

# 소프트웨어 공동 개발 환경을 위한 웹 버전 관리 시스템

## Web Version Management System for Software Cooperation Development Environment

김 수 용\*  
Soo-Yong Kim

최 동 운\*\*  
Dong-Oun Choi

### 요 약

본 논문에서는 UML 편집기에서 생성된 mdl 형태의 파일을 웹 상에서 공유하기 적합한 XML 형태로 변환하여 관계형 데이터베이스에 저장하고, 각종 설계 정보들을 웹 환경에서 팀들간 공동 작업을 위해서 웹 브라우저를 통해 참조하고, 파악되도록 하기 위한 방안을 기술한다. 또한 웹 기반의 소프트웨어 공동 개발 환경에서 얻어진 많은 소스코드와 실행파일 그리고 오브젝트파일들을 프로젝트 팀들간에 공유할 수 있는 환경을 지원하는 웹 기반의 시소러스를 이용한 검색 시스템을 지원하며, 이들 설계정보들 간의 관계를 웹 상에서 유기적으로 향해할 수 있는 항해 도구 시스템에 관한 연구이다.

### Abstract

This paper is to describe how to refer to software objects and reuse them through the web browser so that all the members of the software development team can collaborate with each other on the development procedure. In addition, we describe how to convert the \*.mdl design information generated from the UML editor into their corresponding XML data, which is arranged to be saved in the relational database system. Furthermore, we provide a facet retrieval system based on which makes use of a object-oriented thesaurus, which supports an integrated environment through which all the project team members can share a lot of source codes and execution files as well as object files produced from the web-based collaborative development environment. Finally, we have designed and implemented a web version management system that facilitates software developers to manage in their web-based search for the relationship of design informations.

Keyword : UML(Unified Modeling Language), XML(extensible Markup Language)

## 1. 서 론

최근 대형 소프트웨어 개발 방법론으로 실세계를 자연스럽게 모델링할 수 있고, 인터넷 기반의 네트워킹, 멀티미디어화, 분산화 등과 같이 복잡한 소프트웨어 개발환경을 통합적으로 지원할 수 있는 객체 지향 소프트웨어 개발 방법론이 개발되었다. 객체 기술에 관한 국제 표준화 기구인 OMG(Object Modeling Group)에서 UML(Unified Modeling Language)[1,2,3,4]을 표준 객체지향 모델링 언어로 채택함에 따라 여러 부문에서 이 방법론을 적용

하여 소프트웨어가 개발되고 있다.

현재까지 많은 객체 지향 CASE 도구들이 사용되고 있지만 이들은 지리적으로 인접한 지역에서 독립적으로 소프트웨어 시스템을 개발하는 과정에서 발생하는 설계 정보만을 제어하고 있기 때문에 분산 환경에서 소프트웨어를 개발하는 팀들간 정보를 공유하고, 관리하기에 많은 어려움을 느낀다. 그래서 본 논문에서는 소프트웨어 개발 전체 주기에서 발생하는 버전을 웹 기반에서 개발자들 간에 정보를 공유하기 위한 방법론에 대한 연구이다.

웹은 사용이 용이하고 여러 종류의 자료를 표현하고 전송하기 위한 표준화된 방식을 제공하기 때문에, 소프트웨어 및 하드웨어 플랫폼 종속적인

\* 정 회 원 : 서남대학교 전자계산소 전임교수  
provibe@empal.com(제 1저자)

\*\* 종신회원 : 서남대학교 컴퓨터 정보통신학과 교수  
cdo@seonam.ac.kr(공동저자)

계약들을 극복하여, 정보의 공유와 전달에 효과적인 수단으로 사용되고 있다. 그러나, 점차 방대해지는 정보의 크기로 인해 이들을 구조화하고 이 정보를 효율적으로 서비스하기 위한 요구가 대두되었다. 그 결과, 데이터베이스 시스템이 갖는 데이터의 일관성 보장, 회복, 동시성 제어, 표준 질의어 제공, 백업 등과 같은 기능들을 웹이 가질 수 있도록 하기 위해, 웹에 데이터베이스 시스템을 연동하여 사용자는 웹 브라우저를 통해 데이터베이스에 질의하고, 그 결과를 다시 웹을 통해 볼 수 있도록 하기 위한 연구들이 수행되었다[5, 6].

한편, 객체지향 소프트웨어 개발 방법론은 개발자가 목적 시스템을 작성하기 위한 여러 개념적 도구들을 제공한다. 개발자들은 이 도구들을 사용함으로써 보다 강한 표현력과 검증된 소프트웨어 생산물을 작성할 수 있다. 고안된 소프트웨어 생산물은 소프트웨어 개발 과정에서 고려된 매우 다양하고 풍부한 의미들을 내포하고 있는데, 이들은 다수의 그래픽 도구들을 통해 작성되고 보여지게 된다[7].

그러나, 객체 모델링 기법과 실제 구현될 객체지향 프로그래밍 언어간에는 많은 불일치 문제가 있으며, 구현 코드와 설계 내역들을 서로 연관시키고, 참조할 수 있는 수단을 제공하고 있다. 그러나, 이들 도구들은 모두 하나의 호스트 위에서 운영되며, 이기종 혹은 원거리에서 접근하는 개발자에게는 이 정보들에 접근할 수 있는 수단을 제공하지 않고 있다. 본 연구팀은 웹을 기반으로 한 분산 소프트웨어 환경에서 하나의 프로젝트를 수행하는 과정에서 발생하는 다양한 설계정보들을 공유할 수 있는 웹 기반의 항해 모델을 이용하여서 팀들간에 설계정보를 유기적으로 항해할 수 있는 항해 모델을 설계하였다. 그럼으로써, 개발자들 간의 공동 작업을 원활하게 하기 위한 다양한 기능들을 제공하게 될 것이다. 본 연구의 최종 목적은 기존의 원시 코드 중심의 버전 관리 시스템에 원시 코드 이전 단계인 UML을 기반으로 한 소프트웨어 개발 환경의 전체 개발 주기에서 발

생하는 다양한 설계 객체들을 웹 상에 분산된 개발 팀들 간에 공유하기 위해 웹 기반의 항해 도구를 개발하는 것이다. 본 연구팀은 웹을 기반으로 한 분산 소프트웨어 환경에서 하나의 프로젝트를 수행하는 과정에서

발생되는 원시 코드 중심의 버전뿐만 아니라 원시 코드 이전 단계의 UML기반의 설계 객체들을 XML 파일 형식으로 변환하여 관계형 데이터베이스에 저장하여 객체지향 설계 내역과 원시 코드를 웹 브라우저를 통해 참조하고, 분석함으로써 웹 상의 분산 환경에서 개발 팀들 간의 공동 작업이 가능하고, 또한 소프트웨어 객체를 공유할 수 있는 항해 도구를 개발하였다.

본 논문에서 전개될 내용은 다음과 같다. 먼저 2장에서는 관련연구를 살펴보고, 3장에서는 웹 기반의 항해 모델 전체적인 구조를 기술하고, 4장에서는 웹 기반의 항해 모델을 구현한 내용에 관해서 기술한다. 마지막으로 5장에서는 결론 및 향후 연구과제를 논의한다.

## 2. 관련 연구

### 2.1 기존의 검색 시스템

소프트웨어를 재사용하는 과정에서 소프트웨어 부품을 체계적으로 분류하여 관리하고, 사용자가 적합한 부품을 검색하는 과정은 소프트웨어 개발자의 생산성을 향상시키는 데 매우

중요하다. 이때 부품의 검색 방법은 등록된 부품의 분류 방식에 매우 의존적인데, 주로 사용하는 방법으로는 색인에 의한 방법, 열거형 방법, 그리고 패킷 방법 등이 있다[8].

색인에 의한 방법은 한 시스템의 기능과 특성을 잘 반영한 시스템 명세서나 요구 분석문과 같은 문서로부터 정보를 대표할 수 있는 색인어를 추출하여 정보를 대표하는 기술자로 매핑시키는 방법이다. 대표적인 시스템으로 GURU[9]은 자연어를 기술된 시스템 메뉴얼로부터 색인어를 자동

으로 추출하여 정보를 역파일의 형태로 분류하고 있다. 그러나, 이와 같이 색인을 기반한 분류 방법은 방대한 자연어 문서로부터 정보의 특성을 잘 반영할 수 있는 적합한 색인어 선정이 어렵고, 색인어의 의미적 관계를 파악할 수 없기 때문에 정보를 의미 관계에 따라 체계적으로 분류할 수 없어, 색인어와 탐색어 사이의 의미적 불일치성을 해결할 수 없다. 이러한 단점은 사용자가 신규 소프트웨어에 적용하기 위해 검색된 부품을 이해하는데 많은 시간과 노력을 요구한다. 이러한 단점들을 해결하기 위해 시소러스와 같은 지식(knowledge)을 이용한 검색 방법에 관한 연구가 여러 곳에서 진행 중에 있다. 대표적인 연구로 AT&T Bell 연구소에서 개발한 LaSSIE(Large Software System Information Environment)[10]에서는 부품에 대한 색인어와 의미적으로 관련된 개념들을 시소러스로부터 추출하여 부품의 기술자로 간주하고 있다. 그러나, 색인 과정에서 시소러스의 이용은 시소러스가 변함에 따라 방대한 부품에 대해 빈번히 색인 작업을 다시 해야만 하는 단점을 가지고 있다.

분류에 의한 검색 방법에는 열거형(enumerated) 분류와 패시(facet)분류로 나눌 수 있다[11]. 열거형 부품 분류 방법에서는 먼저 전체 부품에 대한 응용 도메인 영역을 정교하게 분석하여 계층적으로 분할한 다음에 각 부품들을 해당 영역에 적절히 할당하여 분류하는 방법이다. 이 방법은 실제로 부품들이 등록되기 전에 미리 모든 응용 영역을 세밀하게 예측하여 분화하여야 한다. 빠른 속도로 변화하는 소프트웨어 개발 분야의 특성 때문에 신규로 추가되거나 삭제되는 영역이 반드시 발생하며, 이에 대한 처리가 필수적이다. 그러나, 정적으로 고정된 영역을 다시 수정·분할하여 부품을 분류해야 하는 단점을 가지고 있다[12]. 패시 분류 방식은 열거형 분류 방식의 단점을 개선하기 위해 재사용 부품들이 갖는 공통적인 특성(기능, 응용 영역, 사용 환경, 프로그램 언어등)을 패시로 정의하고 각각의 패시에 부품의 특성을 반영한 패시값(term: 개념, 용어)을 표현하는 방식이

다. 이 분류 방식은 부품에 대한 여러 속성들을 표현할 수 있는 구조로, 분류되는 부품에 따라 유동적인 특성을 반영할 수 있으며 분류가 간단하며 이해하기 쉽고, 확장이 용이한 장점이 있다. 패시 분류 방법에서는 부품의 일반적인 속성을 패시로 표현하고, 각 부품에 대한 패시들에 적절한 값을 부여함으로써 부품을 분류하는 방법이다. 패시 방법은 부품을 체계적으로 분류할 수 있고, 순차적으로 분류할 수 있다는 장점을 가지고 있다. 또한 도메인 영역을 하나의 패시로 열거형 방법을 수용할 수 있다. 대표적인 시스템인 RSL(Reusable Software Library)[13]은 Ada로 구현한 재사용 부품(함수, 프로시저, 패키지 등)을 패시 분류 방법으로 분류한다.

Prieto-Diaz[14] 시스템은 패시 분류 방식(faceted classification scheme)에 기반한 소프트웨어 재사용 라이브러리이다. 이 시스템에서는 재사용 가능한 소프트웨어 부품의 검색, 추출 기능, 목록작성(cataloging)이나 분류같은 라이브러리언(librarian) 기능을 지원한다. 그러나 Prieto-Diaz의 방법은 확장성은 좋으나 다음과 같은 단점들을 가지고 있다. 첫째, 사용자의 검색 의도를 반영하는 패시 질의가 단지 몇 개의 패시 값을 합성하여 표현되기 때문에 사용자의 다양한 요구를 표현하기 어렵다. 둘째, 등록 과정에서 사용된 패시 값과 질의에 표현된 패시 값을 합성하여 표현되기 때문에 사용자의 다양한 요구를 표현하기 어렵다. 셋째, 등록 과정에서 사용된 패시 값과 질의에 표현된 패시 값에 대한 일관성이 유지되기 어렵다. 넷째, 질의 표현된 패시 값을 정확히 포함하는 부품만을 검색한다. 따라서, 의미적으로 상당히 연관된 패시 값을 포함한 부품들은 검색될 수 없다.

본 논문에서는 패시 검색 방법인 Prieto-Diaz 시스템의 단점을 보완하기 위해 시소러스를 이용한 패시 기반 부품 검색 시스템을 사용하였다. 웹 환경에서 팀들간 상호 협력을 위해서 UML 편집기에서 생성된 \*.mdl 형태의 파일을 웹 상에서 공유하기 적합한 XML 형태로 변환하여 관계형 데

이터베이스에 저장하고, 또한 웹 기반의 분산 소프트웨어 개발 환경에서 얻어진 많은 소스코드 및 실행파일들을 웹 상에서 개발 팀들 간에 협업 작업을 효율적으로 할 수 있는 환경을 지원한다.

### 3. 웹 기반 항해 도구 설계

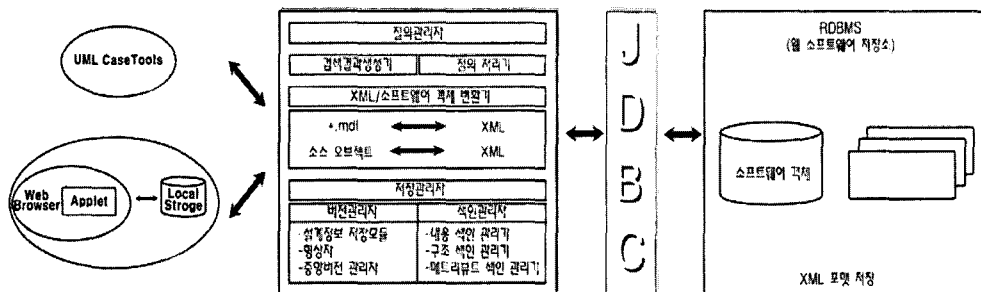
이 장에서는 이미 고안된 바 있는 XML 변환기[15]를 통해 XML 포맷으로 데이터베이스에 저장 관리되는 설계정보를 패킷 기반의 시소러스를 이용한 질의관리자에 대한 내용을 기술한다.

#### 3.1 웹 기반 항해 도구 시스템 구성

웹 기반 항해 도구 시스템은 하부 저장 시스템으로 관계형 데이터베이스 시스템을 사용한다. 이 저장 시스템은 질의 처리 및 검색 결과 생성기로 구성된 질의 관리자와 XML/소프트웨어 객체 변환기, 저장관리자로 구성된다. 본 시스템의 전체 구조는 객체 지향 CASE 편집 도구에서 생성되는 설계 정보들과 소스 코드 및 오브젝트 파일을 웹 저장소에 저장하는 구조이다. 즉, 저장관리자와 JDBC를 통해 XML 형태로 변환[15]된 웹 저장소의 설계 정보들은 웹을 통해서 개발자들 간에 교환이 가능하며, 객체지향 CASE 편집 도구인 UML에서 생성되는 \*.mdl 파일뿐만 아니라 소스 코드와 오브젝트 파일도 상호 공유가 가능하다. 웹 기반 항해 도구 시스템의 전체적인 개괄 구조

는 (그림 1)과 같다.

웹 기반 항해 도구 시스템의 설계 정보 저장 모듈은 설계 객체를 저장시 필요한 테이블들을 자동으로 생성 및 관리하고 소프트웨어 객체로부터 구조 정보인 애트리뷰트들을 추출한다. XML/소프트웨어 객체 변환기는 설계 객체를 웹에서 공유할 수 있도록 XML파일 형태로 변환하며, 이를 웹 브라우저상에서 설계 객체와 소스코드를 보여준다. 구조검색 엔진의 내용 색인 관리자는 내용 인덱스 생성을 위한 정보 추출 및 저장하는 기능과 구조 색인 관리자는 구조 인덱스 생성을 위한 정보추출 및 저장기능을 갖는다. 질의 처리기는 내용 검색, 구조 검색, 애트리뷰트 검색 및 구조 변경 질의처리 기능과 각 해당 질의에 대한 SQL로의 변환 모듈로 구성된다. 질의 처리기에서 시소러스는 분류되지 않은 패킷 값으로 사용자가 질의하였거나 질의에 대한 탐색 결과가 사용자에게 만족스럽지 못할 경우, 질의의 탐색어와 의미적으로 연관성이 있는 개념을 시소러스에 선정하여 질의에 탐색어로 추가하거나 대치함으로써, 사용자에게 보다 만족스러운 결과를 얻도록 하는 목적으로 사용된다. 그리고 검색결과 생성기에서는 검색엔진에서 찾아진 내용을 문서의 전체 혹은 일부분을 사용자에게 보여줄 수 있는 형태로 CASE 도구나 웹 브라우저에 보내진다. 또한 본 웹 저장관리 시스템 HTTP 프로토콜을 이용한 웹 인터페이스를 제공하며, 설계 객체와 소스 그리고 오브젝트를 XML 포맷으로 저장하기 위해 다양한

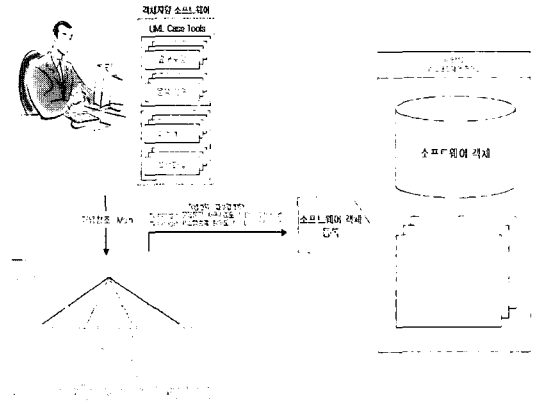


(그림 1) 웹 버전 관리 시스템의 전체 구조

DTD를 설계하였다. 전문검색, 다양한 구조 검색 및 애트리뷰트 검색을 효율적으로 지원하고 설계 객체와 소스 그리고 오브젝트의 수정을 웹 브라우저나 UML 편집기와 개발 도구에 의해 할 수 있으며, 소프트웨어 객체의 히스토리 정보를 가지고 있어 버전을 관리한다. 데이터베이스에 저장된 버전의 목록 및 내용 보기 기능뿐만 아니라 웹 저장소에 저장된 설계 객체나 소스 코드 그리고 오브젝트를 웹 브라우저나 UML 편집기와 개발 도구에서 볼 수 있으며 수정 기능도 가지고 있다. 웹 저장소 시스템의 모듈별 구성 요소를 살펴보면, XML 설계 객체 관리자의 스키마 관리기는 설계객체나 소스 그리고 오브젝트 파일 저장시 필요한 테이블들을 자동으로 생성 및 관리한다. 구조정보 추출기는 설계 객체나 소스 그리고 오브젝트로부터 구조정보를 추출한다. 인스턴스 저장기는 설계 객체, 소스, 오브젝트, 구조정보, 색인정보를 저장한다. 인스턴스 관리기는 소프트웨어 객체를 구성하는 엘리먼트들을 추출한다. 문서 병합기는 추출된 엘리먼트들을 가지고 XML 문서를 생성한다. XML 버전 관리기는 수정된 소프트웨어 객체에 대한 버전 관리 요구 시 버전을 관리한다. XML 색인 관리자의 내용 색인 관리기는 내용 인덱스 생성을 위한 정보 추출 및 저장을 담당하고, 구조 색인 관리기는 구조 인덱스 생성을 위한 정보추출 및 저장을 한다. 애트리뷰트 색인 관리기는 애트리뷰트 인덱스 생성을 위한 정보 추출 및 저장을 한다. XML/소프트웨어 객체 변환기는 \*.mdl 파일을 XML 타입으로 변환하여 웹 저장소에 저장하고, 소스와 오브젝트 파일을 XML 타입으로 변환하여 웹 버전 저장소에 저장한다.

### 3.2 설계 정보의 생성 및 등록

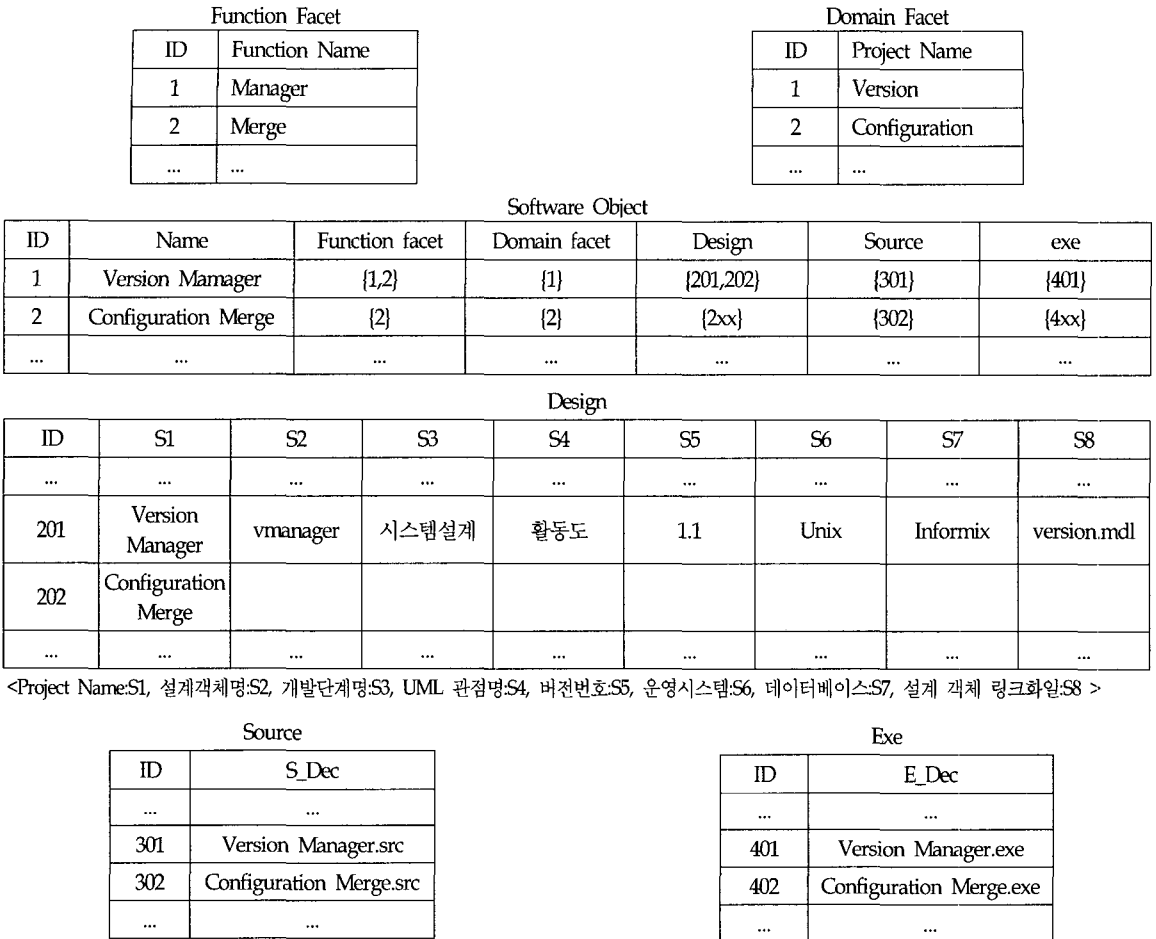
(그림 2)는 재사용 가능한 객체지향 소프트웨어 객체를 생성 및 등록하는 과정을 나타낸다. 재사용될 소프트웨어는 (그림 2)와 같이 설계객체(요구문장, 분석/설계), 소스 코드, 실행파일로 구



(그림 2) 시소러스를 참조한 패킷값 할당

성되어 소프트웨어 등록기에 의해 사용자에게 제시된다. 사용자는 설계 정보의 특징을 나타내는 기능/도메인 패킷값을 시소러스를 참조하여 할당하면 소프트웨어 객체 등록기는 설계 정보를 객체지향 웹 소프트웨어 저장소에 저장한다. 웹 소프트웨어 저장소에 저장된 설계 정보들 중 사용자가 원하는 설계 정보를 정확히 검색하기 위해서는 등록시에 명확하고 구체적인 개념을 각 패킷값으로 할당해야 한다. 그러나 일반적으로 사용자는 소프트웨어 설계 정보의 패킷값으로 표현할 수 있는 구체적인 개념들을 모두 알고 있지 못한다. 따라서 설계 정보의 패킷값으로 표현될 수 있는 개념들을 모두 알고 있지 못한다. 따라서 설계 정보의 패킷값으로 표현될 수 있는 개념이 체계적으로 구성된 시소러스를 참조하여 설계 정보를 명확히 표현할 수 있다.

시소러스를 참조하여 설계 정보의 패킷값을 할당하는 과정은 다음과 같다. 먼저 사용자는 설계 정보의 패킷값으로 표현될 일반적인 개념을 시스템에 입력하면 시스템은 시소러스를 조사하여 입력한 개념과 의미적으로 관련 있는 개념을 시각화하여 사용자에게 제시한다. 사용자는 시각화된 개념 중 설계 정보의 패킷 값으로 가장 적합한 개념을 선택하여 할당한다. 이러한 방법을 통해 각각의 패킷 값이 할당된 설계 정보는 (그림 3)와 같이 웹 소프트웨어 저장소에 저장된다.



(그림 3) 소프트웨어 객체 저장 구조

소프트웨어 객체의 웹 소프트웨어 저장소는 크게 ‘Software Object’, ‘Design’, ‘Source’, ‘Exe’, ‘Function Facet’, 그리고 ‘Domain Facet’ 테이블로 구성되어 있다. ‘Software Object’는 설계 정보의 모든 내용을 저장하는 테이블이며, ‘Function Facet’과 ‘Domain Facet’은 설계 정보의 특징을 표현하는 기능/도메인 패킷값에 식별자가 부여되어 저장된 테이블이다. 소프트웨어 객체의 여러 정보는 ‘Design’, ‘Source’, 그리고 ‘Exe’ 테이블에 저장된다.

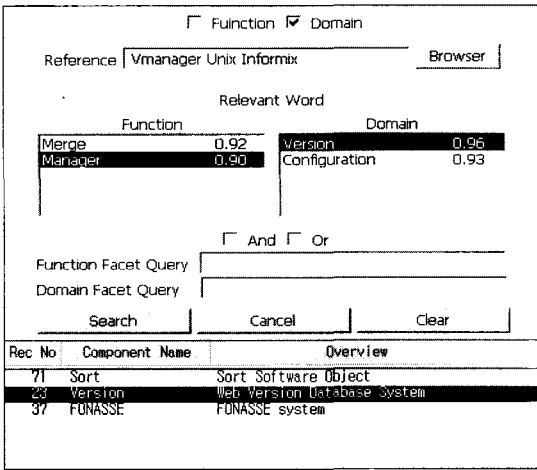
(그림 3)에서 부품 ‘Version Manager’는 패킷값으로 ‘Manager’와 ‘Merge’를 가지며, 도메인 값으로 ‘Version’을 가진다. 여기서 ‘<vmanager, 시스템

설계, 활동도, 1.1, unix, informix>’ 설계 객체에 관한 내용이며 ‘VersionManager.src’는 클래스가 정의되어 있는 소스코드이다. ‘VersionManager.exe’는 소프트웨어 객체에 대한 실행 파일이다.

#### 4. 항해 도구 구현

이 장에서는 소프트웨어 객체를 검색하는 단계에서 각 패킷 값들을 할당하는 과정과 확장된 질의를 통해 개발자가 원하는 소프트웨어 객체를 검색하는 과정을 설명한다.

(그림 4)은 패킷 기반 항해 도구 구현 다음은



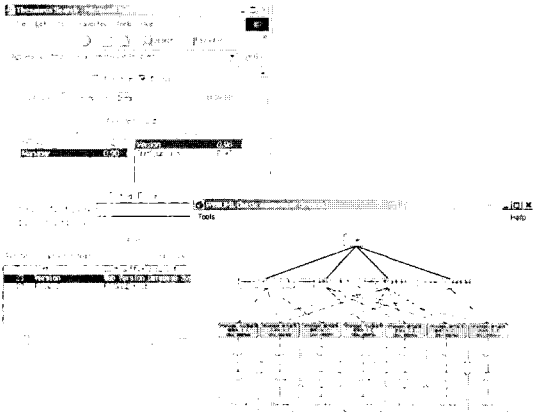
(그림 4) 패킷 기반의 항해 도구

설계정보를 검색하기 위한 메뉴에 관한 화면이다. 위 메뉴는 브라우저 상에서 웹 기반의 설계 정보 저장소에서 설계 객체를 질의하기 위한 화면이며, 예를 들어서 질의가 구성될 패킷 'Domain'을 선택하고, 개발자가 의도한 개념인 설계 객체명을 vmanager, 운영시스템을 Unix, 데이터베이스를 Informix 라 입력 하고자 한다면 "vmanager Unix Informix" 입력하고서 Browser 버튼을 누르면, 시스템은 시소리스를 이용하여 사용자 의도와 의미적으로 상당히 관련된 개념들을 추천 개념으로 사용자에게 제시함으로써, 보다 정교한 사용자 질의를 구성할 수 있다. 즉, "vmanager Unix Informix"의 모든 상위 개념과 하위 개념들을 'Domain' 추천 개념 리스트 화면에 "vmanager Unix Informix"과의 관련 정도와 함께 순위화되어 나열된다. "vmanager Unix Informix"의 모든 상·하위 개념들은 {Version Manager/1.00, Configuration System/0.96}이다. 이때, 도메인에 대한 묵시적인 확장된 검색 질의는 "Version Manager/1.00 OR Configuration Merge/0.96"이 된다. 이 질의는 높은 재현율을 요구하는 검색에 유용하게 이용될 수 있다. 사용자는 'Domain' 리스트로부터 적합한 개념만을 선택하여 AND/OR 연산자의 조합으로 검색 질의를 표현할 수도 있다. 이러한 질의 구성 방법은 높은 정확률이 요구되는

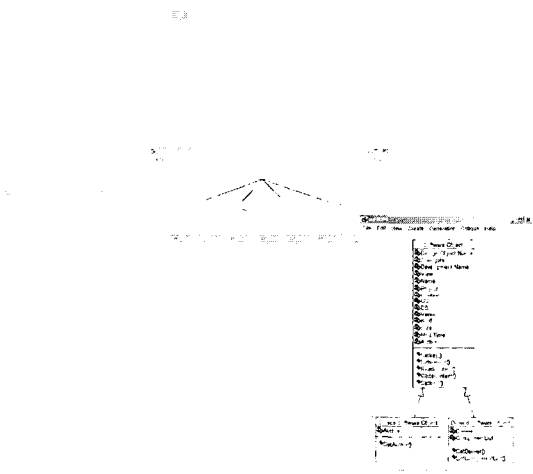
검색 방법에 적합하다. 검색 결과 얻어진 부품은 서로 관련성 있는 다른 형식의 설계 정보를 포함하고 있다. 이 설계 정보들은 하나의 동일한 실세계 대상을 각기 다른 개발 단계 관점에서 보이고, 그 단계의 문맥에 따라 구축된다. 항해 도구 시스템은 관련성 있는 설계 정보들을 서로 연관시키고 이 연관 관계에 따른 관련 설계 정보들간의 원활한 항해 수단을 제공하기 때문에, 개발자는 해당 설계 객체를 보다 쉽게 인식하여 이를 새로운 객체 지향 소프트웨어 개발에 효과적으로 적용할 수 있다. 설계 객체 뷰어는 검색된 설계 정보를 보이기 위해 브라우저, 각종 분석·설계 다이어그램 뷰, 프로그램 코드 뷰, 항해기 등을 포함한다. 먼저 검색기에서 질의 처리 결과 얻어진 설계 정보들은 다음과 같이 관련도에 의해 설계 객체를 순서화(rankng)하여 정렬된다. 개발자는 여기에서 지정된 한 부품에서 시작하여 관련 설계 정보들을 참조할 수 있다. (그림 4)의 설계 객체 뷰어는 브라우저를 통해 구성된 설계 정보목록, 즉 요구문장, 분석·설계 다이어그램, 소스 코드를 시작적으로 보여준다. 설계 정보 내에는 설계 객체의 역할을 나타내는 각종 설계 정보들 외에 설계자 이름, 생성 일자, 설계 객체에 대한 간단한 설명 등을 포함하고 있다.

(그림 4)에서 vmanager를 선택하였을 때 이전과 관련 있는 버전들을 버전트리를 이용해서 보여주는 화면이 (그림 5)이다. (그림 5)은 'vmanager, 분석설계, 순서도, 1.1, Unix, Informix' 설계객체는 한 개의 요구문과 4개의 분석·설계 다이어그램으로 구성되어 있다.

(그림 5)의 화면에서 마우스로 <vmanager, 분석설계, 순서도, 1.1, Unix, Informix>를 선택하여 클릭하면, 선택된 객체도를 (그림 6)과 같이 보여준다. 항해 도구 시스템은 저장소에 저장되어 있는 설계 정보를 XML 형태인 UDXF 파일로 전송 받아서 이를 이용하여서 클래스 다이어그램을 (그림 6)과 같이 그래픽으로 표현한 것이다.

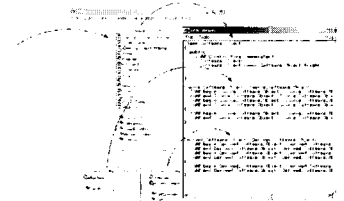


(그림 5) 검색된 설계 객체의 내용



(그림 6) 클래스 다이어그램

위 (그림 6)의 화면에서 마우스로 “Software Object”클래스를 선택하여 클릭하면, 선택된 객체의 소스 창이 (그림 7)와 같이 보여준다. 저장소에 저장되어 있는 설계 정보를 XML 형태인 UDXF 파일로 전송 받아서 이를 이용하여서 해당된 소스를 윈도우 창으로 표현한 것이다. 그래서 개발자들을 설계 정보와 원시 코드를 동시에 참고하면서 개발에 임할 수 있다. 이와 같은 방법을 이용하여 찾고자 하는 설계 정보를 쉽게 찾을 수 있을 뿐만 아니라, 웹 상에서 브라우저를 통해 검색이 가능하므로 공동작업을 원활히 할 수 있어 보다 더 효율적인 개발이 가능해졌다.



(그림 7) 소스창

## 5. 결론 및 향후 연구과제

본 논문에서는 하나의 프로젝트를 수행하는 과정에서 발생하는 원시 코드 이전 단계인 UML을 기반으로 한 소프트웨어 개발 환경에서 다양하게 발생하는 설계 정보들을 공유할 수 있는 웹 기반의 항해 도구를 개발한 내용이다. 이를 이용하여서 팀들간에 설계정보를 유기적으로 항해할 수 있는 항해 도구를 구현하였다. 우리의 항해 도구는 웹 기반의 분산 설계 환경에서 효과적으로 설계 객체를 관리할 수 있도록 고안되었다. 즉, UML 편집기를 이용하여서 소프트웨어를 개발하는 과정에서 발생하는 설계 정보뿐만 아니라 구현 작업에서 발생하는 소스 코드 및 실행 파일을 XML 포맷으로 데이터베이스에 저장 관리한다. 따라서, 웹 기반의 분산 객체 지향 소프트웨어 환경에서는 보다 더 많은 개발자가 자유로운 장소에서 자유로운 시간에 데이터를 공유하며 서로의 의견에 대해 교환할 수 있다. 관계형 데이터베이스에 저장된 설계 정보의 목록 및 내용 보기 기능뿐만 아니라 웹 저장소에 저장된 설계 객체나 소스 코드를 웹 브라우저나 UML 편집기와 개발 도구에서 볼 수 있으며 수정 기능도 가지고 있다. 설계 정보는 웹 버전 관리 시스템을 이용하여 웹 브라우저에서 검색이 가능하여 설계 객체와 소스 코드 그리고 실행파일을 개발자들간에 공유할 수 있게 되었다. 그렇지만, 제한한 항해 모델에서 각 설계 객체의 통합에 대한 세부적인 사항에 대한



생성규칙(production rule)과 같은 기술 언어로 정의하고, 이들의 일치를 통해 설계 객체를 통합하는 방식에 대한 연구가 필요하다.

### 참고문헌

- [1] Martin Fowler, Kendall Cott, UML Distilled, Addison-Wesley Longman, 1997.
- [2] Rational Software et. al, UML Extension for Business Modeling, OMG document number: ad/97-08-07.
- [3] Rational Software et. al, UML Summary, OMG document number: ad/97-08-03.
- [4] 김형준, "UML을 이용한 컴포넌트 설계기술," 한국 래쇼날소프트웨어 Feb., 1999.
- [5] P. C. Kim, "A Taxonomy on the Architecture of Database Gateway for the Web", proc. of the 13th ICAST and 2nd ICMIS, Shaumburg, IL, pp. 226~232, Apr. 1997.
- [6] D. Robinson, The WWW Common Gateway Interface version 1.1 Internet draft, Jan. 1996.
- [7] J. Rumbaugh et al., Object-Oriented Modeling and Design, Prentice Hall, 1991.
- [8] Y. Chiamarella and J. P. Chelvallet, "About Retrieval Models and Logic," The Computer Journal, Vol. 35, No. 3, pp. 233~242, 1992.
- [9] W. Frakes and T. Pole, "Porteus : A Software Reuse Library System that Supports Multiple Representation Methods," CACM, Vol. 24, No. 3, pp. 43~55, 1990.
- [10] P. Devanbu, R. Branchman, P. Selfridge, and B. Ballard, "LaSSIE : A Knowledge-based Software Information System," CACM, Vol. 34, No. 5, pp. 34~49, May, 1991.
- [11] Tomas Lsakowitz and Rovert J. Kauffman, "Supporting Search for Reusable Software Objects", IEEE Transactions on Software Engineering, Vol. 22, No. 6, June 1996.
- [12] R. Rada and B. K. Martin, "Augmenting Thesauri for Information Systems," ACM Transaction on Office Information System, Vol. 5, No. 4, pp. 378~392, 1987.
- [13] Bruce A. Burton et al., "The Reusable Software Library," IEEE Software, Vol. 4, No. 4, pp. 25~33, July 1987.
- [14] R. Prieto-Diaz, P. Freeman, "Classifying Software for Reusability", IEEE Software, Vol. 4, No. 1, pp. 6~16, January, 1987.
- [15] 김진성, 송행숙, 최동운, "XML 기반의 소프트웨어 공동 작업을 위한 UDXF 저장소 관리자", 한국인터넷정보학회 논문지, 제3권 제1호, 2. 2002.

## ● 저 자 소 개 ●



### 김 수 응

1996년 서남대학교 수학과 졸업(학사)

1998년 서남대학교 대학원 전자계산학과 졸(이학석사)

2003년 서남대학교 대학원 컴퓨터정보통신학과(박사수료)

1999년~현재 : 서남대학교 전자계산소 전임조교

관심분야 : 지능형 에이전트, 객체 지향 시스템, 웹 공학, 웹 데이터베이스, XML 저장소

E-mail : provibe@empal.com



### 최 동 운

1984년 전북대학교 전산과 졸업(학사)

1986년 전북대학교 대학원 전산과 졸업(이학석사)

1997년 전북대학교 대학원 전산과 졸업(이학박사)

1994년~1998년 8 서남대학교 전자계산소 소장

1994년~현재 : 서남대학교 컴퓨터 정보통신학과 교수

관심분야 : 지능형 에이전트, 객체 지향 시스템, 웹 공학, 웹 데이터베이스, XML 저장소

E-mail : cdo@seonam.ac.kr