

중소기업형 ERP 시스템 설계 및 구현

Design and Implementation of an ERP System for Medium and Small Enterprises

김 분 희*

Boon Hee Kim

요 약

기업의 경쟁력 향상을 위한 ERP(Enterprise Resource Planning) 시스템 구축은 비용과 시간의 소요량이 크므로 많은 중소기업에서 그 도입이 늦어지고 있다. 본 논문에서는 ERP 시스템의 도입으로 인해 얻을 수 있는 장점을 극대화하고 중소기업에서 그 도입 시기를 앞당길 수 있는 대안으로써 ERP 어플리케이션 서비스에 대해 그 설계 및 구현 방법을 제시하였다. 이 방법은 대부분의 기존 연구에서처럼 어플리케이션 구현 방법 자체만을 제시하는 것이 아니라, 우리 나라 산업의 중추인 중소기업 환경에 적합한 시스템 설계 모델을 제시한다는 점에서 기존의 연구보다 그 실효성이 높다.

Abstract

The introduction of an ERP(Enterprise Resource Planning) system for the competitive enterprise has been late because of a great cost. In this paper, we design and implement ERP application service as an alternative proposal of an ERP system for medium and small enterprises. Unlike most of previous works that show only the implementation of their ERP application, this approach provides the design model of a suitable environment for medium and small enterprises.

1. 서 론

불과 몇 년 사이 국내 100대 기업을 기준으로 업무 효율화를 높이기 위한 수단으로써 ERP(Enterprise Resource Planning) 시스템[1][2][3] 도입이 증가하고 있다. 이는 변화하는 기업 환경에 빠르게 대응할 수 있고, 세계적인 표준과도 일치할 수 있어 기업의 경쟁력이 높아진다는데 기인한다.

이에 따라 우리나라 산업 중추라고 할 수 있는 연간 매출액 50억에서 500억 사이의 중소기업들도 ERP 구축에 큰 관심을 갖고 있다[4]. 그러나 ERP 도입을 위해 투자해야 할 막대한 양의 시간, 비용, 관리인력이 요구되므로 현실적으로 기업 내

구축이 어려운 실정이다. 실제 주요 ERP 업체들은 대부분 13 개월의 설치기간을 빠른 것으로 생각하고 있다[4].

이러한 문제를 해결할 수 있는 방법으로써 ERP 어플리케이션 서비스가 거론되고 있다[4]. 이는 ERP 소프트웨어 패키지들의 단일 라이선스 비용만으로도 시트 당 수천 달러에 이르지만, 실제 비용은 이 복잡한 소프트웨어를 지원하는 IT(Information Technology) 기반 구조를 구축하는데 들어가므로 이러한 비용적인 면을 해결하기 위해 소프트웨어 및 기반구조를 빌려준다는 의미이다[5][6][7][8][9].

이에 본 논문은 비용적인 면에서 우수한 ERP 어플리케이션 서비스에 대해서 그 기반구조와 소프트웨어의 형태를 변화하는 기업 환경에 빠르게 대응할 수 있는 구조로써 제안하고자 한다.

* 학생회원 : 세명대학교 전산정보학과
(flora@altavista.co.kr)

2. 관련연구

언어로 JAVA를 제안하고자 한다.

2.1 JAVA

(1) 개 요

기업용 어플리케이션을 포함해 소프트웨어 산업 전체가 웹과 인터넷기반으로 재편되고 있는 상황에서 웹 기반 ERP 시스템 구현은 당연한 현상이다. 이때 독립된 어플리케이션 개발과 웹 기반 어플리케이션 개발이 동시에 가능한 객체지향 언어로 SUN社의 JAVA[12]가 있다. JAVA의 주요 특성을 정리해 보면 아래와 같다.

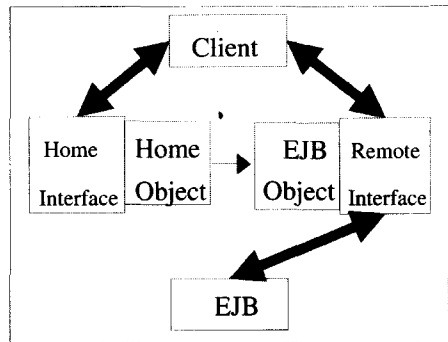
- 플랫폼 독립성 : 보통의 프로그래밍어는 개발한 플랫폼에 종속적인 이진코드(Binary Code) 형태로 실행되나 JAVA는 플랫폼에 독립적인 Byte Code가 JVM(Java Virtual Machine)에 의해 인터프리트 방식으로 실행된다.
- 간단하며 친숙한 언어 : 객체지향언어의 원조격인 C++과 비교해 봤을 때 그 구조가 비슷하여 개발자에게 친숙하며, 각종 버그의 원인이었던 Pointer 연산등의 복잡한 기능이 일부 삭제되어 개발하기가 쉽다.
- Multi-Thread 지원 : 실행할 수 있는 최소의 단위인 thread를 여러 개 두어 각각의 thread 들은 서로 다른 일을 할 있다.

JAVA는 일반적으로 Linux 환경에는 적합하나 마이크로소프트 환경에서는 DB(DataBase) 및 네트워크 작업을 하는데 여러 가지 어려움이 있는 것으로 알려져 있다[4]. 그러나 1995년말 마이크로소프트사와 소스코드 라이선스 계약을 맺은 이후로 그러한 어려움은 점점 해소되고 있어, JAVA의 많은 잇점들의 활용이 용이해지고 있다 [12][13].

이에 본 논문에서는 ERP 어플리케이션 개발

(2) EJB(Enterprise JavaBeans)[16]

JAVA로 웹 어플리케이션을 구현하는데 있어서 시스템 개발과 유지 보수를 용이하게 하며 방대한 시스템에 적용될 수 있도록 3-tier(3계층) 모델을 구현하는데 JSP, Servlet, EJB기술을 포함하는 J2EE가 소개되고 있다[16]. 여기서 EJB 서버는 트랜잭션, 시큐리티, DB연결 풀링과 같은 미들웨어 기능을 담당하여 개발자가 비즈니스로직에만 집중할 수 있도록 한다.



(그림 1) EJB 객체의 생성 및 실행순서

EJB의 구성요소는 다음과 같다. 외부 메소드를 제공하는 Remote Interface, 라이프 사이클과 관련된 메소드를 정의하는 Home Interface, 실제로 비즈니스 메소드를 정의 및 구현하는 Bean Class, 그리고 DB로의 포인터를 제공하는 Primary Key로 이루어져 있다. (그림 1)은 EJB 객체의 생성과 실행순서를 나타낸다.

위 실행순서에서 Remote Interface는 CORBA [11]의 IDL 역할로써 run-time 시 Proxy 객체를 생성해 EJB에 접근하게 한다.

2.2 ERP(Enterprise Resources Planning)

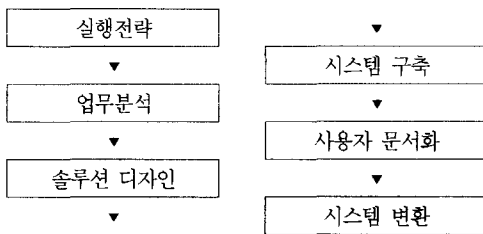
ERP는 각종 조사 평가기관인 가트너 그룹

(Gartner Group)의 정의에 의하면 기업 내의 업무 기능들이 조화롭게 발휘되도록 설계된 응용 프로그램(Application Program)의 집합으로 차세대의 업무 시스템이다. 기업에서 ERP를 도입한 후 기대할 수 있는 효과로는 통합업무처리를 기반으로 한번의 입력을 통하여 모든 부서가 데이터를 공유하여 데이터 일원화라는 업무효율화, 기업 업무 전체에 걸친 정보획득의 실시간화로 신속하고 정확한 정보획득, 업무효율화 추진으로 이익확대 부분에 역량을 집중할 수 있어 기업 경쟁력의 확대, 전사적인 관점에서 접근하므로 정보를 이용한 의사결정이 가능해져 관리수준의 향상 등이 있다.

비즈니스	운영 프로세스	마케팅 및 정보프로세스
		디자인 및 개발프로세스
		조달 및 물류 프로세스
		생산 프로세스
		제품/서비스 인도 프로세스
프로세스	관리 프로세스	정보처리 프로세스
		조정 프로세스
		조절 프로세스
		커뮤니케이션 프로세스
		지식 프로세스

(그림 2) 프로세스의 유형

ERP 시스템의 핵심은 복잡한 비즈니스 프로세스 모델과 프로세스 효율화인데 (그림 2)은 비즈니스 프로세스의 유형을 보여준다[2]. 본 논문에서는 마케팅 및 정보 프로세스에 해당하는 원 부자재 재고추이를 예로 든다.



(그림 3) ERP 프로젝트 추진 6단계

기업의 특징에 맞게 ERP시스템을 구축하는데 있어서 가장 중요한 ERP 프로젝트 추진 절차는 (그림 3)에서와 같이 프로젝트 팀을 구성하여 그 추진 목적 및 추진계획을 수립하는 실행전략 단계로부터 시작하여 운영시스템을 구축하고 현업 사용자에게 교육하는 시스템 변환 단계로 마무리 된다.

(표 1) ERP 패키지 도입과 자체개발의 비교

항목	도입	자체개발
아웃소싱	구축시 전문 컨설턴트의 지원이 필수적임	컨설턴트와 같은 전문 인력에 대한 투자는 적으나 전체적인 인력이 많이 소요됨
사용자교육	패키지의 용어나 개념을 이해하기 위해 사용자에게 대한 교육은 필수적임	사용자용어를 이용한 교육으로 별도의 교육이 필요 없음
예외기능구현	회사만의 고유한 기능에 대하여는 add-on으로 구현가능	개발초기부터 회사 고유의 기능 위주로 설계 및 구현할 수 있음
기간	표준 프로세스만을 적용할 시 단기간이 소요됨	자체분석과정이 필요하므로 통합시스템을 가지기 위해서는 많은 시간이 소요됨
성능	자료의 관계가 복잡한 경우에도 패키지 자체에서 처리할 수 있음	개별적인 자료관리는 용이하나 통합적 관점에서는 관리가 어려움

ERP 도입 형태를 보면 우선 패키지를 구입하여 구축할 수 있고, 자체 프로그래머에 의해 개발할 수가 있다. <표 1>은 아웃소싱, 사용자교육, 예외기능구현, 기간, 성능면에서 ERP패키지를 도입한 경우와 자체 개발한 경우를 비교한 것이다[1].

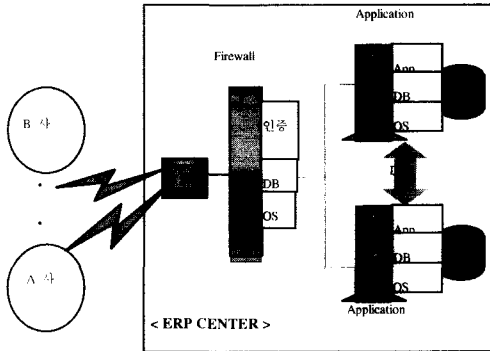
위와 같이 ERP 패키지[10]를 도입하거나 자체 개발을 하는 데에는 ERP 시스템 구축비용과 시간의 소요량이 크므로 많은 중소기업에서 그 도입이 늦어지고 있다.

이에 본 논문에서는 ERP 시스템의 도입으로 인해 얻을 수 있는 장점을 극대화하고 중소기업에서 그 도입 시기를 앞당길 수 있는 대안으

로써 ERP 어플리케이션 서비스에 대해 그 설계 및 구현 방법을 제시하였다.

3. MSERP 시스템 설계

3.1 제 안



(그림 4) MSERP 네트워크 구성도

ERP 어플리케이션 서비스는 기업의 입장에서 봤을 때 ERP 소프트웨어를 빌리는 것이라 할 수 있다. 이는 막대한 비용이 들어가는 IT 하부구조 설비비용이 들어가지 않는다는 점에서 기업의 정보시스템을 ERP화하는데 부담이 경감된다. 하지만 ERP 어플리케이션 서비스 제공 업체는 단일 서버에서 다수의 고객들을 지원하기 위해 철통같은 보안모델과, 네트워크 비용을 내리기 위해 완벽한 웹 운영이 가능한 어플리케이션 세트가 필요하다. (그림 4)은 본 논문에서 제안하는 ERP 센터의 네트워크 구성도이다.

ERP 센터는 비슷한 계열의 회사들을 단위로 동일한 서버를 부분적으로 이용하게 한다. 동일계열 회사끼리 묶는 이유는 ERP프로세서가 거의 동일하여 어플리케이션의 변경을 최소화 할 수 있기 때문이다.

ERP 센터 이용은 먼저 기업의 정보시스템에 대한 철저한 보안을 위해 원격의 이용자는 ERP 시스템 사용을 위한 첫 절차로 방화벽 서버(Firewall

Server)를 통과하여 인증을 받는다. 그 다음으로 어플리케이션 서버에서 각종 ERP시스템의 혜택을 받게된다.

여러 기업에서 동시에 이용하는 어플리케이션 서버는 무엇보다도 견고한 Fault-Tolerance 데이터 센터를 요하게 된다.

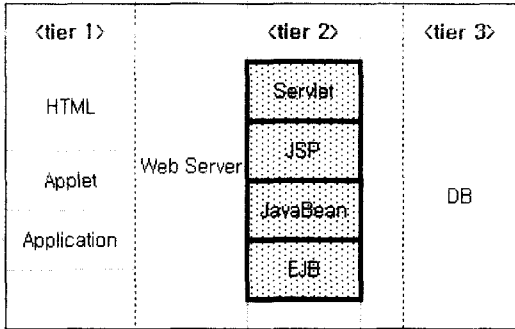
이에 본 논문에서는 동일한 시스템을 Dual-System으로 두어 어플리케이션 서버 1에 고장이 일어나면, 어플리케이션 서버 2가 동작하는 방식으로 시스템의 안정성을 높일 수 있도록 네트워크를 구성하고자 한다.

3.2 구 조

ERP Application Service를 제공하는데 있어 접속하는 사용자의 수를 고려하는 것은 매우 중요하다. 기존의 단순한 클라이언트/서버 형태의 2-tier 모델에서는 DB 요청이 집중되는 경우 뚜렷한 성능저하를 가져왔다. 이에 MSERP 시스템은 효율성(Efficiency), 확장성(Extensibility), 신뢰성(Reliability)을 만족할 수 있는 컴포넌트 기반의 3-tier 환경을 제안한다[3][16].

3-tier는 어플리케이션을 프리젠테이션(Presentation), 비즈니스 로직(Business Logic), EIS(Enterprise Information System) 계층으로 나누고, 이 세 계층을 네트워크로 묶는 것을 말한다. 이때 프리젠테이션 계층은 ERP 시스템 사용자에게 정보를 보여주고, 사용자와 상호작용하는 계층으로 HTML(HyperText Markup Language)와 웹 환경에 적합한 JAVA 애플릿(Applet) 기술들이 여기에 포함된다.

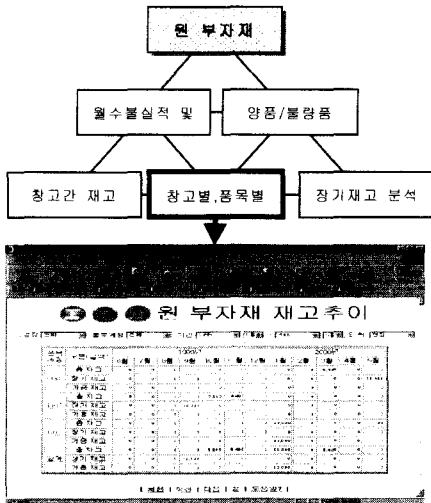
비즈니스 로직 계층은 각종 업무 규칙, 요구사항 등을 처리하는 계층으로 트랜잭션(Transaction)을 처리하는 서버가 위치하게 된다. EIS 계층은 각종 데이터를 저장, 관리하는 계층으로 DBMS(DataBase Management System)가 위치한다. 아래(그림 5)는 3-tier MSERP 어플리케이션 구성을 나타낸다.



(그림 5) MSERP 어플리케이션 구성

위의 MSERP 어플리케이션 구성에서 JAVA 기반 컴포넌트 개발 표준인 EJB(Enterprise JavaBeans)[14], 컴파일 과정 없이 동적인 웹 페이지 작성을 편리하게 하는 기술인 JSP(JAVA Server Pages)[15] 등은 J2EE(JAVA2 Enterprise Edition)[16]에 포함되어 있다. J2EE는 서버 측 JAVA 기술을 집대성하여 복잡하고 방대한 시스템 개발을 가능하게 한다. 데이터베이스 서버는 J2EE 설치 시 디폴트로 만들어지는 CloudScape를 사용하였다.

4. 시스템 구현



(그림 6) 원 부자재 재고추이 작성

ERP 시스템의 복잡한 비즈니스 프로세스 가운데 (그림 6)는 마케팅·정보 프로세스에 해당하는 원 부자재 재고관리 부문을 구성하는 모듈들이다. 그 중 창고별, 품목별 재고관리 추이 모듈은 회사의 재고자산 최소화를 유도하기 위하여 사업장별, 품목계정별로 해당시기의 원 부자재 재고추이를 월별, 연별 마감시점에서 보여준다.

```

...
public class SEntityBean implements EntityBean{
    ...
    public int.ejbCreate(int st, int st2, int st3)
        throws CreateException{
        this.st = st;
        this.st2 = st2;
        this.st3 = st3;
    }
    ...
}
    
```

(그림 7) 엔티티 빈 클래스

```

...
SSession = SessionHome.create();
SEntity = SSession.getEntity(st);
...
System.out.println("<tr>" + SEntity.getStock() +
    "</tr>");
...
    
```

(그림 8) 클라이언트 Servlet 코드

MSERP는 3-tier JAVA 어플리케이션 구현을 위해 J2EE 기술을 적용했는데 (그림 6)과 같은 결과를 얻기 위한 구현과정을 설명하면 다음과 같다. 먼저 서버측 미들웨어 역할을 하는 EJB는 두가지 형태로 이루어져 있는데, 그 중 엔티티빈(Entity Beans)은 DB의 실제로 저장되는 데이터를 표현하고 있으며, 엔티티빈의 상태값이 변경되면 곧바로 데이터베이스에 변경 내용이 반영된다.

(그림 7)에서 엔티티 빈 클래스인 SEntityBean의.ejbCreate 메소드는 EJB를 설치 및 테스트 할 수

있게 해주는 프로그램으로 J2EE 내에 있는 deploytool 을 실행하여 CloudScape 관련 SQL문으로 자동 생성된다.

서버측 미들웨어 역할을 하는 EJB의 다른 형태로 세션빈(Session Beans)은 엔티티빈과는 달리 DB에 존재하는 데이터에 접근만이 허용되어 클라이언트의 요구에 따라 조작의 역할을 한다.

(그림 8) 클라이언트의 요구를 받아들여 EJB의 세션빈과 엔티티빈으로부터 클라이언트가 원하는 정보를 원하는 폼으로 생성해주는 클라이언트 Servlet 프로그램의 부분이다.

위 코드는 객체 접근 시 필요한 Remote Interface 를 이용하는데, 먼저 세션빈을 생성한 후 엔티티빈의 Primary Key인 st 값을 이용해 DB에 저장되어 있는 값을 대입시켜 그 결과값을 출력한다.

이 Servlet 프로그램 내에서 데이터베이스에서 요구되어진 데이터를 가져와 사용자의 웹 브라우저에 나타나도록 HTML 코드를 생성하기 위한 클래스가 생성되는데 다음은 이 클래스 코드의 일부이다.

```

OutputStreamWriter outS = new
OutputStreamWriter(res.getOutputStream());
PrintWriter outSW = new PrintWriter(outS);
res.setContentType("text/html;charset=edu-kr");
out.println("<html>");
//...
out.println("</html>");
    
```

(그림 9) HTML 코드 생성

5. 결 론

중소기업 전산화 구축의 문제점으로 과중한 하드웨어 비용부담, 구성원들의 전산의식 결여, 자체 전산 인력 보유미비 및 운영상의 어려움등이 있으나 기업의 경쟁력 향상을 위해서는 ERP 시스템이 요구된다. 그러나 ERP는 구축 예산, 인력, 시간의 소요량이 매우 크므로 많은 중소기업에서는 그 도입을 늦추고 있는 실정이다. 본 연구에서

는 ERP 시스템의 도입으로 인해 얻을 수 있는 장점을 극대화하고 중소기업에서 그 도입시기를 앞당길 수 있는 대안으로써 ERP 어플리케이션 서비스에 관하여 그 설계 및 구현 방법을 제시하였다.

기존의 연구에서는 ERP 어플리케이션 구현 방법 자체에 초점을 두어 자칫 어플리케이션 개발 시 꼭 필요한 시스템 설계부분을 간과하는 경우가 있었다. 특히 ERP와 같은 기업의 대규모 통합 정보 시스템은 변화하는 기업의 요구에 발빠르게 대처해야 하는데, 이는 시스템 설계가 선행된 후 어플리케이션 설계를 해야 한다는 의미이다.

본 논문에서 제안하는 MSERP의 ERP 어플리케이션 서비스는 기업의 입장에서 봤을 때, ERP 소프트웨어를 빌리는 것이라 할 수 있다. 이는 막대한 비용이 들어가는 IT 하부구조 설비비용이 들어가지 않는다는 점에서 기업의 정보시스템을 ERP화하는데 부담이 경감된다. 그리고 단일 서버에서 다수의 고객들을 지원하기 위해 철통같은 보안모델과, 네트워크 비용을 내리기 위해 완벽한 웹 운영이 가능한 어플리케이션 세트가 필요하다. 그래서 MSERP는 먼저 기업의 정보시스템에 대한 철저한 보안을 위해 원격의 이용자는 ERP 시스템 사용을 위한 첫 절차로 방화벽 서버(Firewall Server)를 통과하여 인증을 받는다. 그 다음으로 어플리케이션 서버에서 각종 ERP시스템의 혜택을 받게된다. 또한 여러 기업에서 동시에 이용하는 어플리케이션 서버는 무엇보다도 견고한 Fault-Tolerance 데이터 센터를 요하게 되므로 MSERP는 동일한 시스템을 Dual-System으로 두어 어플리케이션 서버 1에 고장이 일어나면, 어플리케이션 서버 2가 동작하는 방식으로 시스템의 안정성을 높일 수 있는 네트워크 구성을 제안한다.

MSERP Application은 접속하는 사용자의 수를 고려하여 DB요청이 집중되어도 성능 발휘가 될 수 있도록 효율성, 확장성, 신뢰성을 만족할 수 있는 컴포넌트 기반의 3-tier 환경을 제안한다. 이

때 구현언어로 웹기반 애플릿과 독립된 어플리케이션 개발이 동시에 가능한 JAVA로 하였다.

JAVA는 플랫폼에 독립적이어서 어떠한 운영체제나 마이크로프로세서에도 자유로워서 변화가 많은 기업환경에 필요한 요건을 지녔다. 그리고 3-tier 환경에 적합한 J2EE(JAVA2 Enterprise Edition)를 서버 측 어플리케이션 개발에 이용하여 복잡하고 방대한 시스템 개발을 가능하게 하였다.

본 연구에서는 한국 중소기업형 ERP 시스템에 초점을 두어 소프트웨어 및 기반구조를 설계했는데, 다양한 환경에서 ERP 시스템을 개발하여 그 성능을 비교 평가하고, HCI(Human Computer Interaction)적인 측면에서 GUI(Graphic User Interface)를 설계하여 편리한 사용자 환경을 만든다면 ERP 시스템 설계의 완성도를 높일 수 있을 것이다.

참 고 문 헌

[1] 김영중, 전사적 자원관리 시스템(ERP)의 성공 요인에 대한 연구, 홍익대학교, 석사학위논문, MIS 전공, 1999.
 [2] 최경일, 차세대 ERP와 산업공학, ie매거진, 1999, 제 6권 제2호(통권 17호).
 [3] 정태원, 중소기업의 성공적인 ERP 도입 방안에 관한 연구, 서강대학교, 석사학위 논문, 1997.
 [4] <http://www.kyungcom.co.kr/>

[5] Lee R., An Enterprise Decision Framework for Information System Selection, Information Systems Management, Fall 1998, pp.7-13.
 [6] Vollman, T. E., Manufacturing Planning and Control Systems, IRWIN, 1991.
 [7] Eliot S. Elfner, Integrating ERP DWKM to Create a Quality Management Culture, QGCF, 1999.
 [8] G. Berton Latamore, Flexibility Fuels the ERP Evolution, APICS, 1999.
 [9] Lucas, H. C., Implementing Packaged Software, MIS Quarterly, Dec. 1998, pp.537-549.
 [10] SAP, SAP R/3 System Overview, Introduction to Partner Academy, 1998.
 [11] <http://www.omg.org/>
 [12] <http://java.sun.com/>
 [13] K. Mani Chandy, Adam Rifkin, Paolo A.G. Sivilotti, Jacob Mandelsonk, Matthew Richardson, k Wesley Tanaka, and Luke Weisman, A World-Wide Distributed System Using Java and the Internet, HPDC5'96, August, 1996.
 [14] Michael Morrison, R.W., P.C.,and J.L. How to program JavaBeans, ZiffDavis Press, 1997.
 [15] Jason Hunter, Java Servlet Programming, O'REILLY, 1998.
 [16] <http://java.sun.com/j2ee/>

◎ 저 자 소 개 ◎

김 분 회

1997년 2월 세명대학교 전자계산학과 이학사.
 1999년 2월 세명대학교 전산정보학과 이학석사.
 관심분야 : 전자상거래, 분산객체 컴퓨팅, 경영정보시스템.

