

수리적 분석을 위한 웹 기반 의사결정지원시스템의 설계와 구현

김승권 · 김태형

고려대학교 산업공학과

A Design and Implementation of a Web-based DSS for Mathematical Analysis

Tae-Hyung Kim · Sheung-Kwon Kim

An architecture of a Web-based Decision Support system for mathematical analysis is presented. Front-end modules provide web-client GUI environment for mathematical analysis. The networking architecture is built upon client/server system by Java socket and accesses database by JDBC in WWW. Back-end modules provide decision supporting service and data management for mathematical programming analysis. In the back-end any analysis tools, such as mathematical optimizer, simulation package, or statistics package can be used.

As an application example for this implementation, optimal facility replacement decision problem is selected. In the implementation the optimal facility replacement decision problem is formulated as a shortest path problem. It uses Oracle DB and CPLEX package as the mathematical optimizer. While ORAWeb is designed and implemented on the optimal facility replacement problem, it can easily be extended to any decision supporting problems that would require mathematical optimization process.

1. 서론

인터넷이 글로벌 정보 인프라의 기본 골격이 됨으로써, 인터넷은 이제 단순히 정보를 제공하고 공유하는 기능을 넘어서 인트라넷 어플리케이션이나 웹 기반의 정보 시스템 등과 같은 산업 사회의 여러 부문에서 경영이나 관리를 위한 의사결정의 수단으로서 널리 이용되고 있다(Intranets Redefine Corporate Information Systems, 1996).

네트워크의 하드웨어 기술 및 클라이언트/서버구조로서의 분산 컴퓨팅의 발달로 인해 인트라넷이나 인터넷 환경에서 다수의 컴퓨팅 자원을 접근하고 공유를 통해 연계 운영하게 하는 컴퓨팅 기술이 발달하게 되었다. 이에 소켓 프로그래밍, HTTP/CGI, 서블릿, RMI뿐만 아니라 최근에 CORBA, DCOM 등이 상업적으로 성공하여 표준이 됨으로써 분산 컴퓨팅 및 분산 객체 컴퓨팅 환경에서의 응용 프로그램 개발이 용이해졌다. 따라서, 인트라넷이나 인터넷 환경에서 클라이언트/서버로 구조화하여 의사결정을 지원하는 시스템에 대해 수리모형의 최

적화 기법을 적용하거나, 경험적 접근방법, 시뮬레이션, 또는 통계적 분석 등의 수리 분석을 도입하는 것이 가능하게 되었다.

본 연구에서는 웹 상에서 대상 시스템의 수리적 분석을 통한 의사결정지원시스템을 설계하고 그에 따른 특성을 고찰하였다. 그리고, 구현 사례로서 장비의 최적 교체 정책 결정 문제를 다루는 DSS를 웹 상에서 구축하였다.

2. 기존 연구의 고찰

웹 상에서 의사결정 지원을 목적으로 하는 정보시스템의 구조 설계에 대한 연구와 분산 컴퓨팅 기술에 대한 연구가 최근에 활발하게 진행되고 있다. 그 중에서 Bhargava(1997)는 가정이나 산업 폐기물의 효과적인 재활용 정책 결정을 웹 상에서 지원할 수 있는 시스템을 구현하였다. Bhargava는 재활용 정책 결정 지원을 위해 각 폐기물 처분시 발생하는 처분수익과 각 폐기물 처리장까지의 운송 비용을 고려하여 폐기물의 경로 결정 문제(TSP)로 수리모형화 하고 최적화 환경을 제공하는 AMPL

(Fourer, 1993)을 호출함으로써 해결하도록 하였다. Bahrgava는 웹 기반에서 이를 구현함에 있어서 CGI-HTML에 의해 DBMS와 AMPL을 호출하는 방법을 사용하였다.

Chou(1998)는 주식 거래에 대한 의사결정지원이 단독시스템(stand-alone) 기반에서만 가능했던 기존 시스템(FIFES)을 웹 기반에서 서비스를 할 수 있는 시스템으로 확장하였다. Chou는 이를 구현함에 있어서 전방경계면 모듈에서 데이터베이스 서버의 접근을 JDBC와 미들웨어인 Dataramp(Dataramp) 그리고 JDBC-ODBC 브리지를 사용한 3계층 구조로 구현하였고, 서버의 서비스 모듈에 대한 요청을 CGI에 의해 수행되도록 하였다. FIFES에서의 핵심 모듈은 인공지능 기법을 적용하는 모듈인데, CGI에 의한 접근 방법을 사용함으로써 FIFES의 인공지능 모듈 구현 프로그램을 거의 변함없이 웹 기반 시스템에 그대로 사용할 수 있었다.

본 연구에서는 장비의 최적 교체 정책 결정을 위한 웹 상에서의 DSS인 ORAWeb(Optimal Replacement Analysis by Web)을 구축함에 있어, 최적화 기법을 적용하여 최적화 프로세서인 CPLEX 패키지를 연동시켜 시스템을 구현하였다. Bhargava나 Chou의 웹 기반 시스템과는 달리 ORAWeb에서는 서버의 서비스 모듈에 대한 요청을 자바 소켓에 의해 수행되도록 하였다. 또한 최적화 기법을 적용하기 위해 필요한 모든 모듈들은 자바 어플리케이션으로 구현되었다. 따라서 ORAWeb은 자바 애플릿으로 구현된 전방경계면, 자바 어플리케이션으로 구현된 후방경계면, 그리고 서버의 서비스 모듈에 대한 요청을 자바 소켓 프로그래밍에 의해 구현한 시스템으로써, Bhargava나 Chou가 사용한 CGI에 의한 방법보다 언어적 호환성을 가지므로 시스템의 갱신, 변경, 또는 확장에 보다 융통성을 갖는 시스템이라 할 수 있다.

또한, 최근에 웹 어플리케이션과 객체지향개념의 등장에 따라 자바언어의 급속한 발전이 이루어졌는데, 그 중에서도 기존의 CGI의 많은 제약을 극복한 서블릿(Servlet)이나 서블릿을 기초로 XML과 HTML과 연결된 동적인 웹 페이지를 생성시키는 JSP(JavaServer Pages)를 사용하여 웹 기반 정보시스템을 구축하는 사례가 많다. 하지만 이들만으로는 효과적인 GUI를 구성하기에는 한계를 갖는다. 따라서 ORAWeb을 구축함에 있어 강력한 GUI를 제공하는 애플릿으로 전방경계면을 구현함으로써 수리 분석을 하기에 적합한 GUI를 제공한다.

3. 수리적 분석을 위한 웹 기반 DSS 설계

웹 상에서의 의사결정지원시스템(WDSS: Web-based Decision Supporting System)은 단일 플랫폼(stand-alone)의 DSS가 갖추어야 할 GUI, 모델 기반 하부시스템, 그리고 데이터 매니지먼트 하부시스템과 같은 요소 이외에 인터넷/인트라넷 환경에서의 클라이언트/서버 시스템의 구조를 포함해야 한다. 따라서 클라이언트/서버 시스템의 측면에서 볼 때 전방경계면과 후방경계

면으로 구분할 수 있다(Chou, 1998). 웹 상에서 수리모형의 최적화 기법을 적용하거나, 경험적 접근방법, 시뮬레이션, 또는 통계적 분석 등의 수리 분석을 도입함으로써 구현되는 의사결정을 지원하는 시스템은 일반적인 WDSS와 약간 다른 기능과 구조를 갖게 되는데, 전방경계면은 수리 분석을 하기에 적합한 GUI를 제공하도록 설계해야 한다. 그리고, 후방경계면의 어플리케이션 서버는 수리 분석의 방법론에 따라 이를 위한 대상 시스템의 분석과 설계 모듈을 포함하게 되며 후방경계면에는 필요에 따라 최적화 프로세서, 시뮬레이션 패키지, 또는 통계 패키지 등이 존재할 수 있다.

3.1 전방경계면(Front-end)의 설계

전방경계면은 클라이언트/서버 구조에서 클라이언트의 역할을 하는 부분으로 HTML이나 애플릿의 형태로 존재한다. 전방경계면은 웹 클라이언트들과 일대일로 대화를 함으로써 그들의 서비스에 대한 요청을 받아들이는 접수 창구의 역할을 한다.

웹 클라이언트는 서버 사이트의 수행과는 완전히 분리된 상태로 전방경계면을 통해서만 대화를 하기 때문에 전방경계면에서 다음과 같은 기능 수행을 위한 모듈을 포함해야 한다. 첫 번째, 웹 클라이언트들의 의사결정을 위한 프로젝트의 생성 모듈, 두 번째 의사결정시 필요한 데이터들의 입력 모듈, 세 번째 프로젝트의 실행을 위한 어플리케이션 서버의 호출 모듈, 네 번째 데이터베이스 입출력 모듈, 그리고 마지막으로 결과 및 필요 정보 제시 모듈이 그것이다. 따라서 이러한 기능을 수행하기 위해서 전방경계면은 몇 개의 요소들로 구성된다. 프로젝트 매니지먼트(Project Management), 자바 소켓 인터페이스, JDBC 인터페이스, 그리고 사용자 인터페이스(Graphical User Interface)가 이에 해당되며 각각은 다음과 같은 역할을 수행한다. 우선 프로젝트 매니지먼트는 앞서 설명한 기능 수행을 위한 모듈들을 통제하면서 프로젝트의 전체 수행과정을 총괄하는 역할을 한다. 자바 소켓 인터페이스는 자바 소켓을 사용하여 특정 문자열을 주고 받음으로써 실질적인 DSS 서비스를 수행하는 어플리케이션 서버를 호출하고 서버로부터 서비스 수행에 대한 응답을 받는 역할을 한다. JDBC 인터페이스는 JDBC를 사용하여 프로젝트 생성 및 수행에 필요한 데이터를 후방경계면의 데이터베이스에 저장 또는 갱신하고, 후방경계면의 수행과정에서 생성된 정보 및 데이터를 참조하여 웹 클라이언트에게 제시하는 역할을 한다. 마지막으로 이러한 전체 수행 모듈들은 GUI를 통해 웹 클라이언트와 대화를 하게 된다. 전방경계면의 전체 구조는 <그림 1>에 잘 나타나 있다. 여기서 전문지식이 없는 사용자라 할지라도 프로젝트 생성과 수행을 손쉽게 할 수 있도록 설계하는 것이 중요하다.

3.2 후방경계면(Back-end)의 설계

후방경계면은 클라이언트/서버 시스템에서 서버의 역할을

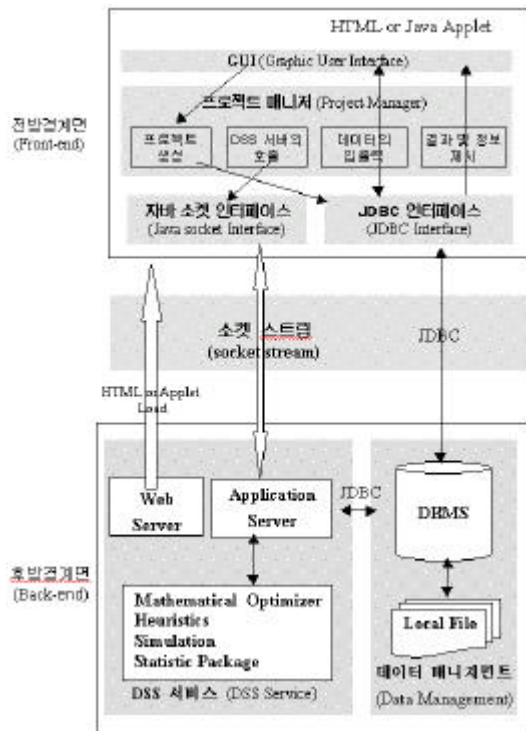


그림 1. 수리적 분석을 위한 WDSS 아키텍처.

하는 부분으로 실질적인 DSS 서비스를 수행하는 DSS 서비스 컴포넌트와 프로젝트의 생성 및 수행에 따른 데이터를 저장하고 관리하는 데이터 매니지먼트 컴포넌트로 구성된다. DSS 서비스 컴포넌트는 크게 웹 서버와 어플리케이션 서버로 구성되며 필요에 따라 최적화 프로세서(Mathematical Optimizer), 시뮬레이션 패키지, 통계 패키지 등의 수리 분석 도구(Mathematical Analysis tools)가 존재할 수 있다. 웹 서버는 HTML이나 애플릿 기반의 전방경계면을 생성하며 애플릿이 다운로드 될 때 데이터베이스 드라이버도 같이 다운로드 시키는 역할도 한다. 어플리케이션 서버는 실질적인 DSS 서비스를 수행하는 부분으로 전방경계면의 자바소켓 인터페이스를 통한 서비스 호출을 받아들이는 서버소켓 구현 모듈과 수리 분석을 통해 실질적인 DSS 서비스를 수행하는 모듈로 구성된다. 위의 수리 분석의 방법에 따라 수리 분석 도구(Mathematical Analysis Tools)가 존재하게 되고 어플리케이션 서버는 이들 수리 분석 도구를 호출하여 실행시키는 역할을 한다. 데이터 매니지먼트 컴포넌트는 프로젝트의 생성과 수행에 따른 데이터들을 저장, 관리한다. 즉 웹 클라이언트에 의한 프로젝트의 생성에 필요한 데이터들과 DSS 서비스 컴포넌트에 의해 프로젝트의 수행시 처리되는 데이터를 DBMS와 로컬 파일 시스템에 저장, 관리하는 역할을 수행한다. 데이터들의 저장과 갱신은 전방경계면의 JDBC 인터페이스를 통해 이루어지기도 하고 DSS 서비스 컴포넌트에 의한 수행 과정이나 결과 데이터의 생성 과정에서 이루어지기도 한다. 두 가지 경우 모두 JDBC를 사용하여 데이터베이스에 접근한다. JDBC를 사용하여 접근하는 방식에는 네 가지 방식이

있는데, 이는 JDBC 드라이버 종류에 따른 방식으로서 (1) JDBC-ODBC 브리지 드라이버, (2) 데이터베이스 API(A Native-API party-Java) 드라이버, (3) 네트워크 프로토콜(A Net-protocol all-Java) 드라이버, 그리고 (4) 데이터베이스 프로토콜(A Native-protocol all-Java) 드라이버를 사용하는 방식이다. 여기서 첫 번째와 두 번째 방식은 클라이언트 쪽에 ODBC 드라이버나 특정 데이터베이스 코드가 존재해야 하기 때문에 웹 기반의 애플릿에서 사용하기엔 부적합하다. 반면 세 번째와 네 번째 방식은 웹 서버로부터 애플릿을 다운로드 받을 때 드라이버도 동시에 같이 다운로드 받게 된다. 그 후에 애플릿이 JDBC API를 호출하면 다운로드 받은 드라이버가 프로토콜로 전환해서 서버에 있는 데이터베이스에 전송하여 접근함으로써 클라이언트의 요청을 처리하게 된다(박지훈, 1997). 애플릿으로 존재하는 전방경계면의 JDBC 인터페이스를 통해 데이터베이스 서버에 접근하여 요청을 처리하려면 세 번째 또는 네 번째 방식을 사용하는 것이 좋다.

4. 장비의 최적 교체 정책 결정을 위한 WDSS 구현

4.1 ORAWeb 시스템의 개요

현대 산업에서 노후화가 일어나는 장비에 대한 최적 교체시점(optimal replacement time)을 찾는 문제는 산업계의 다양한 부문에서 요구되고 있다. 따라서 장비교체문제에 대한 의사결정을 지원하는 시스템을 World-Wide-Web 환경에서 구축한다면 이러한 다양한 부문에서의 분석 요구를 지원할 수 있을 것이다. 본 논문에서는 '각 기간별 예산의 한계(budget constraints)'나 '장비의 제한수명(Life limits)'이라는 제약조건을 갖는 장비의 최적 교체 정책 결정에 관한 분석을 대항해 줄 수 있는 ORAWeb(Optimal Replacement Analysis by Web)의 설계에 대한 것을 WDSS의 구현 사례로 소개한다. ORAWeb은 두 가지 유형의 장비의 최적 교체 정책 결정 문제에 대해 서비스를 제공하는데, 첫 번째는 기간별 단일 도전장비가 존재하는 경우이고(Gillert), 두 번째는 기간별 다수의 도전장비가 존재하는 경우이다(김태현, 1999). 수리 분석을 위한 방법으로는 각 유형에 대해 최적화 프로세서(Mathematical Optimizer)를 사용하여 해결하도록 하였다. 앞서 언급한 두 가지 문제 유형은 동적계획모형에 의해 해결하였는데, ORAWeb에서 사용한 수리 분석은 Karabaki 등(Karabaki, 1994)에 의해 제시된 최소경로문제 모형을 사용하고 각 기간별 예산제약을 추가할 수 있도록 함으로써 이진정수계획 문제로 모형화 하는 방법론을 사용하였다. 다만, 장비의 교체시점 초에 발생하는 교체비, 유희비용, 그리고 이 전 장비의 잔존가치를 교체 바로 전 기간 말에 발생한다고 가정하여 현금흐름의 순 수익을 계산함으로써 모형화하였다(김태현, 1999). 그리고 후방경계면의 최적화 프로세서로 CPLEX 패키지를 사용하였다.

4.2 ORAWeb의 전방정계면

ORAWeb의 전방정계면은 자바 애플릿으로 구현되었다. 전방정계면에 있어서 첫 번째로 설계해야 할 컴포넌트는 전체 장비교체 프로젝트를 통제하고 총괄하는 프로젝트 매니저인 트 컴포넌트이다. 이는 추상적인 컴포넌트로서 다음의 모듈들을 통제한다. 의사결정을 위한 장비교체문제를 정의하는 모듈, 현금요소 데이터 입력 모듈, 프로젝트의 실행을 위한 어플리케이션 서버의 호출 모듈, 후방정계면의 데이터베이스 입출력 모듈, 입력된 현금요소 데이터 중 운영 수익, 운영 유지비, 또는 교체비에 대한 추정 결과 제시 모듈, 최적화 프로세서에 의한 최종 결과 제시 모듈, 그리고 그 밖의 필요정보 제시 모듈이다. 프로젝트 수행과정의 통제는 버튼이나 텍스트 필드 등의 GUI 객체들을 활성(active)상태 또는 비활성(inactive)상태로 만들거나 메시지 창을 통해 메시지를 전달하는 등의 방법을 통해 웹 클라이언트로 하여금 프로젝트를 정해진 순서대로 수행하게 한다. 또한 서버에서 특정 모듈의 수행 시작과 작업 완료 시점은 자바 소켓 인터페이스를 통해 문자열을 주고 받음으로써 알 수 있다.

두 번째로 설계해야 할 부분은 웹 클라이언트(웹 브라우저)에게 사용자 환경(Graphical User Interface)을 제공하는 구조로서 장비교체문제의 설정 및 문제 유형에 따른 옵션 기능을 제공하는 창, 각 장비의 현금요소 데이터 입력 창, 추정 및 결과 제시를 위한 창, 그리고 프로젝트 수행을 위한 창 등을 설계하는 모듈이다. 실제 구현은 자바 애플릿을 사용하였는데, JDK

1.1.X기반으로 작성되었기 때문에 자바가 포함된 브라우저(넷스케이프 4.5, 익스플로워 4.0 등)에서 별도의 플러그인 없이 장착(loading)이 된다. 웹 클라이언트는 웹 서버로부터 초기 애플릿을 다운로드 받음과 동시에 GUI를 구성하는 모든 클래스 파일들을 다운로드 받게 된다.

세 번째는 GUI에 의해 클라이언트로부터 웹 브라우저를 통해 입력받은 데이터를 서버의 데이터베이스에 저장하기 위한 JDBC 인터페이스와 장비 교체 정책 결정을 위한 데이터베이스 스키마 구조를 설계한다. 여기서 입력받은 데이터는 장비 교체문제 설정 및 문제 유형에 따른 옵션 데이터, 각 장비의 종류나 수명 또는 기간에 따른 초기 구입비(purchase cost), 수익(revenue), 유지비 및 수리비(maintenance cost), 잔존 가치(salvage value), 그리고 교체비 및 유휴비용(replacement cost)이다. 애플릿에서 데이터베이스 서버로의 접근은 JDBC 드라이버를 사용하였는데 3.2절에서 설명한 네 번째 방식, 즉 데이터베이스 프로토콜(A Native-protocol all-Java) 드라이버를 사용하는 방식으로 접근한다. 즉, 웹 서버로부터 JDBC API를 사용한 애플릿을 다운로드 받을 때 데이터베이스 프로토콜 드라이버도 같이 다운로드 된다. 그 후에 애플릿이 JDBC API를 호출하면 그것을 특정 데이터베이스의 프로토콜로 전환해서 서버에 있는 데이터베이스에 전송한다. 데이터베이스 서버는 클라이언트의 요청대로 데이터베이스로 접근하여 쿼리를 수행한다. 그리고 수행 결과 값을 클라이언트의 데이터베이스 프로토콜 드라이버에게 전송하면 자바 애플릿은 그 값을 얻을 수 있게 한다. 여기서 데이터베이스는 오라클 8.0을 사용하였고 데이터베이스 프로토콜 드라이버는 오라클사에서 제공하는 오라클 썬 드라이버(Oracle thin driver)를 사용하였다. 예를 들어 오라클 서버의 주소가 '163.152.17.174'이고 포트 넘버가 '1521'이며 오라클 서버의 사용자 계정이 'scott'이고 패스워드가 'tiger'라 하자. 이 오라클 서버에 접근하기 위해 JDBC API를 사용한 애플릿은 우선적으로 오라클 썬 드라이버를 메모리에 올려놓는다. 이 드라이버는 HTML 문서 내 애플릿 태그의 Archive 속성에 의해 애플릿과 동시에 다운로드 받은 것이다. 드라이버가 메모리에 올려지면 JDK 1.1.X의 클래스에 포함되어 있는 드라이버 관리자(DriverManager)에 의해 오라클 썬 드라이버와 오라클 서버를 연결시킨다. 이때 애플릿 내의 JDBC API는 접근하려는 오라클 서버의 IP 주소, 포트 넘버, 사용자 계정, 그리고 패스워드의 정보를 가지고 있다. 따라서 이 정보를 가지고 특정 오라클 서버에 접근할 수 있는 것이다.

<그림 3>의 도스 실행창의 첫 번째 줄은 ORAWeb에서 오라클 썬 드라이버가 메모리에 로드 된 결과를 보여주는 것이고 두 번째 줄은 드라이버 관리자에 의해 오라클 썬 드라이버를 통해 오라클 서버에 접근한 결과를 보여준다. 일단 이렇게 오라클 서버에 연결되면 JDBC API는 SQL 질의를 실행하기 위한 객체를 생성하고 생성된 객체의 메소드를 통해 SQL 문장을 미로소 실행하고 반환 값을 얻을 수 있다.

네 번째는 웹 클라이언트의 애플릿이 실제 DSS 서비스 수행

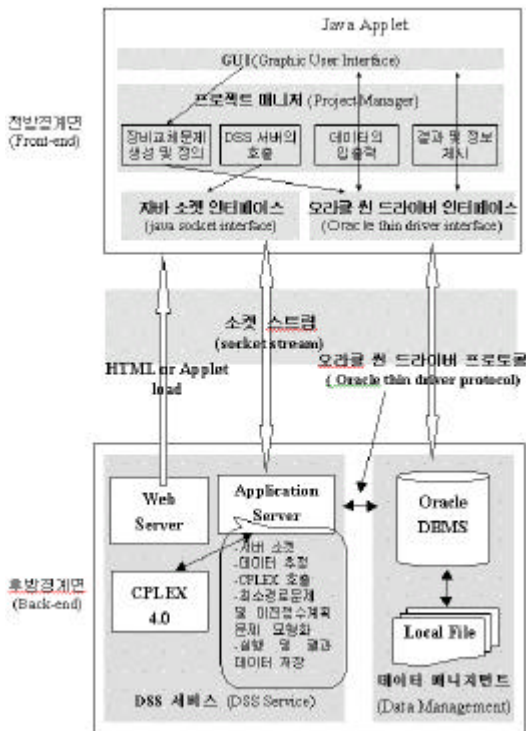


그림 2. ORAWeb의 아키텍처.

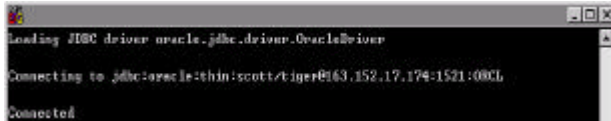


그림 3. 오라클 웹(bin)드라이버에 의한 오라클 서버의 접근.

을 위해 후방경계면의 어플리케이션 서버를 호출하는 자바 소켓 인터페이스의 설계로 자바 스트림 소켓 프로그래밍을 사용하였다(Orfali, 1998). 즉, 웹 클라이언트에서 애플릿을 웹 서버로부터 다운로드 받아 실행시키게 되면 애플릿은 어플리케이션 서버의 서버 소켓 주소, 즉 IP(Internet Protocol) 주소와 포트 번호(Port Number)를 가지고 서버 소켓(server socket)을 찾게 되는데 이미 실행되어 있는 서버 소켓(server socket)을 열고, 클라이언트의 접속을 기다리는 후방경계면의 어플리케이션 서버와 세션을 맺음으로써 전방경계면의 애플릿과 후방경계면의 어플리케이션 서버 사이에 소켓 통신을 하게 된다. 이렇게 세션을 맺게 되면 전방경계면과 후방경계면은 문자열을 주고 받을 수 있는데, 프로젝트의 수행시 특정 문자열을 전송하고 이를 인식하여 전방경계면이나 후방경계면의 특정 메소드를 실행하도록 하였다. 애플릿이 제거(destroy)되면, 세션도 해제된다.

마지막으로 결과 제시 모듈의 구현으로써 DSS 서비스 컴포넌트의 최적화 프로세서인 CPLEX에 의한 결과가 저장된 오라클 서버로부터 결과를 읽어 분석하고 해석하여 전방경계면의 애플릿 상에서 다시 웹을 통해 웹 클라이언트에게 제공하도록 하는 구조이다. 여기서 웹 클라이언트에게 결과에 대한 해석을 용이하게 제공하기 위해 각 기간별로 장비의 종류, 정책, 소요 비용, 그리고 기간별 예산 제약의 정보를 테이블 형식으로 보여주고 최적 정책은 네트워크 구조의 그림으로 나타낸다.

4.3 ORAWeb의 후방경계면

후방경계면의 어플리케이션 서버는 자바 어플리케이션으로 구현되었으며 클라이언트/서버 시스템 구조는 자바 소켓 프로그래밍에 의해 형성되었다.

DSS 서비스 컴포넌트 : DSS 서비스 컴포넌트는 크게 웹 서버와 어플리케이션 서버, 그리고 수리 분석 도구인 최적화 프로세서로 구성된다. 웹 서버는 초기 애플릿을 포함한 HTML 문서에서 애플릿 태그의 Archive 속성값으로 지정된 오라클 웹 드라이버와 전방경계면을 구성하는 클래스 파일들을 애플릿과 함께 다운로드 받도록 한다. 이때 ORAWeb의 전방경계면이 비로소 생성되며 다운로드 받은 드라이버를 통해 전방경계면에서 오라클 서버로의 접근이 가능하게 한다. ORAWeb의 어플리케이션 서버의 모든 모듈은 자바 어플리케이션으로 구현되었는데, 구현의 첫 번째 단계는 서버 소켓(server socket) 프로그래밍이다. 웹 클라이언트가 애플릿을 웹 서버로부터 다운로드 받아 애플릿이 실행되면 애플릿은 서버 소켓(server socket)의

IP(Internet Protocol) 주소와 포트 번호(Port Number) 정보를 가지고 있어서 서버 소켓을 찾아 세션을 맺을 수 있다. 이때 서버 소켓은 이미 실행되어 클라이언트의 접속을 기다리고 있는 것이다. 이렇게 세션을 맺게 되면 소켓 스트림으로 문자열을 전송할 수 있는데, 전방경계면에서 특정 문자열이 전송되면 이를 인식하여 어플리케이션 서버의 특정 메소드를 실행시킴으로써 최적화 프로세서를 실행시킬 수도 있고 데이터 매니지먼트 컴포넌트를 갱신할 수도 있다. 또한 반대로 어플리케이션 서버에서 서버 소켓을 통해 이미 세션을 맺은 전방경계면으로 문자열을 전송하여 특정 메소드를 실행시킬 수 있다. 따라서 서버 소켓의 설계는 전방경계면에서 웹 클라이언트의 서비스 요청에 대해 이를 받아들이고 응답을 할 수 있도록 한다는 면에서 중요하다. 두 번째는 오라클 서버에 저장된 운영 수익, 운영 유지비, 그리고 교체비 데이터를 어플리케이션 서버에서 읽어서 회귀분석을 통한 추정을 하고 추정된 데이터를 다시 오라클 서버와 로컬 파일에 저장하는 모듈의 설계이다. 로컬 파일에 저장된 데이터들은 그래프의 형식으로 보여주는데 각 현금요소에 대해 각 장비별로 실제 운영 데이터 그래프와 추정 데이터 그래프의 비교가 가능하게 하였다. 그리고 오라클 서버에 저장된 데이터들은 테이블의 형식으로 웹 클라이언트에게 보여주는데 추정된 데이터 값을 하나하나 모두 볼 수 있다. 데이터 추정은 최소제곱법(method of least squares)을 사용한 1차 회귀분석과 2차 회귀 분석을 사용한다. 두 가지 중 웹 클라이언트는 선택하여 추정을 할 수 있다. 세 번째는 웹 클라이언트에 의해 생성된 장비교체문제에 대해 입력받은 실제 운영 데이터나 추정된 데이터로 최소경로문제나 이진정수계획 문제로 수리모형을 만드는 모듈의 설계이다. 웹 클라이언트에 의해 문제의 옵션 기능으로 제공되는 각 기간별 예산 제약을 추가하는 경우에 이진정수계획 문제로 모형화 된다(김태형, 1999, Karabakci, 1994). 수리 모형화는 최적화 프로세서로 사용한 CPLEX 4.0 패키지가 인식할 수 있는 lp포맷으로 만들어진다. 네 번째는 세 번째 모듈에서 만든 lp포맷을 서버 사이트의 CPLEX가 인식하여 실행하도록 하고 실행 결과를 오라클 서버에 저장하는 모듈의 설계이다. 어플리케이션 서버에서의 오라클 서버의 접근은 서버 내에 존재하는 오라클 웹(bin) 드라이버에 의해 이루어진다.

두 번째, 세 번째, 그리고 네 번째 모듈 모두 서버 소켓을 통해 웹 클라이언트의 요청을 받아들이고 이를 어플리케이션 서버의 특정 메소드를 실행함으로써 요청에 대한 응답을 하는 모듈이다.

현재 ORAWeb 구현에 사용된 CPLEX는 일인용(1-user) CPLEX를 사용하였기 때문에 다수의 웹 클라이언트가 동시에 접근하여 서비스를 받는 것은 사실상 불가능하다. 다만 CPLEX가 이미 실행중에 있는 경우에는 대기 메시지를 보내주거나 다중 사용자(multi-user) CPLEX를 사용함으로써 이를 극복할 수 있다. 또한 각 웹 클라이언트의 접속 시점에서 사용자 인증 과정을 통해 고유 데이터베이스 공간이나 로컬 파일을 할당함으로써 데이터

트랜잭션에 따른 데이터의 무결성을 유지할 수 있다.

데이터 매니지먼트 컴포넌트: 데이터 매니지먼트 컴포넌트는 장비 교체에 대한 의사결정 지원을 요구하는 웹 클라이언트에 의해 생성되는 프로젝트에 대한 데이터들을 관리한다. 즉 프로젝트 설정과 실행에 필요한 데이터들을 DBMS인 오라클과 로컬 파일에 저장, 갱신함으로써 프로젝트를 수행한다. 전방경계면에서의 데이터들의 저장과 갱신은 다운로드 받은 데이터베이스 프로토콜 드라이버인 오라클 씬 드라이버(Oracle thin driver)를 통해 직접 이루어지고 후방경계면에서 DSS 서비스 컴포넌트에 의한 수행 과정에 따른 데이터의 저장과 갱신은 서버 내에 존재하는 오라클 씬 드라이버를 통해 이루어진다. 오라클 서버에 존재하는 테이블은 프로젝트 설정과 실행을 위한 MASTER 테이블, 실제 운영 현금요소 데이터를 저장하기 위한 PURCHASE, REVENUE, MAINTENANCE, SALVAGE, REPLACEMENT 테이블, 추정 데이터와 예제 데이터를 저장하기 위한 테이블(EST_MAINTENANCE, EX_MAINTENANCE 등)이 존재하며 CPLEX 수행 결과 저장을 위한 RESULT 테이블이 있다. 로컬 파일시스템은 각 현금요소에 대한 추정 데이터를 그래프로 도시하기 위한 파일들과 수리 모형을 위한 lp 포맷 파일 그리고 CPLEX 결과 저장을 위한 파일로 구성된다.

4.4 ORAWeb의 구현

ORAWeb은 다음의 모듈별로 구현하였는데, 구현 순서는 <그림 4>와 같다. 여기서 첫 번째 모듈과 다섯 번째 모듈부터 일곱 번째 모듈까지는 후방경계면에 대한 구현에 해당되며, 두 번째 모듈부터 네 번째 모듈, 그리고 마지막 모듈은 전방경계면의 구현에 해당된다.

4.5 ORAWeb의 적용가능 문제 유형

기간별 단일 도전장비가 존재하는 경우: 본 문제 유형은 유한 수명의 단일 현유 장비에 단일 도전장비가 존재하는 경우의 교체 정책에 관한 문제이다. 본 모형의 특징은 매 의사결

정기간마다 서로 다른 하나의 도전장비가 존재한다는 것이다. 즉, 계획기간의 수와 도전장비의 수가 같게 된다. 또한 장비의 교체 시점에서 교체 이전 장비에 대한 처분 비용을 의미하는 교체비를 고려한다. 이러한 현금요소 흐름의 특성 때문에 Gillette는 동적계획모형으로 모형화 하여 이를 해결하려 하였다. 본 문제 유형은 Gillette의 문제 유형에 장비의 구입시점에서 구입비, 그리고 장비의 노후화에 따른 감가상각을 현금요소의 고려 대상으로 추가한 형태이다.

기간별 다수의 도전 장비가 존재하는 경우: 본 문제 유형은 유한 수명의 단일 현유 장비에 복수 도전장비 교체 정책에 관한 문제를 다루었던 기존 Oakford, Lohmann, Salazar(1984)의 문제 유형과 유사하다. 다만, Oakford(1984) 등이 고려한 장비의 감가상각으로 발생하는 세금 절감의 효과를 배제한 대신에 장비 운영에 따른 계절별 가동률과 유희비용, 장비의 제한 수명 등을 고려한 김태형(1999) 등의 문제 유형을 ORAWeb의 분석 대상으로 한다. 여기서 교체 기간 동안에 발생하는 휴차에 따른 기회 손실 비용인 유희비용 등의 현금요소를 고려하는데 이를 위해 김태형(1999) 등은 국소 최적해를 찾을 가능성을 줄이기 위해 후향 동적 계획 문제로 모형화 하였다.

ORAWeb은 이러한 현금요소의 특성을 아크의 길이에 반영함으로써 Karabaki 등에 의해 제시된 최소경로문제(Karabaki, 1994)로 모형화하고, 각 기간별 예산계약을 추가할 수 있도록 함으로써 아진정수계획 문제(Karabaki, 1994)로 모형화 하는 방법론을 사용하였다. 다만, 장비의 교체 시점 초에 발생하는 교체비나 유희비용, 그리고 이전장비의 잔존가치를 교체 바로 전 기간 말에 발생한다고 가정하여 현금요소의 순 수익을 계산함으로써 모형화 하였다(김태형, 1999).

4.6 ORAWeb의 수행 순서에 따른 예시

- 첫 번째 단계 : ORAWeb의 전방경계면 생성

우선 후방경계면의 웹 서버와 어플리케이션 서버를 실행시킨다. 웹 서버는 웹 클라이언트에게 아래 <그림 5>의 애플릿 창을 포함한 HTML 문서, 전방경계면을 구성하는 모든 클래스 파일, 그리고 오라클 씬 드라이버를 다운로드 받을 수 있도록 한다. 웹 서버의 URL 주소는 http://163.152.17.174이다. 어플리케이션 서버에는 서버 소켓 프로그래밍이 포함되어 있다. 따라서 이를 실행시킴으로써 서버 소켓을 열어 클라이언트의 접속을 기다린다. 여기서 소켓의 주소에서 IP 주소는 웹 서버와 동일한 163.152.17.174이고 포트 넘버는 9999이다. <그림 5>의 애플릿에는 소켓의 주소, 즉 서버 소켓의 IP주소와 포트 넘버의 정보를 가지고 있어서 다운로드 받은 애플릿이 실행됨과 동시에 소켓의 주소 정보를 가지고 후방경계면의 어플리케이션 서버와 세션을 맺게 된다.

<그림 5>의 오른쪽 제일 윗창에 나타난 HanServer는 어플리케이션 서버이다. HanServer의 도스 실행 창과 애플릿의 자바

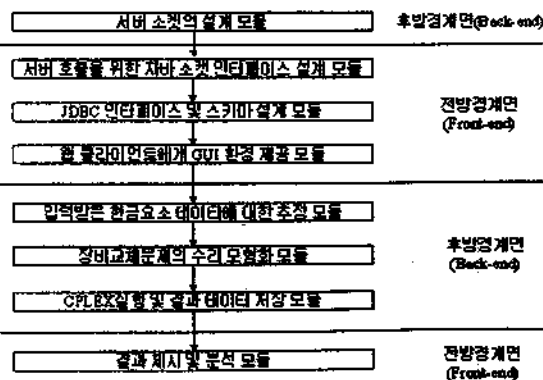


그림 4. ORAWeb의 구현 순서.

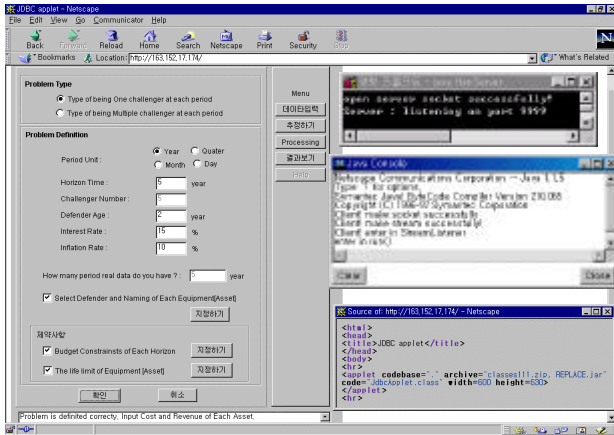


그림 5. 프로그램 예시(1): ORAWeb의 기본 정보 입력 창(예인 메뉴).

큰술을 보면 후방계면과 전방계면 사이에 소켓 세션을 맺었음을 알 수 있다. 그리고 오른쪽 제일 아래 창의 HTML 문서의 소스 중 애플릿 태그에서 classes111.zip이 바로 오라클 쉐드 라이브를 구성하는 클래스 파일이며 REPLACE.jar는 전방계면을 구성하는 클래스 파일에 해당된다.

• 두 번째 단계 : 수리 분석을 위한 장비교체 프로젝트 생성
 <그림 5>의 기본 정보 입력 창에서 다음을 입력하여 장비 교체문제를 정의함으로써 장비교체 프로젝트를 생성하고 시작한다.

- 장비교체문제의 유형 즉, 단일 도전장비가 존재하는 경우와 다수의 도전장비가 존재하는 경우의 두 가지 문제 유형 중 하나를 선택한다.
- 기간의 단위, 계획기간, 장비의 수, 현유장비의 나이, 시장이율, 인플레이션율을 입력한다.
- 현유장비를 정하고 각 장비의 이름을 입력한다.
- 필요에 따라 각 기간에서의 예산에 대한 제약과 각 장비의 제한수명을 입력한다.

입력이 끝나면 확인 버튼을 누른다. 이 때 다운로드 받은 오라클 쉐드(tbin) 드라이버(classes111.zip)를 사용하여 위의 장비교체문제 정의에 필요한 데이터가 후방계면의 오라클 서버에 직접 저장되고 프로젝트가 생성된다.

• 세 번째 단계 : 현금요소 데이터를 입력받는다.
 각 장비의 나이 및 기간별로 실제 운행하여 얻은 현금요소 데이터를 입력한다. 현금요소는 장비의 구입시점에서의 구입비, 장비의 운영시의 운영 수익과 운영 유지비, 장비의 노후화에 따른 잔존가치, 그리고 장비의 교체시점에 발생하는 이전장비의 교체비용이나 유희비용을 포함한다. 입력된 데이터는 후방계면의 Oracle Server에 직접 저장된다. 문제 유형에 따라

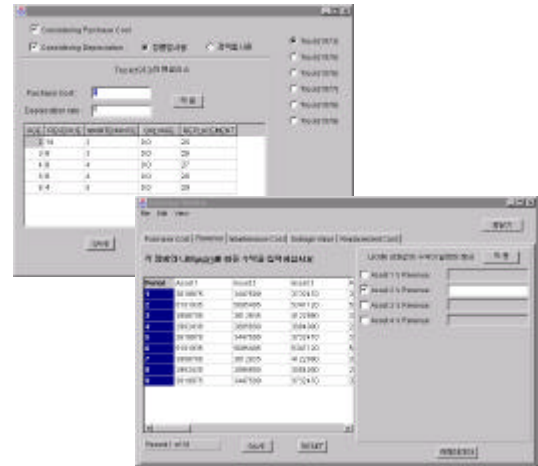


그림 6. 프로그램 예시(2) : ORAWeb의 현금요소 데이터 입력 창(Data Input Window-Salvage tab).

현금요소 데이터 입력 창이 서로 다르다.

• 네 번째 단계 : 데이터 추정

현금요소 데이터 입력 창으로 입력된 장비의 운영 수익, 운영 유지비, 그리고 교체비 중에 데이터 추정이 필요한 현금요소를 선택하고 추정을 하는 단계이다. 데이터 추정이 필요 없으면 이 단계는 무시하고 넘어간다.

<그림 7>에서 보듯이 추정방법을 1차 최귀나 2차 최귀방법 중에서 하나를 선택하고 추정하고자 하는 현금요소를 선택한다. 추정을 하지 않은 현금요소는 현금요소 데이터 입력 창에서 입력된 데이터를 수리모형화에 그대로 적용한다. 추정시작 버튼을 클릭하면 세션을 맺은 스트림 소켓을 통해 'Estimate'라는 문자열을 전송한다. 그러면 후방계면의 어플리케이션 서버(Han Server)는 서버 소켓을 통해 문자열 'Estimate'를 인식하고 이 때 어플리케이션 서버의 추정 메소드를 수행하게 된다. 추정이 끝나면 어플리케이션 서버에서 추정된 데이터를 Oracle Server에 저장한다. 모든 작업이 끝나면 이번엔 반대로 스트림 소켓을 통한 문자열 전송으로 전방계면에 작업이 끝났다는 메시지를 전달한다.

• 다섯 번째 단계 : 수리 분석을 위한 모형화 및 CPLEX 실행
 현금요소 데이터 입력 및 데이터 추정이 끝나면 <그림 5>

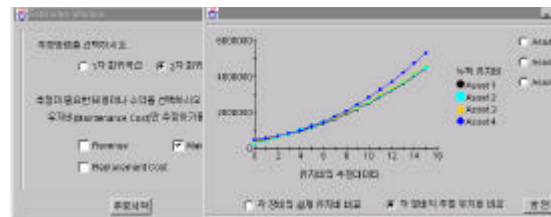


그림 7. 프로그램 예시(3) : ORAWeb의 추정 창 (Estimates Window).

```

Maximize
obj: -19 X001 - 15 X002 - 12 X003 - 11 X004 + 17 X005 - 34 X101
- 21 X102 - 8 X103 + 3 X104 + 38 X105 - 7 X212 + 4 X213 + 15
X214 + 50 X215 - 5 X323 + 8 X324 + 46 X325 - 5 X434 + 32 X435
+ 19 X545
Subject To
e1: X001 + X002 + X003 + X004 + X005 + X101 + X102 +
X103 + X104 + X105 = 1
e2: - X001 - X101 + X212 + X213 + X214 + X215 = 0
e3: - X002 - X102 - X212 + X323 + X324 + X325 = 0
e4: - X003 - X103 - X213 - X323 + X434 + X435 = 0
e5: - X004 - X104 - X214 - X324 - X434 + X545 = 0
e6: - X005 - X105 - X215 - X325 - X435 - X545 = -1
End
    
```

그림 8. CPLEX가 인식할 수 있는 lp 포맷.

의 기본 정보 입력 창의 Processing 버튼을 클릭하여 최적화 작업을 한다. 추정에서와 마찬가지로 서버 소켓을 통해 'Optimize' 라는 문자열을 전송하면 어플리케이션 서버에서는 문제 정의 데이터와 현금요소 데이터를 가지고 최소경로문제나 이진정수계획문제로 모형화하여 CPLEX 패키지가 인식할 수 있는 포맷의 파일(*.lp)을 만든다. 그리고 CPLEX 패키지가 이 파일을 읽어 최적화 작업을 하도록 한다. 아래 <그림 8>은 모형화 모듈에 의해 CPLEX 패키지가 인식할 수 있는 LP 포맷 파일이다. 작업이 끝나면 실행 결과를 분석, 해석하여 데이터를 Oracle Server에 저장한다. 모든 작업이 끝나면 스트림 소켓을 통한 문자열 전송으로 전방경계면에 작업이 끝났다는 메시지를 전달한다.

• 여섯 번째 단계 : 최적 정책 결과 제시

<그림 5>의 기본 정보 입력 창의 결과보기 버튼을 클릭하면 Oracle Server로부터 직접 결과 데이터를 읽어 들여 전방경계면의 결과 창에 결과를 보여준다. 이때 사용자들이 분석을 손쉽게 하기 위해 <그림 9>와 같이 테이블 형식 및 네트워크 그림으로 나타냄으로써 최적 정책 결과를 보여준다.

5. 결론 및 향후 연구과제

이 연구에서는 인터넷이나 인터넷 환경에서 수리모형의 최적화 기법을 적용하거나, 경험적 접근방법, 시뮬레이션, 또는 통계적 분석 등의 수리 분석을 도입함으로써 의사결정을 지원하는 시스템을 설계하였다. 따라서 전방경계면과 후방경계면은 기능과 구조적 특징을 갖게 되는데, 전방경계면은 수리 분석을 위해 적합한 GUI를 제공하도록 설계해야 하며 후방경계면의 어플리케이션 서버는 수리 분석의 방법론에 따라 이를 위한 대상 시스템의 분석과 설계 모듈을 포함하게 된다. 또한 후방경계면에는 필요에 따라 최적화 프로세서, 시뮬레이션 패키지, 또는 통계 패키지 등이 존재할 수 있다. 따라서, 이를 중심으로 한 시스템의 구조 설계에 초점을 두고, 이에 대한 구현

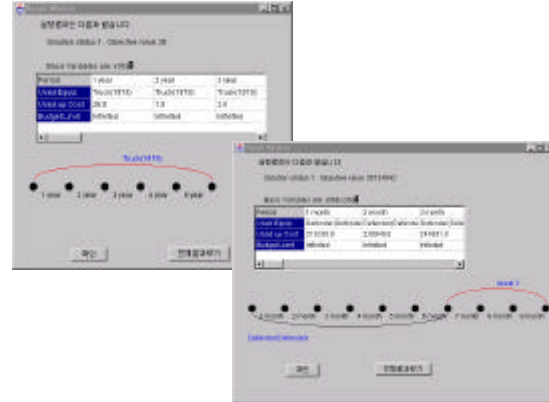


그림 9. 프로그램 예시(4) : ORAWeb의 결과 창(Result Window).

사례로서 장비의 최적 교체 정책 결정을 위한 웹 상에서의 DSS를 구축하였다.

본 시스템의 구현에 있어서 서버와 클라이언트 사이의 통신은 분산 컴퓨팅 기술 중에서 자바 소켓을 사용하였고 데이터베이스는 JDBC를 사용하여 직접 접근하였는데, 최근의 분산 객체 기술, 즉 CORBA, COM, JavaBeans를 사용하여 시스템을 구축한다면 보다 유연하고 동적인 분산 시스템이 될 것이다.

참고문헌

김태형, 김승권 (1999), 다수의 도관 장비론재시 설비의 경제적 수명과 최적 교체 결정을 위한 동적 계획 모형, *대한산업공학회지*, 25(4).

김태형 (1999) 수리적 분석을 위한 웹 기반 의사결정 지원시스템의 구현과 설계 -장비의 최적 교체 정책 결정을 위한 WDSS 구현을 중심으로-, *고려대학교 석사학위 논문*.

박지훈 (1997) *네트워크 컴퓨팅의 핵심 기술 최신자바 1.1*, 127-137, 사이버출판사.

Bhargava, Heunant K. and Tettebach, Clay(1997), A WEB-BASED DECISION SUPPORT SYSTEM FOR WASTE DISPOSAL AND RECYCLING, *Comput., Environ. and Urban Systems*, 21(1), 47-65.

Chou, Seng-cho T. (1998), Migrating to the Web: a Web financial information system server, *Decision Support Systems*, 23, 29-40.

Dataramp, <http://www.dataramp.com>.

Fouer, R., Gay, D., and Kernighan, B. (1993), *AMPL: A modeling language for mathematical programming*, Duxbury, MA: Boyd & Fraser.

Gillett, Billy E., *OR Application*, McGraw-Hill Series in industrial Engineering & Management Science, 51-55.

Intanets Redefine Corporate Information Systems(1996), http://home.netcape.com/campbellat_work/pubs/paper/intanets.html.

Karabakal, N., Lohmann, J. R., and Bean, J. C. (1994), Parallel Replacement under Capital Rationing Constraints, *Management Science*, 40, 305-319.

Oakford, R. V., Lohmann, J. R., and Salazar, A. (1984), A Dynamic Replacement Economy Decision Model, *IIE Transactions*, 16(1), 86-96.

Orfali, R., and Hankey, D. (1998), *Client/Server Programming with JAVA and CORBA*, Second Edition, John Wiley & Sons, Inc., 265-486.



김승권

서울대학교 기계공학과 학사
Stanford University 산업공학과 석사
UCLA 시스템공학과 박사
현재: 고려대학교 산업공학과 교수
관심 분야: 경제성 분석, 생산 및 설비 관리의
DBMS를 활용한 의사결정지원 시스템 개발,
SCM, 수자원시스템 분석 등의 시스템 분석
기법의 현실적인 적용



김태형

인하대학교 산업공학과 학사
고려대학교 산업공학과 석사
현재: (주)코아링크에서 B2B 전자상거래 e-Procurement Solution 개발중
관심분야: Web Application 설계 및 구현,
Supply Chain Management & e-Commerce