

# CBD 기반의 웹 시스템 개발 적용을 위한 컴포넌트 기술 비교 평가

Comparison and an evaluation of a component technology for a Web system development and application based on CBD

나윤지

충북대학교 컴퓨터공학과 박사과정

Yun-Ji Na

Ph. D. Course, ChungBuk National University, Dept. of Computer Engineering

중심어 : Component, CBD, Web Base System, Application Software

## 요 약

정보화 사회가 고도화될수록 소프트웨어의 개발은 복잡해지고 대규모화되고 있다. 또한 소프트웨어의 라이프사이클 단축으로 개발기간의 단축이 절실하게 요구되고 있는 실정이며 개발된 소프트웨어의 생명주기도 짧아지고 있어 개발 및 유지보수가 더욱 어려워지고 있다. 이에 따라 소프트웨어의 개발에서 미리 구현된 컴포넌트를 사용하여 개발 비용과 시간을 단축하려는 노력이 증가하고 있다. 컴포넌트 기술은 현재 매우 빠른 속도로 발전하고 있으며, 기존의 소프트웨어 개발 방법 뿐 아니라 웹 기반 시스템의 개발에도 큰 변화를 가져오고 있다.

본 논문에서는 CBD 기반의 웹 시스템의 개발을 위한 관련 기술을 살펴보고, 특히 중소기업의 웹 시스템 개발에 대한 적용을 위해 기존 컴포넌트 기술들을 비교 분석하여, 문제점을 도출하고 이를 바탕으로 컴포넌트를 기반으로 한 웹 시스템의 개발·활용의 효율성을 높이기 위한 방안을 제안하였다. 본 연구결과의 활용을 통해 CBD 기반의 중소기업 웹 시스템 구축에서 개발비용을 감소시키고 유지 관리의 부담을 경감시킬 수 있을 것으로 기대된다.

## Abstract

As for the software development, it is complicated, and size of software is growing larger. And contraction of a software development period is required acutely because it is short a period of software life cycle. Also, it becomes harder development and a maintenance because it is short a life cycle of developed software. According to this an effort is increased in order to reduce a development cost and time as using a component to have been implemented previously in development of software. A current component technology is developing into a very fast speed and is bringing a large change into development of a Web base system in addition to existing software development method. In this paper, we were handling a related technology for Web system development of a CBD base systematically and derived an issue to apply it to development of a Web system and presented an efficient practical use method of a component technology. First of all we looked into a component technology and a CBD methodology systematically and described a component technology in Web. Also, we proposed a method to increase development and efficiency of practical use of a Web system based on a component. The this study result can decrease a development cost of a Web system and a load of maintenance management through the re-use of a code.

## I. 서론

정보화 사회가 고도화될수록 소프트웨어의 개발 과정

은 점차 복잡해지고 있고 개발되는 소프트웨어의 크기는 더욱 커지고 있다. 또한 소프트웨어 개발에서 라이프사이클의 단축으로 개발기간의 단축이 절실하게 요구되고 있는 실정이다. 더욱이 개발된 소프트웨어의 생명주기도 짧아지고 있어 개발 및 유지보수가 더욱 어려워지고 있으며 이로 인해 소프트웨어 산업의 위기로 이어지는 결과를 낳게 되었다. 이러한 위기의 수습을 위해 객체지향개발 방법론이 나타났고 현재는 대부분의 경우 객체지향 프로그래밍(OOP) 언어를 기반으로 개발이 이루어지고 있다. 그러나 OOP 역시 재사용성의 문제가 발생하였고, 이것을 해결하는 방법으로 어플리케이션을 여러 개의 컴포넌트로 분할하여 각 컴포넌트에 고유 기능을 부여하고, 이들을 하나의 어플리케이션에 통합함으로써 재사용성과 유지보수의 용이성을 확보할 수 있는 방법인 컴포넌트 기반의 개발 방법론이 나타나게 되었다. 개발된 각각의 컴포넌트는 컴포넌트가 제공하는 기능이 필요한 어플리케이션에 재사용 되고 있기 때문에 전체 개발의 비용과 노력을 줄일 수 있다.

컴포넌트 소프트웨어는 컴포넌트 소프트웨어의 사용자나 다른 컴포넌트 소프트웨어에 대해 구체적인 구현정보는 숨기고 정의된 인터페이스를 통하여 해당 기능을 제공하는 단위 소프트웨어이다[1]. 컴포넌트 소프트웨어는 기본적으로는 객체지향 특성을 갖고 있다. 또한, 인터넷의 급속한 보급으로 다양한 플랫폼 환경 아래서 다양한 소프트웨어의 획득이 가능해져 이들 이질적인 소프트웨어 컴포넌트들을 효과적으로 분류, 유통 및 조립하는 기술이 필요하게 되었다. CBD(Component-Based Development)는 어플리케이션 개발의 신속성, 용이성, 플랫폼 독립성 등에 의해 개발비용의 절감, 제품의 시장진입의 신속성, 생산성 향상 및 품질 향상을 기할 수 있는 방법이다[4]. 결국 컴포넌트 기반 소프트웨어 개발이란 독립적인 기능을 담당하는 다양한 컴포넌트 소프트웨어의 집합으로부터, 해당 업무의 수행에 필요한 기능을 담당하는 하나 이상의 컴포넌트를 결합하여 해당 업무를 위한 소프트웨어를 개발하는 기술을 말한다[4],[5]. 컴포넌트 소프트웨어의 특징은 다음과 같다[1],[2].

- 컴포넌트를 구별하는 식별자를 가진다.
- 컴포넌트의 사용 상황을 알 수 있어야 한다.
- 동일한 서비스를 제공하는 새로운 버전으로의 변경이 해당 컴포넌트를 사용하는 응용 소프트웨어나 다른 컴포넌

트에 영향을 주지 않아야 한다.

- 서비스는 인터페이스를 통해서 제공하고 컴포넌트의 인터페이스 변경이 없어야 한다.
- 자신의 인터페이스에 대한 명확한 설명을 함께 제공하여야 한다.
- 개발 도구나 언어, 그리고 플랫폼에 독립적인 사용이 가능하여야 한다.
- 실행시점에 동적 재사용이 가능하여야 한다.

본 논문에서는 CBD 기반의 웹 시스템의 개발을 위한 관련 기술을 체계적으로 다루고 있으며, 중소규모의 웹 시스템 개발에 대한 적용을 위해 기존 컴포넌트 기술들을 비교 분석하여, 문제점을 도출하고 효율적인 응용 방안을 제시하였다. 본 연구 결과의 활용을 통해 컴포넌트를 기반으로 한 중소규모의 웹 시스템에서 개발 및 관리의 효율성을 높일 수 있을 것으로 기대된다.

## II. 컴포넌트 기술

### 1. 컴포넌트

컴포넌트 소프트웨어는 모듈화 설계의 개념에서부터 시작되어 객체지향 설계 기법의 등장과 마이크로소프트의 복합문서(compound document) 기술의 성공으로 활용이 촉진되었다. 복합문서 기술은 OLE/COM(Object Linking and Embedding/Component Object Model)이라 불리고 있다. 또한 복합문서 기술은 MS 워드 작업에서 엑셀 등의 다른 소프트웨어에서 작성된 문서를 워드 문서 내에 삽입하는 경우, 기능과 삽입된 문서에 대한 작업은 워드 상에서 해당 문서를 작성할 때 사용한 소프트웨어의 인터페이스를 그대로 사용할 수 있게 하는 기술이다.

현재 컴포넌트 소프트웨어 시장에서 가장 큰 위치를 점하는 모델 및 운영 구조는 MS의 COM[9]과 Sun의 JavaBeans 이다. 그리고 IBM이 지원하고 있는 OMG의 CORBA(Common Object Request Broker Architecture)[10], Oracle의 NCA(Network Computing Architecture) Cartridge 등이 있다. 이들 모델을 살펴보자.

- COM/DCOM/COM+는 MS사의 운영체제인 윈도우 NT, 윈도우 2000과 같은 윈도우 계열을 플랫폼으로 하는 환경에 제한적인 기술이다. 그러나 윈도우

운영체제가 차지하는 시장 규모가 매우 커서 현재 가장 많이 이용되는 컴포넌트 기술이다.

- **JavaBeans**은 클라이언트 측의 Java 기반 GUI의 구현을 위한 기술로 컴포넌트 작성 언어가 Java로 제한되어 다른 컴포넌트 기술에 비하여 언어 독립성이 떨어진다.
- **CORBA**는 원래 분산 객체 미들웨어 명세로, COM에 비하여 플랫폼에 대한 독립성이 높으며 JavaBeans에 비하여 언어 독립성이 높다.
- **NCA**는 DBMS 개발 업체로는 유일하게 독립적인 컴포넌트 명세를 보유하려는 Oracle사의 전략에 따라 제안된 것이지만 CORBA 규격을 따르는 분산 시스템 위에 Oracle 고유의 컴포넌트 양식을 구현한 것이다.

## 2. CBD 방법론

컴포넌트 기반 개발(CBD: Component Based Development)은 컴포넌트 생산, 선택, 평가 및 통합으로 구성되는 소프트웨어 개발 방법의 가장 새로운 패러다임이다 [11],[13],[15]. OOP를 통해 생성된 컴포넌트가 재사용성이 있어도 소스 형태의 클래스나 객체는 사용하기 쉬운 컴포넌트라고 볼 수가 없다. OOP는 바이너리(Binary)나 실행시 간단계(Run-time level)에서 컴포넌트의 상호운영성을 제공하지 않고 소스를 컴파일 해야 하기 때문에 사용하기가 불편할 뿐만 아니라 특히 클래스의 캡슐화가 완전하지 않다. 이에 반해, 컴포넌트는 복잡한 내부와 인터페이스가 분리되어 컴포넌트의 사용자는 단순히 컴포넌트의 인터페이스를 정의 및 수정함으로써 쉽게 프로그래밍을 할 수가 있다. 그러나 한편으로 컴포넌트는 OOP로 생성된 여러 객체로 구성이 가능하다. 따라서 CBD는 완전히 독립적인 방법론으로서 발전해 온 것이 아니라 기존의 OOP에 내재하는 재사용의 한계를 극복하고자 OOP와 소프트웨어 공학 관련 제 기술의 통합을 통한, 보다 진보된 통합 방법론이라 할 수 있다. CBD 방법론을 정리하면 다음과 같다[3],[5].

- **Catalysis** 방법은 기본적으로 UML 표기법을 사용하여 ActiveX를 기반으로 한 컴포넌트 응용 어플리케이션 제작을 위한 표준공정을 제공함으로써 시스템 설계자나 제작자가 쉽고 빠르게 소프트웨어에 대한 설계를 진행할 수 있도록 돕는다. 객체 모델링, 명세 활동, 구성에 의한 설계 모델을 지원하기 위해서 확립되었다. Catalysis는 컴포넌트와 프레임워크 기반의 소프트웨어 개발을 위해서 UML 및 OMG CORBA를 준수하는 방법이다.

**Catalysis** 방법은 사용자에게 비즈니스 모델, 많은 공통적인 설계 패턴, 매우 근본적인 정의 등의 다양한 영역에 적용될 수 있는 프레임워크를 제공한다.

- 앤더슨컨설팅은 아키텍처 중심적이고 컴포넌트의 플러그 앤 플레이(plug & play)스타일을 사용해서 소프트웨어 개발을 가능하게 하는 컴포넌트 기반의 개발도구를 구축하기 위한 연구 프로젝트를 수행하고 있다. 앤더슨컨설팅의 CBD 도구는 소프트웨어 컴포넌트와 아키텍처 분야에 관한 학계와 산업계의 많은 진보된 연구 아이디어와 결과를 포함하고 있다.
- **Sterling Software**사의 CBD96[12] Standard는 Sterling Software의 프로젝트 COOL: Gen을 사용해서 소프트웨어 컴포넌트를 명세화하고 납품하기 위한 표준이다. 이것은 표준컴포넌트를 위한 납품 요구조건, 고객에게 제공해야 되는 컴포넌트 명세 모델, 컴포넌트 인터페이스를 위한 표준, 컴포넌트 구현 모델의 표준, 컴포넌트 실행 모듈의 표준, 컴포넌트 문서, 컴포넌트 업그레이드의 관리 방법, 명명법 표준(Naming Convention Standards) 등을 포함하고 있다.

## 3. 컴포넌트 기반 소프트웨어 개발 과정

웹서버의 개발에 있어 컴포넌트 기술은 웹이 분산 하이퍼미디어 시스템의 단계를 넘어서 기존의 전통적 응용 서비스 통합 서비스가 가능한 웹 서버의 개발을 가능하게 하며 자바애플릿(Java applet), 액티브 X(Active X) 등을 통한 활용과 개발이 보편화되고 있다. 현재까지의 컴포넌트 기술은 대부분 분산 응용에서 클라이언트 측의 기능 확장과 개발에 중점을 두고 있다. 하지만 점차 COM+, Enterprise JavaBeans, CORBA 등과 같은 서버 측의 컴포넌트 기술에 많은 개발이 이루어지고 있다. 컴포넌트 기술은 MS 계열의 기술과 IBM이 지원하고 있는 JavaBeans 및 Sun이 지원하고 있는 CORBA가 상호 호환성을 보장하기 위해 노력하고 있다.

그림 1은 컴포넌트 기반 응용 소프트웨어의 개발 과정[1]을 보이고 있다. 먼저 업무 분석 및 컴포넌트 기반 설계 단계에서는 요구사항을 분석하고 이를 바탕으로 컴포넌트 모델 기반 설계를 수행하여, 해당 응용을 구현하는데 필요한 컴포넌트의 명세를 인터페이스와 그 서비스 기능을 추출한다. 소용 컴포넌트의 획득 단계에서는 응용 구현을 위해 필요한 컴포넌트를 표 1과 같은 방법을 통하여 확보할 수 있다.

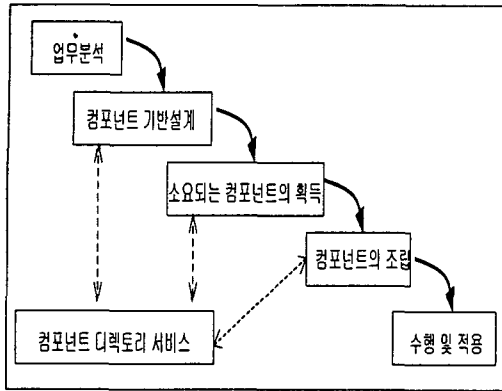


그림 1. 컴포넌트 기반 소프트웨어의 개발 과정

표 1. 컴포넌트의 확보

구분	확보 방법
구입	필요한 컴포넌트를 구입하여 확보
사용신청	필요한 컴포넌트를 제공하는 사업자로부터 사용 횟수나 기간에 따라 비용을 지불할 것을 신청하여 확보
변경	기 확보한 컴포넌트의 기능 확장을 통한 확보
랩핑 (Wrapping)	이미 확보하고 있는 일반 응용 소프트웨어를 컴포넌트화하여 확보
생성	기 확보하고 있는 전통적(비 컴포넌트) 응용 소프트웨어를 컴포넌트화하여 확보

컴포넌트의 조립단계에서는 확보한 컴포넌트를 조립하여 업무 수행을 위한 응용 소프트웨어 구축하며 수행 및 적용 단계에서는 컴포넌트 기반 응용 소프트웨어의 수행과 업무에 적용한다.

### III. 웹과 컴포넌트

#### 1. 컴포넌트와 분산 응용 소프트웨어

웹이 등장하기 이전에는 응용 소프트웨어 환경은 각 응용소프트웨어 로직의 수행이 하나의 범용 컴퓨터에 집중되

는 중앙 집중 구조와 클라이언트/서버 구조가 사용되었다. 클라이언트/서버 구조는 통신 기술의 발전과 개인용 컴퓨터의 확산에 따라 응용 소프트웨어가 서버의 로직과 클라이언트의 로직으로 분리되어 수행되는 2계층(two tier) 구조이다. 서버의 로직은 데이터 베이스 서비스, 트랜잭션 처리 서비스 등이며, 클라이언트 로직은 응용에 고유한 자료 처리와 보다 발전한 사용자 인터페이스 기능이었다. 클라이언트/서버 형태의 2계층 구조는 객체 지향 기술의 발전과 분산 처리 기술의 발전 및 인터넷의 보급으로, 이기종 분산 서버와 이기종 분산 클라이언트 플랫폼을 상호 연동하는 미들웨어를 분산 서버 로직과 분산 클라이언트 로직 사이에 두는 3계층(three tier) 구조로 발전하였다.

웹은 개방된 표준인 HTML을 이기종 서버와 클라이언트 사이의 통신 매체로, HTTP를 통신 프로토콜로 사용하여 등장하였으며, 초기에는 문자 기반의 문서 정보 서비스 시스템이었다. 멀티미디어의 발전으로 HTML에 다양한 멀티미디어 정보가 포함되기 시작하였으며, 이에 따라 주로 클라이언트의 멀티미디어 기능확장을 위하여 외부 뷰어(external viewer)의 플러그 앤 플레이 방식이 도입되었다. 이것이 웹에서 컴포넌트 사용의 초기 단계이며, 이후 자바애플릿과 자바 가상머신(JVM: Java Virtual Machine), 또는 COM 기반의 ActiveX를 채용하여 보다 발전된 컴포넌트의 사용으로 진행되어 현재에 이르고 있다.

현재 웹은 XML(eXtended Markup Language)을 통해 새로운 전기를 맞고 있다. 객체의 정적 특성과 동적 특성(DOM: Document Object Model)을 모두 지원할 수 있는 XML은 웹 기반의 분산 응용 소프트웨어의 개발을 가능하게 한다. 특히, 지난해부터 시작된 분산 응용 소프트웨어의 서버 측 로직의 웹 기반 개발이 올해 더욱 본격화할 것으로 예상된다. 이러한 웹 응용 서버(Web application server)는 DOM을 COM 또는 CORBA 기반으로 구현하고 이를 컴포넌트로 활용하여 응용의 서버로직을 구현한 것이다. 차세대 웹에서의 컴포넌트 활용은 크게 컴포넌트 유형 기반 구조로서의 웹과 컴포넌트 기반 웹 응용 개발의 두 가지 진행될 것이며 그림 2는 응용 소프트웨어의 개발 및 운용환경, 웹, 그리고 컴포넌트 기술의 변화와 이들 사이의 관계를 나타낸 것이다[1].

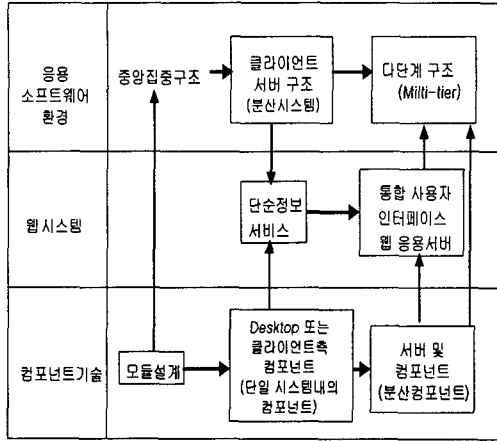


그림 2. 응용소프트웨어, 웹, 컴포넌트 기술

2. 컴포넌트 기반 웹 개발

재사용성 및 확장성을 지원하는 컴포넌트 기반 소프트웨어 개발 방법을 적용한 웹 시스템 개발은 크게 세 가지 형태로 나타나고 있다[1]. 첫 번째는 웹 클라이언트의 개발에 대한 적용이다. 자바애플릿(Java applet)와 ActiveX를 기반으로 웹 클라이언트인 브라우저의 개발은 이미 보편화되고 있다. 두 번째는 웹 기반 응용 서버 개발에 대한 적용이다. 랩핑을 통한 전통적인 상업적 응용 소프트웨어의 컴포넌트화가 가능해지고 XML의 개발과 분산 처리 환경에서의 데이터교환 양식으로의 채용으로 웹이 기업과 기업 사이의 정보 교환 및 서비스의 이용에 활용되고 있다. 이것을 가능하게 하는 것은 XML을 기반으로 하는 웹 기반 응용 서버이다. 세 번째는 웹사이트 구축에 대한 적용이다. 컴포넌트 기반 소프트웨어 개발 기술은 이미 존재하는 소프트웨어 컴포넌트를 재사용하여 새로운 응용을 위한 대형 소프트웨어 개발 기술이다. 이것은 코드의 재사용으로 개발비용을 줄이고, 소프트웨어 유지 관리의 부담을 경감시킨다[8].

3. 웹에서의 컴포넌트 유통

웹 기술과 데이터베이스 연동 기술을 근간으로 웹은 MMM(Middleware for Method Management)[1],[7]과 같이 분산 컴포넌트 소프트웨어의 저장 및 유통을 위한 기반 구조로 활용될 것이다. MMM은 사용 신청(subscribe)을 통한 컴포넌트 소프트웨어의 ppt(payment per transaction) 형태의 유통을 원하는 컴포넌트 소프트웨어 개발자를 위한 미

들웨어이며 요소는 표 2와 같다. 먼저 컴포넌트 베이스는 개발한 컴포넌트 소프트웨어의 저장고 역할을 담당한다. 서비스 엔진은 컴포넌트의 사용을 원하는 응용에 제공되며, 응용의 컴포넌트 사용 요구를 컴포넌트 수행 엔진에 전달하고, 그 결과를 응용에 제공하는 역할을 수행하며 XML을 이용하여 응용과 통신한다. 또한 각 서비스 엔진의 위치 정보는 서비스 엔진 등록 관리기에 등록한다. 컴포넌트 수행 엔진은 응용에서 필요로 하는 컴포넌트의 수행을 담당하며, 다수의 응용에서 요구하는 수행의 스케줄링 등의 자원관리를 수행한다. 각 컴포넌트 사용 응용과는 서비스 엔진을 통하여 컴포넌트 수행요구와 수행결과를 주고받는다. 서비스 엔진 등록관리기는 각 서비스 엔진의 위치, 네트워크 특성, 플랫폼 정보를 관리한다.

표 2. MMM의 구성

구성요소	역할
컴포넌트DB	· 컴포넌트 소프트웨어의 저장고
서비스엔진	· 컴포넌트의 사용을 원하는 응용에 제공 · 응용의 컴포넌트 사용 요구를 컴포넌트 수행 엔진에 전달 · 전달 결과를 응용에 제공하는 역할 · 응용과는 XML을 이용하여 통신함
컴포넌트 수행 엔진	· 응용에서 필요로 하는 컴포넌트의 수행 담당 · 자원관리 수행 · 다수의 응용에서 요구하는 수행의 스케줄링 · 각 컴포넌트 사용 응용과 서비스 엔진을 통하여 컴포넌트 수행요구와 수행결과를 주고 받음
서비스엔진 등록관리기	· 각 서비스 엔진의 위치 관리 · 네트워크 특성 관리 · 플랫폼 정보 관리

4. CBD 도구 비교

현재까지 개발된 CBD 도구들은 여러 가지 문제점을 가지고 있으며 특정 어플리케이션을 구축하기 위해 바이너리 컴포넌트를 생성하기는 쉽지만 컴포넌트의 조립방법은 직접적인 코딩을 통해 결합되기 때문에 실행환경하에서 컴포넌트대체(replacement, deletion/ connection)가 어렵다.

이들 컴포넌트들의 인터페이스를 이해하고 관련 컴포넌트들을 'plug-&play'를 해서 컴포넌트 시스템을 구축할 수 있는 소프트웨어 아키텍처 기반의 컴포넌트 개발 도구가 아직까지 없는 상황이다. 컴포넌트 기술이 보다 강력한 기능을 제공하기 위해서는 아키텍처 기반으로 컴포넌트의 생성, 조립 및 추출이 가능해야 하며 현재 해외에서도 소프트웨어 아키텍처 기술을 컴포넌트 기술 분야에 적용하려는

연구가 진행 중이다[2],[6]. CBD 도구와 그 특징을 정리하면 표 3과 같다.

표 3. CBD 도구 및 특징

개발사	제품명	특징
Computer Associates	Cool Products	· 컴포넌트와 어플리케이션 아키텍처의 모델링 도구로서 행위분석과 인터페이스 기반 설계의 개념을 사용
Compu-ware	Uniface	· 컴포넌트구축/legacy wrapping/컴포넌트조립/컴포넌트매치 등의 모듈 구성 · 생성된 컴포넌트를 선택하고 인터페이스를 통해 상호 링크시켜, 어플리케이션을 통합하는 GUI 접근 방식 제공
IBM	San Francisco Framework	· 프레임워크 기반의 개발 도구 · 인터넷, 인트라넷/엑스트라넷과 같은 컴퓨팅 환경의 중요한 비즈니스 솔루션이 빠른 개발을 위해 이용될 수 있는 서버 측 EJB컴포넌트 제공
Together Soft	Together	· e-solution 개발을 위한 컴포넌트 모델링/설계패턴/편집/컴파일/디버깅/버전 관리/문서화/조립/배치/실행 과정 지원

#### IV. 비교 및 분석

기존 웹 시스템 개발에서는 재사용성과 유지보수의 어려움이 발생하였고, 이에 따라 컴포넌트 기반의 개발 방법론이 나타나게 되었다. 개발된 각각의 컴포넌트는 컴포넌트가 제공하는 기능이 필요한 어플리케이션에 대해 재사용되기 때문에 전체 개발의 비용과 노력을 줄일 수 있다. 특히 중소규모의 시스템 개발에 있어서는 동일한 코드의 반복적인 사용이 많아, 이에 대한 효율성을 높이기 위해 컴포넌트의 사용이 절실히 요구된다. 하지만 기존의 컴포넌트 기반 개발 방법들이 메이저 급의 소프트웨어 개발 업체들을 위주로 그 효율성이 검증되고 있어 중소규모의 개발에 대한 효율성의 검증이 필요하다.

본 연구에서는 개발자가 15인 이하인 중소규모의 IT 관련 업체에서 5년 이상의 실무 개발 경험을 가진 개발자 15명과 IT 관련 학과의 3년 이상 교육 경력을 가진 교수 5명에 대해 각 컴포넌트 기술의 편의성, 시스템에 대한 적응성, 지원하는 컴포넌트의 다양성에 대해 1점에서 10점까지 점수를 부여하여 분석하였으며 그 결과를 그림 3, 그림 4, 그림 5에 나타내었다.

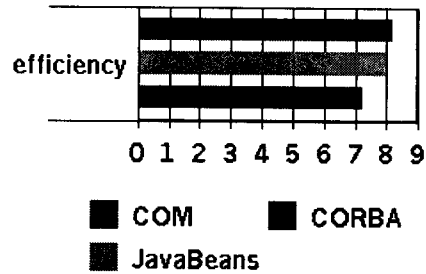


그림 3. 컴포넌트 사용 편의성

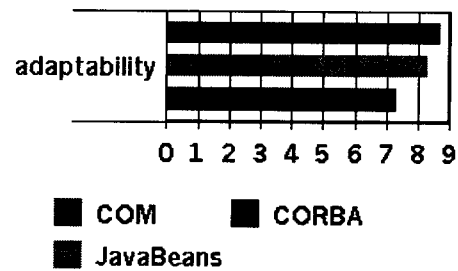


그림 4. 컴포넌트 적응성

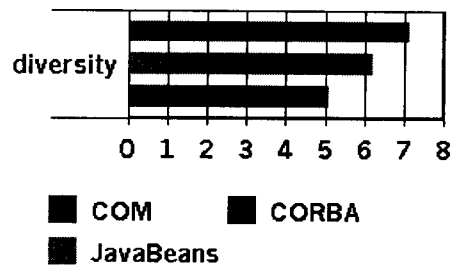


그림 5. 컴포넌트 다양성

따라서 표 4와 같이 MS 계열의 COM/DCOM/COM+와 자바빈즈 및 CORBA의 세 가지 컴포넌트 기술간의 비교 결과를 도출할 수 있다.

표 4. 컴포넌트 기술 비교

구분	COM/DCOM/COM+	JavaBeans	CORBA
특징	· 시장지배력 큼 · MS플랫폼 종속적 · 언어 독립성 높음	· 시장지배력 큼 · JVM/자바만 허용 (언어 종속적) · 플랫폼 독립성 높음	· 시장지배력 작음 · 플랫폼 독립성 높음 · 언어 독립성 높음
사용자 편의성	높음	비교적 높음	낮음
컴포넌트 지원 다양성	다양함	비교적 다양함	다양하지 않음
개발규모	중/소규모	중/소규모	중대규모

플랫폼의 독립성과 언어 독립성은 CORBA가 가장 높지만 중소기업의 개발 환경인 경우 MS사의 윈도우즈 2000 서버, 윈도우즈 NT4와 같은 운영체제가 거의 대부분을 이루고 있다. 또한 현재 시장에서 사용이 빈번한 개발환경은 비주얼 스튜디오5(Visual Studio5), 비주얼스튜디오6, 비주얼스튜디오 닷넷 등과 같은 MS 계열의 개발 환경들이다. 이들은 다양한 COM/COM+/DCOM의 지원하고 있으며, 다양한 컴포넌트의 개발이 가능하다. 따라서 MS 계열의 COM/DCOM/COM+가 편의성, 적응성, 상용으로 제공되는 컴포넌트가 다양한 것으로 나타났다. 또한 근래에 들어 자바의 확산으로 자바빈즈의 사용이 증가하고 있으며 그 결과 편의성과 적응성, 다양성이 증가하고 있는 것으로 나타났다. 이에 반해 CORBA는 상대적으로 비교 항목에 대해 약하게 나타났다. CORBA의 경우 개발 및 관리 운영의 환경이 일반적인 중소기업의 업체에서 다루기가 힘들다는 인식이 지배적이며, 실제 중소기업의 경우 CORBA를 기반으로 개발을 해본 개발자가 거의 없는 실정이었다. 따라서 중소기업의 웹 시스템 개발의 경우에는 MS 계열의 COM/DCOM/COM+의 사용이 개발에 대한 노력과 시간, 경비를 절감할 수 있을 것이다.

본 연구의 실험은 중소기업의 업체에 대해 컴포넌트 기반의 개발에 대해 이루어졌으며 대규모의 시스템에 대한 개발에 대한 결과는 차이가 있을 것이다. 실제 IT 현장에서는 대부분의 경우 중소기업의 웹 시스템의 개발이 주도적이며, 따라서 본 실험의 결과의 효용성을 기대할 수 있다.

## V. 결론

본 논문에서는 CBD 기반의 웹 시스템의 개발을 위한 관련 기술을 살펴보고, 특히 중소기업의 웹 시스템 개발에 대한 적용을 위해 기존 컴포넌트 기술들을 비교 분석하여, 문제점을 도출하고 이를 바탕으로 컴포넌트를 기반으로 한 웹 시스템의 개발·활용의 효용성을 높이기 위한 방안을 제안하고 실험하였다. 본 연구의 실험은 중소기업의 업체에 대해 컴포넌트 기반의 개발에 대해 이루어졌으며 실험 결과 중소기업의 웹 시스템 개발의 경우에는 MS 계열의 COM/DCOM/COM+의 사용이 개발에 대한 노력과 시간, 경비를 절감할 수 있을 것으로 나타났다.

본 연구결과와 활용을 통해 CBD 기반의 중소기업 웹 시스템 구축에서 개발비용을 감소시키고 유지 관리의 부담을 경감시킬 수 있을 것으로 기대된다.

## 참고 문헌

- [1] 홍기형, 서동수, "차세대 웹에서의 컴포넌트 소프트웨어", 정보처리학회지, 제6권 제3호, pp.45-51, 1999.
- [2] 전주현, 홍찬기, "컴포넌트기반의 웹 기반 교육 시스템 설계에 관한 연구", 정보처리학회논문지 D, 제8-D권 제6호, pp.673-680, 2001.
- [3] 서종화, 김진수, 김치수, "웹상에서 운영되는 원격교원연수 시스템", 정보처리학회논문지 A, 제9-A권 제1호, pp.121-128, 2002.
- [4] 김원영 외, "분산 환경을 위한 상호작용적 실시간 교육 시스템의 개발", 한국멀티미디어학회논문지, 제3권 제5호, pp.506-515, 2000.
- [5] 신규상 외 3인, "CBD지원도구의 설계 및 프로토타이핑", 한국정보과학회지, 2001.
- [6] R. N. Taylor, N. Medvidovic, K. M. Anderson, et al., "A Component and Message-Based Architectural Style for GUI Software," IEEE Transactions on Software Engineering, Vol.22, No.6, pp.390-406, June, 1996.
- [7] G. Riessen, H.A. Jacobsen and O. Gunther, MMM - Middleware for Method Management on the WWW, Web Engineering '99 International Workshop, 1999. ([http://budhi.uow.edu.au/webengineering99/accepted\\_pap](http://budhi.uow.edu.au/webengineering99/accepted_pap))

- ers/riessen.html)
- [8] Martin Gaedke, Joern Rehse, Gunuam Graef, A Repository to facilitate Reuse in Component Based Web Engineering, Web Engineering '99 International Workshop, 1999.  
([http://budhi.uow.edu.au/web-engineering99/accepted\\_papers/gaedke.html](http://budhi.uow.edu.au/web-engineering99/accepted_papers/gaedke.html))
  - [9] Don Box, Essential COM, Addison-Wesley, 1998.
  - [10] Michi Henning, and Steve Vinoski, Advanced CORBA Programming with C++, AddisonWesley, 1999.
  - [11] Butler Group, Component-Based Development: Application Delivery and Integration Using Componentised Software, 1998.
  - [12] Namkyu Cho, "Building the EJB component with CBD96," March, 2000.
  - [13] J.Penix, "Automated Component Retrieval and Adaptation Using Formal specifications," Ph.D Dissertation Dept. of Electrical and Computer Engineering and Computer Science, University of Cincinnati, Apr. 1998.
  - [14] Terry Quatrani, Visual Modeling with Rational Rose and UML, Addison-Wesley, 1998.
  - [15] Felix Bachman, Len Bass, "Technical Concepts of Component-Based Software Engineering," Technical Report CMU/SEI-2000-TR-008, 2000.

나 윤 지(Yun-Ji Na)

정회원



1994년 : 경북대학교 생명공학부  
(이학사)  
2001년 : 충북대학교 컴퓨터공학  
(공학석사)  
2001년 ~ 현재 : 충북대학교 컴퓨터  
공학(박사과정)

1999년 ~ 2000년 : 뉴욕공과대학(NYIT)대학원  
Communication ART 전공(석사과정 2학기 수료)

전)대전보건대학 컴퓨터정보처리과 초빙교수

<관심분야> : CBD, 멀티미디어, 전자상거래, 게임,  
네트워크