

# 컴포넌트기반 방법론을 사용한 프레임워크 개발에 관한 연구

김 행 곤<sup>†</sup> · 한 은 주<sup>††</sup>

요 약

프레임워크를 재사용함으로써 개발자들은 클래스 코드뿐만 아니라 도메인에 대한 폭넓은 도메인 지식을 재사용할 수 있다. 프레임워크를 재설계하는 과정을 통해 컴포넌트를 재정의하는데 이를 위해 기존의 리체지향 방법론이나 카타르시스 방법론 등이 제시되었다. 하지만 기존의 방법들은 전체과정이 폭포수형이거나 인터페이스의 설계기 구현단계에 많이 치중한다는 단점이 있다.

따라서 본 논문에서는 프레임워크 재사용을 위해 컴포넌트기반의 방법론을 제시하고 프레임워크 구축환경을 개발한다. 즉, 컴포넌트 기반의 소프트웨어 생성을 위해 실제의 도메인 지식을 입력으로 도메인을 분석하고 분석된 결과를 통해 hotspot을 식별하여 사용자와 개발자의 추가정보를 첨가하는 재설계 과정을 거친다. 이후 도메인에 따라 도메인 프레임워크와 애플리케이션 프레임워크를 생성한다. 이러한 컴포넌트 지향 방법론은 내부적으로는 컴포넌트 라이브러리 저장소를 통해 정보를 검색, 이해하여 추출하거나 혹은 합성으로 얻어진 정보는 각각 컴포넌트 정보로 분류되고 이것은 재설계시 추가정보로 사용되어진다. 이로 인해 사용자는 프레임워크 컴포넌트를 이용해 자신의 애플리케이션에 쉽게 적용 가능한 개발 환경 즉, 본 논문에서는 CORBA(Common Object Request Broker Architecture) 환경하에 컴포넌트 관리 시스템(Component Management System : CMS)을 통해 컴포넌트를 추가, 갱신, 삭제하는 기반 환경을 구축함으로써 재사용성, 사용의 용이성과 이식성을 가질 수 있다

## A Study on the Development of Framework Using Component Based Methodology

Haeng-Kon Kim<sup>†</sup> · Eun-Ju Han<sup>††</sup>

### ABSTRACT

Developers can reuse not only class code but also wide range of knowledge on domain by reusing framework. Existing Object-Oriented Methodology and Catalysis Methodology were presented when redefining component in the course of redesigning framework. However, existing methodologies have weakness that entire process is waterfall model or design of interface lays too much stress on implementation stage

So, this thesis will present Component-Oriented Methodology for the reuse of framework, and construct the environment for framework and domain development. That is, domain is analyzed by input of domain knowledge on real world to create software based on component, and hotspot is identified through analyzed information, and refactoring by putting additional information on users and developers. After that, I will create domain framework and application framework depending on domain. In this Component-Oriented Methodology, information is searched, understood and extracted or composite through component library storage internally. Then this information is classified into the information on component, and used as additional information in redesigning. With this, developer can obtain reusability, easiness and portability by constructing infrastructure environment that allows to register, update and delete component through Component Management System(CMS) under the development environment which can be easily applied to his own application using framework component, in this thesis, CORBA(Common Object Request Broker Architecture) environment

<sup>†</sup> 동신회원 대구효성기불력대학교 컴퓨터공학과 교수  
<sup>††</sup> 준 회원 대구산업정보대학 진산정보계열 겸임교수  
논문집수 1999년 3월 30일, 심사완료 2000년 3월 6일

## 1. 서 론

기존의 라이브러리를 기반으로 한 재사용은 컴포넌트의 조합 과정이 쉬운 작업이 아니라는 점에서 또한, 코드 재사용의 한계를 갖는다는 점에서 재사용의 효과를 극대화하지는 못하였다. 따라서 한 영역에서의 재사용만을 지원하는 프레임워크 기반 애플리케이션의 연구에 초점이 모아졌다. 이는 한 도메인에 속한 다양한 애플리케이션 개발에 공통적으로 필요한 컴포넌트들과 컴포넌트들간의 상호 연결관계 및 상호 작용 방법까지도 포함하고 있기 때문에 한 영역에서의 기본 애플리케이션 행위를 정의하고 있다는 점이 라이브러리를 기반으로 한 재사용과는 다르다.

이러한 재사용성의 기여에도 불구하고 재사용성으로 인한 소프트웨어 생산성이 사용자들의 서비스에 대한 요구를 만족시키지 못했고 소프트웨어 품질이 향상되지 않았으며 유지보수에 문제점을 가지고 있다. 이러한 위기를 해결하기 위한 한가지 대응책으로 최근 컴포넌트웨어(componentware)가, 등장하였다. 이러한 컴포넌트기반의 소프트웨어는 객체 지향적인 소프트웨어 모듈이므로 재사용성을 높였으며 이로 인해 개발에 소요되는 비용을 상당히 줄일 수 있으며, 적은 개발비용 사용자의 요구를 충족시킬 수 있다는 장점을 지니고 있다. 또한, 컴포넌트기반의 소프트웨어 경우 모듈화 되어 있으므로 기존의 소프트웨어처럼 유지보수에 막대한 비용을 들일 필요 없이 분체가 되는 모듈 하나 하나를 유지보수에 나가면 되므로 유지보수 비용을 상당 부분 절감할 수 있다는 강력한 대응책을 가지고 있다.

따라서, 본 논문에서는 컴포넌트 기반의 소프트웨어 생성을 위해 실제계의 도메인 지식을 입력으로 도메인을 분석하고 분석된 정보를 통해 hotspot을 식별하여 사용자와 개발자의 추가정보를 첨가하는 재설계 과정을 거친다. 이후 도메인에 따라 도메인 프레임워크와 애플리케이션 프레임워크를 생성한다. 이러한 컴포넌트 지향 방법론은 내부적으로는 컴포넌트 라이브러리 저장소를 통해 정보를 검색, 이해하여 추출하거나 합성으로 얻어진 정보는 컴포넌트에 대한 정보로 분류되고 이것은 재설계시 추가 정보로 사용되어진다. 이로 인해 사용자는 프레임워크 컴포넌트를 이용해 자신의 애플리케이션에 쉽게 적용 가능한 개발 환경을 통해 컴포넌트를 추가, 갱신, 삭제하는 기반 환경을 구축한다.

## 2. 관련연구

### 2.1 프레임워크 재사용

컴포넌트의 재사용은 사용자에게 제공되는 인터페이스 환경의 일약함과 유지보수를 위한 높은 비용 때문에 실패를 거듭하고 있다. 그러므로 한 단계 높은 프레임워크 연구의 필요성이 반드시 요구되며, 재사용 가능한 부품을 분류하여 저장하거나 부품 검색, 이해, 그리고 수정단계에 대한 연구도 매우 필요하다.

프레임워크는 추상화된 클래스들의 집합 또는 특정 도메인에서 재사용 가능한 설계를 구성하는 상호 연관된 클래스들의 집합이다. 프레임워크를 기반으로 한 애플리케이션의 개발은 기존의 개발방법과는 차이점을 가진다. 즉, 기존 애플리케이션 개발 환경에서의 재사용이란 함수와 클래스들의 집합체인 라이브러리를 제공하고 개발자들은 라이브러리에서 필요한 컴포넌트들을 식별한 후 이들 컴포넌트들을 조합함으로써 새로운 애플리케이션을 개발한다.

라이브러리를 통한 재사용에서 가장 중요한 요소는 잘 정의된 컴포넌트를 식별하고 컴포넌트의 조합 방법을 정확하게 파악하는 것이므로 라이브러리의 분류 기법 및 검색 시스템의 정확성이 재사용의 성공을 결정하는 요인이다. 그러나 이러한 라이브러리를 기반으로 한 재사용은 컴포넌트의 조합 과정이 쉬운 작업이 아니라는 점에서 또한 코드 재사용의 한계로 인해 재사용의 효과를 극대화하지는 못하였다. 이를 극복할 수 있는 방법으로 프레임워크 기반 애플리케이션을 개발한다.

이는 라이브러리와 달리 한 영역에서의 재사용만을 지원한다. 한 도메인에 속한 다양한 애플리케이션 개발에 공통적으로 필요한 컴포넌트들과 컴포넌트들간의 상호 연결관계 및 상호 작용 방법까지도 포함하고 있기 때문에 한 영역에서의 기본 애플리케이션 행위를 정의하고 있다는 점이 라이브러리를 기반으로 한 재사용과는 다르다. 그러므로 프레임워크를 재사용하여 애플리케이션을 개발하는 것은 프레임워크가 이미 애플리케이션의 행위를 포함하고 있으므로 컴포넌트들의 조합을 정의하는 것이 아니라 애플리케이션의 기본 행위를 구성하게 될 컴포넌트를 재정의하는 것이다. 프레임워크 자체를 개발하는 것은 애플리케이션을 개발하는 것보다 더 많은 노력과 검증이 필요하다.

이러한 프레임워크는 애플리케이션을 개발할 수 있는

아키텍처를 제공하는 면에서 라이브러리와는 차이가 있다. 프레임워크 설계와 코드 부분을 재사용함으로써 애플리케이션 개발비용과 시간이 감소되며 사용이 용이하지만 프레임워크의 구현이 힘들다. 일반적으로 프레임워크 도메인은 다음과 같이 내부 구조에 의한 분류와 문제 영역에 의한 분류로 나눈다.

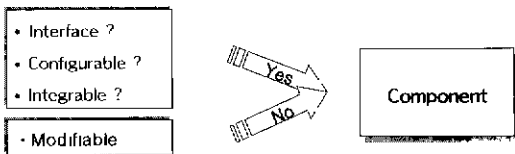
- 내부 구조에 의한 분류 - 블랙/화이트/그레이 박스
- 문제 영역에 의한 분류 - 응용/도메인/지원 프레임워크

2.2 CBSE(Component-Based Software Engineering)

컴포넌트는 넓게는 접근 가능한 인터페이스를 제공하는 모듈을 나타내고 좁게는 일반적으로 널리 알려진 CORBA, COM(Component Object Model), JavaBeans 등을 의미한다[1]. 컴포넌트라는 개념은 뚜렷이 명시되어있지는 않으며, 최근 VB(Visual Basic), Delphi 등이 나오면서 많이 거론된 용어로 다음과 같이 정리할 수 있다[2, 3, 4]

- Philippe Krutchen : 잘 정의된 아키텍처상에서 어떠한 기능을 수행하는 시스템의 독립적이면서 대체 가능한 부분이다. 또한, 인터페이스들의 집합에 대한 물리적인 구현을 제공한다.
- Kozaczynski : 자발적인 비즈니스 객체 또는 비즈니스 로직을 소프트웨어로 구현한 것이다

다음 (그림 1)은 컴포넌트의 정의를 도식화한 것이다



(그림 1) 컴포넌트의 정의

소프트웨어 공학이 대두된 1970년 이래, 대부분의 사람들이 컴포넌트와 컴포넌트기반 설계에 관한 것을 논의하면서 CBSE의 개념을 정의하였다. 일반적으로 이러한 컴포넌트와 연관된 전체적인 패턴의 범주는 단순히 컴포넌트를 개발하거나 이용하는 것뿐만이 아니라, 소프트웨어 집약적인 시스템의 생산과 개발로 조립식 컴포넌트를 유도하는 데 있다[5].

2.3 미들웨어(Middleware)

현재 대표적인 컴포넌트 모델은 크게 다음 세 가지로 나눌 수 있다. OMG(Object Management Group)에서 제안한 CORBA와 Microsoft사에서 제안한 DCOM(Distributed COM)/OLE(Object Linkage Embedded)/ActiveX 그리고 Sun Microsoft사에서 제안한 JavaBeans가 그 예이다. 본 논문에서는 CORBA[6, 7]에 대한 내용만을 다룬다.

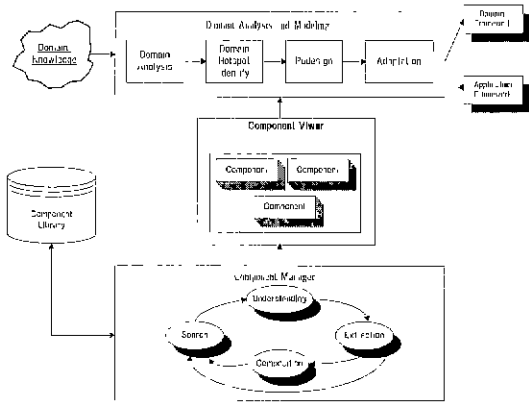
CORBA는 OMA(Object Management Architecture)의 일부로서 분산 환경 하에서 객체간의 통신 기능과 필수 기능을 담당하며, 서로 다른 프로그램 사이에서 통로 역할을 하는 모듈이다. OMA는 분산 환경 하에서 요구되는 객체 솔루션을 제공한다. 또한, ORB(Object Request Broker)는 CORBA의 핵심기술이다. ORB를 구성하는 객체 구현에서 구현 스펙레톤은 클라이언트의 서비스 요청에 해당하는 객체 구현을 직접 호출하는 역할을 하고, 객체 이펙터는 다른 구성 요소들과 상호작용을 통해서 클라이언트 요구에 대한 서비스를 제공한다. 또한, ORB 인터페이스는 클라이언트와 객체 구현에 ORB에 대한 통신 기능, 메소드 호출 등의 기본적인 기능 제공한다.

클라이언트에서 클라이언트 IDL 스템은 메소드 호출이 컴파일시 정해지지는 정적 호출을 하고 DII(Dynamic Invocation Interface)은 인터페이스 저장소에 해당 클래스의 인터페이스를 저장해 두고 동적호출시 이를 참조하여 해당 메소드를 수행한다. IDL(Interface Definition Language)은 특정 프로그래밍 언어와 컴파일러에 독립적이며, 다중 상속을 지원하는 등 대부분의 객체지향 개념을 지원하고 구현을 위한 언어가 아니며 동적 호출 메커니즘을 제공하는 특징을 가진다

3. 컴포넌트기반 방법론

컴포넌트 기반의 소프트웨어 생성을 위해 실제계의 도메인 지식을 입력으로 도메인을 분석하고 분석된 정보를 통해 hotspot을 식별하여, 사용자와 개발자의 추가정보를 첨가하는 재집계 과정을 거친다. 이후 도메인에 따라 도메인 프레임워크와 애플리케이션 프레임워크를 생성한다. 이는 내부적으로 컴포넌트 라이브러리 저장소를 통해 정보를 검색, 이해하여 추출하거나 합성으로 얻어진 정보는 컴포넌트에 대한 정보로 분류되고, 이것은 재설계시 추가정보로 사용되어진다. 또한, 컴포

먼트 관리 시스템을 통해 컴포넌트를 검색, 갱신, 삭제  
가 가능하다 다음(그림 2)은 전체 시스템 구성도를 나  
타낸다



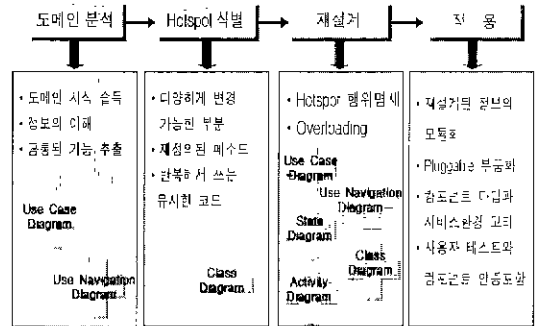
(그림 2) 시스템 구조도

### 3.1 도메인 분석과 모델링

일반적으로 프레임워크 개발을 시작하는 단계에서 우  
선적으로 해야할 일은 도메인을 정의하는 것이다. 단  
일 프레임워크로 일반적인 문제점을 모두 해결하기는  
불가능하므로 도메인을 정의하는 것은 중요하다. 프레  
임워크 개발에 있어서 결정된 도메인에 따라 요구사항  
이 그 도메인에 있는지 아닌지를 판단할 수 있다. 또  
한, 프레임워크가 필요한 애플리케이션에 대해 제사용  
을 할 수 있는지 없는지를 결정할 때도 도메인을 정하  
는 것이 필요하다.

도메인이 정해지면 이 도메인에 대한 분석이 이루어  
진다. 이때 여러 애플리케이션에 대해 일반적인 요구사  
항과 각 애플리케이션에 특정한 요구사항 중 어디에 초  
점을 맞추느냐는 무엇보다 중요하다. 요구사항을 얻는  
다는 것은 애플리케이션들 사이에 일반적인 모든 요구  
사항들을 고립시키는 것과 이러한 요구사항들을 프레  
임워크에 부과된 요구사항으로 만드는 것이다.

분석된 정보는 공통된 기능을 추출하는데 있어서 일  
반적으로 Use Case Diagram을 사용하게 되는데 이때  
객체와 그들간의 협동관계가 정의되고 클래스와 그들의  
메소드를 자세히 기술한다. 프레임워크를 설계하는 동  
안 여러 가지 유사한 애플리케이션을 적용하게 되는데  
이것은 구현에 대한 기초를 제공해 주기도 한다. 다음  
(그림 3)은 본 논문에서 제시한 방법론을 단계별로 나  
타낸다.



(그림 3) 컴포넌트지향 방법론

### 3.2 컴포넌트 뷰어

식별된 정보는 사용자가 요구하는 것에 대한 해결책  
을 찾기 위해 유사한 정보볼 보기(viewing) 원한다. 즉,  
컴포넌트 라이브러리에서 추출된 컴포넌트들은 각각 개  
발 프로세스에 조립되어서 실제계 도메인 지식과 함께  
세로운 정보를 구축할 수 있는 기반 지식이 된다 컴포  
넌트로부터 시스템을 구축하기 위하여 서로 다른 유형  
의 서비스들을 제공하는 광범위한 컴포넌트로부터 신  
택하는 것이 가능해야만 한다. 또한, 더 높은 추상화 수  
준에서 컴포넌트는 특별한 도메인에 특장적인 많은 기  
능을 제공할 수 있다

애플리케이션을 생성하기 위하여 컴포넌트를 조립하  
는 것은 컴포넌트에 의해서 제공되는 서비스에 대한 세  
부 사항이 잘 묘사되고 완벽할 것을 요구한다 대규모  
의 기업에서 애플리케이션을 구축하기 위해서는 컴포넌  
트 인터페이스를 기술하기 위한 풍부한 인이가 모든 컴  
포넌트기반 개발 접근 방법의 핵심적인 측면이다[8].

### 3.3 컴포넌트 관리

사용자는 원하는 컴포넌트의 검색(searching)을 위해  
서 제시되어진 페킷 항목을 선택함으로써 가능한데 페  
킷 검색은 시스템에서 제공되는 여러 가지 페킷들 중 찾  
으려 하는 컴포넌트에 알맞은 속성은 나타내는 항목들  
을 선택하여 조합함으로써 특정 컴포넌트를 검색한다.

원하는 컴포넌트를 검색한 후 자신의 프로그램에 통  
합이 가능한지를 결정하는데 도움을 주기 위해 사용자  
의 이해 정도에(understanding) 따라 컴포넌트 정보를  
즉각적이고 구체적인 시각적 형태의 정보를 제공하는데  
그 형태는 일반적으로 파일, 하이퍼링크, 텍스트 그리  
고 그래픽 등이 있다

제공된 정보를 통해 사용자가 원하는 컴포넌트를 추출(extraction)한다. 본 논문에서는 추출시 두 가지 패킷 항목인 운영체제(OS : Operating System)와 언어(Language) 모두를 만족하는 것을 추출한다. 이렇게 획득되어진 재사용 컴포넌트들은 사용자가 원하는 요구를 충분히 만족시킬 수 있도록 합성(composition)을 통해 얻을 수 있으며, 원시코드 수준의 클래스간의 합성이 아닌 실행 가능한 컴포넌트를 서로 다른 언어, 운영체제 등에 독립적으로 사용할 수 있다.

4. 사례연구

본 논문에서는 컴포넌트기반의 소프트웨어 생성을 위해 기존의 애플리케이션을 분석하고 설계한 후, 사용자의 반복되는 요구 사항을 hotspot으로 인식하여 기존의 분석 결과를 기반으로 추가할 분석 사항을 작성하는 재설계 과정을 거침으로 사용자와 개발자의 요구 사항을 적절히 수용하는 새로운 프레임워크 컴포넌트를 구성한다

4.1 도메인 분석 및 모델링

컴포넌트 프레임워크 구축을 위해 본 논문에서는 블랙 박스 형태의 프레임워크를 구성한다. 애플리케이션 개발자는 클래스를 인스턴스화하고 그들의 멤버 함수를 호출함으로써 프레임워크의 built-in 기능을 사용하고(black-box), 새로운 클래스를 파생시키고 멤버함수를 오버라이드(override)함으로써 기능을 확장 및 수정(white-box)하게 된다. 또한, 프레임워크를 사용하여 애플리케이션들이 개발되어짐에 따라, 선택할 수 있는 서브클래스들의 라이브러리가 만들어진다.

4.1.1 도메인 지식을 통한 도메인 분석

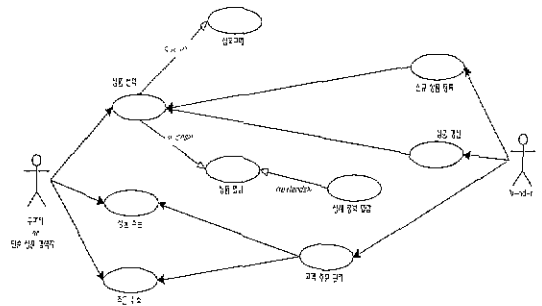
기존의 웹상에 부유상태의 광고 컬럼 애플리케이션 중 하나를 선택하여 분석하였다. 이것은 다음(그림 4)과 같은 문제 명세서를 기반으로 한다

광고 컬럼 애플리케이션은 사용자가 여러 종류의 상품중 원하는 상품을 선택하게 되면 해당상품을 서버에서 검색한 후 요약정보와 그 목록이 제시된다. 제시된 정보보다 좀더 상세한 내용을 원할 경우 링크된 부분을 클릭하면된다. 제시된 상세정보는 상품의 이미지의 상품의 계원에 대해서 나타나며 구매를 원할 경우 색상이나 개수, 지불방법등에 대한 구매서를 작성한 후 주문을 하게된다.

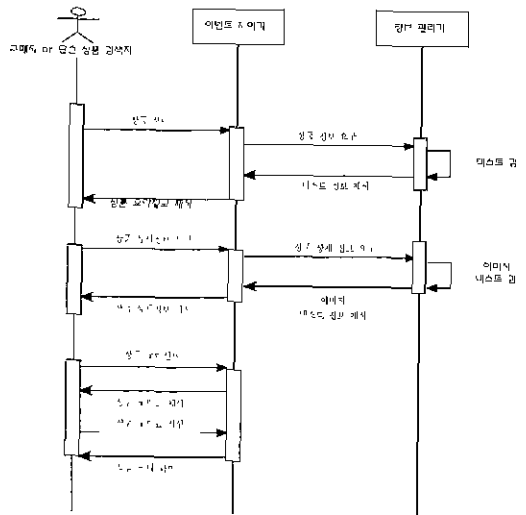
(그림 4) 문제 명세서

도메인 분석을 통해 Use Case Diagram을 다음(그림 5)과 같이 UML(Unified Modeling Language) 표기법을 사용하여 나타내었다. 또한, 시스템 내의 객체 상호작용을 모형화하기 위해 다음(그림 6)과 같이 수직선상의 여러 개의 객체와 순서나 함수로 시간이 지나가는 것에 따라 객체를 사이에 메시지 교환을 나타내는 Sequence Diagram으로 표현하였다.

그리고 추출된 공통적인 클래스와 객체들은 다음(그림 7)과 같이 시스템에서 다루어지는 클래스들의 정적인 구조를 연관, 종속, 특수화된 또는 패키지화된 여러 가지 방법으로 다른 것들과 각각 관계를 가지는 것을 표현한 Class Diagram으로 나타내었다.

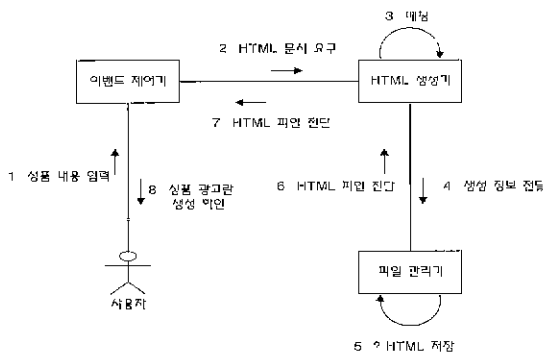


(그림 5) 사용사례도

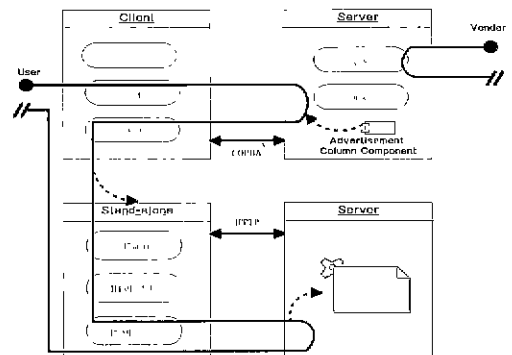


(그림 6) Sequence Diagram

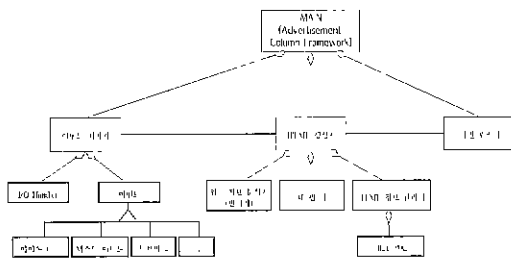




(그림 13) Collaboration Diagram



(그림 15) Use Navigation Diagram



(그림 14) Class Diagram

추가적으로 UML지식 기반으로 동적이고 다중적인 경로를 통해 네비게이션 되는 정보와 구성요소에 초점을 두고, 웹 애플리케이션에 대한 분석과 설계를 위한 Navigation Diagram을 표현한다. 우선 Simple Use Navigation Diagram은 인터넷/인트라넷뿐만 아니라 클라이언트/서버 환경에서 네비게이션 되는 정보의 흐름과 연결에 관련된 전반적인 사항을 나타내기 위한 것이다. 전송되는 자료는 시스템 이벤트와 사용자 선택에 의해 정해지며 이때 메시지는 웹 구성요소사이에서 네비게이션 된다.

또한, Use Navigation Diagram은 전체 시스템에서 사용되는 오퍼레이션 구성요소의 사용을 나타내며 네비게이션 되는 동안 전달되는 자료의 목적지를 명시하는 것을 원칙으로한다. 이는 사용자에게 시스템의 전반적인 구조와 실 매개체를 직관적으로 파악하고 이해할 수 있는 장점을 가진다

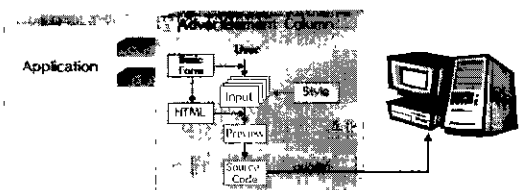
다음 (그림 15)는 제시된 표기법을 이용해 본 논문에서 적용된 Navigation Diagram이다[9].

#### 4.1.4 프레임워크 구축

본 논문에서는 분석과 설계를 통해 정의된 소프트웨어를 하나의 모듈화로 정의하기 위해 앞 절에서 정의된 분석과 설계 정보를 이용해 직접 컴포넌트 프레임워크를 구성한다

구성할 광고 컬럼 애플리케이션 프레임워크는 도메인 프레임워크(domain framework)이고, 이러한 프레임워크는 각 애플리케이션 영역마다 다양하게 개발 가능한 도메인이지만 텍스트 항목, 이미지, 오디오, 동영상 등을 지원하는 환경은 모두 유사하기 때문에 공통적인 아키텍처를 정의할 수 있다. 또한, 분야마다의 특징에 따라 사용자가 임의로 변경할 수 있는 hotspot 역시 미리 추출이 가능하고 이를 캡슐화하거나 정련화된 알갱이의 객체를 생성하기 때문에 블랙 박스 프레임워크의 재사용 부품인 pluggable한 객체 개발이 가능하다.

다음 (그림 16)은 광고 컬럼 애플리케이션에 대한 전체 구성도이며 stand-alone환경에서 사용자의 입력에 대한 디자인 폼을 제시하며, html문서가 자동 생성되는 것을 볼 수 있다. 또한, 사용자가 쉽게 웹상에 프레임워크 컴포넌트를 올려서 사용하는 것을 보여준다. 본 논문에서는 VB 5.0에서 ActiveX 컴포넌트를 사용하여 생성하였다



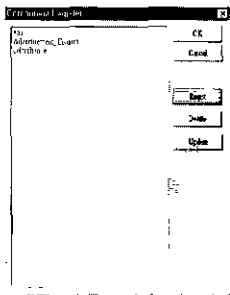
(그림 16) 광고 컬럼 애플리케이션 구성도

4.4 컴포넌트 관리

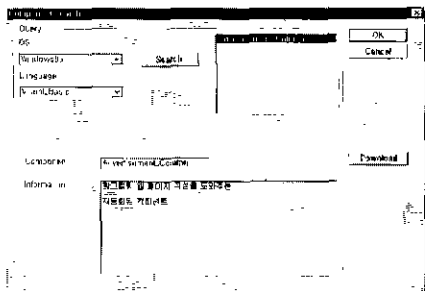
사용자의 요구에 맞는 컴포넌트를 검색, 추출하여 plug-in함으로써 사용자는 쉽게 그 컴포넌트를 사용할 수 있다. 사용자는 컴포넌트 관리 시스템을 통해 컴포넌트를 관리하며, 서버에 있는 컴포넌트를 선택하여 사용자 환경에서 stand-alone으로 그 컴포넌트를 쉽게 사용할 수도 있고, 웹상에 올려서 사용할 수도 있다. 본 논문에서는 앞 절에서 구성한 광고 컬럼 애플리케이션 프레임워크 컴포넌트를 선택하여 웹상에 올려지는 것을 제시한다.

본 논문에서 클라이언트 측의 운영체제와 구현언어는 Win 95와 VC++ 5.0[10]을 선택했으며, CORBA상에서 운용되도록 컴포넌트 관리 시스템을 개발하였다. 다음 (그림 17, 18)은 컴포넌트를 검색하거나 사용자가 프레임워크를 통해 생성한 컴포넌트를 등록할 수도 있는 것을 나타낸다. 컴포넌트 관리 시스템의 서버측 구현 또한, Win 95와 CORBA상에서 운영된다.

다음 (그림 18)에서 컴포넌트 분류는 일반적으로 컴포넌트들의 공통된 특성을 수집하여 페킷을 구성한 후 컴포넌트에 알맞은 속성을 페킷 항목으로 선택하고 조합하는 과정을 사용한다



(그림 17) 컴포넌트 관리 시스템 - 컴포넌트 등록



(그림 18) 컴포넌트 관리 시스템 - 컴포넌트 검색

각 페킷은 컴포넌트들의 대표 특성들을 나타내는 항목들로 이루어져 있으므로 새로운 컴포넌트 추가에 융통성을 제공하여 항목이 지속적으로 증가할 때 신속성 있게 대처할 수 있다. 또한, 컴포넌트에 대한 효과적인 표현이 가능해 정확한 컴포넌트의 검색이 용이하다. 컴포넌트의 공통적인 속성을 모아 하나의 페킷으로 모으고, 여러 페킷들 중 찾으려하는 컴포넌트에 알맞은 속성을 나타내는 항목들을 선택하여 조합함으로써 특정 컴포넌트를 검색할 수 있다.

본 논문에서 구성한 CORBA[11] 응용 프로그램 개발 과정은 다음과 같다.

① IDL 작성

인터페이스 부분을 정의하는 인이로 CompMgr.idl을 작성한다.

```

ifndef _CompMgr_IDL
#define _CompMgr_IDL
#include "CommonDataType.idl"
interface CompMgr{
    oneway void start();
    void GetComponentInfoList(out ComponentInfoList clst);
    void searchComponentList(in Facet facet, out ComponentInfoList clst);
    void registComponent(in ComponentInfo mfo, in File file);
    void removeComponent(in ComponentInfo mfo);
    void updateComponent(in ComponentInfo mfo, in File file);
    void downloadComponent(in ComponentInfo mfo, out File file);
};
#endif
    
```

② IDL 인터페이스 컴파일

작성된 인터페이스 CompMgr.idl파일은 컴파일 과정을 거치면서 C++ 매핑 규칙에 따라 C++ 파일로 변환되어진다. 이때 네트워드의 서로 다른 곳이나 서로 다른 메모리 공간 내에 위치된 클라이언트와 서버간의 통신 기능 묘사를 위해 CompMgr.hh, CompMgrS.cpp, CompMgrC.cpp, CompMgr.ih, CompMgr.ic파일이 자동으로 생성되며 CompMgr.hh를 CompMgr\_i.hh로 CompMgr.ic는 CompMgr\_i.cpp로 바꾼 후 서버를 구현한다. 분산 객체에서 클라이언트 부분은 서버에 정의된 클래스의 메소드를 원격지에서 호출하고 이 결과를 전달받는 과정을 통해 작업한다. 그러므로 서버에서는 요구에 대한 메소드를 구현해야 한다. 이는 일반적인 C++ 프로그램 작성에서와 같이 인터페이스 설명을 위한 헤드 부분과 그 메소드의 실제 구현하는 두 개의 부분으로 이루어진다.



③ 서버와 클라이언트 메인 작성

클라이언트 부분과 서버 부분에 필요한 모든 클래스 직성을 완료한 후 각 부분에 필요한 2개의 메인 프로그램을 작성한다. 클라이언트의 메인 프로그램은 해당 클래스를 선언하고 서버의 구현 부분과 해당 클래스를 바인딩 한 후 필요한 클래스를 호출하는 것이며, 서버의 메인 프로그램은 클라이언트가 요구 즉시 사용할 수 있도록 환경을 설정하고 메소드의 구현 부분을 완성하는 부분이다

검색된 결과를 컴포넌트 관리 시스템의 인터페이스를 반환하게 되는데 이때는 작성된 인터페이스에 맞는 드라이버를 구동한다. 또한, 컴포넌트 관리 시스템을 실행시키며 컴포넌트를 추가, 삭제, 등록할 수 있다

5. 평가

본 절에서는 기존의 개발 방법론들과 구축된 시스템 자체의 특성을 개발자가 제시한 기준으로 유사한 시스템과의 상대적인 비교를 통하여 구축 시스템이 가진 독창적인 성격을 중심으로 평가한다.

우선, 개발 방법론 측면에서 살펴보면, 기존의 객체지향 방법론을 기본으로 하는 Rose의 경우는 그 자체로 모든 작업을 수행한다. 즉, 컴포넌트 인터페이스 설계와 컴포넌트 구현 설계가 모두 객체지향 모델링 도구인 Rose 안에서 수행되므로 작업을 일관되게 반복적으로 수행할 수 있지만 인터페이스 설계가 구현 설계에 의하여 지나치게 변경될 여지가 많다 또한, Sterling의 Cool Family의 경우는 작업이 단계별로 목표를 갖고 분할되어 수행할 수 있지만 전체과정이 폭포수형으로 수행될 수 있다는 단점이 있다.

본 논문에서 제시한 방법론은 도메인 분석과 모델링 과정에서 실세계의 도메인 지식을 입력으로 도메인을 분석하는데 이때, UML을 기반으로 하나 인터넷/인트라넷에서의 네비게이션 되는 정보의 흐름 순서는 시스템 이벤트에 의해 정해진다. 이 경우의 네비게이션 되는 메시지가 장치사이의 흐름을 명시하기 위한 것으로 Navigation Diagram을 제시하였다

분석된 자료를 통해 hotspot을 식별하여 애플리케이션마다 있을 수 있는 다양한 변화의 부분을 정의할 수 있다. 이것은 재 정의된 메소드에 대응되며 융통성과 일반성을 지니며 잘 설계된 프레임워크 개발에 도움을 준다.

두 번째로 재사용 시스템 측면에서의 평가는 본 논문에서 구축한 CMS를 다른 재사용 시스템과 비교해 보았다. 비교한 내용은 다음<표 1>에 제시한다.

<표 1> 다른 시스템과의 비교

1	KAIST	충일대	진남대	CMS
검색 모델	질의+브라우저	질의	질의+브라우저	질의+브라우저
분류 모델	열거+패킷	패킷	패킷	패킷
컴포넌트 이해 정보	○	×	×	○
패턴 이해 정보	×	×	×	○
사용자 인터페이스	풍부한 GUI	GUI	×	GUI

6. 결론

특정 목적을 위해 재사용이 일반적으로 수행되어 오 는 동안, 더 체계적인 접근 방법을 통해 생산성과 품질 의 상당한 향상이 이루어졌다. 최근 이러한 접근 방법 들의 초점은 재사용 가능한 컴포넌트였고 이것은 CBD 를 등장하게 하였다. 따라서, 컴포넌트 기반 개발 방법 들은 재사용 가능한 컴포넌트를 구축하고 애플리케이션 을 조립하기 위하여 이들 컴포넌트를 사용하는 것에 초 점을 두고 있다.

따라서, 컴포넌트 기반의 소프트웨어 생성을 위해 실 세계의 도메인 지식을 입력으로 도메인을 분석하고, 분석된 정보를 통해 hotspot을 식별하여 사용자와 개발자 의 추가정보를 첨가하는 재설계 과정을 거친다. 이후 적용범위에 따라 도메인과 애플리케이션 프레임워크를 생성한다. 컴포넌트 지향 방법론은 내부적으로는 컴포넌트 라이브러리 저장소를 통해 정보를 검색, 이해하여 추출하거나 조합으로 얻어진 정보는 컴포넌트에 대한 정보로 분류되고 이것은 재설계시 추가정보로 사용되 어진다. 이로 인해 사용자는 프레임워크 컴포넌트를 이 용해 자신의 애플리케이션에 쉽게 적용 가능한 개발 환경 즉, 본 논문에서는 CORBA환경 하에 컴포넌트 관리 시스템을 통해 컴포넌트를 추가, 갱신, 삭제하는 기반 환경을 구축할 수 있다.

향후 연구 방향은 개발환경을 컴포넌트에 국한해서 작업하였던 것을 패턴 관리로의 전환이 요구되며, 이를 통합하는 IDE(Integrated Development Environment) 가 요구된다.

참 고 문 헌

[1] Component, Component Development Strategies, Vol.8, No 7, July, 1998.

[2] Philippe Kruchten, "Modeling Component Systems with the Unified Modeling Language," Available Web server from <http://www.sei.cmu.edu/cbs/icse98/papers/pl.html>.

[3] Dsouza D. F., Wills A. C., Objects, Components. and Components with UML, Addison-Wesley, 1998.

[4] Kozaczynski Wojtek, Booch G, "Component-Based Software Engineering." IEEE Software, Sept/Oct., pp.34-36, 1998.

[5] Sterling, "Component Based Software Development," Available web server from <http://www.sterling.com>

[6] IONA, Programming Guide : Orbix 2.3, IONA Technologies Ltd, 1998.

[7] IONA, The Common Object Request Broker : Architecture and Specification, IONA Technologies Ltd, 1998.

[8] Alan W, Brown and Keith Jaeger, "The Future of Enterprise Application Development with Components and Patterns," STERLING SOFTWARE, Aug., 1998.

[9] 김행곤외 2인, "확장된 UML기반의 E-Mailing System 개발 프로세스", 정보처리학회 논문지, 제2권 5호, pp.461-464, 1998.

[10] Microsoft Press. Microsoft Visual C++ 6.0 Manual, 1998.

[11] Orfali Harkey, JAVA and CORBA, WILEY, 1998.



김 행 곤

e-mail : hangkon@cuth.catacgu.ac.kr  
 1985년 중앙대학교 전자계산학과 졸업(공학사)  
 1987년 중앙대학교 대학원 전자계산학과 졸업(공학석사)  
 1991년 중앙대학교 대학원 전자계산학과 졸업(공학박사)

1978년~1979년 미 항공우주국 객원 연구원  
 1987년~1989년 AT&T 객원 연구원  
 1990년~현재 대구효성가톨릭대학교 컴퓨터공학과 부교수  
 관심분야 : 객체지향 시스템 설계, 사용자 인터페이스, 소프트웨어 제공학, 유지보수 자동화 툴, CASE



한 은 주

e-mail : ejhan@cuth.catacgu.ac.kr  
 1994년 경일대학교 전자계산학과 (공학사)  
 1996년 대구효성가톨릭대학교 대학원 전산통계학과 졸업 (이학석사)

1999년 대구효성가톨릭대학교 대학원 전산통계학과 졸업 (이학박사)  
 1999년~현재 대구산업정보대학 전산정보계열 겸임교수  
 관심분야 : CBSE, 컴포넌트 합성, 요구공학, 비즈니스 컴포넌트