에 대한 유니캐스트(Unicast) 메시징 서비스 및 서브넷으로 제한되는 IP 환경에 대하여 주기적으로 실시하는 멀티캐스트 (Multicast) 메시징 서비스로 구성된다.

2.2 Join / Publish Service

Discovery Service를 통하여 LUS를 검색한 후 해당 Jini Federation에 사용자 정의 Jini 서비스를 멤버로서 Join하고, 관련 서비스 Proxy 오브젝트를 등록하기 위한 시스템 서비스이다. Discovery Service와 마찬가지로 유니캐스트 메시징 서비스 및 멀티캐스트 메시징 서비스를 이용한다.

2.3 Event Service

분산 이벤트를 처리하기 위한 시스템 서비스이며, Event Mailbox Service의 형태로 구현되어 있다. 발생하는 네트워크 이벤트의 저장과 포워딩 서비스, 그리고 연결 해제된 클라이언트 혹은 사용자 정의 서비스에 대해서는 비동기 (Asynchronous) 이벤트 공지가 가능하다.

2.4 Leasing Service

Jini Federation를 구성하는 다양한 서비스 오브젝트 자체 및 소모 리소스에 대하여 지정 시간동안 그 유효성이 보장되도록 필요한 관리작업을 제공하는 시스템 서비스이다. 사용자 정의 서비스가 그 자신의 유효성을 보장 받기 위해서는 정기적 Lease Renewal 요청을 LUS에 내야 한다. 그렇지 않을 경우 LUS에서는 해당 오브젝트를 삭제하게 된다.

2.5 Transaction Service

Jini Federation를 구성하는 모든 서비스와 그 리소스에 대하여 ACID 속성을 유지하는 트랜잭션 서비스를 제공하기 위한 시스템 서비스이다.

2.6 Persistence Service

가상 공간 (= Tuple Space)상에 정의되는 공유메모리 모델 및 리모트 이벤트 모델에 근거하여 Jini Federation를 구성하는 각 요소 오브젝트에 대한 데이터 저장 서비스를 제공하는 API 서비스이며,

JavaSpaces라고 불린다. Transient Space 및 Persistent Space의 2가지 구현모델을 지원하는데, Persistent Space는 보조 저장장치를 이용하여 서비스 오브젝트를 저장하며 시스템 재시작시 이전의 서비스를 자동 복원한다.

3. Jini 옥션 시스템 모델

본 논문에서 제안하는 Jini 기술 기반 P2P 옥션 시스템은 AuctionInterface, JiniSystem, AuctionSystem 등 3개의 Tier로 구성된다.

3.1 AuctionInterface

경매참가자 (입찰참가자, 물품판매자)가 사용하는 자바 어플리케이션이며 유, 무선 정보입력장치에 설치된다.

GUI를 통하여 입력되는 참가자의 신원정보 및 입찰정보를 속성 (Service Attributes)으로 하는 사용자 정의 Jini 서비스 오브젝트 (자바 클래스 파일이며, 패키지 java.rmi.server의 UnicastRemoteObject를 확장한 오브젝트)가 생성되어, Jini Federation에 등록하기 위한 Jini API 서비스 (즉, Discovery Service 및 Join/Publish Service)를 경유하여 네트워크에 전송된다.

네트워크에서 옥션에 관한 중요 정보를 비동기 수신하기 위하여 RemoteEventListener 인터페이스를 구현한다. 지정된 조건의 경매절차를 완료할 수 있는 상태에 도달되었을 경우 사용자의 개입에 의해 경매절차를 종료 혹은 취소할 수 있다. 이러한 사용자의 개입내용 역시 JiniSystem의 Event Mailbox Service를 통하여 원격 수신하게 된다.

3.2 JiniSystem

Jini Federation을 구성하고 운영하기 위한 시스템 서비스를 제공하며, Lookup Service (LUS), Event Mailbox Service, Leasing Service, Transaction Service, Persistence Service로 구성된다. 이러한 서비스는 썬 마이크로 시스템즈사에서 공표한 표준서비스 (Reference Implementation)를 사용하며, 네트워크 접속된 임의의 노드에 설치할 수 있다. 그 밖에, 각 Jini 서비스 클래스의 정의를 위한 메타정보를 저장하기 위한 별도의 HTTP Web Server가 필요하다.

LUS는 AuctionInterface로부터 전송되는 사용자 정의 서비스 오브젝트의 레지스트리를 등록, 관리하며, Event Mailbox Service는 LUS에 신규 정보등록, 공지대상 경매 진행정보, 경매절차에 관한 사용자 개입내용을 자바 이벤트 오브젝트 (net.jini.core.event 패키지의 RemoteEvent 오브젝트)에 포장하여 비동기적으로 Jini Federation 멤버 (즉, RemoteEventListener 인터페이스를 구현하고, LUS에 등록된 모든 Jini 서비스 오브젝트)에 송신한다. 그 밖의 JiniSystem Services는 제2절에서 소개한 표준적인 시스템 서비스와 같다.

3.3 AuctionSystem

입찰과 낙찰의 조건 처리, 그리고 거래대상 물품과

응찰자 관련 정보를 위한 DB 관리, 경매 진행 결과 저장을 위한 DB 관리 등 전자경매 진행을 위한 전통적인 프로토콜을 관리하고, JiniSystem 과 통신하기 위한 Tier 이다.

AuctionSystem 은 통상적으로 JiniSystem 을 지원하기 위한 Web Server 가 설치된 동일한 하드웨어에 데이터베이스 시스템과 함께 설치하는 것이 좋다. 이것은 J2EE (Java 2 Enterprise Edition) API 를 지원하는 자바 오브젝트의 설치와 실행을 서버사이드에서 지원하는 J2EE Container (서버 어플리케이션의 일종)[4] 오브젝트와의 연동을 통해, 서비스 가능한 Jini Federation 의 범위를 확장하고 기존 시스템 및 타 프로토콜의 채용 가능성을 열어둘 수 있기 때문이다.

3.4 J2EE Enterprise Java Beans (EJB) Supports

본 논문에서는 JiniSystem 및 AuctionSystem 을 구성하는 주요 서비스 오브젝트에 대하여 EJB 2.0 Specification[5]을 준수하고 지원하도록 요건화 하였다. 제안되는 P2P 옥션 시스템을 구현함에 있어서 표준적인 웹서비스[6]로서 제한받는 상황을 방지하기 위함이다.

다.

즉, 기존 클라이언트-서버 기반 옥션시스템으로의 변환, Legacy System 과의 원활한 통합, CORBA 오브젝트와의 통신 가능성, MQ 를 기반으로 하는 Messaging Service 와의 인터페이스, SOAP, WSDL, UDDI 등 기타 프로토콜의 채용 편의성, AuctionInterface 를 JSP, ASP 등으로 제작하고 이것을 Jini Service 오브젝트로 변환하기 위한 Servlet 의 사용가능성 등을 고려해야 하기 때문이다.

4. Jini 옥션 시스템 설계내용

본 논문에서 제시하는 P2P 옥션 시스템 모델의 UML[7] Sequence Diagram 을 그림 1 에 보인다.

물품 판매자가 전자경매 과정을 통하여 물품을 판매하기 위해서는 AuctionInterface 의 GUI 를 통하여 제공물품에 관한 상세사항 (낙찰조건 포함)을 사용자 정의 Jini Service 의 형태로 JiniSystem 에 공개, 등록하여야 한다. 전자경매 과정을 통하여 물품을 구매코자 하는 입찰참가자 역시 동일한 절차를 거친다.

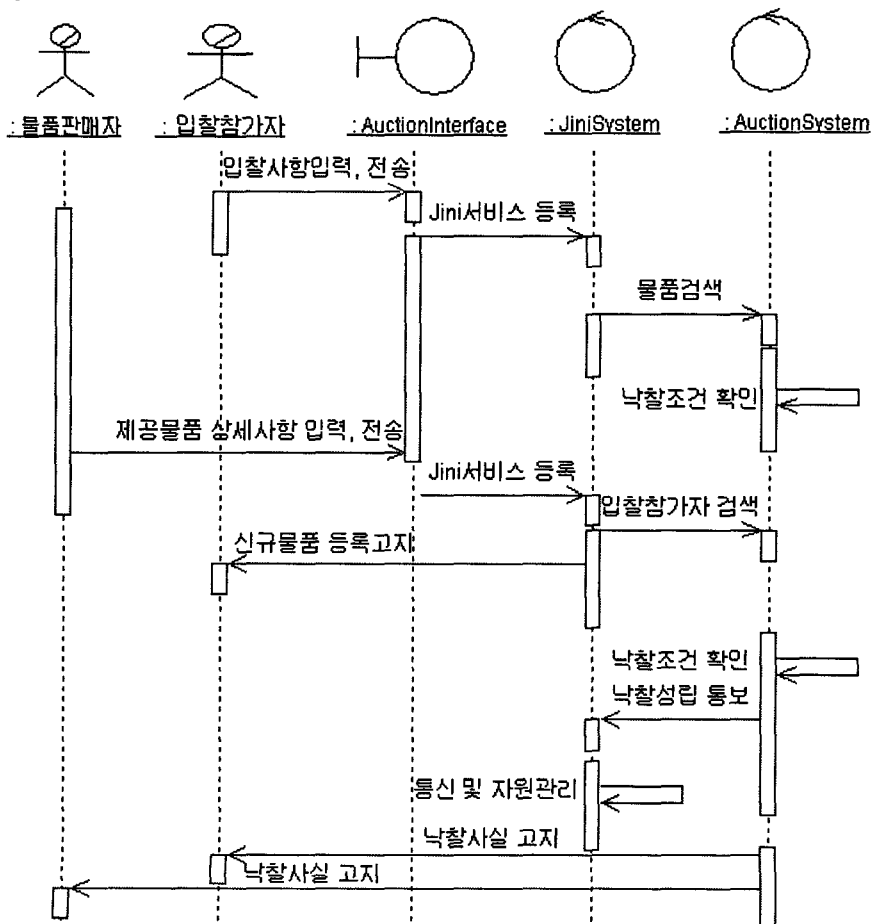


그림 1. P2P 옥션 시스템 모델에 대한 UML Sequence Diagram

물품과 입찰 참가자에 대한 상세 데이터를 속성으로 하는 사용자 정의 Jini Service 의 레지스트리를 유지, 관리하는 LUS 는 모든 입찰참가자 및 물품판매자에게 현재 관리중인 내용중에 원하는 경매 조건을 충족하는 서비스 오브젝트를 반환한다. 경매 조건은 net.jini.core.lookup.ServiceTemplate 오브젝트로서 작성된다. 신규 정보가 네트워크에 등록될 경우, JiniSystem 의 Event Mailbox Service 는 RemoteEventListener 인터페이스를 구현한 사용자 정의 Jini Service 에게 비동기(Asynchronous)로 이벤트 오브젝트를 전송한다.

AuctionSystem 은 낙찰조건을 확인한 후 낙찰성립을 판단하고, 필요시 낙찰성립을 통보하기 위한 알고리즘을 구현하며, 경매참가자가 네트워크에 공개한 경매정보의 데이터베이스 저장 및 관리를 담당하게 된다. 그림 2 를 참조하자. 이와 같은 과정을 보여주는 UML use-cases Diagram 이 그림 2.에 나타나 있다.

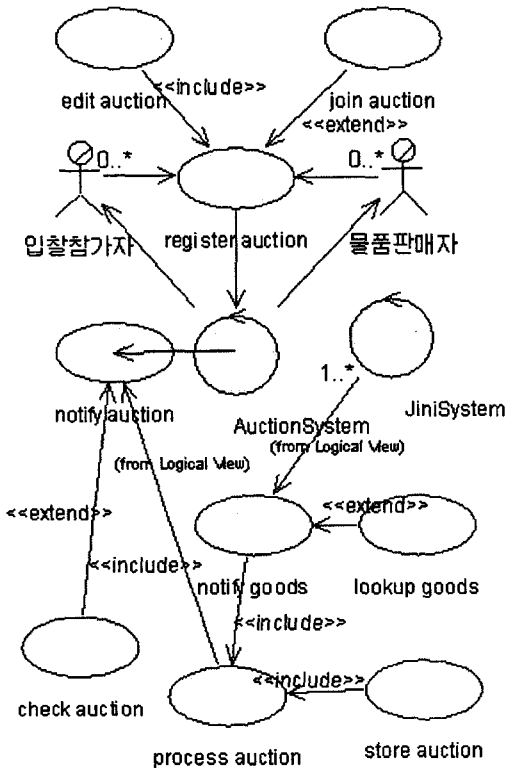


그림 2. P2P 옥션 시스템 모델 UML use-case Diagram

그림 2 에 보이는 Use-Case 는 다음과 같다.

edit auction : AuctionInterface 에서 제공되는 GUI 를 통하여 경매의 참가조건을 작성한다. 작성된 상세내역을 속성 (Service Attributes)으로 하는 Jini Service 오브젝트가 생성된다.

join auction : net.jini.lookup.JoinManager 오브젝트를 통하여 Jini Service 를 Jini Federation 의 멤버로 참여, 네트워크에 공개한다.

register auction : Jini Service 오브젝트에 필요한 클래스

정의등 필요한 메타 데이터를 HTTP Web Server 에 설치하고, 리소스 Lease 내역 설정 등 JiniSystem 에 필요한 서비스 (제 2 절 참조)를 설정한다.

notify auction : 낙찰, 유찰 여부를 RemoteEvent 오브젝트의 형태로 Jini Federation 에 공지한다.

check auction : 해당 경매 조건을 만족하는 경매과정을 종료할 것인지 확인한다. 경매 참가자에게 Prompt 를 제시하고 그 회신결과에 근거하거나 지정된 시간이 경과한 후에도 낙찰되지 않은 경매과정은 종료한 것으로 본다.

process auction : 경매 참가자간 조건을 검색, 낙찰 조건 충족여부를 판단한다.

store auction : Jini Service 의 형태로 공개된 낙찰 조건과 상세내역을 저장한다.

notify goods : 지정 조건을 충족하는 것으로 검색된 물품정보 Service 를 Jini Federation 에 공지한다.

lookup goods : 조건을 충족하는 물품정보를 나타내는 Jini Service 를 검색한다.

입찰 참가자 및 물품판매자가 AuctionInterface 를 통하여 네트워크에 공개하는 모든 정보는 Jini Service 오브젝트의 형태로 변환되어야 한다는 점에 주목하자.

5. 결론

본 논문에서는 Jini 기술을 사용하여 P2P 방식의 전자 경매 시스템을 구현할 수 있는 3 Tier 모델을 제시하였다. 경매 과정에 참여하는 모든 정보는 Java bytecode 로 작성되는 Jini Service 오브젝트로서 네트워크에 공개, 등록되며, 등록된 정보에 근거한 경매과정은 자동화된다.

Jini 기술을 채택함으로써 얻을 수 있는 장점은, 모든 LUS (Jini Lookup Service) 하드웨어가 제거되지 않는 한 네트워크 다운 등의 상황에서도 서비스를 지속할 수 있으며, Java 를 개발언어로 하고 있으므로 하드웨어의 운영체제에 무관하게 실행할 수 있으며, 별도의 경매 서버를 운영할 필요가 없다는 점 등일 것이다.

시스템에 등록되는 모든 Jini Service 가 J2EE 를 지원하도록 구현되므로, 표준적인 웹 서비스로서 J2EE 가 보장하는 신뢰성과 확장성을 지원한다.

참고문헌

- [1] W. Keith Edwards, "Core Jini", 2nd Ed., Prentice Hall, 2001
- [2] Sing Li, et al., "Professional Jini", Wrox Press, 2000
- [3] Robert Flenner, "Jini and JavaSpaces Application Development", Sams Publishing, 2002
- [4] Michael Girdley, et al., "J2EE Applications and BEA WebLogic Server", Prentice Hall PTR, 2002.
- [5] Richard Monson-Haefel, "Enterprise Java Beans", 3rd Ed., O'Reilly & Associates, 2001
- [6] Steve Graham, et al., "Building Web Services with Java", Sams Publishing, 2002
- [7] CT Arrington, "Enterprise Java with UML", John Wiley & Sons, 2001