

XML 기반의 소프트웨어 공동 작업을 위한 UDXF 저장소 관리자

UDXF Repository Manager for Software Collaborative Based on XML

김진성*
Jin-Sung Kim

송행숙**
Haeng-Sook Song

최동운***
Dong-Oun Choi

요약

UML은 소프트웨어 시스템을 그래픽 보델링하는 객체 지향 방법론이다. UML 그래픽 편집기를 개발하였는데, 소프트웨어 개발 주기 동안에 다양한 설계 객체들이 발생한다. 이는 의미 정보와 뷰에 관한 정보들로 구성되는 설계 객체는 관계형 데이터 베이스 시스템이 관리한다. 기존에 UML을 지원하는 많은 도구들이 존재하는데, 이들은 설계 객체들을 웹 상에서 공유하기 위한 명확한 양식을 가지고 있지 못하다. 본 논문에서는 이런 문제점을 극복하여 UML 설계 객체를 상호 교환하고, 공동으로 공유하기 위한 XML 기반의 UDXF(UML Design object eXchange Format) 교환 양식을 개발하였다. UDXF는 UML 설계 객체들을 단순하고 구조화된 양식으로 변환한다. XML 표준에 의한 모델 정보의 교환에 의하여 공동작업과 설계 객체의 재사용이 가능하다. 또한 UDXF 형식의 설계 객체를 위한 설계 객체 저장소 관리자를 제안한다.

Abstract

UML is an object-oriented development methodology to graphically modeling, a software system. We are developed UML graphics editor and its generating various design objects during software development cycle. Its design object compose of semantics information and view information manage using of RDBMS. Extant tools supporting UDML, however, it does not have an explicit format to describe and interchange its design object intentionally.

In this thesis, to overcome the drawback, the UML design object interchange and presents our efforts to make UML design objects highly interoperable. We developed an interchange format called UDXF(UML Design object eXchange Format) based on XML. UDXF is a simple and well-structured format to encode UML design objects. It leverages the tool interoperability, team development and reuse of design objects by interchanging the model information with the XML standard. Also, we propose design object repository manager for design object of UDXF format.

1. 서론

최근에 개발되고 있는 소프트웨어들은 규모가 대용량이고 복잡하여지는 경향을 가지게 되는데, 이를 위해서 객체 지향 개발 방법론이 소개되어 많이 이용되고 있다[1,2,3]. 또한 개발 방법론의 전체 개발 주기 동안에 체계적으로 지원하기 위한 통합 CASE 환경에 관한 연구가 진행되고 있다. 이

와 같은 통합 CASE 환경에 기반 한 소프트웨어 개발 환경에서는 개발하는 과정에서 다양한 유형의 산출물들이 생성된다. 이들은 지리적으로 분산된 서로 다른 영역 전문가들 사이에 상호 협력을 위해서 공유해야 하는 필요성이 대두됨에 따라 네트워크를 이용한 새로운 형태의 공동작업이 필요하게 된다. 특히 복잡한 시스템 개발을 위해서는 장기간에 걸쳐 서로 다른 전문가들의 협력 작업을 요구하고 있으며, 소프트웨어 설계 정보 공유, 상호 의존적 업무 수행, 공동의 산출물 생성 등 개발 작업에 있어 웹을 이용한 환경에 적용하기 위한 시도가 많이 이루어지고 있다[4,8]. 분산 환경에서는 소프트웨어 개발하는 팀들이 설계 정보

* 비 회 원 : 원광대학교 정보·전자상거래학과 교수
iqzero@wonkwang.ac.kr

** 비 회 원 : 한일장신대학교 컴퓨터정보통신학과 교수
songhs@hanil.ac.kr

*** 종신회원 : 서남대학교 컴퓨터정보통신학과 교수
cdo@tiger.seonam.ac.kr

를 공유하고, 관리하기에 많은 어려움을 느낀다.

본 논문에서는 객체 지향 소프트웨어 개발 방법론이 제안되고 있는데, 이 개발 방법론 중에서 UML(Unified Modeling Language)[5,6,7]을 사용한 분산된 소프트웨어 개발 환경에서 상호간에 설계 정보를 공유하기 위한 방법론에 대한 연구이다. UML은 웹 상의 분산 개발 팀들 간의 설계 정보를 공유할 수 있는 방법을 제공하고 있지 못하고 있다. UML 그래픽 편집기를 개발하고, 이 편집기를 이용해서 소프트웨어를 개발하는 과정에서 발생하는 설계 정보들을 RDBMS를 이용하여서 저장 관리한다. DBMS에 관리하는 소프트웨어 설계 정보를 웹 상에 분산된 개발 팀들 간에 공유하기 위한 방법으로서 XML(eXtensible Markup Language)[11,12]을 이용한다. XML DTD(Document Type Definition)로 정의한 UDXF(UML Design data eXchange Format)을 이용하여서 UML 설계 정보를 XML 화일로 변환하여서 인터넷 공간에서 공유할 수 있는 환경을 제공하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 UML과 XML에 관한 연구 내용을 기술하였으며, 3장에서는 본 논문에서 제안하는 UDXF DTD에 관한 설계 내용을 기술하고, 4장에서는 UDXF 설계정보 저장소에 관해서 기술하고, 5장에서는 결론 및 향후 연구 방향에 대해서 기술하였다.

2. 관련 연구

2.1 UML에 의한 소프트웨어의 모델링

통합 모델링 언어인 UML은 먼저 요구명세를 하고, 분석하고, 설계하고, 구현하고, 테스트하는 소프트웨어 개발 주기의 전 과정에 단계 정의 및 단계별 업무들을 정의한다. 단계 및 업무들 간의 효율적인 상호 관계를 식별하기 위한 단계별 업무들 간의 업무 흐름을 정의한다. 기본적으로 UML에서는 도메인을 기능적으로 분할 할 수 있는 도구인 사용 사례도와 사용 사례 명세서를 제공하

고 있으며, 시스템의 정적, 동적, 기능적인 부분을 모델링할 수 있는 기법으로 클래스 다이어그램, 순차도 또는 협력도, 상태 전이도, 그리고 활동도를 제공하고 있다. 관련 있는 클래스들을 결합할 수 있는 장치로 패키지도를 제공하고 시스템에 대한 물리적 장치를 표현할 수 있는 기법으로 다형도를 제시하고 있다. 이 모델링 언어는 문제를 표현하는 문장 형식인 요구 명세로부터 시작한다. 이를 위해 요구 명세에서는 사용 사례도와 간단한 클래스도 그리고 활동도로 표현이 되어진다. 그런데 대개 문제 문장은 불완전하고 규정에 맞지 않기 때문에, 분석 단계에서 요구 사항의 모호함과 불일치성을 해결하여 실세계에 대한 모델을 명확히 해주어야 한다. 이를 위해 분석 단계에서는 클래스도, 순차도, 협력도, 상태도 그리고 활동도가 만들어진다. 설계 단계에서 분석 단계의 결과물에 기술적인 부분을 첨가하여 확장하는 것이다. 기술적인 확장이란 시스템을 어떻게 구현할 것인지에 중점을 두고 어떻게 동작하고 어떤 제약이 있어야 하는 지에 관하여 생각하는 것이다. 설계 단계에서 클래스도, 순차도 그리고 협력도 그리고 상태도 그리고 활동도, 구성요소도, 배치도가 만들어진다. 마지막으로 설계의 결과에 따라 구현 및 테스트 활동이 수행된다. 그리고 필요에 따라 이전 단계에서 앞 단계로 피드백이 일어날 수 있다.

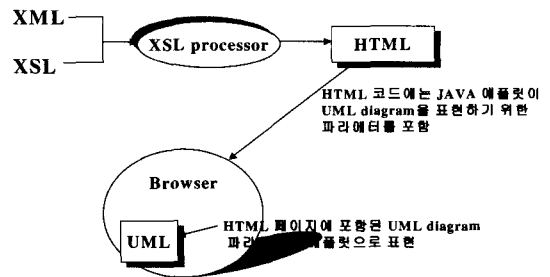
2.2 기존의 설계 객체 교환 방법론

분산되어 있는 소프트웨어 개발 환경에서는 설계 정보들을 개발 도구들 간의 연계성을 고려하고, 개발자들 간의 설계 정보를 상호 교환할 수 있어야 하고, 이들을 보다 보편화된 웹 환경을 위해서 확장되어야 한다[10]. 그러기 위해서는 설계 정보를 스트림 데이터로 기술할 수 있는 양식이 필요하게 된다. UML의 설계 정보들은 웹을 이용하여서 공유하고자 할 경우에 의미 정보와 표기법에 관한 두 가지 측면의 정보를 제공하여야 한

다. 첫째, 의미(Semantics) 정보는 각 모델의 구성 요소들이 무엇을 포함하고 있는가와 각 구성 요소들이 가지고 있는 의미들, 각 구성 요소들 간의 관련성을 기술하며 둘째, 표기법(notation)은 각 구성 요소의 시각 표현을 위한 그래픽 표기법이다.

웹 상에서 UML 설계정보를 상호 교환하기 위한 방법으로 HTML[9] 문자 스트림 안에 UML 다이어그램을 GIF나 JPG 이미지 형태로 변환하여서 교환하는 방법이 사용되고 있다. 그러나 이 방법은 그 이미지들을 제작, 판독, 유지보수 등에 많은 시간을 낭비하게 한다. 또한, 모델의 정보들이 이미지 안에 숨겨져 있으며, 다른 개발 도구들 간의 설계 정보를 상호 교환하여 사용이 불가능하다. 그래서 최근에 CASE 설계 정보를 공유하기 위한 방법으로 CDIF(CASE Data Interchange Format)[13], UXF(UML eXchange Format)[8] 등이 제안되었다. 이중에 UXF는 차세대 웹 표준안인 XML을 이용하여 소프트웨어 설계 정보를 저장한 후에 웹 상에서 상호 교환하는 방법론에 대해서 연구되고 있다. 이의 방법론은 UML의 의미 정보를 XML의 DTD를 이용하여서 XML 파일로 변환한 후에 웹 상에서 개발 도구나 개발자들이 공유할 수 있도록 하였다. 그런데 이 방법론은 설계 정보의 의미 정보만을 변환하여 사용한다. 그렇기 때문에 클라이언트의 사용자들은 UML 다이어그램을 GIF, JPG 이미지 형태로 변환한 후 교환하는 방법이 사용되는데, 이는 다이어그램을 재사용할 수 없어 많은 문제점을 야기한다.

본 논문에서는 제안하는 UDXF는 UML의 의미 정보와 다이어그램을 XML 안에 정의하고, 사용자들이 브라우저 이용하여 디스플레이 할 수 있는 방법을 제공한다. 이를 위해서는 UML 설계 정보의 설계 객체(즉 클래스, 상태), 설계 객체들 간의 연결 관계 등의 두 가지의 엔티티들을 관리하여야 한다. DTD 안에 설계 객체 템플릿이 가져야 할 속성은 타입, 이름, 위치, 설계 객체에 삽입된 서브 설계 객체 등이다. 이와 같은 설계 정보들은 UDXF의 이용하여 XML 파일로 변환되어서



(그림 1) UML 설계 정보의 변환 과정

저장되고, 사용자 웹 브라우저를 이용하여서 디스플레이 된다.

그림 1은 UML 설계 정보가 UDXF DTD를 이용하여서 HTML 형태의 XML 파일로 변환 된 후에 인터넷상에서 개발자나 개발 도구들 사이에 공유되는 과정을 보여주고 있다.

3. UDXF의 설계

개발도구들 사이에서 UML의 설계 정보를 교환하기 위해서는 모델에 관련된 의미 정보와 뷰에 관한 표기 정보를 XML의 DTD를 이용하여서 변환하여야 한다. 모델에 관한 정보는 의미(Semantics) 정보인데, 이들은 각 모델의 구성요소들이 무엇을 포함하고 있는 가와 각 구성요소들이 가지고 있는 의미들, 각 구성 요소들 간의 관련성들을 말한다. 뷰에 관한 정보는 모델에 관한 정보의 표기법(notation)인데, 이는 각 구성요소의 시각 표현을 위한 그래픽 표기법이다. 표 1은 UDXF 표현과 UML의 모델을 관계형 DBMS에 저장할 때 속성 요소들과 사상 관계를 나타낸 표이다.

그림 2는 개발한 UML 편집 도구를 이용하여서 작성한 예제의 클래스 다이어그램이다. 이와 같이 그래픽 편집 도구를 이용하여서 모델링 된 설계 정보들을 기존의 Rational Rose의 UML은 화일 시스템을 이용하여서 저장하는 반면에 본 시스템에서는 관계형 데이터베이스를 저장 관리한다. 즉, 설계 정보의 의미 정보와 표기 정보를 관계형 테이블에 사상하여 관리한다.

(표 1) UML 요소와 UDXF 요소의 비교

UML 패키지	UML 모델 요소	UDXF 표현
Core	Class	<ClassObject>
	Attribute	<Attribute>
	Method	<Method>
	ClassRept	<ClassRept>
	Association	<AssociationRept>
	CommentRept	<CommentRept>
	AggregationRept	<AggregationRept>
	generalization	<generalRept>
	Interface	<Interface>
	ConstraintRept	<ConstraintRept>
Extension Mechanism	TaggedValue	<TaggedValue>

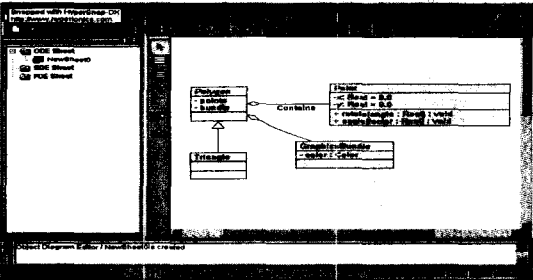
계형 데이터베이스를 이용하여 개발한 저장소에 저장된 데이터베이스는 표 2에서부터 표 7까지와 같다. 여기에는 설계객체의 의미정보와 표현정보들이 함께 저장되어 관리된다. 또한 이들은 XML 형태의 화일로 변환되어서 웹 환경에서 서로 다른 개발자나 유사한 개발도구들 상호 간에 설계 정보를 교환하기에 적합하다.

(표 2) omtmodule 테이블

modulename	next_oid
varchar(255)	integer
uml	18

(표 3) objectsheet 테이블

pid	object_id	sheet_name
varchar(255)	integer	varchar
uml:1	1	newsheet0



(그림 2) UML 편집 도구를 이용한 클래스

(표 4) classobject 테이블

pid	object_id	class_name	attribute_list	mthd_list
varchar(255)	int	varchar(255)	varchar(255)	varchar(255)
uml:3	3	Polygon	- points - bundle	
uml:5	5	Point	-x: Real = 0.0 -y: Real = 0.0	+rotate (angle: Real):void +scale (factor: Real):void
uml:7	7	Triangle		
uml:9	9	GrahpicBundle	-color: Color	

3.1 UML 설계정보의 저장된 예

다음 표들은 개발한 UML 편집 도구를 이용해서 모델링한 예제의 클래스 다이어그램의 저장된 예이다. 이와 같이 그래픽 편집 도구를 이용하여서 모델링 된 설계 정보들을 기존의 Rational Rose의 UML은 화일 시스템을 이용하여서 저장하는 반면에 본 시스템에서는 관계형 데이터베이스를 저장 관리한다. 즉, 설계 정보의 의미 정보와 표기 정보를 관계형 테이블에 사상하여 관리한다.

위의 그림 2의 편집도구를 이용하여 클래스 다이어그램을 작성한 내용에 관한 설계 정보를 관

(표 5) classrept: 테이블

pid	rept_id	object_id	width	height	x	y
varchar			integer			
uml:1:2	2	3	87	76	131	110
uml:1:4	4	5	276	92	344	103
uml:1:6	6	7	87	60	130	257
uml:1:8	8	9	150	60	292	232

(표 6) aggregationrept 테이블

pid	rept_id	object_id	first_node	second_node	name	f_role	f_mul	s_role	s_mul	line_lst
varchar	int	int	int	int	varchar	varchar	int	varchar	int	varchar
uml:1:10	10	11	4	2		Contains	1		1	uml:1:10:11
uml:1:16	16	17	8	2			1		1	uml:1:16:17

(표 7) generalrept 테이블

pid	rept_id	object_id	first_node	second_node	name	line_lst
varchar	integer	integer	integer	integer	varchar	varchar
uml:1:14	14	15	6	2		uml:1:14:15

3.2 UML을 위한 DTD

위 절에서의 UML 설계 정보들이 관계형 데이터베이스에 저장된 형태를 XML 파일 양식으로 변환하기 위한 DTD를 설계한 내용을 보면 표 8과 같다.

(표 8) UML 설계 정보를 UDXF 파일로 변환하기 위한 DTD의 일부

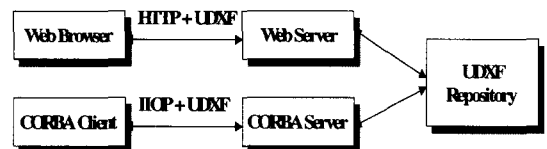
```

<model>
<Objectsheet NAME = newsheet >
  <ClassObject>
    <ClassObject PID = uml:3
      OBJECT_ID = 3
      OBJECT_NAME = polygon>
    <Attribute_LIST VISIBILITY = private
      TYPE = Point
      NAME = points />
    <ClassRept PID = uml:12
      REPT_ID = 2
      OBJECT_ID = 3
      WIDTH = 87
      HEIGHT = 76
      X = 131
      Y = 110 />
    <AggregationRept PID = uml:1:10
      REPT_ID = 10
      OBJECT_ID = 11
      FIRST_NODE = 4
      SECOND_NODE = 2
      F_ROLE = contains
      F_MUL = 1
      S_ROLE = 1
      S_MUL = 1
      LINELIST = uml:1:10:11/>
    <GeneralRept PID = uml:1:14
      REPT_ID = 14
      OBJECT_ID = 15
      FIRST_NODE = 6
      SECOND_NODE = 2
      LINELIST LINELIST = uml:1:14:15/>
  </ClassObject>

```

3.3 설계 정보의 유통구조

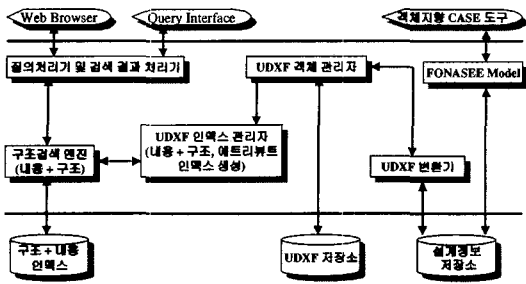
본 시스템은 웹 환경에서 UDXF의 DTD를 이용하여서 변환된 XML 형태의 UDXF 파일들은 분산된 개발 도구들과 개발자들에게 공유할 수 있는 환경을 제공하여 즐겁만이 아니라 CORBA(Common Object Request Broker)의 규약을 따르는 자바 기반의 ORB(Object Request Broker)에서도 설계 정보의 공유가 가능하도록 개발하였다. 또한 XML의 파일 형태를 Relational ROSE의 *.mdl 형태의 파일로 변환하여 직접 개발 도구에서 사용할 수 있다. 지금 현재에는 UML의 클래스 다이어그램의 설계 정보만을 지원한다. 그림 3의 그림은 웹 브라우저 사용자와 코바 클라이언트를 각각 웹 서버와 코바 서버를 경유하여서 UDXF의 저장소에 접근하는 경로에 대한 내용을 도식화한 내용이다.



(그림 3) UDXF 파일의 유통구조

4. UDXF 저장관리 시스템

UDXF 저장 관리자의 하부 저장시스템으로 관계형 데이터베이스 시스템을 사용하였다. 이 저장



(그림 4) UDXF 저장관리시스템의 전체 구조

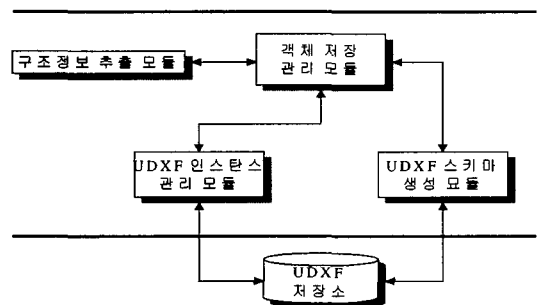
시스템은 질의 처리 및 검색 결과 생성기, UDXF 객체 관리자, UDXF 인덱스 관리자, UDXF 변환기 등으로 구성된다. 본 시스템의 전체의 구조는 객체 지향 CASE 편집 도구에서 생성되는 설계 정보들은 관계형 데이터 베이스인 설계 정보 저장소에 저장된다. 설계 정보 저장소의 설계 정보들은 UDXF의 DTD를 이용하여서 변환기가 XML 형태의 화일로 변환되어서 UDXF 저장소에 저장된다. UDXF 저장소의 설계 정보들은 웹을 통해서 개발자들 간에 상호 공유하고, 개발 도구를 위해서는 CASE 편집 도구의 *.mdl 화일 형태로 변환되어서 공유한다. UDXF 저장관리 시스템의 전체적인 개괄 구조는 그림 4와 같다.

UDXF 저장관리자의 UDXL 객체 관리자는 실제 XML 화일 형태로 변환된 설계정보를 저장하기 위한 스키마 생성 및 UDXF 문서 인스턴스의 저장 및 검색을 담당한다. UDXF 인덱스 관리자는 구조 검색 및 애트리뷰트 검색을 위한 인덱스를 생성하고 관리한다. 구조 검색 엔진은 구조 검색, 애트리뷰트 검색, 혼합 검색을 지원하기 위해 자체 구현하였다. 질의 처리기는 웹 브라우저를 이용하여 사용자가 질의할 때 이를 분석하여 구조 검색, 애트리뷰트 검색, 내용과 구조 검색 등 혼합 검색은 구조 검색 엔진을 사용한다. 그리고 검색결과 생성기에서는 검색엔진에서 찾아낸 내용을 문서의 전체 혹은 일부분을 사용자에게 보여 줄 수 있는 형태로 변환하여 사용자의 웹 브라우저에 보내진다. 본 논문에서는 설계 정보 저장소에 저장된 설계 정보를 UDXF DTD를 이용하여 XML

화일 형태로 변환하여 UDXF 저장소에 저장한 후에 웹을 이용하여서 개발자들이 공유하는 부분에 관한 연구인데, 관계형 데이터베이스시스템에 저장된 UML 설계 정보를 XML 화일 형태로 변환하는 UDXF와 변환된 정보를 UDXF 저장소에 저장하는 UDXF 객체 관리기에 한해서 설계하고 구현한 내용을 기술한다.

4.1 UDXF 객체 관리자

UDXF 객체 관리자는 설계 정보를 XML 화일로 변환한 UDXF 문서를 관리하기 위해서 개발되었다. UDXF 객체 관리자는 UDXF 문서의 데이터 모델링에 의해 UDXF 문서가 포함하고 있는 여러 가지의 특성을 고려하여 스키마를 설계하여 관계형 데이터베이스에 생성한다. 또한 UDXF 문서, 구조 정보 등을 데이터베이스에 저장하여 이를 사용자가 원하는 문서나 이의 일부분 검색하는데 사용한다. UDXF 객체 관리자의 구성요소를 보면 객체 저장관리 모듈, 구조 정보 추출 모듈, UDXF 인스턴스 관리 모듈, 스키마 생성 모듈 등으로 구성되어 있다. 객체 저장 모듈은 UDXF 객체 관리자의 모든 모듈들에 대한 통합 인터페이스를 제공한다. 또한 구조 정보 추출 모듈은 UDXF 인스턴스 문서를 저장하는 필요한 구조 정보를 추출하며, UDXF 스키마 생성 모듈은 DTD 테이블을 시스템 초기에 생성하고, 새로운 프로젝트 문서를 수용할 때마다 문서 테이블, 엘리먼트 테이블 등을 동적으로 생



(그림 5) UDXF 객체 관리자

성한다. UDXF 인스턴스 관리 모듈은 구조 정보 추출 모듈에서 추출된 문서의 구조 정보, UDXF 문서 등을 UDXF 저장소에 저장하고, 이들을 사용자가 요구하는 문서 전체 혹은 문서의 일부를 저장소로부터 추출하는 일을 수행한다.

4.2 UDXF 질의 처리기

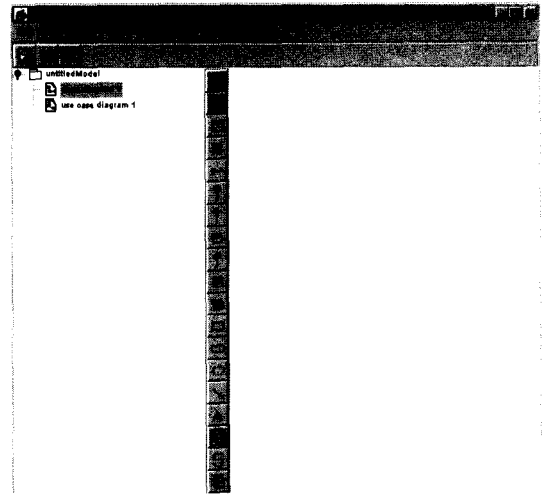
내용 검색과 구조 검색이 사용자에게서 요구될 경우 검색 엔진에서 처리하고 처리된 결과는 검색 결과 생성 모듈에서 UDXF 객체 관리자의 UDXF 인스턴스 관리 모듈 저장소에 저장되어 있는 설계 정보의 일부 혹은 전체를 사용자에게 웹 브라우저를 통해서 보여 준다. 구조 검색, 애트리뷰트 검색, 혼합 검색이 요청될 경우에 구조 검색 엔진이 검색한 결과를 이용하여 설계 정보를 사용자에게 제공한다.

5. UDXF 저장소 인터페이스

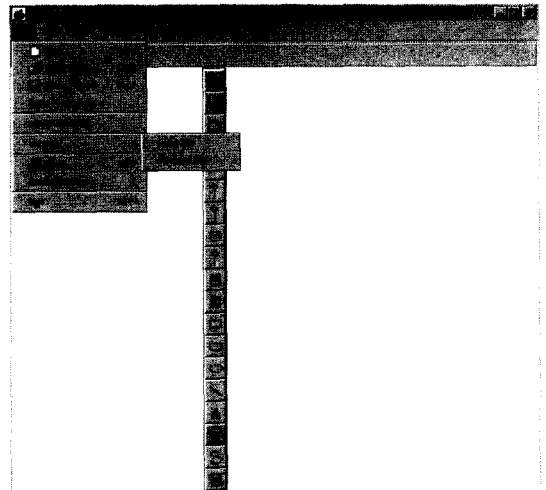
대형 소프트웨어를 개발하는 과정에서 개발에 참가하는 개발자들이 지리적으로 분산되어 있는 경우에 이들 간의 소프트웨어의 분석에 관한 정보와 설계 정보들을 상호간에 공유할 수 있도록 지원해 주도록 웹 기반의 UDXF 저장소 관리자를 개발하였다. 분산 환경의 개발자들은 기존의 Rational ROSE사의 UML에 웹 기반의 정보 저장소를 접근할 수 있는 기능을 추가한 Web-UML을 이용하여서 자신이 개발한 컴포넌트를 등록할 수 있을 뿐만 아니라 공동 팀원들이 개발한 컴포넌트를 참조하면서 지역적인 환경에 제약을 받지 않고서 하나의 프로젝트를 공동으로 개발하여 나갈 수 있다. 그림 6은 개발자들을 위한 저장 관리자에 접속하여서 JAVA 애플릿으로 개발된 개발 도구인 Web-UML을 다운로드 받아서 실행된 화면을 볼 수 있다. 이 도구는 현재 클래스 다이어그램만을 지원하고 있지만 향후에는 기존의 UML에서 지원하는 모든

다이어그램을 지원할 수 있도록 개발 예정이다.

지금까지 기술한 웹 기반의 설계 도구가 어떠한 방식으로 UML 설계 객체들을 사용자가 검색할 수 있도록 지원하는가를 쉽게 이해할 수 있도록 하기 위해, 전형적인 예들을 사용자 인터페이스 상에서 설명한다. 그림 6은 웹 기반의 설계 도구에서 설계 정보를 관리하기 위한 Web-UML의 화면이다.



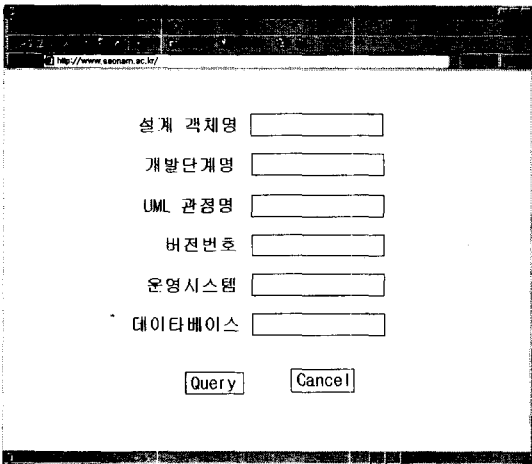
(그림 6) 웹 기반의 설계 도구



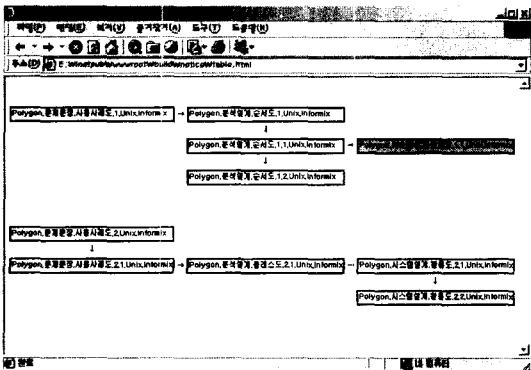
(그림 7) 설계객체 질의 화면

그림 8은 브라우저 상에서 웹 기반의 설계 정보 저장소에서 설계 객체를 질의하기 위한 화면이며, 예를 들어서 설계 객체명을 Polygon, 운영시스템을 Unix, 데이터베이스를 Informix라 입력하고서 Query 버튼을 누른 다음, 설계 정보 트리를 차례대로 선택하면 앞에서 입력한 설계 객체를 중심으로 한 기원 버전에서 파생된 설계 정보들의 버전들이 검색된다. 이때에 설계정보 트리는 버전 규칙에 의해 설계 정보 트리가 생성되어 그림 9에서 보여주고 있다.

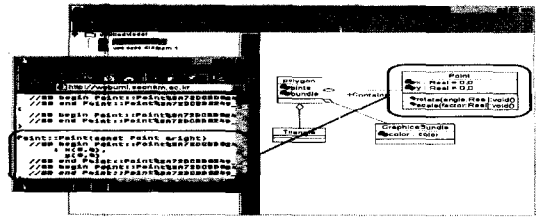
그림 9의 화면에서 마우스로 <Polygon, 시스템 설계, 활동도, 1.1, Unix, Informix>를 선택하여 클릭하면, 선택된 객체도를 보여주기 위한 객체도



(그림 8) 설계객체 질의 화면



(그림 9) 설계정보 버전 트리



(그림 10) 선택된 설계 객체를 보여주는 화면

윈도우와 소스 창이 그림 10과 같이 이 윈도우에서 해당 Web-UML은 저장소에 저장되어 있는 설계 정보를 XML 형태인 UDXF 파일로 전송 받아 이를 이용하여 클래스 다이어그램을 그래픽으로 표현한 것이며, 좌측의 윈도우는 이 설계 정보의 구현된 원시코드를 보여주고 있다. 그래서 개발자들을 설계 정보와 원시 코드를 동시에 참고하면서 개발에 임할 수 있다.

이와 같은 방법을 이용하여 찾고자 하는 설계 정보 버전을 쉽게 찾을 수 있을 뿐만 아니라, 웹 상에서 브라우저를 통해 검색이 가능하므로 공동 작업을 원활히 할 수 있고, 개발 기간을 단축할 수 있고, 보다 더 효율적인 개발이 가능해졌다.

6. 결론 및 향후 연구

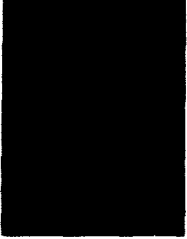
기존의 설계정보를 웹 상에서 UML 설계정보를 상호 교환하기 위한 방법으로 HTML 문자 스트림 안에 UML 다이어그램을 GIF나 JPG 이미지 형태로 변환하여서 교환하는 방법이 사용되고 있다. 이 방법론은 다이어그램을 재사용할 수 없는 단점을 내포하고 있다. 반면에 개발한 UML을 위한 그래픽 편집기에서는 클래스 다이어그램을 설계하는 과정에서 발생하는 설계정보를 데이터 베이스에 저장하게 된다. 이 UML의 클래스 다이어그램 설계 정보를 XML 형식의 UDXF 파일 형태로 변환하여 분산된 환경에서 작업하는 개발자들에게, 웹을 통해서 공유할 수 있는 환경을 제공하였다.

향후에는 모든 UML 다이어그램들에 관한 UDXF DTD를 정의하여, UDXF를 이용 자바, C++, IDL의 원시 코드들을 생성하는 모듈이 개발되어야 한다.

참 고 문 헌

- [1] Grady Booch, Object-Oriented Analysis and Design 2nd Edition, The Benjamin/Cummings Publishing, 1994.
- [2] Rumbaugh, J et. al, Object-Oriented Modeling and Design, prentice Hall, 1991.
- [3] Jacobsen, I. Object-Oriented Software Engineering : A Use Case Driven Approach, Addison-Wesley, 1995.
- [4] A. Versey, A. P. Sravana, "CASE as Collaborative Support Technologies", Communication of the ACM, Jan. 1995, pp. 83-94.
- [5] Rational Software et. al. UML Summary, OMG document number: ad/97-08-03.
- [6] Rational Software et. al. UML Extension for Business Modeling, OMG document number: ad/97-08-07.
- [7] Rational Software et. al. OA&D CORBAfacility, OMG document number: ad/97-08-09.
- [8] J. Suzuki, Y. Yamamoto, "Managing the Software Design Documents with XML", <http://www.yy.ics.keio.ac.jp/~suzuki>, 1999.
- [9] T. Berner-Lee, Hypertext Transfer Protocol - HTTP/1.0, Internet RFC 1945, May 1996.
- [10] M. Gaedke, Hans-W. G., A. Schmidt, Ulf S., Wolfgang Kurr, "Object-oriented Web Engineering for Large-scale Web Service Management", Proc. of the 32nd Hawaii international conference on System Science, IEEE, 1999.
- [11] Eric van Herwijnen, Practical SGML Second Edition, KLUWER Academic Publishers, Stevens Printing, 1994.
- [12] Bray, T et. al. Extensible Markup Language (XML) 1.0, W3C, <http://www.w3.org/XML>
- [13] A series of CDIF specifications are available at <http://www.cdif.org>.

◎ 저자 소개 ◎



김진성

1975년 전북대학교 물리학과 졸업(이학사)
1984년 전북대학교 대학원 물리학과 졸업(이학석사)
2002년 서남대학교 대학원 컴퓨터정보통신학과(박사수료)
1984년 9월~현재 : 원광대학교 정보·전자상거래학과 교수
관심분야 : 전자상거래시스템, 프로그래밍언어, 데이터베이스, etc.
E-mail : iqzero@wonkwang.ac.kr



송행숙

1985년 우석대학교 수학과 졸업(이학사)
1988년 전북대학교 대학원 전산학과 졸업(이학석사)
1995년 아주대학교 대학원 컴퓨터공학과 졸업(공학박사)
1997년~현재 : 한일장신대학교 컴퓨터정보통신학과 교수
관심분야 : 컴퓨터그래픽스, 멀티미디어 시스템, 데이터베이스.
E-mail : songhs@hanil.ac.kr



최동운

1984년 전북대학교 전산학과 졸업(학사)
1986년 전북대학교 대학원 전산학과 졸업(이학석사)
1997년 전북대학교 대학원 전산학과 졸업(이학박사)
1994년~1998년 8월 서남대학교 전자계산소 소장
1994년~현재 : 서남대학교 컴퓨터정보통신학과 교수
관심분야 : 전자상거래시스템, 데이터베이스, 지능형에이전트, 웹공학, XML 저장소
E-mail : cdo@tiger.seonam.ac.kr