전자상거래 시스템 구축을 위한 자바 애גל릿 기반 설계

Java Aglet-based Design for Electronic Commerce System

요 약

전자상거래는 글로벌 생산과 분산 공급 체인을 통합하고 최대화하도록 기회를 제공한다. 고객 선호도와 요구 변화에 따라 신속한 대처 능력과 새로운 기술의 활용 능력은 가장 중요한 요소이다. 다양한 회사의 컴퓨티는 상호 통신하면서 컴퓨턴트의 가격과 유용성을 결정하고, 주문과 확인을 수행하고, 배송 시간을 합성한다. 본 논문에서는 이동 에이전트인 자바 애글릿을 기반으로 상거래 시스템 구축을 기술한다. 구매자를 대신한 애글릿들은 판매자를 대신한 애글릿들은 상품 거래소 시장으로 보내낸다. 시장에서 자율적으로 주문과 배송을 합성하고, 최종 매매 결과와 실패코드를 구매자 및 판매자에게 보낸다. 우리는 이러한 상호 작용을 시뮬레이션으로써 전자상거래 시스템이 자바 애글릿 기술을 이용하여 적절히 구축될 수 있음을 보여준다.

Abstract

Electronic commerce offers the opportunity to integrate and optimize the global production and distribution supply chain. Rapid response to changes in demand and customer preference, and the ability to exploit new technologies, are becoming critical. The computers of the various corporations communicate with each other to determine the price and the availability of the components, to place and confirm orders, and to negotiate delivery time scales. In this paper, we describe a trading system that is based on mobile agent technology, called aglet. Aglets for the buyer and sellers are dispatched to the various marketplace, where they negotiate autonomously orders and deliveries, returning to the buyer and seller with their best deals for approval. We show that the electronic commerce system is feasibly built by using the Java aglet technology to demonstrate for simulating a traditional retail marketplace.

I. 서론

인터넷과 World Wide Web(WWW)은 전자상거래를 촉진시키고 있다. 상품을 보여주고 주문 받는 웹 기술은 비즈니스의 공통적인 양식이 되고 있다. 전자상거래는 컴퓨티에서 네트워크를 통하여 정보, 제품, 서비스 등을 요청 받고 시행하는 형태로 정의할 수 있고, 정보통신기술을 활용한 모든 종류의 거래 행위들을 포함한다. 전자상거래는 글로벌 생산과 분산 공급 체인을 통합하고 최대화하도록 기회를 제공한다. 고객 선호도와 요구 변화에 따라 신속한 대처 능력과 새로운 기술의 활용 능력은 가장 중요한 요소이다. 다양한 회사의 컴퓨티는 상호 통신하면서 컴퓨턴트의 가격과 유용성을 결정하고, 주문과 확인을 수행하고, 배송 시간을 합성한다.

웹 기술을 이용하여 인터넷 세상에서 사용자가 원하는 상품을 효율적으로 검색하여 구매 또는 판매하는 것은 일관화된 현실로 받아들여지고 있다. 상품 검색을 용이하게 할 수 있는
도록 식별 비교 사이트가 출현하는 데, 개인들의 식별에 따라 증가하는 경제적, 또는 사이트들의 서비스 수도 급속히 늘어나고 있다(3.4).

이동 에이전트기술은 실행 코드가 목표 달성을 위해 호스트를 이동하여 작업을 수행한다. 이동 에이전트는 네트워크 연결을 계속 유지한 상태로 작업하지 않고 실행 코드 자체가 시스템간을 이동하며, 임무를 수행하기 때문에 네트워크 연결이 불안정하거나, 호스트가 많이 걸리는 환경에서 유용하게 활용될 수 있다(4.5.6.7).

본 논문에서는 이동 에이전트의 자바 에글릿[8]을 기반으로 상자 시스템 구축을 기술한다. 구체적으로 자바 에글릿을 생성한 에글릿들은 상품 거래소인 시장으로 보낸다. 시장에서 자동적으로 가격과 매매를 합산하고, 최종 매매 결과와 실행코드를 구매자 및 판매자에게 전달한다. 우리는 전통적인 소매 시장에 오픈레이어 현으로서 전자상거래 시스템이 자바 에글릿 기술을 이용하여 적절히 구축될 수 있음을 보여준다.

본 논문의 구성은 다음과 같다. 제3장의 전자상거래 시스템의 이동 에이전트 활용에서 자바 기반 이동 에이전트 시스템을 비교 분석하고 전자상거래 시스템에 적용된 대표적인 예를 고찰하고 제4장에서는 자바 에글릿 이동 에이전트 시스템 개발이 기술한다. 제5장에서 이론 기반으로 전자상거래 시스템 구축을 이동 에이전트 기반으로 설계하고 간단히 시험 구현하였다. 마지막으로 시험 구현한 전자상거래 시스템을 고찰하고 향후 연구 과제에 대하여 기술한다.

II. 전자상거래 시스템의 이동 에이전트 활용


이동 에이전트 시스템들은 이동 기법에 따라 점프(jump) 모델과 진입점(know entry-point) 모델 등을 사용하고 있다. 첫째, 점프 모델은 시스템이 점프하는 포트폴리오 연산을 제공한다. 이 연산은 자동적으로 에이전트의 코드, 객체 상태, 및 제어 상태를 포함하여 다른 시스템으로 보내고, 도착 시스템에서는 상태를 복원하여 접근할 지점에서부터 실행을 계속한다. 이러한 시스템은 D'Agent[5], Arq[9], Tacoma[10] 등이 있다. 둘째, 진입점 모델은 에이전트의 코드와 객체 상태만을 포함하여 다른 시스템으로 보내고, 도착 시스템에서는 진입점(특정 메소드)에서 실행을 시작한다. 이러한 시스템으로 Aglets[8], Concorde[11], Voyager [12], Odyssey[13] 등이 있다. 자바 기반 이동 에이전트 시스템은 자바가상머신에 아무 변경 없이 실행시키는 진입점 모델을 사용하고 있다. 점프 모델이 에이전트 프로그래밍에 흔히 편리하지만, 제어 상태를 포함하는 플랫폼이 기존 인터프리터에 추가되어야 한다는 단점이 있다.

Table 1. 자바 기반 시스템의 특징별 비교

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote creation of agents</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Remote Java message</td>
<td>None</td>
<td>State-of-the-art</td>
<td>None</td>
</tr>
<tr>
<td>Messaging to mobile agents</td>
<td>None</td>
<td>Transparent</td>
<td>None</td>
</tr>
<tr>
<td>Messaging mode</td>
<td>Sync, Future</td>
<td>Oneway, Future, sync</td>
<td>None</td>
</tr>
<tr>
<td>Life spans</td>
<td>Explicit</td>
<td>5 different modes</td>
<td>Explicit</td>
</tr>
<tr>
<td>Mobile naming service</td>
<td>None</td>
<td>Integrated</td>
<td>None</td>
</tr>
<tr>
<td>Move to program</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Move to object</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Itineraries</td>
<td>Yes, special API</td>
<td>Yes, no special API</td>
<td>Yes, special API</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Restricted</td>
<td>Full</td>
<td>Restricted</td>
</tr>
<tr>
<td>Security</td>
<td>Security manager</td>
<td>Security manager</td>
<td>None</td>
</tr>
<tr>
<td>Partial connection</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Voyager는 원격의 이동 에이전트 클러스터를 생성할 수 있는 데 비하여 에플리나 Odyssey는 원격 생성이 불가능하다. Voyager에서 일단 클러스가 생성되면 네트워크의 어디에서나 쉽게 객체를 생성할 수 있고, 그 객체에 자바 메시지를 보낼 수 있다. Voyager는 synchronous, oneway, 및 future message 모드를 제공한다. 객체가 이동한 새로운 위치로 이동된 객체에게 메시지를 전송하기 위하여 이동하기 전의 위치인 secretary를 넘겨주고 이동한다. 이에 비해 애플리에는 특정 URL에 존재하는 고정 에이전트에게 스트링 명령어를 보낼 수 있지만 그 명령어를 수행하는 책임은 에이전트에게 달려있다[14].

전자상거래 시스템은 간단한 온라인 쇼핑 시스템으로부터 종합적이고 복합적인 시스템까지 다양하다. 이 시스템은 고객 구매 행태 모델(15)의 6단계 구분에 따라 구분될 수 있고, 대표적인 전자상거래 시스템은 다음과 같다.


MAGMA(Minnesota Agent Marketplace Architecture)[18]는 인터넷에서 전송될 수 있는 아이템들을 대상으로 거래하는 가상 시장 프로토콜이다. 이들은 Java 기반의 에이전트 (구매 에이전트와 판매 에이전트) 분류된 광고 서비스 시장을 제공하는 광고 서비스, 및 재정과 지불을 담당하는 은행으로 구성된 다. 각 에이전트는 유일한 ID를 지정받고, 에이전트간 메시지 경로 설정을 위해 중계 서버에 도착한다.

MagNET[9]는 구매자, 구매 이동 에이전트(자바 애플리) 및 공급자로 구성된 전자상거래 시스템을 구축하고 있다. 기본 목적은 구매자와 판매자가 상호 관심이 있는 품목을 직접 거래하도록 한다. 구매 애플리에는 판매자의 공급자들을 점검 하는 pull 모델을 근거하고 있다. 구매자는 상품에 따라 제계적인 공급자 리스트를 갖고, 공급자에게 가치 최고 결과를 거리는 후 결과를 구매자에게 보고한다.

III. 자바 애플리 이동 에이전트 시스템


이 패턴에 애플리는 하드웨어/소프트웨어 간 객체 상태를 저장하는 자속상 기능, 즉 투명성을 제공하기 위한 객체 간의 데이터가 proxy 기능, 이동 에이전트를 진행하는 흐름 기능, 이동 에이전트를 통신을 위한 메시지 전달 기능 등을 포함하고 있다. 애플러의 보안 모델은 D’Agenet[5], ARA[9] 및 다른 자바 기반 시스템 등에 유사하고, 이동 에이전트가 특정 미션의 제어 문맥의 내에서도 구현할 수 있도록 적절한 기동에 따라 자원 접근 면허를 준다.

IV. 전자상거래 시스템의 자바 애플리 기반 설계
전자상거래 시스템의 이동 에이전트 기반 설계는 구매자
에이전트, 판매자 에이전트, 구매 에글릿, 판매 에글릿, 및 시장 에이전트로 구성된다고. 판매 에이전트의 에이전트와 판매자 에이전트. 및 시장 에이전트는 사용자를 위한 고정 에이전트(stationary agent)이기. 구매 에글릿과 판매 에글릿이 이동 에이전트 (mobile agent)이다. 이동 에이전트는 대응 관계(peer-to-peer) 모델이기 때문에 마스터(MMmaster)와 서버(MMserver)로 나누어 설계한다.

그림 1에서 이동 호스트에 마스터가 설치되고 이동 서버에 서버가 설치된다. 공동적으로 자바 가상 머신(JVM)과 에글릿 서버가 설치되어 있다. 에글릿 서버는 JAVA(Agent Transfer Protocol) 사용, 에글릿 제어문맥, 및 에글릿 브로커(Tahiti)로 구성된다. JAVA은 현재 동작중인 에글릿과 통신하거나 전송하기 위하여 MM 메시지를 처리한다. 에글릿 제어문맥은 서버에 존재하는 에글릿을 수행시키며, 에글릿의 생명 주기를 제어한다. 에글릿 브로커는 이동 에이전트 자바 클레스를 생성하여 동작시키는 사용자 인터페이스와 이동 에이전트의 생명 주기를 제공하는 사용자 인터페이스를 갖는다.

MMmaster는 이동 호스트의 고정 에이전트로서 사용자의 인터페이스 하면서, MMslave의 인스턴스(instance)인 자바 에글릿을 생성시켜 이동 서버인 MMserver로 보내고, 처리 결과를 보고 받아 사용자에게 그 결과를 보여준다. MMslave는 MMmaster로부터 생성되는 자바 에글릿으로서, 방문하고자 하는 서버 목록에 따라 사용자의 구매 또는 판매 물품에 대한 요구 사항을 실고 이동 서버인 MMserver로 직접 이동하여 이동 서버에서의 처리 결과를 가지고 이동 호스트로 다시 돌아와 MMmaster에게 처리 결과를 보고한다. 한편, MMserver는 이동 서버에 존재하는 고정 에이전트로서, MMmaster가 보내는 MMslave가 도착하면 이를 대기 큐에 넣고 요구 사항에 따라 처리하고 그 결과를 MMslave에게 넘겨준다.

그림 2는 이동 호스트에서 동작하는 MMmaster의 MMslave의 수행 흐름을 보여준다.

① Tahiti 사용자 인터페이스를 사용하여 MMmaster를 실행 시켰을 때 MMslave 화면을 띄운다. MMslave 화면은 사용자 정보와 물품 정보의 입력을 받는다. 정보 입력을 마친 후 buy 버튼 또는 sell 버튼을 누른다.
② 사용자가 buy 버튼 또는 sell 버튼을 누르면 MMmaster는 이동 에이전트인 MMslave 인스턴스를 생성시켜 목적지로 보낸다. 이 때 우선 순위에 따라 복수개의 MMslave 인스턴스를 생성시켜 복수개의 목적지로 보낼 수 있다.
③ MMslave 인스턴스가 임무 수행을 완료한 후 MMmaster로 되돌아와 수행 결과를 보고한다. MMmaster는 수행 결과를 MGresult 화면에 보여준다.

그림 3은 이동 서버에서 동작하는 MMserver와 MMslave의 수행 흐름을 보여준다. MMserver는 고정 에이전트로서 이동 에이전트인 MMslave로부터 사용자의 요구를 받아들어 전자상거래 서비스를 수행한다.

그림 1. 전자상거래 시스템의 에글릿 기반 전자상거래 설계

그림 2. 이동 호스트에서 MMmaster와 MMslave의 수행 흐름
그림 3. 이동 서버에서 MMserver와 MMslave의 수행 호름

① 이동 예약트한 MMslave가 도착하면 MMserver에 도착 신고를 하고 대기 큐에서 기다린다. MMserver는 상태 완도우 MMslave의 상태를 보여준다.

② MMserver는 MMslave가 가져온 사용자 요구를 해석하여 Item list에 표시하고 새로운 스크렌을 생성하여 Matchmaking을 수행하고 그 결과를 MMslave에게 보낸다.

③ MMslave는 수행 결과를 받아서 사용자 요구 사항과 함께 이동 호스트의 MMmaster에게 되돌라간다.

1. MMMaster

MMMaster는 사용자 인터페이스 MMwin으로부터 사용자의 요구 사항을 입력받고 Buy 버튼이나 Sell 버튼이 눌려졌다면 MMslave 인스턴스를 생성시킨다. MMslave로부터 수행 결과를 받으면 MMresult 화면을 띄우고 임무 수행 결과를 사용자에게 보고한다. 그림 4는 MMMaster 클래스, MMwin 클래스, MMresult 클래스, 애플릿 서버가 제공하는 API, JDK1.1.8의 제공하는 API, 클릭 (callback)함수, MMslave간 수행 관계를 보여준다.

사용자가 xmas.demo.MMmaster를 Tahlil 화면에서 실행하면, MMmaster는 MMS 화면의 MMwin을 보여준다. MMwin 화면에서 사용자의 이름, 소속, 전화번호, e-mail 등의 사용자 정보와 물품 종류, 제목, 수량, 단가 등의 물품 정보를 입력하면, MMmaster는 MMtrp 인스턴스를 생성시켜 면 Buy 버튼이나 Sell 버튼을 누르면 MMslave를 실행시킨다. MMMaster는 생성된 MMtrp 인스턴스를 이동 예약트인 MMslave에게 전달하고 받은 목록에 등록된 곳 (MMserver)으로 dispatch한다.

MMslave는 주어진 임무를 완수하고 홀 예약트인
MMserver로 입력되어 수행 결과를 보고하려고 MMmaster는 MMserver ID와 MMmaster URL을 갖고 있다.

MMserver는 생성된 MMslave의 정보(MMslave id, MMmaster id)를 관리하고, 수행 후 돌아와야 할 수 있도록 처리를 유지한다.

MMslave가 가져야 할 수 있는 자료가 되면 MMmaster에게 보고 메시지를 띄워 보고한다. MMmaster는 MMresult 화면을 통하여 수행 결과를 보여주고 MMslave를 제거한다.

그림 5는 MMmaster 클래스의 메소드들을 보여 준다. MMmaster 클래스에는 메글릿 관련, 사용자 인터페이스, 메시지 제어, 및 MMslave 제어 관련 메소드들을 포함한다.

2. MMslave

MMslave 클래스는 이동 에이전트를 구성하는 것으로서 에글릿 서버에서 제공되는 API들을 기본적으로 이용한다. 주요 기능은 여러 지역에 따라 MMserver로 이동하여 사용자에 의한 입수를 수행하고, MMserver에 도착하여 등록하고, MMserver에게 가져온 MMId 정보를 전달하여 수행시키고, 그리고 수행 결과를 받아 MMmaster로 되돌아와 결과를 보고하는 기능 등을 갖는다. 그림 6은 MMslave 클래스의 기본형 메소드들을 보여 준다.

3. MMserver

이동 서버에 존재하는 MMserver는 그림 7과 같이 MMserver 클래스를 중심으로 구성된다. MMserver에서는 여러 개의 MMslave들이 이동 방사하여 사용자의 구역 또는 판매 정보를 제공한다. 이동 레이어로부터 MMserver를 발동 한 이동 에이전트의 MMslave들은 MMserver에 등록한 후 상대 목록에 표시한다. MMserver는 MMslave의 요구 정보에 따라 구역에 MMslave이면 구역에 대기, 판매 정보에 MMslave이면 판매자 대기의에 넣고 가져온 정보를 해석한다. 각 MMslave가 가져온 MMslave ID와 사용자 정보, 판매 정보들은 Item List 에 추가하여 표시한다. MMserver는 MMslave가 가져온 정보에 따라 MatchMaking를 처리한다. MMserver는 판매 정보가 일치되면 바로 그 결과를 MMslave에게 전달하는다. MatchMaking은 각 MMslave가 도착할 때마다 수행되고 Item List에 표시한다. MMslave가 만족한 결과를 얻었을 경우, MMserver는 MMslave들을 블록 에이전트의 MMmaster에게 되돌아오도록 한다.

MMserver 클래스는 에글릿 서버에서 제공되는 API들을 기본적으로 이용하여 설계하였으며, MMserver 클래스의 기본형 메소드들은 그림 8과 같다.
V. 전자상거래시스템의 이동에이전트 기반 시범구현

전자상거래 시스템은 운영 체제로 Windows NT 4.0, 프로 그레밍 언어로 자바 JDK1.1.8, 이동 에이전트 시스템으로 IBM 애플리케이션 Workbench 1.0.3 버전을 사용하여 구현하였다. 구성 시스템은 애플리케이션 기반의 이동 에이전트 기술을 이용하여 MMS(Mobile Market System)를 구현하였다. MMS는 이동 에이전트가 사용자 구매 또는 판매 정보를 가지고 시장에 참여하면서 적절한 구매자 또는 판매자를 찾아 관련 정보 및 상품을 협상하거나 교환하는 서비스이다.

MMS에서는 물건을 구매하거나 판매하고자 하는 사람을 MMaster로, 구매자 또는 판매자들의 물품 매매를 대행하는 사람을 MSlave로, 물품 매매가 이루어지는 장소를 MServer로 서열리턴 해야 한다. MServer에는 MMaster들의 물품 매매 대리인인 MSlave들이 모여 매매를 체결하는 장소가 된다.

사용자는 그림 9에서와 같이 Tahiti를 통하여 이동 호스트의 MMaster와 이동 서버의 MServer를 각각 기동시킨다. MMaster는 그림 10의 MMain 화면을 띄우고 MServer는 그림 11의 화면을 띄운다.

MMaster 클래스는 이동 호스트에서 동작하는 모든 기능을 제어하는 핵심 클래스이다. 사용자가 MMS를 invocation하여 MMaster 클래스의 onCreation() callback 함수가 불리고, 이 함수 내에서 MMain 클래스를 호출함으로써 전자상거래 서버와 화면이 나타난다. 이 화면에서 Buy 버튼이나 Sell 버튼을 누르면 전자상거래 서버 화면의 사용자 정보(사용자의 이름, 소속, 전화번호)의 물품 정보(물품 종류, 물품 이름, 수량, 단가)를 인스턴스에 저장하여 MMaster 클래스에게 전달한다. 이 클래스는 buyPressed()나 sellPressed() 함수를 호출하여 매매를 생성한다. 매매론은 요구한 정보를 받아 주어진 목적지로 실행 코드와 함께 이주한다.

이동 서버의 MServer는 도착한 Buy 또는 Sell 이동 에이전트를 프로세스로 생성하여 초기화시키고 주어진 임무를 수행하도록 한다. 이동 서버에 도착한 이동 에이전트는 그림 11의 화면과 같이 Market Status List에 도착을 알리는 메시지를 보내 주고 구매 또는 판매 의뢰에 따라 Buying Item List 또는 Selling Item List에 물품 정보를 등록하고 새로운 셀렉트를 생성하여 요구된 작업을 수행한다.

그림 9. MMaster 실행

그림 10. MMain 화면으로부터 정보 입력

그림 11. 전자상거래 서버에 도착한 Buy 자바 애플리케이션
MMserver 클래식은 이동 서버에서 동작하는 모든 기능들을 제공하는 핵심 클래스이다. 사용자가 전자상거래 서버용 프로그램을 실행시킨 MMserver 클래식의 onCreation() callback 함수가 끝나고, 이 함수 내에서 MMserverwin 클래스를 호출함으로써 전자상거래 서버 화면이 나타난다. 그리고 run() callback 함수가 불린다. 외부로부터 메시지를 접수받기 위하여 handleMessage() callback 함수를 사용한다.

이동 서버에서 구매자와 판매자 사이에 매매가 체결되면 그 결과를 그림 12와 같이 Status List, Item List, Waiting Place에 적절한 메시지를 표시한다. 매매가 체결된 이동 에이전트들은 실행 결과를 알고 자신의 홈 에이전트인 이동 호스트의 MMmaster로 되돌아간다. 이동 서버에서의 매매 체결은 구매자와 판매자의 상품 정보(상품명과 가격)를 비교함으로써 이루어진다.

Buy 또는 Sell 이동 에이전트가 처리 결과를 가지고 이동 호스트로 되돌아오면 MMmaster는 Status List에 상태를 표시하고 그림 13의 MMresult 화면에 매매 결과를 보여 준다.

VI. 결론 및 향후연구

본 논문에서는 전자상거래 시스템을 구축하기 위해서 자바 에글릿을 이용하여 설계할 수 있음을 보였다. 사용자를 대신하여 복수 개의 자바 에글릿들이 상거래 서버 시점으로 직접 이용하여 서비스나 상품을 자동적으로 협상하거나 매매하고, 그 결과를 사용자에게 실행 코드와 함께 결과를 보고하였다. 자바 에글릿은 실행 코드와 함께 사용자 요구사항을 가지고 직접 이상해능성을 때문에 이동 콜러팅의 전자상거래 시스템 구축에 응용을 할 수 있었다. 이것은 전자상거래의 소비자 구매 행동 모델에서 구체적으로 '상품 탐색(무엇을 살 것인가)'과 '판매자 탐색(누구로부터 살 것인가)'에 해당되는 과정을 자바 에글릿 기술을 사용하여 구현한 것으로서, 특히, 사용자가 상품을 검색할 경우 복수개의 자바 에글릿을 생성하여 복수개의 전자상거래 서버로 병렬 전송하여 수집함으로써 병렬 강도 검색 가능성을 보여주고 있다.

향후 연구로는 전자상거래 서비스 중에서 판매자 팀색에서 가격과 수량을 이용하였으나, 매출 조건, 유자 보수/지불 방법 등을 고려한 4단계의 협상 과정을 추가 구현하여야 한다. 또한, 사이트 구현된 전자상거래 시스템은 자바 에글릿 기술을 전자상거래에 적용할 수 있음을 보인 초기적인 서비스이며, 향후 사용자의 구매 패턴(구매 빈도가 높은 물품 종류, 협상 조건 등)을 이용한 이동 서버의 선택, 이동 에이전트의 이동 서버 탐색 방법에 대한 연구, 협상 과정의 구현, 웹상의 소형 사이트들로부터 상품 정보의 자동 추출, 필터링 기법 등의 연구를 진행할 예정이다.

참고 문헌


1999.


김정웅(Phyong-Jung Kim)
정보화
1985년 2월 충남대학교 계산통계학과 (학사)
1995년 2월 KAIST 대학원 입학
전산학과(공학석사)
2000년 2월 충남대학교 대학원 컴퓨터과학(이학박사)
1995년 10월 전자제조기술조직응용기술학과 취득
1998년 2월 ~ 1998년 2월 한국전자통신연구원 멀티미디어연구부 신입연구원
1998년 1월 ~ 2000년 12월 한국정보처리학회 학회지 편집위원
1998년 3월 ~ 현재 충북과학대학교 컴퓨터정보학과 교수
<관심분야> : 이동에이전트, 전자상거래, 컴퓨터네트워크, 멀티미디어
김정호(Jeong-Ho Kim) 정회원
1980년 2월 경북대학교 전자공학과(학사)
1983년 경북대학교 대학원 전자공학과 (공학석사)
1994년 단국대학교 대학원 컴퓨터공학과(공학박사)
1983년 3월 ~ 1996년 2월 한국전자통신연구소 (책임연구원, 실장)
1989년 정보처리기술사
1990년 공업계획제어기술사
1991년 정보통신기술사
1996년 3월 ~ 현재 국립한밭대학교 정보통신·
컴퓨터공학부 교수
<관심분야> : 데이터통신, 컴퓨터네트워크, 통신서비스

박진영(Jin-Yang Park) 정회원
현재 인하공업전문대학 컴퓨터정보과 교수