

개인사용자를 포함하는 JXTA 기반의 Service Oriented Architecture 구현

김태호*, 이홍철**, 천현재***, 김준룡****

Implementation of a Service Oriented Architecture based on JXTA

- an SOA framework involving users -

TaeHo Kim*, HongChul Lee**, HyeonJae Cheon***, JunRyong Kim****

요약

본 논문에서는 상대적으로 B2B 환경만을 고려하고 있는 SOA 프레임워크를 대신하여 일반 사용자들에게도 원활한 서비스 제공을 가능하게 하는 SOA 프레임워크를 제시한다. 이를 위해 SOA의 개념과 특징, 구성요소에 대해 살펴본 후 이를 차세대 P2P 플랫폼으로 부각되고 있는 JXTA를 사용하여 구현한다. 제안된 프레임워크는 다른 SOA 프레임워크들의 기본적인 기능들을 포함하면서, 다른 프레임워크들에 비해 서비스 공급과 사용이 쉽고, 서비스가 데이터와 그 데이터를 제어할 수 있는 어플리케이션이 실행 가능한 형태로 공급되어 보안 문제를 해결할 수 있으며, 각각의 peer들이 서비스 공급자, 서비스 사용자, 서비스 저장소의 역할을 동시에 수행할 수 있는 다양한 장점이 있다.

Abstract

This paper presents an SOA framework that enables smooth provision of service to ordinary users while also replacing the current B2B environment-oriented SOA framework. To that end, the proposed SOA will be reviewed for its concept, characteristics, and components. Key to its design is that it will be implemented using the prospective next-generation P2P platform known as JXTA. The proposed framework includes all the basic functions of other SOA frameworks, and will make provision and use of services more easy compared with other frameworks. Under this service, data and applications controlling these data are provided in an enabling format, thereby addressing security problems. Another advantage of this proposed framework is that each peer is able to play all roles of a service provider, service client, and service registry.

▶ Keyword : Service Oriented Architecture (SOA), JXTA

• 제1저자 : 김태호 교신저자 : 이홍철

• 접수일 : 2007. 6.22, 심사일 : 2007. 7.20, 심사완료일 : 2007. 8.22

* 고려대학교 정보경영공학부 ** 고려대학교 정보경영공학부 교수

*** 고려대학교 BK21 유비쿼터스 정보보호 사업단 연구 전임 강사 **** 중앙대학교 컴퓨터공학과

※ 이 논문은 2007년도 두뇌한국 21사업에 의하여 지원되었음

I. 서론

최근 고객의 요구가 다양해지고, 기업의 사업 환경이 점점 복잡해짐에 따라 기업의 정보 시스템 또한 그 복잡성을 더해가고 있다. 기업이 지속적으로 발전하고 고객의 만족도를 높이기 위해서는 고객의 다양한 요구에 대해 빠르게 대응하고 시대의 흐름에 맞는 다양한 서비스를 기민하게 제공해 주어야 한다. 이러한 시대적인 요구 사항에 맞춰 최근 Service Oriented Architecture(이하 SOA)[1]가 부각되고 있다. SOA는 1996년 Gartner Group에 의해 소개된 개념으로 일관되고 표준화된 규칙으로 이기종 시스템들 간에 정보공유가 가능하게 하는 기술이다. Gartner Group에 따르면 2008년까지 신규 개발 프로젝트의 80%이상이 SOA를 기반으로 개발될 것으로 전망하고 있다[2].

하지만 대부분의 SOA의 프로젝트가 개인 사용자를 배제한 채 웹서비스를 기반으로, 주로 대규모 기업 업무환경에서의 시스템 통합(System Integration)만을 고려하여 구축되고 있는 실정이다.

따라서 본 논문에서는 SOA의 개념과 특징, 구성요소에 대해 살펴본 후 이를 차세대 P2P 플랫폼으로 부각되고 있는 JXTA[3]를 사용하여 구축하고, B2B에서 서비스 제공을 원활히 하기 위해 주로 활용되는 SOA의 개념을 일반 사용자들에게도 원활한 서비스 제공이 가능하도록 그 개념을 더욱 확장시킨 새로운 비즈니스 모델을 위한 SOA 프레임워크를 제시한다.

II. 관련 연구

2.1 기존 연구

현재 진행되고 있는 많은 프로젝트에서 SOA를 구축하기 위해 보통 웹서비스 기술이 많이 사용되고 있다. Victor Guevara-Masis et. al.[4]은 웹서비스를 사용하여 teleCARE SOA 시스템을 구축하였으나 기초적인 파일럿 시스템 제안에 그쳤다. JaeHoon Cho et. al.[5]은 SOA와 웹서비스의 관계를 설명하고, SOA 구축 방법론에 따라

항공사의 업무에 적용해 보았지만 기업업무만을 고려하였다. Pierpaolo Baglietto et. al.[6] 기업이 SOA를 도입하기 위한 단계적 방법론을 제시하고, 그 방법론을 실제 파일럿 시스템에 도입해보았지만 항만시스템에 한정하였다.

하지만 웹서비스 기술의 부족한 점을 보완하고 보다 효과적인 SOA를 위한 다양한 구축 방법이 요구 되고 있다. Nathalie Furmento et. al.[7]은 JINI[8], JXTA, OGSA[9]을 이용하여 기본적인 SOA 환경을 제안해보고 각각의 장·단점을 제시하였으나 보안 문제는 고려하지 않았다. Aphrodite Tsalgatidou et. al.[10]은 Web, P2P, Grid 환경에서 SOA 서비스 개발을 위한 명세를 제공하였으나 기본적인 로드맵을 제공하는데 그쳤다. Feng Yang et. al.[11]은 JXTA를 이용하여 P2P 서비스를 위한 구조를 제시하였으나, JXTA에서 제공하고 있는 기본적인 구조를 이용하는데 그쳤다.

2.2 SOA

SOA는 문자 그대로 서비스 지향 아키텍처이다. 서비스란 생산된 재화를 운반·배급하거나 생산·소비에 필요한 노무를 제공하는 것을 의미한다. 일반적으로 SOA는 다음과 같이 정의된다. W3C는 SOA를 “호출 가능한 컴포넌트의 집합”으로 정의했다. 여기에서 말하는 컴포넌트란 그 인터페이스의 정의 내용이 공개(Publish) 및 발견(Discovery)가 가능한 것을 의미한다. Mark Colan은 SOA를 “서비스를 요청하거나 제공하여 응용 프로그램 기능이 가능하도록 하는 정책이나 관례 또는 골격이다”고 정의했다[12]. SOA 관점에서 서비스는 인터페이스를 통해 자신이 가진 비즈니스 프로세스를 처리할 수 있는 컴포넌트로 정의된다. SOA의 서비스는 다음의 조건을 만족한다[13].

- 약결합 방식(Loose Coupling): 사용자는 서비스 사용에 있어서 서비스의 내부구조나 실행방식을 알 필요가 없다. 또한 서비스는 플랫폼이나 언어에 독립적이다.
- 서비스는 표준화된 호출 방식으로 제공된다: 웹서비스의 경우 WSDL을 이용하여 서비스를 정의한다.
- 서비스는 자신의 상태를 스스로 유지한다(self-contained).

이러한 특성을 만족하는 SOA의 도입으로 인해, 해당 기업에서 활용 가능한 서비스 단위로 그 기능을 수행하는 소프트웨어의 개발이 가능하다.

III. SOA와 JXTA의 구조

3.1 SOA의 구성요소

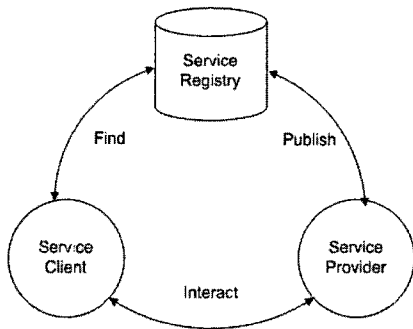


그림 1. SOA 구성요소[12]
Fig 1. Components of SOA[12]

SOA를 구현하기 위해서는 <그림 1>에서 보여주는 각 요소들이 필요하다. 각 요소의 특징은 다음과 같다[14].

- 서비스 사용자(Service Client): 서비스 공급자가 공급하는 하나 이상의 서비스를 사용한다.
- 서비스 공급자(Service Provider): 서비스 사용자가 이용할 수 있는 서비스를 공급한다. 자신이 직접 서비스를 사용하여 서비스 사용자가 될 수도 있다.
- 서비스 저장소(Service Registry): 서비스에 대한 정보를 저장하고 있다. 서비스 공급자가 서비스를 등록하고, 서비스 사용자는 자신의 원하는 서비스를 검색하여 사용한다.

이러한 SOA의 구성요소들은 서로간의 서비스 제공·사용을 위해서 반드시 서비스 사용자·공급자간에 서로 알고 있는 표준화된 문서 형식을 사용 해야 한다. SOA를 구체화한 사례 가운데 최근 가장 주목 받고 있는 웹서비스 기술에서는 이러한 표준형식들을 <그림 2>와 같이 WSDL(Web Service Definition Language)[15], UDDI(Universal Description, Discovery and Integration)[16], SOAP(Simple Object Access Protocol)[17] 을 사용하여 구체화 하였다.

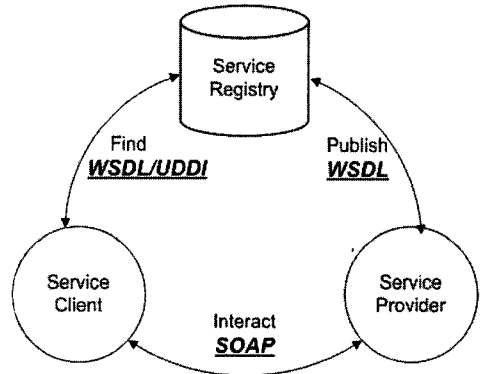


그림 2. 웹서비스에서의 SOA 구현[12]
Fig 2. Components of SOA in Web service[12]

3.2 JXTA의 구성요소

본 논문에서는 차세대 P2P 플랫폼인 JXTA를 이용하여 SOA를 구현하였다. JXTA는 이기종 시스템, 다른 플랫폼 시스템들 간의 분산 컴퓨팅, P2P 컴퓨팅, 유비쿼터스 컴퓨팅을 지원하기 위해 2001년 선 마이크로 시스템에 의해 제안되었다. JXTA는 P2P에 특화된 플랫폼이기 때문에 JXTA를 이용하여 SOA를 구축시, 그 장점을 활용한 다양한 기능 부가가 가능하다. JXTA의 구조는 <그림 3>과 같이 크게 라우팅과 같은 peer 설정과 통신기능의 JXTA Core Layer와 인덱싱과 자료검색, 파일 공유 등의 기능의 JXTA Services Layer, 메일링이나 스토리지 시스템과 같은 JXTA Applications Layer로 구분된다[18].

IV. SOA 구현

4.1 새로운 비즈니스 모델을 위한 JXTA 기반의 SOA 구현

본 논문에서는 새로운 비즈니스 모델을 위한 JXTA 기반의 새로운 SOA 프레임워크를 제안한다. 제안된 프레임워크는 현재 많이 이용되고 있는 웹서비스 기반 SOA가 아닌 범용 P2P 플랫폼인 JXTA를 이용하여 구축한 SOA 프레임워크이다. 기존의 웹서비스를 이용하여 구현한 SOA는 주로 기업 업무환경에서의 시스템 통합만을 고려하여 비교적 작은 단위의 서비스만을 제공하였다고 한다면 본 논문에서 제안하는 프레임워크는 서비스 자체만으로 실행

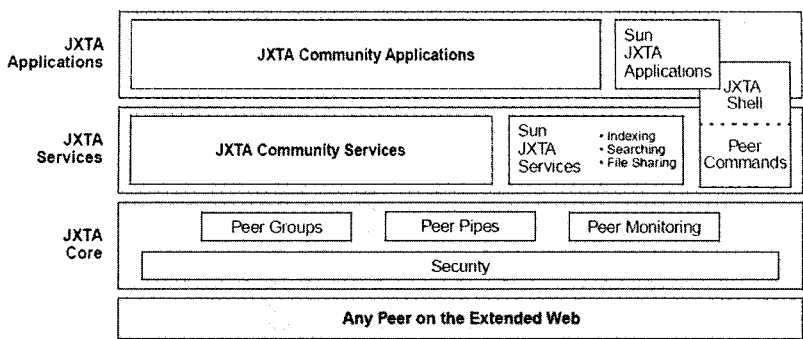


그림 3. JXTA의 구조(18)
Fig 3. Structure of JXTA(18)

가능한 완전한 어플리케이션 형태의 서비스로 공급되기 때문에 기존의 기업의 업무환경뿐만 아니라, 다양한 서비스를 원하는 일반 사용자들에게 공급되어도 아무런 문제가 생기지 않는다. 이러한 특징으로 인해 제안된 프레임워크는 기존의 웹서비스 기반 SOA와는 다른 방법으로 서비스의 보안 문제에 접근 할 수 있다. 제안된 프레임워크는 완전한 형태의 서비스 즉 데이터와 그것을 제어할 수 있는 어플리케이션이 동시에 공급되기 때문에 공급된 데이터의 관리가 용이해진다. 또 서비스가 SOA 프레임워크 위에서 동작하기 때문에 비정상적인 서비스의 제어가 가능해진다. 서비스의 보안 문제를 효과적인 방법으로 제어 할 수 있다. 또한 논문에서 제안하는 프레임워크 상에서 서비스와 서비스들 간의 통신이 가능하기 때문에 기업 환경의 시스템 통합 문제 역시 유연하게 받아들일 수 있는 구조이다. 제안된 프레임워크는 기존의 웹서비스 기반 SOA와는 달리 서비스 공급자·사용자들의 자원을 효과적으로 분배·사용 할 수 있는 P2P 네트워크 상에서 서비스가 이루어

지기 때문에 네트워크 과부하를 줄일 수 있다. 이기종 시스템, 다른 플랫폼 시스템 환경을 지원하는 JAVA기반 프레임워크이기 때문에 기존 웹서비스 기반 SOA에 비해 다양한 환경에서의 서비스 지원이 가능하다.

4.2 새로운 비즈니스 모델을 위한 JXTA 기반의 SOA 구현

제안된 프레임워크는 <그림 4>와 같이 그 기능에 따라 SOA Manager, Network Manager, Service Manager, Plug-in Manager로 구분된다.

SOA Manager는 각 모듈들을 로딩하고 동작 상태를 점검한다. SOA Manager는 어플리케이션 시작시에 Network Manager를 시동하여 P2P 네트워크에 참여하고 나머지 모듈들을 프로그램에 포함 시키는 기능을 한다. 그 외에도 제공받은 서비스의 리스트를 관리하는 역할을 한다.

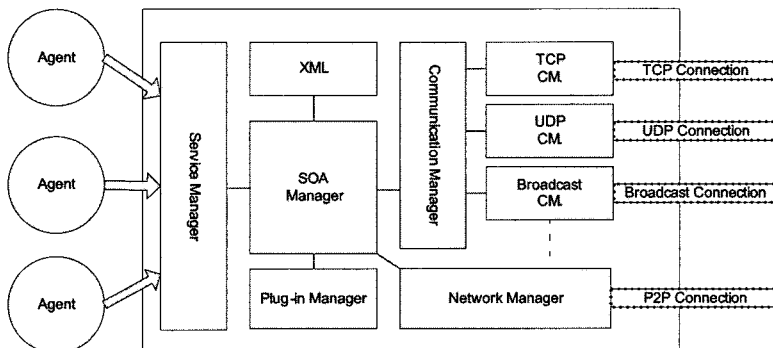


그림 4. 프레임워크 아키텍처
Fig 4. Framework Architecture

Service Manager는 서비스 사용자가 효과적으로 서비스를 사용할 수 있는 방법을 제공한다. 제공받은 서비스를 사용자가 이용하기 위해서는 컴포넌트 상태의 서비스를 프레임 워크 상에서 실행시켜야 한다. 웹 서비스 에서는 XML 형태의 데이터만을 받아 메서드 형태의 간단한 컴포넌트에서 수행하여 그 결과값을 요청자에게 되돌려주는 방식을 사용하고 있다. 이러한 방법은 시스템 통합이라는 측면에서는 효율적인 방법이지만 일반 사용자가 사용하는 어플리케이션에 적용하기에는 무리가 따른다. 그래서 제안된 프레임워크에서는 특정 interface를 상속한 클래스를 직접 전달하고, Service Manager를 이용하여 클래스 파일을 로딩하는 방식을 제공한다. 이러한 방식은 서비스 사용자의 관점에서 실행 가능한 완전한 형태의 서비스를 제공하는 이점이 있다.

Network Manager는 서비스의 등록과 배포를 관리한다. 사용자가 자신에게 필요한 서비스를 이용하려면 서비스에 대한 정보를 전달받고 서비스 수행에 필요한 클래스 파일과 XML 파일을 전송 받는 방법이 필요하다. 기존의 웹서비스를 이용한 SOA에서는 필요한 서비스를 검색하기 위해서 서비스 사용자가 직접 서비스 공급자의 서비스 목록을 찾아서 직접 목적 프로그램에 웹서비스를 등록해 주어야 한다. 서비스 저장소에 등록된 수많은 서비스 목록을 모두 다 알 수 없는 개인 사용자에게 이러한 작업은 매우 불편하다. 따라서 제안된 프레임워크에서는 이러한 단점을 보완하여 Network Manager를 통해 서비스가 등록될 때마다 각 프레임워크에 등록된 서비스를 Broadcasting 한다. Broadcasting된 서비스들은 P2P 네트워크를 통해 다

양한 루트로 서비스 사용자에게 전달된다. 서비스 사용자들의 접속, 서비스의 시작 등 서비스에 관련된 기본적인 행동들은 <그림 5>와 같이 리스너 패턴으로 구현했다.

Plug-in Manager는 추후 확장 가능한 기능을 수용하기 위한 구조이다.

4.3 프레임워크 구현

완전한 형태의 SOA 기능을 구현하기 위해서는 앞서 3.1절에서 언급한 SOA 구성요소들인 서비스 공급자, 서비스 요청자, 서비스 저장소의 기능이 충족되어야 한다. 제안하는 프레임워크 역시 이러한 SOA 필수 구성요소들을 기본으로 구성된다.

<그림 6>는 제안된 프레임워크의 시스템 구조이다. 제안된 프레임워크는 서비스를 제공하는 서버의 과부하를 피하고, SOA의 개념을 더욱 확장하여 개인간의 서비스 공급·사용을 원활히 할 수 있는 P2P의 특성을 적극 활용하기 위하여 범용 P2P 프레임워크인 JXTA를 기반으로 작성되었다. <그림 6>에서와 같이 이 프레임워크는 P2P 네트워크상에서 동작하기 때문에 다른 SOA 구현물과는 다르게 각각의 peer들이 서비스 공급자, 서비스 사용자, 서비스 저장소의 역할을 모두 수행할 수 있는 구조이다. P2P환경에서 동작하는 각각의 peer들은 다양한 P2P Network(TCP, Broadcast, RCP)로 서로 연결되어 특정한 상황에 따라서 서비스를 공급해주는 서비스 공급자, 서비스를 사용하는 서비스 사용자, 서비스를 저장하는 서비스 저장소의 역할을 복합적으로 수행 할 수 있다. 서비스는 <그림 7>의 과정에 의해 공급 된다.

```
class 클래스 이름 implements AhraListener, NetworkListener {
    /*서비스를 시작하기 위한 준비*/
    public void openService(AhraEvent e) {
        ahraService = AhraService.getInstance();
        ...
        //서비스시작을 위한 서버 오픈을 한다.
        new OpenPeer();
    }
    /*서비스에 접속하기 위한 준비*/
    public void acceptService(AhraEvent e) {
        ahraService = AhraService.getInstance();
        //시작된 서비스서버에 접속한다.
        new AcceptPeer(this, ip, port);
    }
    /*내가 게시한 서비스에 새로운 에이전트가 접속했을 때 준비*/
    public void discoveryEvent(String ip) {
        // TODO 자동 생성된 메소드 스템
        System.out.println("아이피 "+ip);
        sender.addPeerInfo(new PeerInfo(ip, 42050));
    }
}
```

그림 5. 리스너 패턴으로 구현된 서비스 인터페이스
Fig 5. Implemented as the listener pattern

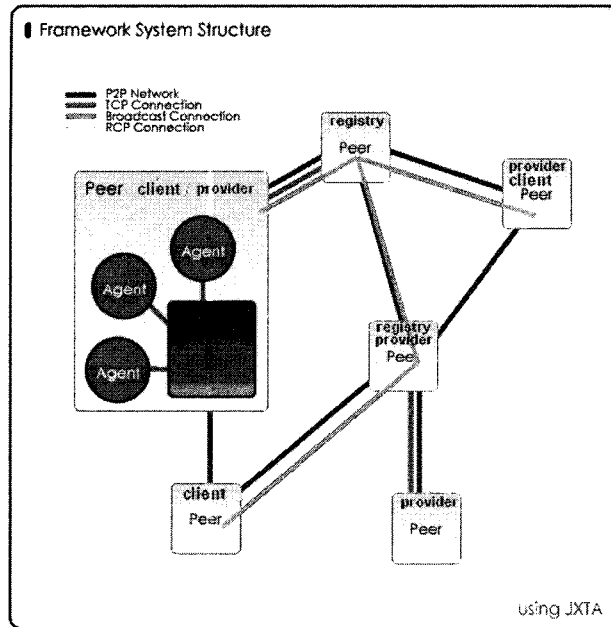


그림 6. 프레임워크 시스템 구조
Fig 6. Framework System Structure

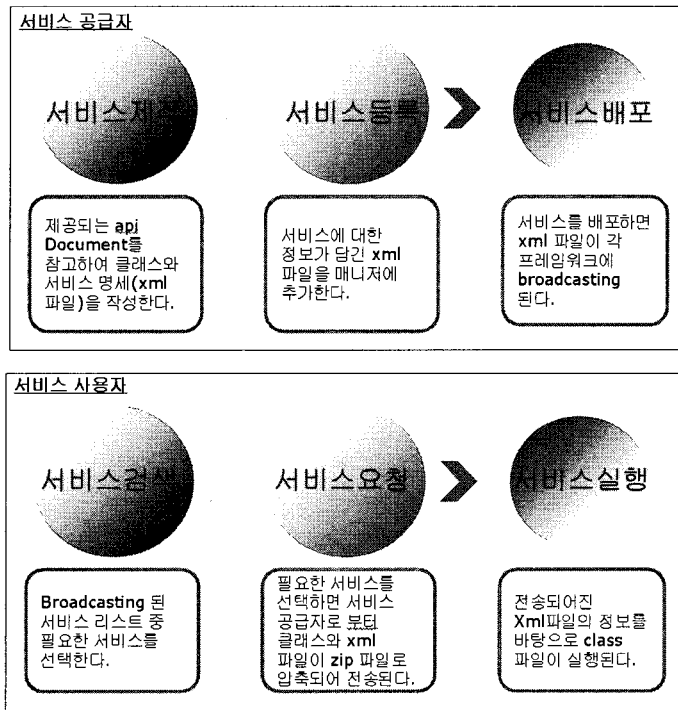


그림 7. 서비스 공급 과정
Fig 7. Service process

SOA 프레임워크 상에서 공급되는 서비스들의 표준화된 서비스 명세를 제공해주는, 웹 서비스의 WSDL의 역할을 하는 서비스는 명세는 (그림 8)의 XML형태로 정의된다.

제안된 프레임워크에서는 서비스 공급자가 서비스를 작성·공급 할 수 있는 API를 제공하여 누구나 쉽게 자신이 제작한 서비스를 P2P 네트워크상에 배포할 수 있다. 배포된 서비스는 서비스 사용자들에게 P2P망을 통해 빠르고 효율적으로 전달된다. 이렇게 배포된 서비스 리스트를 빠르게 확인할 수 있는 방법과 제공받은 서비스를 쉽게 실행시킬 수 있도록 편리한 UI 환경을 제공하여 사용자는 최소한의 노력으로 타인이 제공하는 서비스를 이용할 수 있다. 제안된 프레임워크의 이러한 특징으로 인해 서비스 공급자의 입장에서 서비스의 광고와 서비스의 공급이 매우 간편하고, 서비스 사용자의 입장에서 서비스를 제공받는 일이 간편하다.

4.4 어플리케이션 사용

제안된 프레임워크를 구현하기 위한 어플리케이션은 개인 사용자들의 쉬운 접근성을 확보하기 위하여 (그림 9)와 같이 일반적인 윈도우 어플리케이션과 유사한 UI가 제공된다. 어플리케이션의 각 Tab은 다음과 같은 기능을 한다. Manager Tab은 서비스의 구성을 보여주고 JXTA

Network를 시작시키는 기능이 있다. Agent Tab은 어플리케이션 사용자가 서비스 공급자로서의 역할을 수행하고 싶을 때 사용할 수 있는 기능을 가지는 Tab이다. Agent Tab에서는 현재 자신이 공급하고 있는 서비스의 리스트를 보여준다. 각각의 서비스들에 대해서 추가, 삭제, 그리고 서비스를 시작 할 수 있다. Service Tab은 어플리케이션 사용자가 서비스 사용자로서의 역할을 수행하고 싶을 때 사용할 수 있는 기능을 가지는 Tab이다. 따라서 Service Tab에서는 현재 네트워크상에서 공급되고 있는 서비스들의 리스트를 보여준다. 서비스는 사용자는 해당 리스트들 중에서 원하는 서비스를 선택하여 공급 받을 수 있다. Setting Tab에서는 서비스 공급자, 서비스 사용자의 다양한 환경세팅을 할 수 있다. Help Tab에서는 이용자들에게 도움말을 제공한다.

이 어플리케이션에서 제공 가능한 다양한 서비스 중에서 개인 사용자에게 특화될 수 있고 현재 많이 이용되고 있는 동영상 스트리밍 서비스는 다음과 같은 방식으로 제공된다.

서비스 공급자는 (그림 10)와 같이 스트리밍하고 싶은 동영상 파일을 선택하여 PLAY할 수 있다.

```

<agent>
  <name>Agent Name</name>
  <description>
    this agent is p2p file share.
  </description>
  <files>
    <file type="main">a.class</file>
    <file>b.class</file>
    <file>c.txt</file>
  </files>
  <environment>
    <version>1.0</version>
    <communications>
      <communication>tcp</communication>
    </communications>
    <plug-ins>
      <plug-in url = "http://abc.com/file.jar">file search</plugin>
    </plug-ins>
  </environment>
  <service>
    <author>service provider</author>
    <root>auto</root>
  </service>
</agent>

```

그림 8. 서비스 명세 XML
Fig 8. Service specification XML

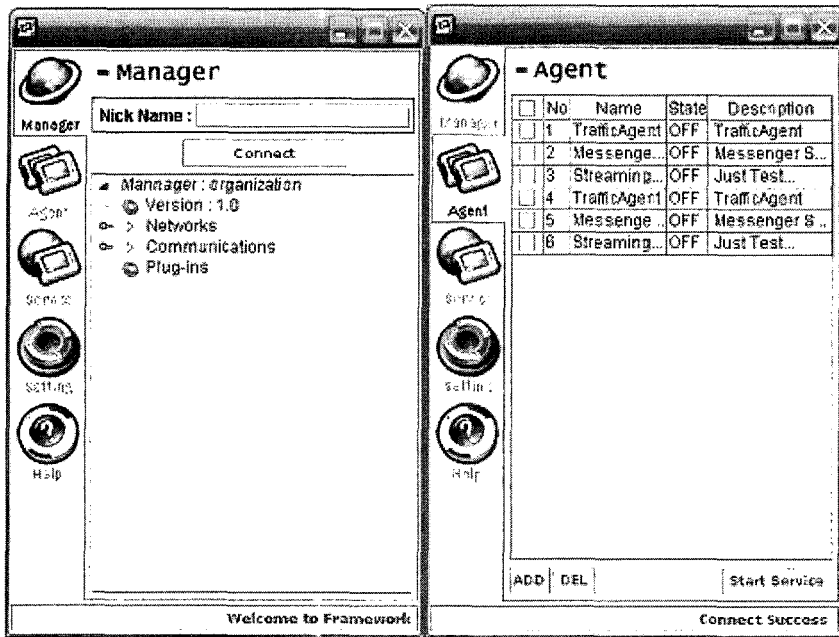


그림 9. 어플리케이션 UI
Fig 9. Application UI

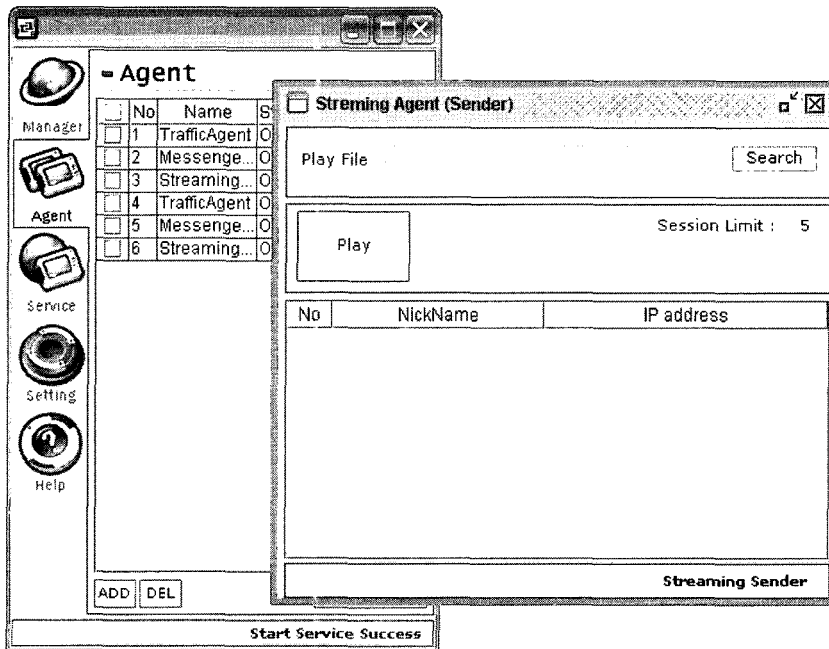


그림 10. 서비스 공급자: 동영상 스트리밍 서비스 UI
Fig 10. Service provider: Video streaming service UI

서비스 공급자가 공급하는 동영상 서비스는 P2P 네트워크를 통해 연결된 peer들에게 Broadcasting 된다. 동영상 서비스를 이용하고 싶은 서비스 사용자는 Service Tab의 여러 서비스 목록들 중에서 동영상 서비스를 선택하여 서비스를 공급받을 수 있다. 이때 공급되어지는 동영상 서비스는 앞서 언급한 대로 서비스 자체가 실행 가능한 완전한 형태로 공급되기 때문에 별도의 동영상 플레이어나 코덱등이 필요하지 않다. 따라서 서비스 사용자 입장에서는 서비스의 종류나 그 형태에 상관없이 쉽게 서비스를 이용할 수 있는 장점이 있다. 또한 <그림 11>과 같이 서비스가 실행 가능한 형태로 공급되기 때문에 서비스의 보안문제도 고려 할 수 있다.

서비스 공급자가 공급하는 동영상 서비스는 P2P 네트워크를 통해 연결된 peer들에게 Broadcasting 된다. 동영상 서비스를 이용하고 싶은 서비스 사용자는 Service Tab의 여러 서비스 목록들 중에서 동영상 서비스를 선택하여 서비스를 공급받을 수 있다. 이때 공급되어지는 동영상 서비스는 앞서 언급한 대로 서비스 자체가 실행 가능한 완전한 형태로 공급되기 때문에 별도의 동영상 플레이어나 코덱등이 필요하지 않다. 따라서 서비스 사용자 입장에서는 서비스의 종류나 그 형태에 상관없이 쉽게 서비스를 이용할 수 있는 장점이 있다. 또한 <그림 11>과 같이 서비스가 실행 가능한 형태로 공급되기 때문에 서비스의 보안문제도 고려 할 수 있다.

V. 결론

현재 각 벤더들은 경쟁적으로 SOA를 구축할 수 있는 어플리케이션과 미들웨어들을 쏟아내고 있다. SOA는 다양한 레거시 시스템들을 통합하고, 통합된 환경에서의 새로운 가치창출을 위한 도구로서 그 역할이 증대되고 있다.

본 논문에서는 새로운 비즈니스 모델을 위한 JXTA기반의 SOA 프레임워크를 제안했다. 이를 통해 종래의 기업 업무 환경에서 적용되었던 SOA의 범위에서 벗어나, 개인과 기업, 개인과 개인간의 업무에 까지 그 기능을 확장하였다. 공급되어지는 모든 서비스는 표준화된 XML형태로 그 명세가 제공되며, 모든 서비스는 서비스 데이터와 그것을 제어할 수 있는 어플리케이션이 동시에 배포된다. 이러한 특징은 제공되어지는 서비스의 보안 문제의 새로운 접근을 가능하게 한다. 또한 프레임워크는 JXTA기반으로 SOA를 구현하여서 서비스를 이용하는 각각의 peer들이 서비스 공급자, 서비스 사용자, 서비스 저장소의 역할을 동시에 수행할 수 있다. 이러한 P2P적인 특성으로 인해 네트워크 자원을 효과적으로 활용할 수 있다. P2P특성을 활용하여 서비스 목록을 Broadcasting함으로써 서비스 공급자 입자에서는 서비스의 광고와 서비스 공급을 용이하고, 서비스 사용자는 서비스의 목록 검색 노력이 최소화 된다. 향후 연구로써 온톨로지 기술을 활용하여 서비스 사용자가 서비스를 선택하는 부분을 서비스 사용자의 충분한 정보를 고려하여 선택할 수 있게 하면 진정한 의미의 맞춤형 서비스를 제공할 수 있을 것이다.

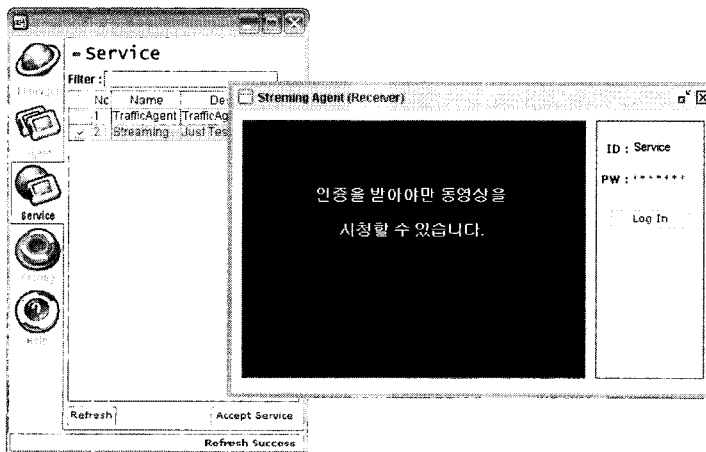


그림 11. 서비스 사용자: 동영상 스트리밍 서비스 UI
Fig 11. Service client: Video streaming service UI

참고문헌

- [1] Service Oriented Architecture,
<http://www-306.ibm.com/software/solutions/soa/>
- [2] Simon Hayward, "Positions 2005: Service-Oriented Architecture Adds Flexibility to Business Processes", Gartner research, ID number: G00126409
- [3] Project JXTA. <http://www.jxta.org/>.
- [4] Victor Guevara-Masis, Hamideh Afsarmanesh, and Louis O. Hertzberger "Service-Oriented Architecture of TeleCARE", OTM Workshops 2004, LNCS 3292, pp. 14 - 16, 2004.
- [5] JaeHoon Cho and SangWan Lee, "Implement SOA with "Patterns for e-business"", 2005 KMIS International Conference, 2005.
- [6] Pierpaolo Baglietto, Massimo Maresca, Andrea Parodi, Nicola Zingirian, "Stepwise deployment methodology of a service oriented architecture for business communities", Information and Software Technology 47, pp. 427 - 436, 2005.
- [7] Nathalie Furmento, Jeffrey Hau, William Lee, Steven Newhouse and John Darlington, "Implementations of a Service-Oriented Architecture on top of Jini, JXTA and OGSA", Lecture notes in computer science(LNCS) 3165, pp. 90 - 99, 2004.
- [8] JINI Network Technology,
<http://www.sun.com/software/jini>
- [9] Globus: OGSA - The Open Grid Services Architecture, <http://www.globus.org/ogsa/>
- [10] Aphrodite Tsalgatidou and Eleni Koutrouli "Interoperability and eServices", DAIS 2005, LNCS 3543, pp. 50 - 55, 2005.
- [11] Feng Yang, Shouyi Zhan, and Fuxiang Shen, "PSMI: A JXTA 2.0-Based Infrastructure for P2P Service Management Using Web Service Registries", GCC 2003, LNCS 3032, pp. 440 - 445, 2004.
- [12] Mark Colan, "Service Oriented Architecture and Web Services", IBM, Technical Presentation Material, 2004
- [13] Service-Oriented Architecture expands the vision of Web services,
<http://www-128.ibm.com/developerworks/web/services/library/ws-soaintro.html>
- [14] Thomas Erl, "Service-Oriented Architecture Concepts, Technology, and Design", Pearson Education, Inc, 2005.
- [15] Web Services Description Language (WSDL), <http://www.w3.org/TR/wsdl>
- [16] Universal Description Discovery and Integration, The UDDI Community,
<http://www.uddi.org>
- [17] Simple Object Access Protocol(SOAP), <http://www.w3.org/TR/soap/>
- [18] Li Gong "Project JXTA: A Technology Overview" Sun Microsystems, Inc. October 29, 2002.

저자소개



김 태 호

2006 고려대학교 산업시스템정보
공학 학사
현재 고려대학교 산업시스템정보공
학과 석사과정
관심분야 : 시스템통합, SOA



이 흥 철

1983 고려대학교 산업공학 학사
1988 Univ. of Texas 산업공학 석사
1993 Texas A&M Univ. 산업공학
박사
현재 고려대학교 정보경영공학부 교수
관심분야 : SCM, 생산 및 물류
정보시스템, PLM



천 현 재

1997 인천대학교 산업공학 학사
1999 고려대학교 산업공학 석사
2006 고려대학교 산업공학 박사
현재 고려대학교 BK21 유비쿼터스 정보보호 사업단 연구 전임 강사

관심분야 : SCM, Data Mining,
Simulation



김 준 룡

현재 중앙대학교 컴퓨터공학과 학사
과정
관심분야 : Semantic Web,
SOA, 시스템통합