

객체지향 프레임워크 재사용을 위한 통합 객체 관리 저장소 설계

선수균*, 송영재**

*동원 대학 사무자동화과

**경희대학교 전자계산공학과

A Design of Integration Object Management Repository for reuse Object-oriented Framework

Su-Kyun Sun* , Yong-Jea Song**

*Dept. of Office Automation, Dong Won Collage, Korea

**Dept. of Computer Engineering, Kyung Hee University, Korea

요 약

최근 전산 환경은 통합되는 개방형 시스템으로 변모하고 있는데, 클래스 라이브러리의 재사용이 재사용을 향상시키지 못하고 있다. 이로 인한 라이브러리 습득에 많은 시간과 노력이 가중되고 개발시간이 길어진다. 따라서, 본 논문에서는 객체지향 프레임워크를 잘 사용하여 재사용할 수 있도록 잘 관리하는 저장소를 제안한다. 객체지향 소프트웨어 공학 프로세스에 의해 생성되는 산출물을 객체 형태로 통합 관리함으로써 프레임워크 재사용을 쉽게 할 수 있고, 객체들을 효율적으로 관리해 주고, 특정한 도메인에서 프레임워크를 재사용하여 새로운 어플리케이션을 개발할 때 시간을 단축하기 위해서 통합 객체 관리 저장소를 제안한다. 이 모델은 프레임워크 클래스 계층 구조 및 클래스 상호간의 상호작용 정보를 유기적으로 관리하고 통합할 수 있다.

1 서론

객체 지향 패러다임의 확산으로 인하여 소프트웨어 개발을 위한 객체 모델의 사용이 일반화되고 있다[1].

대형 라이브러리 전체를 이해하고 기초 클래스를 선정 후 상호 연관성을 제어하는 코드를 개발해야 한다. 클래스 라이브러리는 잘 정의된 규모의 기능성을 클래스와 객체의 형태로 제공해 주지만 재사용을 위해서는 클래스 라이브러리의 구조를 잘 이해하고 각 인터페이스를 파악한 후 전체 프로그램을 만들기 위해서 각 클래스 부품을 조립해야 한다[6]. 기존 시스템들은 새로운 어플리케이션을 개발할 때 프레임워크 재사용을 효과적으로 지원하기 힘들었다[2]. 따라서, 본 논문에서는 객체지향 소프트웨어 공학 프로세스에 의해 생성되는 산출물을 객체 형태로 통합 관리하는 통합 객체 관리 모델(Integration Object Management Model)중 통합 객체 관리 저장소를 만들어 새로운 어플리케이션을 개발할 때 프레임워크 재사용을 효과적으로 지원하고 프레임워크 클래스 계층 구조 및 클래스 상호간의 상호작용 정보를 유기적으로 관리하고 통합할 수 있는 구조로

프레임워크의 검색도 가능하게 해 준다. 이 모델은 분산 데이터 처리를 위한 표준화 작업이 이루어져 현존하는 다양한 플랫폼 및 응용 시스템을 그대로 살리면서 통합환경에 적합한 모델로 만드는 것이 향후 과제이다.

본 논문의 목적은 통합객체 관리 모델을 중심으로 기존 시스템을 통합에 필요한 환경을 만들어 효율적으로 관리함으로써 소프트웨어 재사용성을 향상시켜 생산성을 극대화시키는 것이 목적이다.

또한 객체지향 소프트웨어 공학 프로세스에서 발생하는 다양한 산출물을 데이터 베이스화하여 필요한 산출물을 관리하고, 다양한 CASE 도구들과 서로 호환성 있게 유지, 보수관리 할 수 있는 통합 객체 관리모델 중 객체 프레임워크를 재사용할 수 있게 입력 검색, 관리, 분류하기 위한 통합 객체 관리 저장소를 설계한다.

이 설계는 객체 저장소의 객체지향과 객체관계 DBMS 와 매핑하는 것이다. 이 과정은 프레임워크 클래스 계층 구조 및 클래스 상호간의 상호작용 정보를 유기적으로 관리하고 통합할 수 있는 구조로 프레임워크의 검색도 가능하다.

2. 관련 연구

프레임워크 구분은 기반 어플리케이션을 개발할 때 반복되어 작성되는 유사코드가 발생한다. 프레임워크내의 이러한 hot spot부분을 채우는 방법에 따라 화이트박스 프레임워크와 블랙박스 프레임워크로 구분할 수 있다[1][2][4][5][6].

저장소의 기본 기능으로는 프레임워크 저장소에서 일반적으로 제공해야 하는 기능은 저장소내의 객체에 대한 모델링 검색 및 관리기능등이다. 첫째, 저장소를 사용하는 작업이 장기 작업인 경우, 전체 작업을 하나의 트랜잭션으로 취급하는 것은 비현실적이므로 공용 저장소내의 객체를 사용자의 개인 작업 공간으로 복사하고 작업이 끝나면 체크인 연산을 수행하여 개인 작업 공간에서 고용 저장소로 다시 객체를 복사하는 기능이 다.

둘째, 저장소내의 객체는 일반적으로 일련의 재구성 작업을 거치게 되는데 임의의 시점에서 의미 있는 객체의 상태를 버전이라 하며 이를 관리하는 버전 제어 기능이 필요하다.

셋째, 저장소내의 객체는 다른 객체들의 계층적인 집합으로 복합 객체라 불리 우며 이들은 여러 버전을 가질 수 있으며 이들 사이의 연결을 제공하는 형상 관리기능이 필요하다.

넷째, 저장소내의 여러 객체들은 서로 관련이 있으며 한 객체에 연산이 적용될 경우 다른 관련된 객체에도 연산이 적용되어야 할 경우를 고려한 통지 기능이 필요하다. 이 모델을 적용하여 프레임워크 저장소 구조를 설계한다면 인터넷 환경을 기반으로 한 클라이언트에서 관리자는 프레임워크의 정보를 등록하고 관리 검색등의 요구를 서버 측에 전송하면 서버는 클라이언트로부터 요청 받은 프레임워크 관리를 위한 연산을 수행하여 클라이언트로 응답을 한다[6].

- 1)프레임워크등록: 클라이언트에서 전달받은 프레임워크 정보를 평가과정을 거쳐 분류하고 저장소에 등록한다. 등록 모듈은 서버모듈로 구성된다.
- 2)프레임워크 편집: 특정 프레임워크 또는 특정 조건을 만족하는 프레임워크에 대한 이해과정을 통하여 용도에 맞게 수정을 하고 응용 프레임워크에 적용을 한다.
- 3)프레임워크 정보검색: 클라이언트 사용자는 브라우저를 통하여 검색조건에 부합하는 프레임워크를 검색한 후 편집기를 통하여 수정작업을 한다.
- 5)프레임워크 제어기: 저장소내의 프레임워크에 대한 메타정보를 관리한다.
- 6)프레임워크 객체 관리자(Object manager):프레임 워크 정보를 물리적 하부 저장 구조를 사용하여 관리하는 하위레벨 인터페이스를 제공한다.

3. 통합 객체 관리 모델

각 CASE도구들의 유기적인 연관 관계를 맺어서 전체 도구들이 가지고 있는 기능을 극대화시키는 방법은 도구들을 통합하는 것이다.

[5] 단점인 이 기종간의 통합가능과 서비스 품질 향

상을 위한 제어 기능 부재로 효과적인 관리를 할 수가 없었고 웹과 연동 할 수 있는 환경이 매우 미약했다.

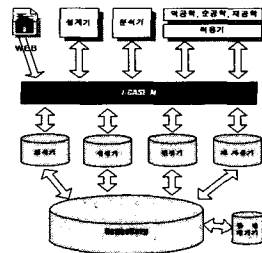
따라서 본 논문에서는 이 기종간의 시스템을 통합 할 수 있고, 보다 효율적으로 관리해 주고 급변하는 소프트웨어 산업에 능동적으로 대처하여 소프트웨어 개발에 시간을 단축함으로써 통합 분산 환경에서 시스템 성능 저하 문제점을 해결하고, 웹과 통합하여 효과적인 관리를 할 수 있는 모델인 통합 객체 관리 모델(Integrated object Management Model)를 제안한 것 중 세 번째 계층인 통합 객체 관리 저장소를 설계한다.

통합 객체관리 모델은 세 계층으로 나누어지는데 (1)분류계층과 (2)통합 관리기 계층, (3)저장계층이다.

첫 번째 계층으로는 분류하고 상호작용에 필요한 여러 정보를 유기적으로 분류하는 View 객체라 하는데 사용자에게 정보를 디스플레이하는 객체이다.

두 번째 계층으로는 통합 관리기라는 제어객체를 두어 검색에 필요한 여러 사항을 제어하는 기능이 있고, 사용자가 입력을 처리하는 객체이다.

마지막으로는 실질적인 저장기능을 담당하는 저장소 객체로 검색한 결과를 재사용할 수 있도록 효율적으로 저장관리를 하고, 모델 객체라 하는데 데이터를 갖는 모델 객체이다.



(그림 3) 통합 객체 관리 모델 전체 구성도

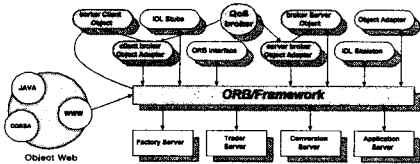
개발기간을 단축시켜 소프트웨어 산출물들을 효율적으로 유지보수 하고 저장 관리하여 생산성을 극대화시키는 것이 본 논문에서 제안한 목적이다. (그림 3)은 통합 객체 관리 모델 전체 구성도를 나타낸 것으로 세 가지 계층으로 구성되는데 프로세싱 관리계층, 객체 저장소 관리 계층, 통합 관리기 계층(I-CASE Manager)이다.

3.1 프로세싱 관리 계층

객체지향 소프트웨어 개발과정 중에는 문서, 다이어그램, 원시코드, 방법론 정보, 설계 패턴의 프레임워크, 사용자 인터페이스 객체, 데이터 베이스 스키마 객체 등의 다양한 산출물이 있는데 이런 산출물들을 분류하는 방법을 제시하는 것이 프로세싱 관리 계층이라 하고 분류객체라 한다. 분류를 View 객체라 하는데 사용자에게 정보를 출력하는 객체이다. 즉 여러 산출물들중 어디에 저장시킬 것인가를 결정시키고, 재사용할 수 있도록 분류하고 저장하는 단계를 결정하는 것이 프로세싱 관리 계층이다.

3.2 통합 관리기 계층

[5]에서는 제어 메카니즘이 없고, 분산 객체 시스템에서 클라이언트와 서버에 각각 프로세스와 데이터의 형태와 불균형으로 통합에 필요한 적절한 구조가 절실하게 필요하게 되었다. 통합 관리기는 이러한 단점을 보완하고 프레임워크 기반의 통합을 실현한다. 통합 관리기(I-CASE Manager)의 구조를 나타낸 것이 (그림 4)과 같다.



(그림 4) 통합 객체 관리기 구성도

3.3 통합 객체 저장소 설계

효율적인 검색과 재사용 할 수 있도록 객체저장소에 저장해야한다. 이런 객체 저장소의 주요기능으로는 많은 산출물을 관리해야 하는데 저장소 관리 계층은 이런 관리를 하는 계층이다. 또한 다양한 도구와 사용자의 동시 공유를 위한 병행 처리, 주요 산출물 객체에 대한 등급별 접근 처리와 보안처리, 시스템 장애 발생에 대한 회복, 객체의 올바른 상태를 유지하기 위한 무결성 처리 기능이 제공되어야 한다. 이러한 기능은 객체 저장소를 파일 시스템을 기반으로 하지 않고, DBMS를 기반으로 개발함으로써 제공받을 수 있다.

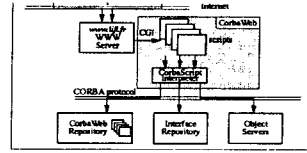
통합 객체 저장소 설계는 객체 저장소의 객체지향과 객체관계 DBMS 매핑으로 구성할 수 있는데, 객체지향 DBMS나 객체관계 DBMS는 다음의 방법을 따라 객체 저장소를 설계할 수 있는데 OID를 제공하는 제품은 O ID 필드로 시스템 OID를 사용해야 하고, 텍스트 형태의 데이터는 Text 관련 클래스로 표현한다.

또한, 그래픽 형태의 데이터는 Graphic 관련 클래스로 표현한다. Set value 데이터 타입을 갖는 속성은 그대로 set value 속성으로 매핑한다.

객체 저장소 관련 메소드를 DBMS의 method나 function으로 설계 구현할 수 있다. 또한 텍스트, 그래픽 등에 대한 인덱싱과 내용 검색을 지원하는 시스템의 경우, 내용 검색을 위한 인덱스를 해당 필드에 대해 생성한다.

클래스간의 종속 관계를 스키마상에서 표현할 수 있는 경우는 직접 그 기능을 이용한다. 이것은 스키마상에 종속 관계나 공유/독점 관계를 직접 표현한다.

마지막으로 DDL/DCL로 반영되지 못한 사항은 응용 프로그램에서 설계 구현할 수 있다.



(그림 5) 통합 객체 관리 저장소 인터페이스

위 모델을 적용하여 프레임워크 저장소 구조를 연결할 수 있고, 인터넷 환경을 기반으로 한 클라이언트에서 관리자는 프레임워크의 정보를 등록하고 관리 검색등의 요구를 서버 측에 전송하면 서버는 클라이언트로부터 요청 받은 프레임워크 관리를 위한 연산을 수행하여 클라이언트로 응답을 한다. 다음은 객체저장소 스키마의 설계로 관계 DBMS에서 구현 가능한 관계 스키마의 구조는 다음과 같다.

```
create table Object
( oid number, // OID를 숫자
  version_number float,
  patent_version number,
  primary key(oid));
create table Child_Ver //set value를 표현
( oid number,
  child_version_id number,
  primary key(oid, child_version_id));
```

5. 결론 및 향후 연구과제

본 논문에서는 소프트웨어 생명주기의 효율적인 관리와 분산객체 환경에서 소프트웨어 산출물들과 연결, 기존의 제어 메카니즘 부재로 인해 객체 저장소에 효율적인 저장을 하기 위해서 통합 관리 모델을 제시한다. 객체지향 프레임워크 개발에 필요한 프레임워크 저장소와 저장소를 구성하는 각 모듈별 필요한 기능을 정의하였다. 향후 설계된 구조와 기능을 제공하는 객체지향 응용 프레임워크 저장소를 구현하고 관련된 클래스들의 하이퍼텍스트 구조를 사용하여 분산 저장소에 대한 연구가 필요하다.

참고 문헌

[1]Ralph E Johnson, Brian Foote, "Designing reusable classes", june/july 1991
 [2]r.e. jhpson, et al, "reusing objected-oriented designs", univ. od illinoise tech, report uiucdcs91-1696, 1991
 [3]J. Bosch, Peter molin, michadl mattson, "object-oriented frameuorks problems & experiences", july,1997
 [4]최한석, 이상돈, "응용 프레임워크 재사용 정보관리 방안" 98추계 학술 발표논문집, 정보과학회 526-528, 1998
 [5] 선수균, 송영재, "통합객체지향 관리기 중점을 둔 F77/J++ 생성기 설계", 99년 추계 학술발표집 한국정보과학회, 1999.
 [6]김행근,현창문,김성원 "객체지향 프레임워크 재사용을 위한 정보저장소의 설계에 관한 연구", 99년 춘계 학술발표집 한국정보처리학회, 1999.