

영역 프레임워크와 프레임워크 지원 도구 개발에 관한 연구

김 강 태[†] · 배 제 민^{††} · 이 경 환^{†††}

요 약

본 연구에서는 웹을 기반으로 웹 클라이언트 간의 의사소통과 협동 작업을 가능하게 하는 웹 기반 협동 시스템의 구축에 필요한 상위레벨의 지식들과 재사용 컴포넌트들인 자바 애플릿과 어플리케이션들을 컴포넌트로 갖는 객체지향 프레임워크를 개발하였다. 프레임워크의 부품들은 재사용을 지원하기 위해서 설계 정보, 소스 코드, 실행 코드 등을 포함한다. 관련된 연구를 통해서 얻어진 영역 프레임워크 구축 프로세스를 통해서 웹 협동 시스템 프레임워크를 구축하였다. 웹 클라이언트 간의 의사소통과 협동 작업을 지원하기 위해서 웹 기반 협동 시스템의 서브 시스템으로 사용자 관리 시스템, 채팅, 게시판, 질의 응답 시스템, 전자 게시판 등을 정의하였다. 프레임워크의 사용성을 높이기 위해서 지원 도구로 소스 코드 생성기, 클래스 편집기, 이해지원 도구 등의 비주얼 도구를 개발하였다.

A study on Construction of Domain Framework and Framework Supporting Tools

Kang-Tae Kim[†] · Je-Min Bae^{††} · Kyung-Whan Lee^{†††}

ABSTRACT

In this paper, we built an Object Oriented Framework for Web collaboration system which contains high level analysis information and design knowledge for java applets and applications that enable web clients to communicate and collaborate each other. Components of framework contain design information, source codes and executable codes for reuse. We had defined a development method for domain framework in related works and built a web collaboration system framework following it. We defined subsystem of web collaboration system : user management system, chatting, BBS, Q&A system, board service system for the communication and collaboration between web clients. We also developed visual tools for framework usability : source code generator, class editor, knowledge supporting tools.

1. 서 론

소프트웨어 개발 방법은 개발자로 하여금 소프트웨어를 좀 더 빨리 개발할 수 있고 사용자에게 좀 더 품

질 높은 프로그램을 전달할 수 있도록 개발되어 왔다. 그리고 이러한 요구에 부응할 수 있는 객체지향 기술이 광범위하게 사용되었으며, 객체지향 프레임워크는 객체지향 기술을 가장 효과적으로 적용할 수 있는 기술 중의 하나이다.[1]

객체지향 소프트웨어를 설계하는 일은 매우 어려운 일이며, 재사용 가능한 객체지향 소프트웨어를 설계한다는 것은 더욱 어려운 일이다. 적절한 객체를 찾고,

† 준 회원 : 중앙대학교 대학원 컴퓨터공학과
†† 중신회원 : 관동대학교 컴퓨터교육과 교수
††† 정 회원 : 중앙대학교 대학원 컴퓨터공학과 교수
논문접수 : 1998년 9월 28일, 심사완료 : 1999년 3월 29일

그 객체를 알맞게 분류하여 클래스를 만들고, 클래스의 인터페이스와 상속관계를 정의하고, 클래스간의 관계를 정의한다는 것은 결코 쉬운 일이 아니다. 객체지향 재사용시스템에서의 재사용 대상은 클래스 라이브러리이며, 이를 재사용하기 위해서는 전체 인터페이스에 대한 이해가 필요하고, 응용 프로그램개발을 위해서는 미리 정의된 조건에 따라 각 클래스들을 조립하는 프로세스가 요구된다. 결국 대형 라이브러리 전체를 이해하고 기초 클래스를 선정한 후 상호 연관성을 제어하는 코드를 개발하게 된다. 이로 인하여 재사용성은 증가되었지만 클래스 라이브러리의 습득에 대한 부담과 라이브러리 재사용으로 인한 추가 부담이 생기게 되어 실질적으로는 재사용을 향상시켜주지 못하며, 상호 운영성에도 많은 제약점을 갖고 있다[2].

이러한 문제에 대한 해결책을 제시할 수 있는 것이 바로 객체지향 프레임워크이다. 객체지향 프레임워크는 특정한 영역에 필요한 서로 연관성이 있는 객체들과 그들의 상호작용을 함께 나타내는 일종의 설계안이다. 객체지향 프레임워크는 객체들과 그들의 상호작용은 물론 영역 내의 어플리케이션들에 사용된 공통된 지식들을 모두 포함한 메타 해결안이다[4]. 본 논문에서는 객체지향 프레임워크를 편의상 프레임워크로 명명하기로 한다.

본 논문에서는 인터넷에서 웹 접속자들이 상호 협동 작업을 할 수 있도록 지원하는 웹 기반 협동 시스템의 구축을 지원하는 영역 프레임워크를 구축한다. 본 논문과 연관된 연구에서 정의한 프레임워크 구축 프로세스에 따라 디자인패턴을 사용하여 프레임워크의 아키텍처를 설계하였으며, 프레임워크의 컴포넌트 라이브러리를 구축하였고, 프레임워크의 사용성을 높이기 위한 도구로 프레임워크 지원 시스템을 개발하였다.

본 논문에서 제시하는 바는 다음과 같다.

- 첫째, 프레임워크 구축 프로세스를 이용하여 프레임워크의 해당 영역에 필요한 컴포넌트 클래스 라이브러리를 구축하였고,
- 둘째, 해당영역의 설계지식을 제공하여 개발자의 프레임워크 사용성을 높이는 설계지원 도구와 컴포넌트 라이브러리의 사용성을 높여주는 프레임워크 지원 도구를 개발하였다.

컴포넌트 라이브러리를 구성하는 각 컴포넌트의 구

현언어는 JAVA 1.1을 이용하며 컴포넌트 라이브러리는 마이크로소프트사의 데이터베이스를 사용한다. 본 논문은 2장에서 프레임워크의 전반적인 내용을 관련연구를 통해서 알아보고, 3장에서는 관련 프레임워크 구축 프로세스에 따라서 웹 기반 협동 시스템의 영역을 지원하는 프레임워크를 구축하며, 4장에서는 프레임워크의 지원 도구를 구현한다. 그리고 끝으로 5장에서 결론 및 향후 연구 방향을 제시하도록 한다.

2. 프레임워크 관련 연구

2.1 프레임워크

프레임워크는 소프트웨어 개발 프로세스를 개선시키기 위한 도구라고 할 수 있다[9]. 프레임워크의 사용자는 프레임워크에서 제공하는 클래스와 오퍼레이션을 현재의 문제를 해결할 수 있도록 수정, 확장하여 프레임워크의 행위를 특성화하여 어플리케이션을 개발할 수 있다[5]. 프레임워크는 특정한 영역의 문제를 해결하는데 필요한 프로그래밍 지식을 가지고 있다. 이러한 지식 중에서 해당 영역에서 일반적인 지식들은 개발자에게 숨기고 문제영역에서 특성화에 필요한 부분만을 노출시키게 된다. 프레임워크의 재사용성을 이해하기 위해서는 기존의 객체지향 개발 방법과는 구별되는 프레임워크의 특징을 이해해야 한다. 즉, 프레임워크는 이미 검증된 소프트웨어 설계안과 이미 개발되어 있는 컴포넌트를 이용해서 소프트웨어의 개발비용을 절감시키고 소프트웨어의 품질을 높일 수 있는 기술이지만 일반적인 객체지향 방법의 재사용 영역과는 달리 특정한 어플리케이션의 영역에서의 재사용성을 지원한다는 것이다[3]. 프레임워크는 어플리케이션의 구현의 뼈대가 되는 추상 클래스(abstract class)와 구체 클래스(concrete class) 그리고 그들 간의 협동관계의 집합으로 정의할 수 있다[6]. 또한 프레임워크는 클래스 라이브러리의 컴포넌트에 대한 재사용 정보 및 어플리케이션 설계안들을 제공한다[7].

2.2 화이트박스 프레임워크와 블랙박스 프레임워크

프레임워크는 제공되는 라이브러리의 사용성에 따라서 화이트박스 프레임워크와 블랙박스 프레임워크로 나눌 수 있다. 프레임워크에서 제공되는 라이브러리 내의 클래스에서 상속을 받아 새로운 객체를 생성하는 방식으로 사용되는 프레임워크를 화이트박스 프레임워크라고 한다. 즉, "상속"에 그 기초를 두고 있기 때문

에 화이트박스 프레임워크에서의 객체들은 "Is_A" 관계를 갖는다[8]. 결론적으로 화이트박스 프레임워크는 해당 영역의 어플리케이션의 구현에 대한 뼈대이며, 사용자가 재정의한 서브클래스들이 그 뼈대에 첨가된다.

블랙박스 프레임워크는 객체 "합성"에 그 기초를 두고 있다. 즉, 새로운 기능이 새로운 객체들을 첨가하거나 합성함으로써 얻어진다는 것을 의미한다. 어플리케이션의 개발자는 어플리케이션의 특정한 행위를 구현할 때 이미 존재하는 여러 컴포넌트들을 조합한다. 객체의 합성은 상속에 비해서 유동적으로 변경될 수 있기 때문에 일반적으로 상속에 비해서 그 유연성이 더 뛰어나다고 알려져 있다. 객체 합성은 "Has_A" 관계를 갖는다. 일반적으로 프레임워크에는 상속기반의 화이트박스 프레임워크와 합성기반의 블랙박스 프레임워크가 공존한다. 일반적으로 화이트 박스 프레임워크로부터 시작하여 시스템이 개발되는 프로세스에서 시스템의 영역에 대한 지식의 이해도가 높아지면서 점차 블랙박스 프레임워크로 진화해 나간다. 블랙박스 프레임워크는 결국 프레임워크의 개발의 최종목표가 되지만 결코 블랙박스만의 프레임워크만은 존재할 수 없으며, 화이트박스 프레임워크 역시 최종 프레임워크가 될 수 있다[8].

2.3 핫 스팟(hot-spot)

프레임워크 개발의 성공요소는 해당 영역에서의 핫 스팟의 식별여부에 달려있다. 핫 스팟은 어플리케이션의 개발자가 프레임워크를 확장할 수 있는 장소를 말하며 이는 컴퓨터에 장착하는 카드를 설치하는 슬롯과 같은 것으로 생각할 수 있다. 핫 스팟은 어플리케이션 영역의 슬롯으로 이 슬롯에 어플리케이션의 개발자가 개발하고자 하는 어플리케이션에서 고유하게 구현되어야 할 구현물을 끼워 넣어 어플리케이션 영역의 공통 구현요소를 재사용하여 개발하고자 하는 어플리케이션 고유의 행위를 정의할 수 있는 곳으로, 핫 스팟은 프레임워크에 유연성을 제공한다[14].

2.4 컴포넌트 라이브러리

프레임워크에서의 컴포넌트 라이브러리는 프레임워크에서 제공하는 어플리케이션 영역에서 필요한 구현 결과물로서의 객체들을 가진다. 프레임워크는 잘 정의된 구체 컴포넌트들의 라이브러리를 제공함으로써 프레임워크의 사용성을 높인다. 하지만 프레임워크에서 제공되는 객체들이 실제 프레임워크의 사용자에게 유용한

것인지에 대한 판단은 쉬운 것이 아니다. 그러므로 프레임워크의 컴포넌트 라이브러리는 일반적인 라이브러리의 개발 프로세스처럼 단순하고 작은 라이브러리의 개발을 기초로 하여 그 라이브러리를 계속 확장하는 프로세스를 갖는다. 프레임워크 컴포넌트 라이브러리의 확장 프로세스에서 많은 객체들이 라이브러리에 등록되고 삭제되는데 이를 통해서 프레임워크에서 반드시 제공되어야 할 객체들이 식별될 수 있으며, 이를 통해서 프레임워크의 영역에서 정의되어야 할 영역정보에 대한 추상화가 가능하다[12]. 영역정보에 대한 추상화는 추상 클래스를 통해서 제공되는데 이러한 클래스에 대한 식별과 클래스를 구현하는 것이 프레임워크 컴포넌트 라이브러리의 확장 프로세스라고 할 수 있다.

3. 프레임워크 구축

3.1 영역 프레임워크 구축 프로세스

본 논문과 관련하여 영역 프레임워크를 구축하는 프로세스로서 중앙대학교 소프트웨어 공학 연구실에서 제안한 '영역 프레임워크 개발 프로세스'를 사용하도록 한다[15].

이 절에서는 '영역 프레임워크 개발 프로세스'에 대해서 간략히 설명하도록 하며, 그 프로세스는 다음과 같다.

① 프레임워크의 영역 분석

프레임워크가 지원할 영역을 정의하고 분석하는 단계로 프레임워크의 개발 공정의 초기 입력을 정의한다. 개발할 프레임워크가 다루는 문제영역을 명시하는 영역 정의와 정의된 영역 내에 존재하는 주요 클래스들을 나타내는 정적 모델들이 이 단계에서의 출력물이 된다. 성공적인 프레임워크의 구축을 위해서 기존에 존재하고 있는 문제 영역에 대한 해결책에 대한 조사 또한 필요하다.

② 영역내의 공통 요구사항 추출 및 분석사항 추출

프레임워크에서 제공할 일반적인 설계안과 프레임워크의 컴포넌트 라이브러리를 구성하는 컴포넌트를 개발하기 위한 일련의 분석 프로세스이다. 영역내의 여러 어플리케이션 간에 존재하는 공통 사항을 추출하기 위한 방법을 도입하며 이를 위해서 문제영역에서 3개의 어플리케이션에 대한 요구사항과 분석사항을 추출한다[4]. 이때 3 examples 디자인 패턴을 도입하여 3개의 영역 어플리케이션을 구축

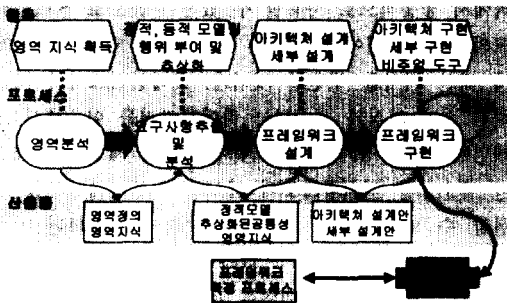
한 지식과 경험을 바탕으로 프레임워크를 구축한다.

③ 프레임워크의 설계

컴포넌트 개발을 위해 요구사항과 분석사항으로부터 객체 모델을 추출하여 작성하고, 이에 디자인 패턴을 적용하여 문제영역에 대한 해결책으로서의 소프트웨어 아키텍처와 설계안을 제시한다. 디자인 패턴은 일반적으로 사용되는 객체지향 설계에 대한 추상화된 형태의 지식으로 객체지향 분석 및 설계 작업에서 재사용될 수 있다. 본 논문에서는 프레임워크의 구축과정에서 얻어진 분석 및 설계에 대한 정보를 디자인 패턴의 형태로 표현하여 프레임워크의 사용자의 이해성을 높이고자 하였다[13]. 프레임워크는 문제영역의 어플리케이션들의 공통적인 부분에 대한 구현물들의 집합으로 생각할 수 있다. 프레임워크의 설계는 프레임워크의 아키텍처 설계와 세부 설계의 서브 프로세스를 갖는다. 아키텍처의 설계 프로세스는 분석 프로세스의 결과물을 이용해서 프레임워크와 문제영역의 어플리케이션들에 대한 추상화 단계가 높은 설계지식을 추출하는 프로세스이다. 아키텍처의 설계는 시스템의 구현에 핵심적인 객체들과 그 객체들 간의 협동관계를 식별하는 것이며 필요에 따라서 시스템을 분할한 서브 시스템을 대상으로 설계할 수 있다.

④ 프레임워크의 컴포넌트 라이브러리 구축

프레임워크의 컴포넌트 개발하여 컴포넌트 라이브러리를 구축한다. 문제영역의 성격에 따라서 프레임워크의 컴포넌트 및 지원도구를 구축하는 프로세스는 별도로 정의한다. 이는 문제영역의 성격에 따라서 구축해야 할 컴포넌트 및 지원도구가 달라지므로 프레임워크의 개발과정에서는 이를 명시하지 않는다. (그림 1)은 '영역 프레임워크 구축 프로세스'이다.

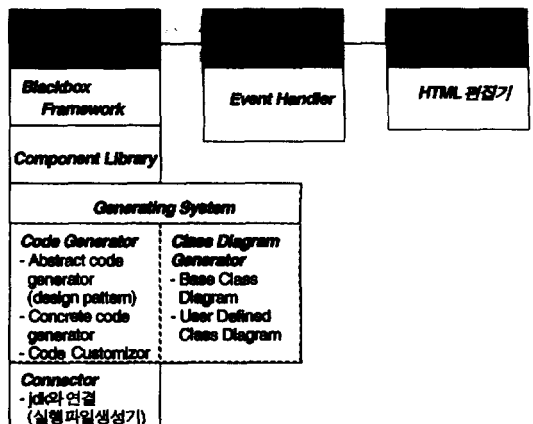


(그림 1) 영역 프레임워크 구축 프로세스

3.2 프레임워크의 구축

3.2.1 웹 협동 시스템 프레임워크 설계

본 논문에서 구축하는 웹 협동 시스템 프레임워크의 전체 구조는 어플리케이션 프레임워크와 웹 콘텐츠 프레임워크의 두 부분으로 이루어진다. 어플리케이션 프레임워크를 통해서 웹 기반 협동 시스템을 이루는 자바 어플리케이션이나 애플릿을 개발하고, 웹 콘텐츠 프레임워크를 통해서 개발한 어플리케이션이나 애플릿을 웹상에서 서비스하기 위해서 HTML tag를 가진 HTML 문서를 생성시킨다. 이때 자바 어플리케이션 및 애플릿의 영역을 웹을 기반으로 여러 원격 사용자 간의 협동작업을 가능하게 하는 웹 협동 시스템으로 한정한다. (그림 2)는 웹 협동 시스템의 전체적인 구조를 설계한 것이다. 어플리케이션 프레임워크는 웹 협동 시스템을 자바 언어로 구현할 수 있도록 지원하는 프레임워크이다. 본 프레임워크는 프레임워크의 성격상 영역 프레임워크로 분류할 수 있다. 웹 콘텐츠 프레임워크는 어플리케이션 프레임워크와 더불어 웹 사이트의 콘텐츠에 해당하는 HTML 문서를 작성할 수 있도록 지원하는 프레임워크로서 타겟 언어는 HTML이다. 연결기는 어플리케이션 프레임워크와 웹 콘텐츠 프레임워크를 연결하는 기능을 가지며 이를 통해서 어플리케이션 프레임워크와 웹 콘텐츠 프레임워크에서 제공되는 컴포넌트를 이용하여 웹 협동 시스템을 구축할 수 있다.

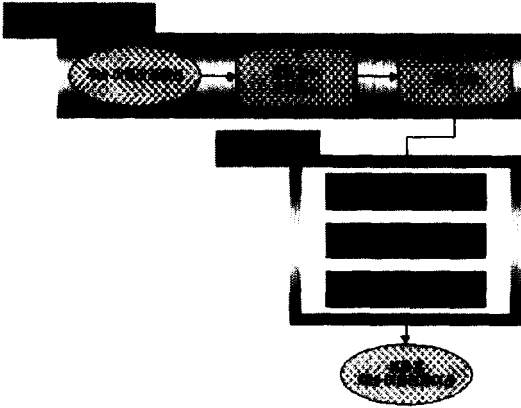


(그림 2) 웹 협동 시스템을 위한 프레임워크 구조

3.2.2 어플리케이션 프레임워크 구축

본 프레임워크는 2개의 서브 프로세스에 의해서 개

발되었다. 프레임워크에서 제공하는 코드 및 설계안에 대한 정보를 추출하여 이를 프레임워크의 컴포넌트화 하는 '프레임워크 컴포넌트 구축 프로세스'와 사용자가 프레임워크를 쉽게 사용할 수 있도록 지원하는 '비주얼 도구 구축 프로세스'이다.



(그림 3) 프레임워크 및 지원도구 구축 프로세스

(그림 3)에서 프레임워크 컴포넌트 구축 프로세스는 프레임워크의 컴포넌트를 구축한 과정을 보여주며, 비주얼 도구 구축 프로세스에서는 본 프레임워크에서 개발한 프레임워크 지원도구를 보여준다.

1) 프레임워크 컴포넌트 구축

프레임워크에서 사용할 코드의 DB를 구축하여 프레임워크의 구조 및 객체들을 정의하고 구현하였는데 해당 활동은 다음과 같다.

- ① 기존의 프로그램 분석
- ② 영역 내의 프로그램 간의 공통성 추출 및 추상화
- ③ 추상화 식별을 통한 코드 분석과 코드 분할
- ④ 공통 객체의 추출과 공통 구조 추출
- ⑤ 추출된 객체에 책임 부여
- ⑥ 객체 간의 협동관계 정의
- ⑦ 코드 DB 구축

2) 비주얼 도구 구축

사용자의 프레임워크에 대한 이해성을 높여주고 이용을 쉽게 해주는 비주얼 도구를 다음과 같이 개발하였다.

- ① 코드 생성기 개발 : 코드 DB에서 사용자가 원하는

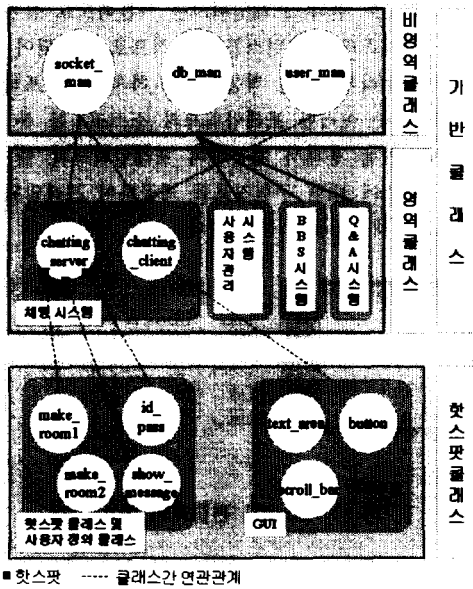
코드를 추출하는 도구로 해당 영역 어플리케이션의 템플릿 컴포넌트와 핫 스팟에 연결될 수 있는 컴포넌트를 구분하고 템플릿 컴포넌트에서 핫 스팟을 명시하여 개발한다.

- ② 객체 관리기 개발 : 코드 DB와 코드 생성기를 연결하는 도구로 실제 DB에 저장되어 있는 객체를 관리하는 도구로 코드 생성기 개발에서 정의한 컴포넌트의 분류안을 통해서 만들어진 컴포넌트를 관리할 수 있도록 개발한다.
- ③ 이해 지원 시스템 개발 : 사용자의 이해를 돕기 위해서 사용되는 디자인 패턴에 대한 문서와 사용된 코드 및 설계에 대한 객체 모형 등을 보여주는 도구로 프레임워크 라이브러리의 구축에 사용된 디자인 패턴과 연관 클래스들의 OMT 모형을 제공할 수 있도록 개발한다.

3.2.3 프레임워크 컴포넌트 라이브러리 구축

프레임워크의 컴포넌트 라이브러리는 서버 시스템으로 정의한 사용자 관리 시스템, 채팅, BBS, 질의 응답 시스템에 대한 컴포넌트를 구현하여 구축한다. 각 서버 시스템에 대한 영역을 정의하고 공통 요구사항을 추출하고 추상화된 설계안을 바탕으로 핫 스팟을 정의할 수 있는 형태의 라이브러리를 구축한다. 컴포넌트 라이브러리의 구축에서 가장 중요한 것은 프레임워크에서 확장이 일어날 수 있는 부분인 핫 스팟을 정의하는 것으로 본 논문에서는 핫 스팟을 정의하기 위해서 자바에서 제공하는 interface 객체를 이용한다.[10,11] interface 객체는 C++의 abstract 클래스와 같이 클래스의 일부분은 구현되어 있고, 나머지 부분 즉 프레임워크의 개발자가 핫 스팟으로 정의하여 추후의 확장을 가능하게 할 부분은 템플릿으로 정의하여 그 설계안을 제공한다. 부분적으로 구현되어 있는 부분은 해당 영역에서 공통적으로 사용할 수 있는 부분으로 프레임워크의 구축 과정에서 산출된 상위레벨의 분석 및 설계에 대한 지식을 구현한 것이다. 프레임워크에서 제공하는 설계 지식과 제공되는 라이브러리를 이용해서 프레임워크의 사용자는 핫 스팟에 연결될 새로운 클래스나 메소드를 정의함으로써 새로운 어플리케이션을 구현할 수 있다. (그림 4)는 프레임워크에 정의된 기반 클래스, 핫 스팟 클래스를 나타낸다.

기반 클래스는 사용영역에 따라서 비영역 클래스와 영역 클래스로 구분하였다. 비영역 클래스는 프레임워



(그림 4) 프레임워크에 정의된 클래스

크의 전체 영역인 웹 협동 시스템의 사용자 관리 시스템, 채팅, BBS, 질의 응답 시스템 어플리케이션에서 공통적으로 사용할 수 있는 클래스를 정의하였고 영역 클래스는 해당 어플리케이션 영역에서만 사용될 수 있는 클래스를 정의하였다. 기반 클래스는 어플리케이션의 구조를 제공하고 어플리케이션의 실행 흐름을 가지고 있으므로 프레임워크의 핵심 클래스라고 할 수 있으며 핫 스팟 클래스는 기반 클래스인 비영역 클래스와 영역 클래스에 정의된 핫 스팟에 연결될 수 있는 클래스로 기본적으로 구현된 연결될 클래스는 미리 정의되어 있으며, 사용자는 코드 생성기의 코드 편집기를 통해서 핫 스팟에 연결될 새로운 클래스를 정의하여 새로운 기능을 제공할 수 있다.

어플리케이션의 기본적인 구조와 제어의 흐름을 가지고 있는 기반 클래스는 3.1의 ②에서 언급된 3 examples 패턴을 이용해서 추상화된 정보를 abstract 클래스 형태로 구현하였다.

채팅 시스템의 경우 기반 클래스의 비영역 클래스로 socket_man, db_man, user_man이 있고 영역 클래스로는 chatting_server, chatting_client 클래스가 사용되고 있다. socket_man 클래스는 웹 협동 시스템에서 TCP/IP 네트워크를 담당하는 클래스이며 웹 협동 시스템의 모든 어플리케이션에서 각 클라이언트와 서버 간의

socket을 생성하고 이곳에 들어오는 메시지를 처리하는 기능을 갖는다. user_man 클래스는 웹 협동 시스템에서 등록된 사용자를 관리하기 위해서 사용자가 등록된 데이터 베이스를 검색하는 기능을 갖는다. 채팅 시스템의 영역 클래스의 chatting_server 클래스는 채팅의 서버 기능을 제공하고 chatting_client 클래스는 채팅의 클라이언트 기능을 제공한다. chatting_server 클래스는 채팅 시스템 서버에서 제공하는 기능에 따라서 정의된 핫 스팟에 핫 스팟 클래스를 연결할 수 있으며, 정의된 핫 스팟은 방의 개수를 정의하는 make_room, 사용자 검증의 유무를 결정하는 id_pass, 메시지 보여 주기의 형태를 결정하는 show_message가 있다. 각 핫 스팟에 새로운 기능을 제공할 필요가 있을 때에는 새로운 클래스를 정의하여 핫 스팟에 연결시킨다. chatting_client 클래스는 메시지의 입력과 메시지를 보는 기능을 제공하므로 기본적인 GUI 클래스를 그대로 이용한다. chatting_server 클래스는 GUI가 없으므로 GUI 클래스와는 연결되지 않는다.

3.2.4 웹 콘텐츠 프레임워크의 구축

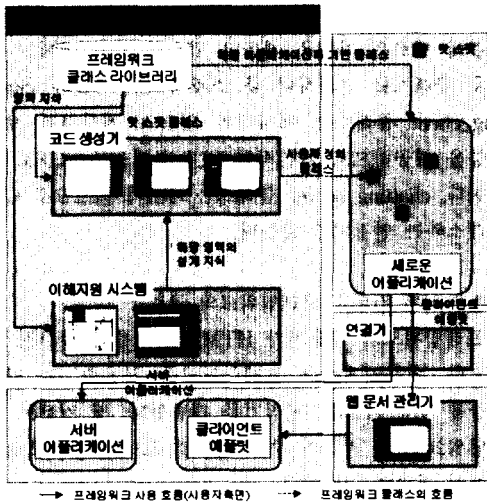
프레임워크의 사용자가 HTML 문서를 생성시킬 수 있으며, 어플리케이션 프레임워크를 사용할 수 있는 기능을 갖는 HTML 편집기를 개발했다. 웹 콘텐츠 프레임워크는 그 기능이 주로 비주얼 도구에 있으므로 4장에서 설명하도록 한다.

4. 프레임워크의 사용과 비주얼 도구

4.1 프레임워크의 사용

구축된 프레임워크의 사용은 비주얼 도구와 밀접하게 관련된다. 프레임워크의 라이브러리 자체는 사용자가 이해하기가 매우 어렵고, 또한 재사용하기도 쉽지 않다. 그러므로 프레임워크의 사용성을 높여주기 위한 비주얼 도구가 필수적이다. 본 프레임워크에서는 (그림 5)의 형태로 프레임워크를 사용한다. 전체적인 프레임워크의 사용은 어플리케이션 프레임워크를 이용하여 어플리케이션 프레임워크에서 제공하는 해당 영역의 기반 클래스를 어플리케이션의 구조로 사용하여 기반 클래스에 정의된 핫 스팟에 연결될 핫 스팟 클래스를 코드 생성기를 통해서 구현하여 프레임워크의 사용자가 원하는 어플리케이션을 구현할 수 있다. 해당 어플리케이션이 클라이언트 애플릿인 경우 웹 문서 관리기

를 통해서 해당 애플릿을 특정한 웹페이지에 위치시키며, 서버 어플리케이션인 경우에는 웹 문서 관리기를 사용하지 않는다. 웹 콘텐츠 프레임워크에서 제공하는 HTML 컴포넌트들을 이용해서 웹 페이지 내에 어플리케이션 프레임워크를 통해서 만들어낸 시스템을 웹 상에서 서비스할 수 있다. 프레임워크의 사용자는 개발하고자 하는 어플리케이션에 대해서 프레임워크에서 제공하는 분석 및 설계 지식을 이해 지원 시스템을 통해서 얻어낼 수 있으며, 설계 지식과 연관된 소스 코드를 소스코드 생성기를 통해서 얻어낼 수 있다.



(그림 5) 프레임워크를 이용한 시스템 구축 프로세스

비주얼 도구는 코드 생성기, 이해 지원 시스템, 연결기, 웹 문서 관리기로 구성된다. 코드 생성기, 이해 지원 시스템, 연결기는 어플리케이션 프레임워크를 사용하여 이미 정의되어 있는 설계지식 및 컴포넌트를 합성하여 프로그램을 개발할 때 사용되는 도구이며, 웹 문서 관리기는 만들어진 웹 서비스 프로그램과 웹 문서에 들어갈 각종 멀티미디어 객체들을 관리하는 도구이다.

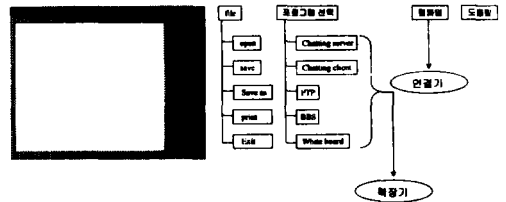
4.2 코드 생성기

프레임워크를 이용하여 사용자가 원하는 프로그램의 Abstract 클래스 코드, Concrete 클래스 코드, Interface 코드를 생성시킨다.

① 코드 편집기

프레임워크가 생성한 코드를 보여주는 편집기이며, 프레임워크의 사용자가 만들어낼 소스코드의 편집

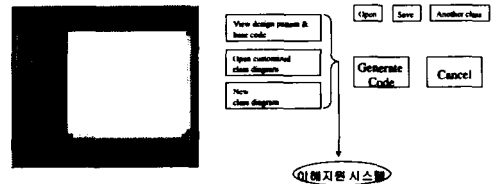
기의 역할도 한다. 프레임워크는 기본적으로 실행 가능한 클래스의 인스턴스들이 동적으로 연결되어 하나의 어플리케이션을 작동하도록 하지만 본 프레임워크에서는 이러한 프레임워크의 소스코드를 본 코드 편집기를 통해서 제공하여 프레임워크의 핫스팟 클래스에 대한 구현정보를 제공하여 프레임워크의 코드 재사용을 가능케 한다.



(그림 6) 소스코드 편집기와 메뉴 구조

② 클래스 편집기

프레임워크에서 사용할 코드가 Concrete 클래스가 아닌, Abstract 클래스를 사용할 때는 그 클래스에서 상속을 받는 사용자가 구현해야 할 클래스에 대한 편집과 Interface를 사용할 경우 사용자가 재정의할 클래스 코드를 편집하는 편집기이다.

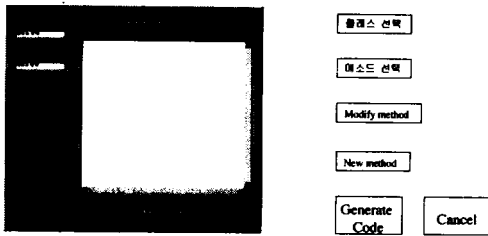


(그림 7) 클래스 편집기와 메뉴 구조

즉, 사용자가 재정의하거나 새로이 정의할 핫스팟 클래스를 편집할 수 있는 기능을 제공하며, 이곳에서 편집된 클래스의 코드는 코드 생성기에 연결되고 이를 컴파일하여 핫스팟을 통해서 프레임워크와 연결된다.

③ 메소드 편집기

프레임워크가 제공하는 클래스내의 특정한 메소드의 구현사항을 바꾸고자 할 때 사용되는 편집기로 위의 클래스 편집기 내의 특정한 메소드만을 새로이 정의할 필요가 있을 때 사용되며 클래스 편집기와 마찬가지로 핫스팟을 통해서 프레임워크에 연결된다.



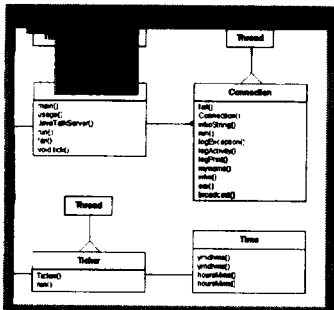
(그림 8) 메소드 편집기와 메뉴 구조

4.3 이해 지원 시스템

이해 지원 시스템은 프레임워크에 사용되는 클래스나 객체, 그리고 그들 간의 연결 형태, 사용된 디자인 패턴 등을 사용자에게 제공함으로써 프레임워크에 대한 이해를 돕고 궁극적으로 프로그래밍을 쉽게 할 수 있는 환경을 제공하는 데에 그 목적이 있다. 결국 이해 지원 시스템은 사용자에게 프레임워크가 제공하는 설계안에 사용되는 클래스와 객체들 그리고 그들의 연결관계를 나타내는 OMT 모형과 그 설계안에 사용되는 디자인 패턴을 보여준다.

① 기반 클래스 다이어그램 생성기

프레임워크에서 제공하는 설계안에서 사용되는 클래스나 객체들 그리고 그들 간의 연결관계를 OMT 모형으로 보여준다. 기반 클래스 다이어그램 생성기를 통해서 프레임워크를 구축할 때 정의한 서브 시스템들의 상위레벨의 지식 즉, 분석 및 설계 지식을 제공할 수 있다. 분석 지식은 요구분석 문서의 형태로 제공되며 설계 지식은 클래스 다이어그램과 디자인 패턴을 통해서 제공된다. 서브 시스템의 설계에 디자인 패턴이 적용된 경우에는 사용된 디자인 패턴에 대한 설명과 해당 영역에서 어떻게 사용되

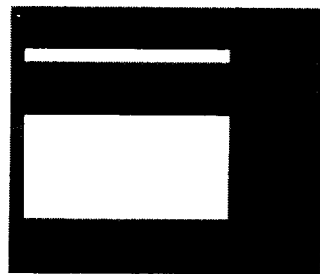


(그림 9) 기반 클래스 다이어그램 생성기

고 있는 가에 대한 정보를 제공한다. (그림 9)는 채팅 서버에서 정의된 설계지식인 OMT 모형을 제공하는 기반 클래스 다이어그램 생성기이다. 메뉴의 내용은 사용된 디자인 패턴을 Gamma의 디자인 패턴 카탈로그의 형식으로 볼 수 있도록 한다.[13]

② 사용자 정의 클래스 다이어그램 생성기

사용자가 새로 정의하여 생성시킨 클래스의 다이어그램을 만들어 내는 도구이다.



(그림 10) 사용자 정의 클래스 다이어그램 생성기

③ 디자인 패턴 도구

상위레벨의 분석 지식과 더불어 본 프레임워크에서는 웹 기반 협동 시스템의 서브 시스템으로 정의한 사용자 관리도구, 채팅, BBS 등에 대한 설계안을 제공한다. 서브 시스템의 컴포넌트에 대한 설계안은 위에서 설명한 기반 클래스 도구에서 OMT 모형의 형태로 제공되며, 컴포넌트들이 연결되어 하나의 서브 시스템을 이루는데 필요한 정보는 디자인 패턴을 이용해서 제공한다. 디자인 패턴 도구는 해당 설계안과 연관되어 있는 디자인 패턴 정보를 제공한다. (그림 9)의 기반 클래스 도구의 서브 시스템 형태로 구현되어 있으며, 기반 클래스 도구를 사용하면서 해당 어플리케이션의 설계에 사용된 디자인 패턴을 찾아 핫 스팟 클래스의 구현에 대한 기반 설계지식으로 사용할 수 있다.

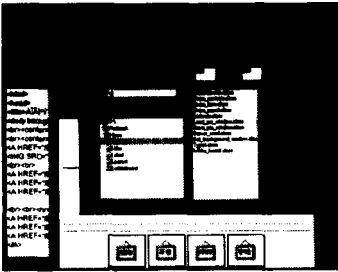
4.4 연결기

코드 생성기를 통해서 생성한 자바코드를 파일로 저장한 후, 연결기를 통해서 그 파일을 자바 컴파일러로 컴파일 할 수 있도록 자바 컴파일러와 연결시킨다. 별도의 윈도우는 존재하지 않으며, 단순히 사용자가 정의한 핫 스팟 클래스를 컴파일할 수 있도록 컴파일러

에 연결시키는 기능을 가진다.

4.5 웹 문서 관리기

웹 문서 관리기는 어플리케이션 프레임워크를 이용해서 개발한 시스템의 클라이언트 부분인 자바 애플릿을 웹 문서에 올리기 위한 목적으로 개발하였다. 웹 문서 편집기의 기능으로는 HTML tag 편집 기능, 브라우저 기능, 애플릿 연결 기능이 있다. HTML tag 편집기능으로 (그림 11)의 버튼에서 보듯이 HTML 문서의 제목, 패러그래프, 링크 등을 쉽게 입력할 수 있는 기능이 있고 브라우저 기능을 인터넷 익스플로러 커넬 포넛트를 이용해서 편집 중인 문서를 브라우저를 통해서 볼 수 있도록 하는 기능이 있다.



(그림 11) 웹 문서 관리기

5. 결 론

본 논문에서는 영역 프레임워크의 구축 프로세스를 이용하여 영역의 분석 및 설계 지식, 어플리케이션의 아키텍처, 영역의 컴포넛트의 재사용을 지원하며, 웹 협동 시스템을 영역으로 하는 프레임워크를 구축하였다. 영역 내의 어플리케이션들에 대한 분석을 통해서 일반적인 공통 객체를 추출하고 이에 행위를 부여하며, 프레임워크의 핵심 사항인 객체 간의 협동관계를 정의하였으며 각각의 객체들은 구현된 서브시스템인 채팅, 게시판, 질의 응답 시스템의 구성요소가 된다. 또한 정의된 소프트웨어 아키텍처에서 미래에 일어날 수 있는 변경사항에 유연하게 대처하기 위해서 핫 스팟을 java 언어에서 제공하는 interface를 통해서 정의하였다. 프레임워크의 사용성을 높여주기 위해서 비주얼 지원 도구를 구현하였으며, 이는 코드 생성기, 컴포넛트 관리기, 이해지원 시스템으로 구성된다. 프레임워크를 이용해서 만들어낸 시스템을 웹 상에서 서비스할 수 있다

록 웹 문서 관리기를 구현하였다. 본 프레임워크에는 웹 협동 시스템 영역에서 영역 내의 어플리케이션에 대한 기본 구조와 제어의 흐름을 제공하므로 프레임워크의 사용자인 프로그램 개발자는 같은 영역에서 같은 작업을 반복하는데 필요한 시간 및 자원을 크게 절약할 수 있었다. 본 프레임워크는 동적인 실행 코드의 재사용 뿐 아니라 소스 코드 형태의 재사용도 제공하여 소스 코드에 익숙한 개발자에게 2가지 형태의 접근 방법을 제시할 수 있었다.

향후 연구과제로는 프레임워크에 대한 별도의 학습을 도와줄 수 있는 방법 및 이를 지원하는 도구의 개발, 프레임워크의 확장이 일어나는 부분인 핫 스팟을 기술하는 방법(스크립트 언어 등), 프레임워크의 컴포넛트 라이브러리의 재사용성에 대한 검증, 프레임워크의 서브 시스템의 확장에 대한 연구를 수행될 예정이다.

참 고 문 헌

- [1] Taligent Inc., "Leveraging Object-Oriented Frameworks," <http://www.taligent.com/Technology/WhitePapers/LeveragingFwks/LeveragingFrameworks.html>, A Taligent White Paper, 1993.
- [2] 김정아, 홍찬기, "프레임워크에 의한 재사용 기법", 한국정보과학회 소프트웨어공학회지, 10(1), pp.46-55, 1997.
- [3] Mohamed E. Fayad, Douglas C. Schmidt, "Object-Oriented Application Frameworks," CACM, pp.32-38, Vol.40, No.10, October, 1997.
- [4] Roberts D. Johnson, "Evolving Frameworks, A Pattern Language for Developing Object-Oriented Frameworks," Proceeding of PLoP'96, 3rd Annual Conference on the Pattern Language Of Programs, 1996.
- [5] 이정환, "객체모델링 기술 개발에 관한 연구", 시스템 공학 연구소 위탁과제 최종 보고서, 1996.
- [6] Desmond D'Souza, "Frameworks in Java and Catalysis," <http://www.iconcomp.com/papers/Frameworks-Catalysis/Co16.frm.html>, ICON Computing Inc., 1997.
- [7] Ralph E. Johnson, "Framework Homepage," <http://www.cs.uiuc.edu/users/johnson/frameworks.html>, 1996.

- [8] Ralph E. Johnson, Brian Foote, "Designing Re-usable Classes," Journal of Object-Oriented Programming, pp.63-76, Vol.1, No.2, June 1988.
- [9] Juha Hautamaki, "A Survey of Frameworks," Technical Report of FRED Project(University of Tampere), <http://www.cs.helsinki.fi/research/fred/reports.html>, 1997.
- [10] Arnold Gosling, "The Java Programming Language : The Java Series," Addison Wesley, pp.65-95, 1996.
- [11] Herbert Schildt, "Teach yourself C++ second edition," osborne, pp.294-342, 1994.
- [12] David G. Firesmith, "Framework : golden path to object nirvana," JOOP, 1996.
- [13] Erich Gamma, Helm Johnson, Vlissides, "design patterns : abstraction and reuse of object oriented design," European Conference on Object Oriented Programming, 1993.
- [14] Mattson M. "Frameworks FAQ's" <http://www.ide.hk-r.se/~michale/fwpages/fwfaq.html>, 1996.
- [15] Kangtae Kim, KyungWhan Lee, "developing O-O framework for web collaboration system," Proceeding of AI'99, 1999.



배 제 민

e-mail : gemini@mail.kwandong.ac.kr
 1991년 중앙대학교 전자계산학과 (이학사)
 1993년 중앙대학교 전자계산학과 (공학석사)
 1998년 중앙대학교 전자계산학과 (공학박사)

1999년~현재 관동대학교 컴퓨터교육과 조교수
 관심분야 : 객체지향 방법론, 소프트웨어 재사용, 객체지향 멀티미디어



이 경 환

e-mail : kwlee@object.cse.cau.ac.kr
 1980년 중앙대학교 대학원 응용수학 전공(이학박사)
 1982년~1983년 미국 Auburn 대학 객원교수
 1971년~현재 중앙대학교 컴퓨터공학과 교수

관심분야 : 소프트웨어 공학, 객체지향 모델링, 소프트웨어 재사용

김 강 태



e-mail : waltz@object.cse.cau.ac.kr
 1996년 중앙대학교 컴퓨터공학과 (공학사)
 1998년 중앙대학교 컴퓨터공학과 (공학석사)
 1998년~현재 중앙대학교 컴퓨터공학과 박사과정

관심분야 : 객체지향 방법론, 컴포넌트웨어, 객체지향 프레임워크, 소프트웨어 아키텍처