

□ 특집 □

차세대 웹에서의 컴포넌트 소프트웨어

홍 기 형[†] 서 등 수^{††}

◆ 목 차 ◆

- | | |
|----------------------------|-------------------|
| 1 서 론 | 4 웹에서의 컴포넌트 활용 동향 |
| 2 컴포넌트 소프트웨어 기술 | 5 결 론 |
| 3 웹, 컴포넌트, 그리고 분산 응용 소프트웨어 | |

1. 서 론

Web은 지난 10년 동안 정보 기술 분야에서 최대의 관심이 쏟아진 분야이다. 사용하기 쉬운 사용자 인터페이스와 개방성 등으로 인하여 양적, 질적으로 폭발적인 성장을 해왔으며, 앞으로도 그 성장세가 꺾이지 않을 전망이다. 초창기의 단순 문자 기반 분산 하이퍼텍스트 시스템에서 현재는 이미지, 애니메이션, 동영상 및 음성 등의 연속 미디어를 포함하는 분산 하이퍼미디어 시스템으로 발전하였으며, 소수의 전문가 층을 위한 정보 서비스 환경에서 불특정 다수를 대상으로 하는 환경으로 발전하였다. 또한, HTML 문서를 기반으로 하는 정보의 수동적 응용환경에서 전자상거래 등을 위한 능동적 응용환경으로 발전하고 있으며, 최근에 등장한 XML은 이러한 발전 방향에 큰 영향을 미치고 있다. 이제 웹은 문서 형태의 분산 정보 서비스 시스템에서 분산 처리 시스템의 한 표준 프레임으로 자리를 잡은 것이다. 웹 브라우저는 다양한 응용 소프트웨어에서 요구하는 다양하고 복잡한 사용자 인터페이스 요구를 통일된 프레임웍에서 지원할 수 있으며, 웹 서버

는 분산 미들웨어와 결합하여 웹 응용 서버(Web application server)로 발전하고 있다.

한편, 기업 활동의 성공을 위해서 상업용 소프트웨어, 또는 비즈니스 응용 소프트웨어의 적기 개발 및 적용은 매우 중요하다. 그러나, 현재까지도 대부분의 응용 소프트웨어 개발이 수(手)작업으로 기존 소프트웨어 개발 기법에 따라 이루어진다. 이에 따라 개발 기간이 길어지고 경우에 따라서는 시의 적절한 기업 활동에의 적용이 불가능해서, 응용 소프트웨어 개발 작업이 취소되거나, 불안정한 응용 소프트웨어의 적용으로 해당 기업 활동에 부정적인 영향을 줌으로써 기업의 손해가 종종 발생한다. 특히, 인터넷 기반 기업 활동의 증가와 전자상거래의 확산은 상업적 응용 소프트웨어에 대하여 다음과 같은 사항을 만족할 것을 요구하고 있다.

- 비즈니스 환경 및 기술 변화를 수용할 수 있는 설계
- 새로운 기능의 손쉬운 확장
- 비즈니스 업무의 변화 요구를 빠른 시간에 반영
- 기능의 확장과 환경의 변화의 반영은 기존의 정보 및 소프트웨어의 재사용을 통하여 저 투자분의 보호

[†] 정회원 : 성신여자대학교 컴퓨터정보학부 전임강사

^{††} 정회원 : 성신여자대학교 컴퓨터정보학부 전임강사

이러한 요구사항을 만족시키기 위하여 소프트웨어의 컴포넌트화(化)와 컴포넌트 기반 응용 소프트웨어의 개발에 관한 많은 연구가 진행되고 있다[1].

컴포넌트 기반 응용 소프트웨어의 개발은 독립적인 기능을 담당하는 다양한 컴포넌트 소프트웨어의 집합으로부터, 해당 업무의 수행에 필요한 기능을 담당하는 하나 이상의 컴포넌트를 결합하여 해당 업무를 위한 소프트웨어를 개발하는 것을 말한다. 현재 대표적인 컴포넌트 기반 응용 소프트웨어 개발 및 운용을 위한 모델은 마이크로소프트의 COM과 선의 JavaBeans, 그리고 OMG의 CORBA가 있다.

현재, 웹에서의 컴포넌트 사용은 웹 브라우저(클라이언트)에 집중되고 있으며, 예로는 마이크로소프트의 ActiveX, Java applet에 의한 브라우저 측 기능 확장을 들 수 있다. 이러한 웹 브라우저, 또는 클라이언트 측에서의 컴포넌트 사용은 이미 성숙 단계이며, 앞으로는 서버 측 서비스에서의 컴포넌트 활용이 보편화 될 것으로 예상된다.

본 고에서는 현재의 웹과 컴포넌트 소프트웨어 기술의 결합 상황에 대하여 알아보고, 앞으로의 웹 환경에서의 컴포넌트 소프트웨어 기술이 어떠한 역할을 할 것인지에 대하여 고찰하고자 한다. 먼저, 컴포넌트 소프트웨어 기술의 현황과 미래를 알아보고, 컴포넌트 소프트웨어 기술을 결합한 웹의 현재와 미래를 기술한다

2. 컴포넌트 소프트웨어 기술

2.1 컴포넌트 소프트웨어의 정의

컴포넌트 소프트웨어(이하 컴포넌트 S/W)는 잘 알려진 기능을 수행하도록 구현한 단위 소프트웨어로, 구체적인 구현은 컴포넌트 S/W의 사용자(다른 컴포넌트 S/W)에게 숨기고 잘 정의된 인터페이스를 통하여 해당 기능을 제공한다. 컴포넌트

S/W는 객체지향 특성을 많이 갖고 있으나, 객체지향언어로만 구현되는 것은 아니다.

컴포넌트 소프트웨어가 갖추어야 하는 특징은 다음과 같다.

- 컴포넌트를 구별하는 식별자를 가짐
- 컴포넌트의 사용 상황을 알 수 있어야함
- 동일한 서비스를 제공하는 새로운 버전으로의 변경이 해당 컴포넌트를 사용하는 응용 소프트웨어나 다른 컴포넌트에 영향을 주지 않아야 함
- 서비스는 인터페이스를 통해서 제공하고 컴포넌트의 인터페이스 변경이 없어야함
- 자신의 인터페이스에 대한 명확한 설명을 함께 제공하여야함
- 개발 도구나 언어, 그리고 플랫폼에 독립적인 (재)사용이 가능하여야함
- 실행시점에 동적 재사용이 가능하여야함

2.2 컴포넌트 기술 동향

컴포넌트 S/W는 약 30년 전에 등장한 모듈화 설계에서부터 시작되었다고 볼 수 있으며, 객체지향 설계 기법의 등장과 마이크로소프트(이하 MS)의 복합문서(compound document) 기술의 성공으로 개발과 확산이 촉진되었다. 초기의(현재도 그렇지만) MS 워드에서 엑셀 등의 다른 소프트웨어에서 작성된 문서를 삽입하는 기능과 삽입된 문서에 대한 작업은 워드 상에서 해당 문서를 작성할 때 사용한 소프트웨어의 인터페이스를 그대로 사용하게 한 것이 복합문서 기술이며, 현재는 OLE/COM(Object Linking and Embedding/ Component Object Model)으로 알려진 것이다.

현재 컴포넌트 소프트웨어 시장에서 가장 큰 위치를 점하는 컴포넌트 모델 및 운영 구조는 MS의 COM[2]과 Sun의 JavaBean 이다. 그리고, IBM이 강하게 밀고 있는 OMG의 CORBA(Common Object Request Broker Architecture)[3], Oracle의

NCA (Network Computing Architecture) Cartridge 등이 있다. (표 1)은 이들을 비교한 것이다.

(표 4) 컴포넌트 기술 비교

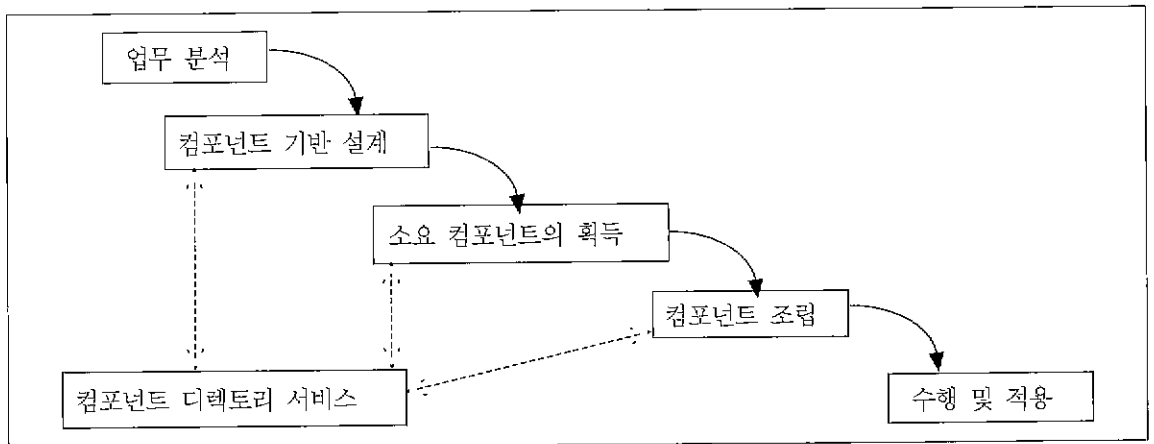
	COM	JavaBeans	CORBA	NCA Cartridges
시장 장악력	강함	커짐	커짐	적음
컴포넌트 운영 주체	운영체제 (기본적으로 Windows)	Java Virtual Machine	다수의 시스템 존재	Oracle 제품군
언어 독립성	높다	Java만 허용	높다.	높다.
플랫폼 독립성	제한적	높다	높다.	높다.

- COM은 개인용 컴퓨터 운영체제인 윈도우 (Windows 98, NT)로 그 플랫폼이 제한적이다. 그러나 윈도우가 차지하는 시장 규모가 매우 커서 현재 가장 많이 이용되는 컴포넌트 기술이며, 앞으로도 이 분야에서 그 위치가 보장되어 있는 기술이라고 할 수 있다.
- JavaBeans은 클라이언트 측의 Java 기반 GUI의 구현을 위한 기술로 컴포넌트 작성 언어가 Java로 제한되어 다른 컴포넌트 기술에 비하여 언어 독립성이 떨어진다.

- CORBA는 원래 분산 객체 중간모(middleware) 명세로, COM에 비하여 플랫폼에 대한 독립성이 높고, JavaBeans에 비하여 언어 독립성이 높다.
- NCA는 DBMS 개발 업체로는 유일하게 독립적인 컴포넌트 명세를 보유하려는 Oracle 사의 전략에 따라 제안된 것이지만 CORBA 규격을 따르는 분산 시스템 위에 Oracle 고유의 컴포넌트 양식을 구현한 것이다.

현재까지의 컴포넌트 기술은 대부분 분산 응용에서 클라이언트 측의 기능 확장과 개발에 초점을 맞추었으나, 앞으로는 서버 측의 컴포넌트 기술에 많은 개발이 이루어질 것으로 보인다. COM+, Enterprise JavaBeans, CORBA 등이 이에 해당한다. 컴포넌트 기술을 독자적으로 확보하고 있는 MS와 IBM, Sun을 주축으로 하는 JavaBeans 및 CORBA가 상호 호환성을 보장하는 방향으로 나아가고 있다.

(그림 1)은 컴포넌트 기반 응용 소프트웨어의 개발 과정을 보이고 있다. 이 개발 과정은 현재 성숙단계에 이르지 않은 것으로, 컴포넌트 소프트웨어 개발 및 유통의 목표라고 할 수 있겠다.



(그림 1) 컴포넌트 기반 소프트웨어 개발 과정

- 업무 분석 및 컴포넌트 기반 설계 : 응용의 요구사항을 분석하고 이를 바탕으로 컴포넌트 모델 기반 설계를 수행하여, 해당 응용을 구현하는데 필요한 컴포넌트의 명세를 인터페이스와 그 서비스(기능)으로 추출함.
- 소요 컴포넌트의 획득 : 응용 구현을 위해 필요한 컴포넌트를 다음의 방법을 통하여 구함.
 - 구입 : 필요한 컴포넌트를 구입하여 확보
 - 사용 신청 : 필요한 컴포넌트를 제공하는 사업자로부터 사용 횟수나 기간에 따라 비용을 지불할 것을 신청하여 확보
 - 변경 : 기 확보한 컴포넌트의 기능 확장을 통한 확보
 - 랩핑(wrapping) : 기 확보하고 있는 전통적 (비 컴포넌트) 응용 소프트웨어를 컴포넌트화하여 확보
 - 생성 : 필요한 컴포넌트를 개발하여 확보
- 컴포넌트의 조립 : 확보한 컴포넌트를 조립하여 업무 수행을 위한 응용 소프트웨어 구축
- 수행 및 적용 : 컴포넌트 기반 응용 소프트웨어의 수행과 업무에의 적용

3. 웹, 컴포넌트, 그리고 분산 응용 소프트웨어

(그림 2)는 응용 소프트웨어의 개발 및 운용 환경, 웹, 그리고 컴포넌트 기술의 변화와 이들 사이의 관계를 간단히 도식화 한 것이다. 웹의 등장 직전의 응용 소프트웨어 환경은 각 응용 소프트웨어 로직의 수행이 하나의 범용 컴퓨터에 집중되는 중앙 집중 구조에서 클라이언트/서버 구조로의 전이가 이루어지고 있었다.

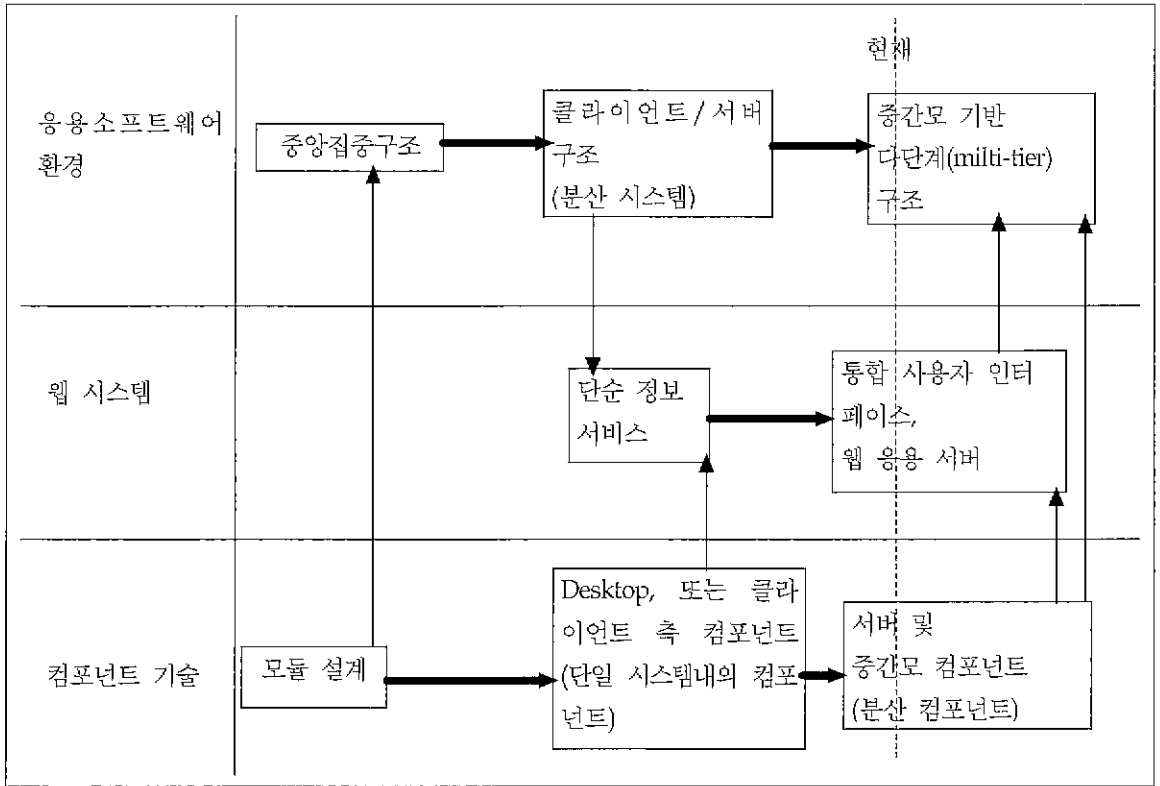
클라이언트/서버 구조는 통신 기술의 발전과 개인용 컴퓨터의 확산에 따라 응용 소프트웨어가 서버의 로직과 클라이언트의 로직으로 분리되어

수행되는 2 단계 구조이다. 서버의 로직은 데이터베이스 서비스, 트랜잭션 처리 서비스 등이며, 클라이언트 로직은 응용에 고유한 자료 처리와 보다 발전한 사용자 인터페이스 기능이였다. 이러한 클라이언트/서버 구조는, 객체 지향 기술의 발전과 분산 처리 기술의 발전, 그리고 인터넷의 보급은 이기종 분산 서버와 이기종 분산 클라이언트 플랫폼을 상호 연동하기 위한 중간모(middleware)를 분산 서버 로직과 분산 클라이언트 로직 사이에 두는 3 단계(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. 멀티미디어 기술의 발전으로 HTML에 다양한 멀티미디어 정보가 포함되기 시작하였으며, 이에 따라 주로 클라이언트의 멀티미디어 기능확장을 위하여 외부 뷰어(external viewer)의 plug & play 방식이 도입되었다. 이것이 웹에서 컴포넌트 사용의 초기 단계이며, 이후 Java applet과 JVM(Java virtual machine), 또는 COM 기반의 ActiveX를 채용하여 보다 발전된 컴포넌트의 사용으로 진행되어 현재에 이르고 있다.

웹은 XML(Extended Markup Language)을 등장으로 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 서버 로직을 구현한 것이다.

차세대 웹에서의 컴포넌트 활용은 크게 다음의 두 가지 방향으로 나타날 것이다.



(그림 2) 응용 소프트웨어, 웹, 컴포넌트 기술

- 컴포넌트의 유통 기반 구조로서의 웹
- 컴포넌트 기반 웹 응용 개발

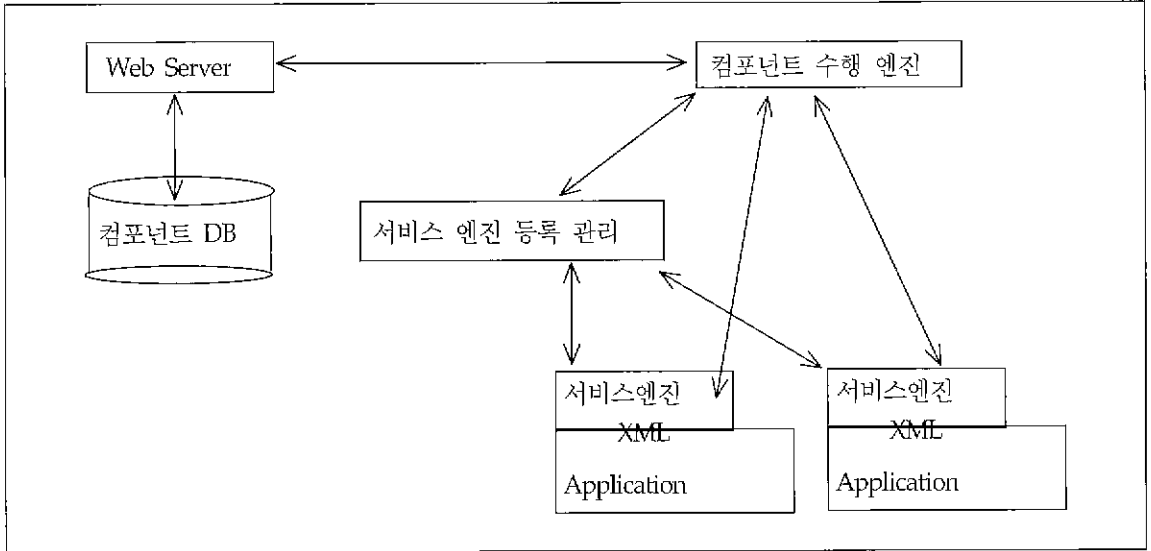
4. 웹에서의 컴포넌트 활용 동향

4.1 컴퍼넌트 유통 기반 구조로서의 웹

web 기술과 DB 연동 기술을 근간으로 web은 분산 컴포넌트 소프트웨어의 저장 및 유통을 위한 기반 구조로 활용될 것이다. [4]에서 제안한 MMM (Middleware for Method Management)는 이러한 예이다. MMM의 구성은 (그림 3)과 같다. 기존의 판매를 통한 소프트웨어의 유통이 아닌, 2장에서 설명한 사용 신청(subscribe)을 통한 컴포넌트 소프트웨어의 유통(payment-per-transaction)을 원하는 컴포넌트 소프트웨어 개발자를 위한 중간

모인 MMM을 구성하는 요소는 다음과 같다.

- 컴포넌트 DB : 개발한 컴포넌트 소프트웨어의 저장고
- 서비스 엔진 : 컴포넌트의 사용을 원하는 응용에 제공되며, 응용의 컴포넌트 사용 요구를 컴포넌트 수행 엔진에 전달하고, 그 결과를 응용에 제공하는 역할을 수행한다. 응용과는 XML을 이용하여 통신한다. 또한, 각 서비스 엔진의 위치 정보는 서비스 엔진 등록 관리기에 등록한다.
- 컴포넌트 수행 엔진 : 응용에서 필요로 하는 컴포넌트의 수행을 담당하며, 다수의 응용에서 요구하는 수행의 스케줄링 등의 자원관리를 수행한다. 각 컴포넌트 사용 응용과는 서비스 엔진을 통하여 컴포넌트 수행



(그림 3) 컴포넌트 소프트웨어 유통 기반 구조인 웹

요구와 수행 결과를 주고 받는다.

- 서비스엔진 등록 관리기 : 각 서비스 엔진의 위치, 네트워크 특성, 플랫폼 정보를 관리한다.

4.2 컴포넌트 기반 웹 개발

재사용성 및 확장성을 지원하는 컴포넌트 기반 소프트웨어 개발 방법은 웹의 개발에도 적용되고 있다. 크게 다음의 세 가지 형태로 나타나고 있다.

(1) 웹 클라이언트

3장에서 언급한 바와 같이, Java applet과 Active X를 기반으로 하여 웹 클라이언트, 즉 브라우저의 개발이 이미 성숙 단계에 이르렀다.

(2) 웹 기반 응용 서버 개발

랩핑을 통한 전통적인 상업적 응용 소프트웨어의 컴포넌트화가 가능해 지고, XML의 개발과 분산 처리 환경에서의 데이터교환 양식으로의 채용으로 웹이 기업과 기업 사이의 정보 교환은 물론이고, 서비스의 이용이 용이해 지고 있다. 이것을 가능하게 하는 것은 XML을 기반으로 하는 웹 기반 응용 서버이다. 웹 기반 응용 서버의 한 예는 webMethods 사의 XML을 기반으로 기업 간 전자

상거래를 위한 기업간 업무 통합 도구[5]를 들 수 있다.

(3) 웹 사이트 구축

컴포넌트 기반 소프트웨어 개발 기술은 이미 존재하는 소프트웨어 컴포넌트를 재사용하여 새로운 응용을 위한 대형 소프트웨어 개발 기술이다. 이것은 코드의 재사용으로 개발 비용을 줄이고, 소프트웨어 유지 관리의 부담을 경감시킨다. 그러면, 이미 존재하는 웹 사이트 (또는 웹 페이지)로부터 보다 크고 새로운 웹 사이트의 개발에도 컴포넌트 기술을 이용할 수 있다. WebComposition [6]은 바로 이러한 웹 사이트 개발을 위한 컴포넌트 기술의 활용이다.

5. 결 론

본 연구에서는 웹에서의 컴포넌트 소프트웨어 기술에 대하여, 현재의 상황과 앞으로의 동향에 대하여 알아보았다. 컴포넌트 기술은 현재 매우 빠른 속도로 발전하고 있으며, 기존의 소프트웨어 개발 방법 뿐 아니라 유통 방법에 일대 변화를

가져 올 것으로 생각된다. 웹은 이러한 컴포넌트 소프트웨어의 새로운 유통 환경의 기반 구조로 컴포넌트 저장, 관리 및 서비스가 웹을 통하여 이루어 질 것이다.

컴포넌트 소프트웨어 개발 기술은 웹 자체의 개발에 적용되어, 이미 브라우저의 경우에는 Java applet, Active X 등을 통한 컴포넌트의 활용이 성숙 단계에 이르렀다. 웹 서버의 개발에 있어서는 컴포넌트 기술은, 웹이 분산 하이퍼미디어 시스템의 단계를 넘어서 기업간 업무 통합과 기존의 전통적 응용 (legacy application) 서비스 통합 서비스가 가능한 웹 서버의 개발을 가능하게 하고 있다. 해외에서는 이미 지난해부터 컴포넌트 기술과 웹의 결합으로 새로운 방식의 소프트웨어의 개발과 유통을 위한 도구의 개발이 본격화하고 있다.

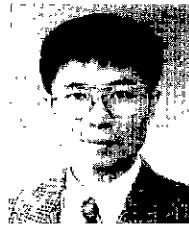
참고문헌

[1] Butler Group, Component-Based Development: Application Delivery and Integration Using Componentised Software, 1998.
 [2] Don Box, Essential COM, Addison-Wesley, 1998.
 [3] Michi Henning, and Steve Vinoski, Advanced CORBA Programming with C++, Addison-Wesley, 1999.
 [4] 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/web-engineering99/accepted_papers/riessen.html)
 [5] Anne Thomas, webMethods B2B Integration

Server: Business-to-Business E-commerce Solutions, 1998.

(http://www.webmethods.com/solutions/products/webM_b2b/tech_wp.html)

[6] Martin Gaedke, Joern Rehse, Guntram 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)



홍 기 형

1985년 서울대학교 전자계산기공학과 (공학사)
 1987년 한국과학기술원 전산학과 (공학석사)
 1994년 한국과학기술원 전산학과 (공학박사)

1987년-1991년 한국과학기술원 전산학과 조교
 1991년-1994년 (주)컴퓨터응용기술 개발실장
 1994년-1998년 한국전자통신연구원, 선임연구원
 1998년-현재 성신여자대학교 컴퓨터정보학부 전임강사
 관심분야 : 데이터베이스 시스템, 웹정보시스템, CTI, 컴포넌트 소프트웨어



서 동 수

1986년 중앙대학교 컴퓨터공학과 (이학사)
 1990년 영국 맨체스터 이공대학 (이학석사)
 1991년 영국 맨체스터 이공대학 (공학박사)

1994년-1998년 한국전자통신연구원 선임연구원
 1998년-현재 성신여자대학교 컴퓨터정보학부 전임강사
 관심분야 : 분산객체기술, 정형기법, 소프트웨어재사용