

모바일 에이전트를 이용한 상품거래서비스에 관한 연구

안 성 옥* 오 기 옥**

A Study of Product Trade Service using Mobile Agent

Sung-Wook, An* Gi-Oug, Oh**

요 약

최근 인터넷의 증가와 함께 인터넷 비즈니스의 상거래 방식 또한 바뀌어 가고 있다. 기존 상거래 방식은 사용자 스스로가 원하는 정보를 찾아 다녔지만 점차 사용자의 편의를 위해서 하나의 사이트에서 사용자에게 맞춤 서비스를 할 수 있어야 한다. 이러한 서비스를 사용자에게 제공하기 위해 서비스 업체들은 모바일 에이전트를 이용한 상거래 시스템 구축이 필요하다. 모바일 에이전트 기반의 상거래 시스템은 사용자에게는 하나의 사이트에서 원하는 정보를 얻을 수 있고, 서비스 업체는 고객들에게 다양하면서도 유익한 정보를 제공할 수 있다. 이는 신뢰성 있는 전자상 거래 환경 구현을 위하여 매우 중요하다.

하지만 일반적으로 사용되는 모바일 에이전트 기반의 플랫폼은 사용자가 URL정보를 알아야 하고, 이러한 기능은 검색 엔진에 의존하고 있다. 이와 더불어 사용자의 요구 수준이 증가하면서 사용자는 단순 데이터를 제공받는 것이 아니라 통합, 요약된 서비스를 제공받기 원하고 있다.

본 논문에서는 사용자가 원하는 정보를 한 사이트 또는 한 페이지에서 제공하고, 적은 횟수의 마우스 클릭으로 사용자의 요구사항을 만족시킬 수 있도록 모바일 에이전트 개념을 적용한 상품 거래 시스템을 구현한다. 이를 위하여 E-Service를 하기 위한 플랫폼인 E-speak를 이용하여 상품 거래 시스템을 구현한다. 또한 콘텐츠는 XML 문서로 저장하고 이를 기반으로 자료를 공유한다.

Abstract

Recently, way of Electronic Commerce is changing with increment of Internet users. In the previous way of commerce, users used to search for information by themselves. However, for user's convience, it can be done for users from one site. Mobile agent-based Electronic Commerce system construction is needed for service companies in order to

* 숭실대학교 컴퓨터공학과 박사과정, HP 연구원

** 강원관광대학 컴퓨터정보세열 선임강사

provide users with such service. By the mobile agent-based Electronic Commerce system, users can get information from one site, and service companies can provide various and useful information to users. This is very important for construction of reliable Electronic Commerce environment.

However, in generally used mobile agent-based platforms, the user should know URL information, and it is based on search engines. And as user's requirement levels become higher, they want to be provided with integrated and summarized service.

In this paper, we implement product-trading system by applying mobile agent concept in order to satisfy user's requirement of getting required information from one site with minimal mouse click.

For implementing the product trading system, E-speak, which is a platform for E-Service, is used. And contents are saved as XML documents and data are shared based on these documents.

I. 서론

최근 인터넷 사용자의 증가와 인터넷 관련 기술의 급속한 발달로 전자상거래 서비스가 활성화되고 있다. 기존의 인터넷 서비스는 사용자 스스로 URL정보를 알아야 하며, 이를 검색엔진에 의존한다. 수백 또는 수천 개의 사이트 또는 페이지에 존재하는 상품 중에서 최적의 정보를 사용자 스스로 결정해야 하므로 사용자는 많은 횟수의 사이트 방문이 필요하며 시간의 소모가 커지게 된다. 앞으로 인터넷 서비스는 사용자의 요구 수준이 증가하면서 단순 데이터를 제공하는 서비스에서 통합, 요약된 서비스를 제공해야 한다. 또한 인터넷상의 신뢰성 있는 전자상거래 환경 구현을 개선해야 한다 [1], [6], [11]. 이를 위해서는 사용자를 대신해서 자율적으로 작업을 수행하는 모바일 에이전트가 필요하다. 모바일 에이전트는 현재의 컴퓨터에서 실행되지 않고 [9], 그 작업을 할 수 있는 다른 컴퓨터로 이동해서 실행되도록 지원하는 프로그램이다 [3], [10], [4], [16]. 모바일 에이전트는 에이전트 간의 정보의 교환과 자료의 공유를 통해 효율적인 인터넷 서비스를 제공할 수 있다. 분산처리와 기존의 네트워크 프로그램 등은 트래픽의 내용이 처리된 또는 처리할 데이터이지만 에이전트의 경우는 에이전트 코드가 이동하는 것이다. 통신망의 대역폭에 따라 작은 여러 개의 패킷을 보내는 것보다 훨씬 적은 양의 패킷이 이동하므로 네트워크 전체의 부하를 줄일 수 있고 자원의 공유가 보다 효율적일 수 있다 [4], [9], [14]. 모바일 에이전트가 사용자 요구에 따라 컴퓨터 사이를 이동하므로 사용자는 서비스를 제공받기 위해 지속적으로 연결 상태를 유지할 필요는 없다. 나중에 두 컴퓨터를 연결하여 처리된 자료의 결과를 가져오면 된다. 그러나 전체 네트워크를 구성하는 각각의 노드가 서로 이질적이고 서로 다른 개발 환경에서 구현된 에이전트가 실제로 이동하는 것은 상당히 어렵다.

그래서 본 논문에서는 클라이언트, 브로커, 정보 제공자를 하나의 가상 조직으로 형성하여 사용자가 하나의 사이트에서 원하는 모든 서비스를 받을 수 있는 환경을 제시한다. 모바일 에이전트 특성을 가능하게 해주는 여러

가지 플랫폼이 있지만 본 논문에서는 E-Speak 플랫폼을 이용하여 상품 거래 서비스 환경에 적용한다. E-speak는 인터넷 기반으로 개방형 인터넷 서비스 플랫폼으로 정보들을 연결하고 교환하며 자동적으로 서비스도 제공해줄 수 있다[7]. 이 플랫폼을 이용하여 상품 거래 서비스 환경을 제시하고 구현한다.

본논문의 구성은 다음과 같다. 2장에서는 모바일 에이전트를 기반으로 한 플랫폼에 관한 연구와 각 에이전트 플랫폼들 간에 비교 분석한다. 3장에서는 모바일 에이전트 플랫폼을 이용하여 전자상 거래 서비스 흐름을 보인다. 4장에서는 3장에서 전자상 거래 서비스 환경을 바탕으로 E-speak를 이용한 상품 상거래 서비스를 구현한다. 5장에서는 결론과 향후 연구에 대해 기술한다.

II. 관련연구

2.1 Voyager

Voyager는 오브젝트스페이스사에서 개발한 ORB(Object Request Broker)를 이용하며, 자바기반의 동적 하부구조 플랫폼이다 [12], [13]. 하부구조 플랫폼은 엔터프라이즈 코드를 자바 이외의 언어나 어떠한 네트워킹 프로토콜로도 접근 가능하게 만들도록 하고, 동적으로 사용될 수 있는 분산 객체들로 파일들을 변형시킴으로써 네트워크가 고정 데이터 구조와 코드로 파일을 삽입할 수 있다. 그림 1은 Voyager 애플리케이션 아키텍처를 보여준다.

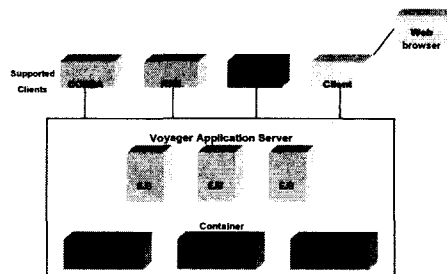


그림 1. Voyager 애플리케이션 아키텍처

2.2 JATLite

JATLite는 스탠포드 대학에서 제작한 에이전트 기반 구조 개발을 위한 자바 패키지의 집합이다 [8], [15]. JATLite는 자바 언어를 사용하고 지식의 전달로는 KQML을 이용한 메시지 교환방식의 통신 도구를 제공한다 [8], [15]. 또한 기반 구조상에서 에이전트 간의 협력 작업을 조정하기 위한 에이전트 라우터를 제공한다.

JATLite 아키텍처는 추상 레이어, 베이스 레이어, KQM레이어, 라우터 레이어, 프로토콜 레이어의 5계층으로 구성된다. 추상 레이어는 상위의 클래스들이 정의될 수 있도록 하는 루트 클래스들을 제공하고, 베이스 레이어는 TCP/IP를 이용한 통신 레이어이며, 에이전트가 메시지 프로토콜을 정의하도록 지원한다. KQML 레이어는 KQML 메시지들의 파싱과 에이전트가 KQML 통신을 하도록 지원하고, 라우터 레이어는 메시지 라우팅을 제공한다. 프로토콜 레이어는 SMTP와 FTP와 같은 표준 인터넷 서비스를 지원한다. JATLite 아키텍처는 그림 2에서 나타난다.

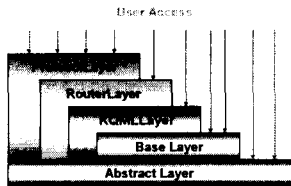


그림 2. JATLite 아키텍처

2.3 E-speak

E-speak는 휴렛 팩커드사에서 개발한 인터넷 기반의 서비스를 생성, 조합, 중재, 관리하며 이를 접근하기 위한 개방형 인터넷 서비스 플랫폼이다 [7]. 코어는 액세스 인정 여부를 결정하고 자원 처리를 가동 시키기 위해서 자원 명세서를 사용한다. E-speak는 조정 아키텍처, 자원모델, 메타데이터 시스템, 명칭(naming)부여 모델, 보안모델, 통신 등으로 구성되어 있다. E-speak는 ESIP(E-Speak Interface Protocol)를 통하여 정보들을 연결하고 교환하며 자동적인 서비스를 제공할 수 있게 된다. 그림 3은 E-speak의 코어와 레파지토리의 아키텍처를 보여주고 있다.

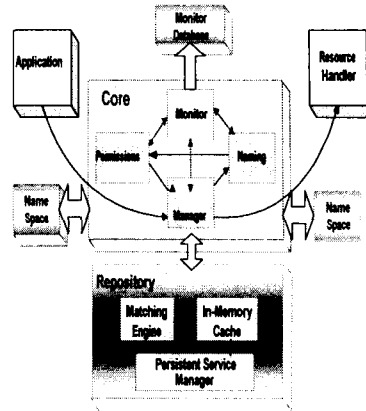


그림 3. E-speak 코어와 레파지토리 아키텍처

III. 모바일 에이전트를 이용한 전자상거래서비스

3.1 전자상거래서비스구조

전자상거래 서비스를 위한 구성 요소로는 클라이언트, 브로커, 서비스 제공자로 구성된다.

○ 클라이언트(Client)

클라이언트는 전자상 거래를 하기 위해 서비스를 제공 받는 사용자를 말한다.

○ 브로커(Broker)

브로커는 여러 서비스 제공자로부터 자료를 수집하여 사용자에게 가장 적합한 정보를 제공해 주는 역할을 하며, 서비스는 클라이언트와 직접 연결되어 이루어진다.

○ 서비스 제공자(Service Provider)

서비스 제공자는 클라이언트에게는 등록을 기반으로 하고 브로커를 통해 다른 서비스 제공자들과 함께 자료를 공유하고 브로커에게 서비스를 제공해 주는 역할을 한다. 콘텐츠를 제공하는 업체들이 여기에 해당된다.

전자상거래를 하기 위한 구성 요소인 클라이언트, 브로커, 서비스 제공자의 관계는 클라이언트는 원하는 정보를 브로커를 통해서 제공받으며 그 전에 브로커는 서비스

제공자와 연결되어 있어야 한다.

본 논문에서는 모바일 에이전트를 기반으로 한 E-service 를 위한 플랫폼인 E-speak를 이용하였다.

3.2 클라이언트와 서비스 제공자간의 관계

클라이언트는 서비스 제공자에 등록된 후 로그인 시 서비스 제공자에서 제공하는 콘텐츠를 이용할 수 있다. 그림 4는 브로커와 서비스 제공자사이에서 사용자가 상품을 구매하는 순차도이다 [2], [5]. 구매자(Buyer)가 먼저 서비스 제공자에 등록을 한 다음 로그인 하여 서비스 제공자에 상품을 등록한다. 그 다음 상품 구매를 하고자 할 때 브로커를 통해 상품 목록에서 상품 구매 주문을 하고 브로커는 구매자가 원하는 정보를 서비스하기 위해 사용자와 타협을 얻어낸다.

3.3 클라이언트와 브로커간의 관계

클라이언트는 모든 서비스를 브로커를 통해 이루어진다. 그림 5는 브로커와 서비스 제공자 사이에서 판매자(Seller)가 상품을 판매하는 순차도이다 [2], [5]. 판매자(Seller)가 먼저 서비스 제공자에 등록을 한 다음 로그인 하여 서비스 제공자에 상품을 등록한다. 그 다음 상품 구매를 하고자 할 때 브로커를 통해 상품 목록에서 상품 판매를 하고 브로커는 구매자가 원하는 정보를 서비스하기 위해 사용자와 동의를 얻어낸다. 동의가 되면 브로커는 서비스 제공자와 연결을 해제하고 사용자에게 매매 상태가 가능함을 알린다.

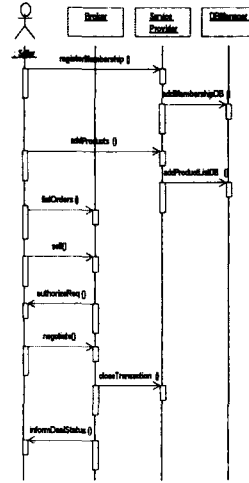


그림 5. 상품판매순차도

3.4 브로커와 서비스제공자간의 관계

브로커는 서비스 제공자와 연결되어야 클라이언트가 브로커를 통해서 정보를 제공받을 수 있다. 이는 E-speak의 코어에 서비스 제공자가 연결되어서 연결을 할 수 있다. 또한 브로커 간의 통신은 E-speak에서 제공하는 ESIP(E-Speak Interface Protocol)에 의해 자동으로 연결된다

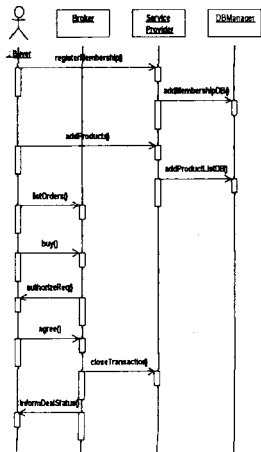


그림 4. 상품구매순차도

IV 상품거래시스템 구현

모바일 에이전트를 기반으로 한 E-speak를 이용하여 3장에서 제시한 아키텍처를 상품 거래 시스템에 적용한다. 시스템 구현에서는 클라이언트/서버 환경만을 고려한다.

4.1 시스템의 기능적 요구사항

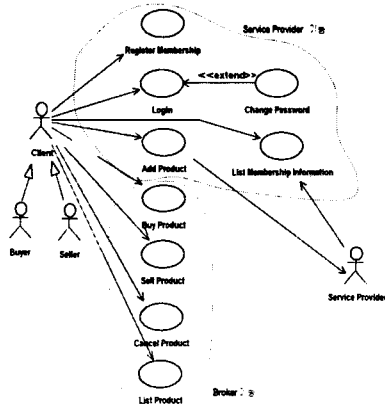


그림 6. 상품 거래 시스템의 Use Case 다이어그램

기능은 크게 2가지로 서비스 제공자 기능과 브로커 기능으로 나눌 수 있으며 그림 6은 상품 매매 시스템의 Use Case 다이어그램이다 [2], [5]. 서비스 제공자의 기능은 클라이언트의 등록, 로그인, 패스워드 변경, 상품 등록, 상품정보 목록이다. 각각의 기능 설명은 다음과 같다.

○ 회원 등록(Register Membership)

서비스를 이용하는 사용자는 먼저 서비스 제공자에 등록을 한다.

○ 로그인(Login)

서비스 제공자에 등록된 사용자는 로그인 정보를 입력하여 서비스 제공자와 브로커의 서비스를 이용하게 된다.

○ 상품 등록(Add Product)

사용자가 서비스 제공자에게 자신의 정보를 등록한다. 이것은 상호간의 서비스를 이용하는데 유용한 정보가 될 수 있다.

○ 등록된 상품 목록(List Membership)

이 기능은 서비스 제공자에 등록된 정보의 목록을 보여준다.

브로커의 기능은 서비스 제공자에 등록된 상품 목록, 사용자의 구매 정보, 판매 정보, 주문 취소가 있다. 각각의 기능 설명은 다음과 같다.

○ 상품 목록(List Product)

이 기능은 사용자가 상품을 구매하거나 판매할 수 있도록 상품의 자세한 목록을 보여준다.

○ 구매 정보(Buy Product)

사용자가 상품 정보를 보고 브로커에 상품을 구매할

수 있는 기능이다.

○ 판매 정보(Sell Product)

사용자가 상품 정보를 보고 브로커에 상품을 판매할 수 있는 기능이다.

○ 주문 취소(Cancel Order)

사용자가 브로커에게 상품을 구매하거나 판매한 주문을 취소하는 기능이다.

4.2 시스템의 분석과 설계

구매자(Buyer)가 상품을 등록하게 되면 상품 정보는 OrderDetail 객체를 통해 ESArray 객체의 RecordTable에 등록되고 이 정보는 MatchThread 객체에 의해 BrokerLogic 객체로 전달되고 구매자에게 주문 번호를 반환한다. 등록된 정보는 다른 브로커에서 가져온 정보와 비교해서 구매자가 원하는 정보에 대한 메시지를 보내준다. 그림 7은 그림 6의 Buy Product Use Case에 대한 상품 거래 시스템의 순차도이다 [2], [5].

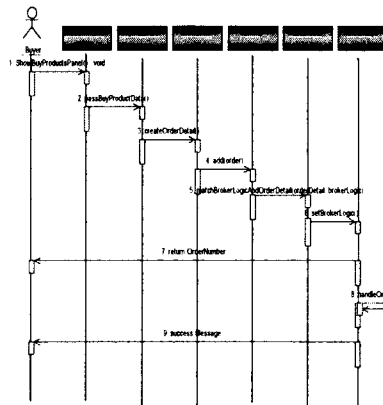


그림 7. 상품거래시스템의 "Buy Product"의 순차도

그림 15는 상품 거래 시스템의 사용자 인터페이스 중심에 대한 클래스 다이어그램을 보여주고 있다 [2], [5].

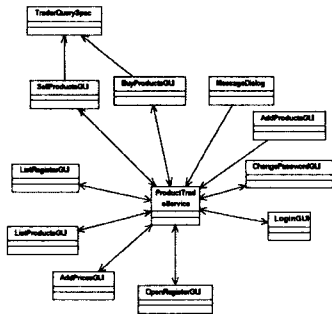


그림 8. 상품거래시스템의 UI중심의 클래스 다이어그램

그림 9는 상품 거래 시스템의 로직 부분에 대한 클래스 다이어그램이다 [2], [5].

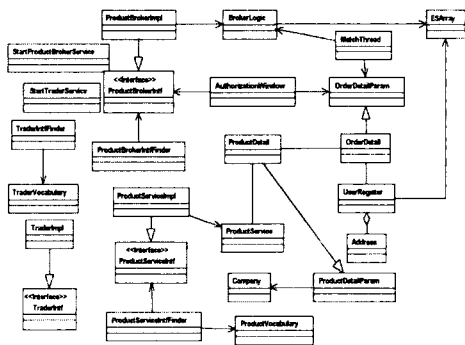


그림 9. 상품거래시스템의로직중심의클래스 다이어그램

4.3 클라이언트/서버기반의 상품거래 시스템의 구현

앞에서 제시한 모바일 에이전트를 이용한 전자상거래 서비스를 근간으로 상품매매 서비스를 구현한다. 시스템의 구현환경으로 운영체제는 Windows 2000, JDK 1.2 모바일 에이전트 플랫폼으로는 E-Speak 3..03이며 시스템의 구성은 클라이언트, 정보제공자, 브로커, 정보저장소의 4부분으로 구분되어지며, 각 부분의 수행절차는 다음과 같다.

○ 클라이언트(Client)

클라이언트 측의 사용자는 포탈 사이트에 등록하거나 서비스 제공자에 등록한다.

○ 서비스 제공자(Service Provider)

서비스 제공자는 그 정보를 클라이언트가 사용할 수 있도록 이름(Name)을 주어야 하며 쉽게 찾을 수 있도록 어트리뷰트로 구분하여 그 정보를 잘 기술하여 쿼어에 연결하여 레파지토리에 등록한다. 정해진 클라이언트 외에 타 클라이언트에게도 서비스를 전달하기 위해 통지(advertise)하고 각 클라이언트에게 동일한 형태의 인터페이스를 제공하여 공동체를 형성한다.

○ 브로커(Broker)

브로커는 상품을 구매하거나 판매하려는 주문 목록을 가지고 있다. 사용자가 원하는 서비스를 서비스 제공자에서 찾아서 사용자가 요청한 기준에 벗어나면 제거함으로써 사용자와 서비스 제공자 사이에 중재하여 사용자의 요구에 가장 잘 맞는 서비스를 연결해 준다.

○ 정보 저장소(Repository)

정보 저장소의 역할은 데이터를 XML문서로 저장하고 추출할 수 있도록 한다. 본 시스템에서 모든 정보는 XML문서로 저장된다. 저장된 XML문서는 재사용과 정보 제공자 간의 데이터를 공유할 수 있으며 액세스하는데도 빠르게 찾을 수 있다. 데이터 베이스에 저장된 정보는 다른 사용자들에게 검색이나 기타 기능을 제공하기 위한 데이터로 사용될 수 있다.

4.4 상품거래시스템의 핵심소스

본 장에서는 상품 매매 시스템을 구현하는데 있어서 핵심이 되는 소스로 서비스 제공자가 E-speak 코어에 연결하고 등록하는 부분을 제시한다. 그림 10은 상품 서비스를 찾기위한 소스를 보여준다. 그림 10에서 ESConnection은 서비스 제공자가 서비스를 제공하기 위해서 먼저 E-speak 코어에 연결하는 부분이다. ESVocabularyFinder는 코어에 연결되어 있는 서비스 이름을 찾는 부분이다. ESQuery는 XML문서로 작성되어 있는 데이터를Name=ProductVocabulary 쿼리로 find()를 통해 상품 서비스를 찾아낸다. 그림 11은 vocabulary 서비스를 하는 XML문서를 보여준다.

```

public ProductVocabulary(ESConnection session, String
vocabXMLFile,
String contractName) {
    ESAccessor accessor = null;
    try {
        // FIND THE Product VOCABULARY IF ONE EXISTS
        ESVocabulary productVocab = null;
        ESVocabularyFinder vf =
            new ESVocabularyFinder(session);
        try {
            productVocab = vf.find(new ESQuery("Name =
'ProductVocabulary'"));
        } catch (LookupFailedException e) {
    
```

그림 10. 상품서비스 찾기소스

```

<?xml version="1.0">
<ESpeak version="E-Speak 1.0beta"
operation="RegisterService" xmlns="http://localhost/e:/es
xml/Schemas/espeak.xsd">
    <resource>
    <resourceDesx mins="name="ProductDescription">
        ...
    <Name xmlns="">ProductVocabulary</Name>
    <Type xmlns="">ESVocabulary</Type>
    <attr xmlns="" name="ProductName" required="true">
    <value xmlns="">Clock Product</value>
    </attr>
    ...
</ESpeak>
    
```

그림 11. Vocabulary 서비스를 위한 XML 문서

그림 12는 서비스 컨트랙트(=contract) 즉, 서비스 인터페이스를 등록하는 절차이다. 먼저 코어에 연결을 하고 ESContractDescription 클래스를 통해서 서비스 인터페이스를 정의하도록 제공한다.

"ProductContract"가 이 생성자에 단 하나의 어트리뷰트 ESConstants.SERVICE_NAME으로 구성되는 서비스 기술을 생성하는데 사용되는 하나의 아큐먼트로 전달된다. 그 상수의 값은 "ProductContract"로set된다. ESContractElement의 register()는 서비스 컨트랙트(인터페이스)를 레퍼지토리에 등록하기위해 실행된다.

```

public ProductVocabulary(ESConnection session, String
vocabXMLFile, String contractName)
// CREATE CONTRACT
ESContractDescription escd = new ESContractDescription();
escd.setInterfaceName(contractName);
escd.addAttribute(ESConstants.SERVICE_NAME,
"ProductContract");
escd.setInterfaceDefinition("ProductServiceIntf.esidl");
ESContractElement escc = new ESContractElement(session,
escd);
ESContract cont = escc.register();
System.out.println("ProductContract registered successfully");
    
```

그림 12. 서비스에 대한소스

그림 13은 찾아낸 상품 서비스를 코어에 등록하는 절차이다. 일단 ESConnection으로 코어에 연결한 상태에서 ESVocabularyDescription로 서비스인 vocabulary들을 정의하도록 제공한다. ESVocabularyElement 의 register() 메소드는 서비스 엘리먼트를 레퍼지토리에 등록하기 위해 실행된다. 이렇게 함으로써 다른 클라이언트들이 서비스를 찾을 수 있게 된다.

```

public ProductVocabulary(ESConnection session, String
vocabXMLFile, String contractName) {
    ...
    System.out.println("Creating ProductVocabulary and Contract");
    ESVocabularyDescription esvd =
        new ESVocabularyDescription(new
        ESXMLFile(vocabXMLFile), session);
    ESVocabularyElement esvc =
        new ESVocabularyElement(session, esvd);
    vocab = esvc.register();
    System.out.println("ProductVocabulary registered successfully");
    
```

그림 13. 상품서비스 생성소스

4.5 상품매매 사용자 인터페이스

그림 14는 클라이언트가 서비스 제공자와의 연결을 위해 회원 등록을 하는 화면이다. 이 데이터는 정보 제공자가 서비스를 제공하는데 유용한 자료가 될 수 있으며, 회원 기반으로 한 콘텐츠 중심 사업에 중요한 자료가 사용될 수 있다. 그림 15는 사용자가 정보 제공자에게 자신이 가지고 있는 상품을 등록하고 등록된 상품 목록을 사용

자에게 보여주는 화면이다. 이 자료는 정보 제공자에 의해 다른 사용자에게도 제공될 수 있는 자료로 사용될 수 있다.

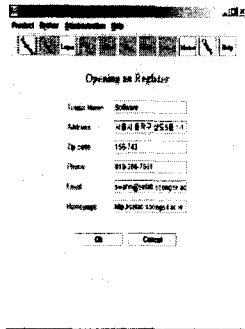


그림 14. 등록서비스화면

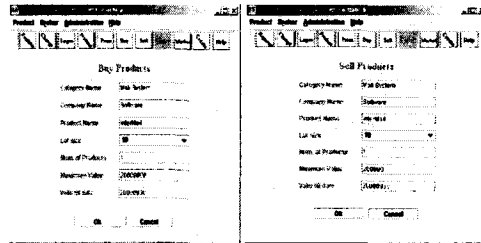


그림 17 상품구매와 판매화면

V. 결론

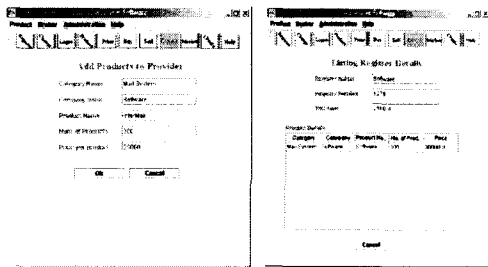


그림 15 상품등록과 등록된 상품목록 화면

그림 16은 사용자가 브로커를 통해 서비스를 할 수 있는 화면이다. 그림 17에서 사용자가 회사와 목록에서 원하는 회사의 목록을 선택하면 사용자에게 자세한 정보를 제공하는 화면이다. 이 화면의 정보를 통해 사용자는 자신이 원하는 상품 거래 업무를 하게 된다. 이 화면이 모바일 에이전트를 이용하여 전자상 거래 서비스를 할 때 중요한 역할을 하게 된다.

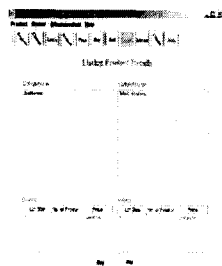


그림 16. 브로커화면

모바일 에이전트는 네트워크상에서 사용자의 유도없이 자유로이 돌아다닐 수 있다는 장점으로 다양한 하드웨어와 운영체제, 그리고 다양한 언어들로 프로그래밍된 프로그램들과 연결되어 있다. 그렇기 때문에 모바일 에이전트는 어떠한 플랫폼에 도달하더라도 문제없이 수행되어야 한다. 모바일 에이전트가 플랫폼과 언어에 독립적으로 사용되기가 불가능하다고 표현해도 과언이 아니다. 그래서 본 논문에서는 동일 플랫폼으로 E-speak를 이용하였다.

본 논문에서는 사용자가 정보를 찾아 다니는 것이 아니라 사용자가 원하는 맞춤 정보를 제공하고자 하기 위해서 E-speak플랫폼을 이용하여 모바일 에이전트 개념을 적용한 상품 거래 서비스를 구현하였다. 상품 거래 서비스의 구성요소로는 클라이언트, 정보 제공자, 브로커의 3가지 역할과 기능을 제시하였다. 모바일 에이전트를 이용하여 상품 거래 서비스를 할 때의 장점은 다음과 같다.

- 사용자가 원하는 서비스를 하나의 사이트에서 제공한다.
- 사용자는 재 로그인 불필요하다.
- 서비스 제공자간에 콘텐츠를 공유한다.
- E-business 등 다양한 분야에 적용 가능한 아키텍처이다.
- Batch 스타일의 인터넷 활용이 가능하다.

향후 연구는 전자상거래 서비스를 웹 기반으로 제공하며 더 나아가 휴대전화기, PDA, 메신저, 페이지 등에 서비스를 지원한다. 수많은 서버 중에서 자신이 원하는 서비스를 제공하는 서버를 어떻게 효율적으로 찾을 수 있

는가에 대한 브로커 기능의 연구가 필요하다. 서비스 제공자의 DTD기반의 XML 문서화로 업종별 카탈로그 시스템을 구축하여야 한다. 이는 ECS(Electronic Category System)에 대한 구축이 필요하다. 마지막으로 E-Business로의 확장을 위한 추가 연구가 필요하다.

참고문헌

- [1] J.P. Baron, M.J. Shaw, and A.D. Bailey, Jr "Web-based E-catalog Systems in B2B Procurement," Communications of the ACM, Vol. 43, No. 5, May 2000.
- [2] Booch, Jacobson, Rumbaugh, The Unified Modeling Language User Guide, Addison Wesley, 1999.
- [3] A. Caglayan, C. Harrison, Agent Sourcebook, John Wiley & Sons, 1999.
- [4] P. Dasgupta, N. Narasimhan, L.E. Moser, and P.M. Melliar-Smith "MAGNET: Mobile Agents for Network Electronic Trading," IEEE Trans. On Knowledge and Data Eng., Vol. 11, No. 4, JULY/AUGUST 1999.
- [5] M. Fowler, UML Distilled, Addison Wesley, 1997.
- [6] R.J. Glushko, J.M. Tenenbaum, and B. Meltzer "An XML Framework for Agent-based E-commerce," Communications of the ACM, Vol. 42, No. 3, March 1999.
- [7] Hewlett-Packard Co., E-speak, at URL: <http://www.e-speak.net>, 2000.
- [8] H. Jeon, C. Petrie, and M.R. Cutkosky, "JATLite: A Java Agent Infrastructure with Message Routing," IEEE Internet Computing, Vol. 4, No. 2, March/April 2000.
- [9] D.B. Lange, and M. Oshima, "Programming and Deploying Java Mobile Agents with Aglets," Addison-Wesley Longman, Reading, Mass., 1998.
- [10] D.B. Lange and M. Oshima "Seven Good Reasons for Mobile Agents," Communications of the ACM, Vol. 42, No. 3, March 1999.
- [11] P. Maes, R.H. Guttman, and A.G. Moukas "Agents That Buy and Sell," Communications of the ACM, Vol. 42, No. 3, March 1999.
- [12] B. McCarty and L. Cassady-Dorion "Java Distributed Objects", SAMS, 1999, pp. 787-826.
- [13] J. Nelson, "Programming Mobile Agents with JAVA", WILEY, 1999, pp. 139-188.
- [14] C. Petrie, "Agent-based Engineering, the Web, and Intelligence," IEEE Expert, Vol. 11, No. 6, Dec. 1996, pp. 24-29.
- [15] Stanford University, JATLite, at URL: <http://java.stanford.edu>, 1998.
- [16] D. Wong, N. Paciorek, and D. Moore "Java-based Mobile Agents," Communications of ACM, Vol. 42, No. 3, March 1999.9.

저자소개



안성욱
 2001년 숭실대학교일반대학원
 컴퓨터공학과 (공학석사)
 2001년 숭실대학교일반대학원
 컴퓨터공학과 박사과정
 현재 HP 연구원



오기욱
 1991년 경원대학교 전자계산학과(공학사)
 1993년 숭실대학교 일반대학원 전자계산학과(공학석사)
 2000년 숭실대학교 일반대학원 컴퓨터공학과 박사과정
 현재 강원관광대학 컴퓨터정보계열 전임강사
 강원관광대학 전산소장