CBD 프로세스 지원을 위한 컴포넌트 저장소의 구축

(Construction of Component Repository for Supporting the CBD Process)

차 정 은 † 김 행 곤 ††
(Jung-Eun Cha) (Hang-Kon Kim)

요 약 컴포넌트 기반 개발(CBD : Component Based Development)은 비즈니스 시스템의 요구를 해결할 수 있는 최상의 전략적 방법으로 빠르게 확산되고 있다. 이는 CBD가 소프트웨어 컴포넌트의 조립을 통해 어플리케이션 개발이 가능한 새로운 패러다임으로서, 비즈니스 프로세스의 빠른 변화에 능동적으로 대처하며, 증가되는 생산성 향상의 요구를 충족시킬 수 있기 때문이다. 특히, 컴포넌트 저장소는 컴포넌트의 개발과 유통 그리고 재사용을 위해 가장 중요한 부분으로 컴포넌트 그 자체뿐 아니라 컴포넌트 개발의 각 단계로부터 생산된 결과물들을 저장하고 관리할 수 있다.

본 논문에서는 CBD 프로세스의 효과적인 지원을 위해 컴포넌트 저장소 구축을 중심으로 실제적인 적용 기술을 제시하고 이를 바탕으로 컴포넌트 저장소 관리 시스템(CRMS : Component Repository Management System)의 프로토타이핑을 개발한다. CRMS는 컴포넌트 아키텍처에 기반하여 다양한 컴포넌트의 형상을 관리하며, 어플리케이션 개발자들은 자신의 프로젝트에 이용하기 위한 컴포넌트를 검색하고 관련 이해 정보를 획득할 수 있다. 본 논문을 통해 CBD 환경을 실제화 하는 공용 컴포넌트 저장소 구축으로의 실제적 접근을 제시함으로써 컴포넌트 개발을 지원하는 CBD 프로세스 및 컴포넌트의 저장소에 대한 선행 연구로 이용하고자 한다.

키워드 : 컴포넌트 기반 개발, 컴포넌트 저장소, 컴포넌트 메타 데이타, 컴포넌트 브라우징 및 검색

Abstract CBD(Component Based Development) has become the best strategical method for the business application. Because CBD is a new development paradigm which makes it possible to assemble the software components for application, it copes with the rapid challenge of business process and meets the increasing requirements for productivity. Since the business process is rapidly changing, CBD technology is the promising way to solve the productivity. Especially, the repository is the most important part for the development, distribution and reuse of components. In component repository, we can store and manage the related work-products produced at each step of component development as well as component itself. In this paper, we suggested a practical approach for repository construction to support and realize the CBD process and developed the CRMS(Component Repository Management System) as implementation product of the proposed techniques. CRMS can manage a variety of component products based on component architecture, and help software developers to search a candidate component for their project and to understand a variety of information for the component. In the paper, a practical approach for component repository was suggested, and a supporting environment was constructed to make CBD to be working efficiently. We expect this work wall be valuable research for component repository and the entire supporting Component Based Development Process.

Key words : CBD(Component Based Development), Component Repository, Component metadata, Component Browsing and Retrieval

† 정 회 원 : ETRI 컴퓨터소프트웨어기술연구소 S/W공학부 선임연구원
mary2743@etr.re.kr
†† 정 회 원 : 대구가톨릭대학교 컴퓨터공학과 교수
hangkon@cuth.cataegu.ac.kr
논문접수 : 2001년 1월 30일
심사완료 : 2002년 5월 20일

1. 서 론

소프트웨어 시스템들이 인터넷과 웹 중심의 인프라스트럭쳐(Infrastructure)로 전개되어감에 따라, 네트워크 상에서의 정보 공유가 보편화되고, EDI, EC, ASP와 같은 새로운 비즈니스 시스템이 출현하고 있다. 따라서 시스템 개발과 운영에서 이러한 급격한 변화를 수용하고 증가되는 생산성 요구를 충족시킬 수 있는 새로운 접근이 필요하다. 특히, 이질적인 환경적, 기능적 요소들 간의 상호운영성을 허용하고 동적 변화에 민첩하게 대응할 수 있는 컴포넌트 기반 개발(CBD : Component Based Development)은 비즈니스 시스템 요구를 위한 최상의 솔루션으로 인식되고 있다[1].

CBD는 "부품화된 소프트웨어의 생산과 재사용"의 실현이다. 즉, 기성품 형태의 이미 완성된 실행 모듈의 조립 및 커스터마이징에 의해 어플리케이션 시스템을 구성함으로써 소프트웨어 대량 생산의 산업화를 추구하는 패러다임이다. 따라서, 기존 방법론에 의한 재사용의 한계를 컴포넌트 부품화 및 실행성에 따른 분산과 독립성을 통해 해결함으로써 DCOM, EJB, CORBA 등과 같은 실제적인 기술적 결과를 제시하고 있다. 이는 잘 정의된 인터페이스를 기반으로 소프트웨어 부품들의 조립에 의해 어플리케이션 개발의 생산성과 유지보수성을 보장할 수 있는 기술적 진보성과, 웹과 자바와 같은 인터넷 기술의 지원을 통한 컴포넌트 보급과 획득의 대중화, 그리고 이에 따른 경제성에 바탕을 두고 있다. 이러한 CBD의 실용적인 완성은 단편적인 개발 단계의 하나의 이론, 하나의 도구에 의해서는 불가능하다[2]. (그림 1)에서 표현된 것과 같이 아키텍쳐 기반의 컴포넌트 명세화 및 구현과 패키지화 그리고 생산된 컴포넌트들의 재사용성 관리 및 배포와 조립에 의한 어플리케이션 생성에 이르는 체계적인 프로세스가 컴포넌트 저장소를 중심으로 병행되어져야만 한다. 즉, 컴포넌트 기반의 개발을 위해서는 저장소를 핵심 축으로 컴포넌트의 추출과 생산, 배포 및 관리 등에 대한 일련의 "컴포넌트 재사용을 위한 개발(Development for Component Reuse)" 프로세스와 생산된 컴포넌트의 검색, 이해, 구매 및 새로운 어플리케이션으로의 적용 등을 포함하는 "컴포넌트 재사용에 의한 개발(Development with Component Reuse)"을 설명할 수 있는 통합된 접근 방식이 필요하다.

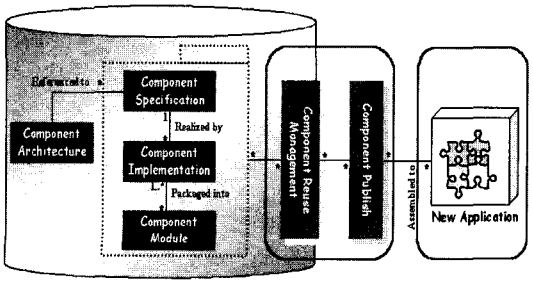

그림 1 CBD 전개를 위한 기술적 접근

본 논문에서는 CBD 프로세스의 효과적인 지원을 위해 컴포넌트 저장소 구축을 중심으로 접근 기술들을 제시하고 이를 바탕으로 CBD 방법론을 지원하는 실제적인 저장소 도구의 프로토타이핑을 개발하였다. 컴포넌트 저장소 관리 시스템(**CRMS** : **C**omponent **R**epository **M**anagement **S**ystem)은 CBD 방법론 적용에 의해 분석, 설계, 구축된 클라이언트/서버 시스템이다.

표 1 두 가지 관점에서의 CBD 프로세스

구 분	설 명
Development for Component Reuse	컴포넌트의 식별, 추출, 개발에 관련된 일련의 행위들로 고수준의 재사용 잠재성을 가진 가능성 있는 컴포넌트를 생산하는 과정
Development with Component Reuse	이미 존재하는 컴포넌트와 관련 결과물들의 활용을 극대화함으로써 비즈니스 요구에 대한 솔루션 시스템 구축을 지원하는 과정

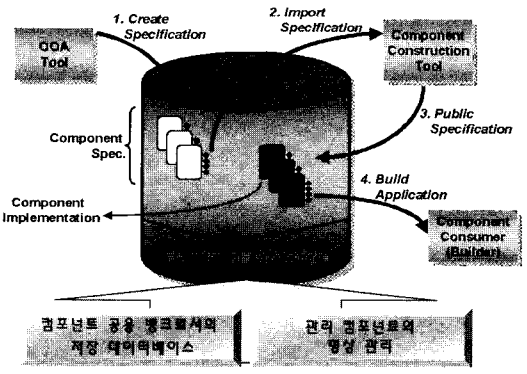

그림 2 컴포넌트 저장소의 전반적인 개요

즉, 컴포넌트 각 생명 주기에서 생산된 결과물 및 이와 관련된 메타 정보들을 형상 관리의 측면에서 체계적으로 관리하기 위한 서버 시스템과 웹 상에서 사용자가 편리하고 효과적으로 컴포넌트를 검색, 이해, 획득 등에 관한 총체적인 유통 체제를 담당하는 클라이언트 시스템으로 구성된다. 특히 사용자 관점에서 커스터마이즈 가능한 객체의 집합으로서 조립 원칙과 같은 경험적 자

산도 저장소에서 관리한다. 뿐만 아니라, 컴포넌트에 관한 다양한 조작 관점에 따른 컴포넌트 메타 데이타를 식별, 정의, 분류하고 이를 저장소의 저장 및 검색 프로세스에 적용하였다.

2장에서는 CBD 프로세스 및 연구 현황에 대해 간단히 제시하고 3장에서는 컴포넌트 저장소를 중심으로 CBD 프로세스 지원을 위한 기술을 제시하며 4장에서는 실제 프로토타이핑 시스템인 CRMS를 개발하였다. 그리고 5장에서 결론 및 향후 연구로 마무리한다.

2. 관련 연구

2.1 컴포넌트 저장소

컴포넌트 저장소는 비즈니스 어플리케이션 구축을 위해 요구되는 컴포넌트를 찾고 공급함으로써 어플리케이션 개발을 위한 환경으로서, 컴포넌트의 개발과 유통 그리고 활용을 지원하기 위한 실질적인 도구가 되는 라이브러리 시스템이다. 따라서 컴포넌트 저장소는 컴포넌트의 분석, 설계 및 구현 결과물에 이르는 컴포넌트 라이프사이클의 모든 정보들을 정의된 아키텍쳐에 따라 저장, 등록 관리하며, 보다 진보된 검색 서비스와 정보의 브라우징 기능을 통해 재사용을 지원하는 도구이다. 컴포넌트 저장소는 (그림 2)와 같이 두 가지 관점에서 컴포넌트의 생성과 활용을 위한 중심 매개체로서 도식화될 수 있다. 따라서 컴포넌트 생성과 검증에 관련된 도구의 개발이나, 형상 관리 및 유통을 위한 프로세스 정의는 컴포넌트 저장소에서 제공하는 일치성 있는 메타 정보 및 사용자 피드백 정보 분석을 통해 성숙된 CBD 환경으로 전개할 수 있다[3]. 현재, 컴포넌트 공유와 툴간의 상호운용성 수행을 목적으로 상용화되어 출시되고 있는 제품으로 MS Repository 2.0, SELECT Component Manager, 그리고 Oracle의 SEDONA 등이 컴포넌트기반 어플리케이션의 단계별 지원을 위한 비주얼 도구의 집합 형태로 제공된다[4][5].

2.2 CBD 프로세스

CBD는 독립적인 비즈니스 프로세싱 모듈로서 컴포넌트들의 통합에 의한 높은 융통성과 유지보수성을 창출한다. 따라서 CBD 프로세스의 진행은 관점에 따라 다음의 두 가지 부분으로 구분된다. 이 관점은 저장소로 상품화된 컴포넌트를 생산, 공급하는 입력 프로세스와 비즈니스 솔루션 개발을 위한 컴포넌트 수요 프로세스를 의미한다. 그러므로 실질적인 프로젝트 수행에서 요구되는 필수적인 작업들을 중심으로 재구성한 큰 그림은 (그림 3)과 같다. 컴포넌트 저장소 중심의 재사용 요소의 "획득-이해-적용"이라는 기본적인 재사용 원칙에

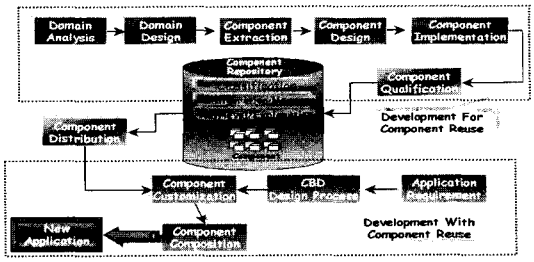

그림 3 CBD 프로세스

준하며 컴포넌트 확보 과정에서는 도메인 공학을 기반으로 하는 재사용 자산의 식별 및 추출을 위한 일련의 세부 프로세스를 포함하고 있다.

2.3 CBD의 연구 현황

소프트웨어의 집중 전략 부분으로서 CBD에 대한 연구 및 프로젝트의 개발이 활발히 진행 중이다. 따라서 컴포넌트 개발 프로세스의 표준화나 도구화, 컴포넌트 저장소 구축이 시도되고 있으며, 비즈니스 도메인에 적절한 컴포넌트의 확보 및 유통, 적용 등을 포함하는 전체 프로세스에 대한 전사적인 접근이 고려되고 있다. 이미 외국에서는 상업적인 컴포넌트 유통 시장이 형성되어 활용되고 있으나 명확한 접근 방식 및 표준 형태의 부재로 생산성 있는 CBD 전개가 어려운 실정이며, 이에 대한 해결책으로 컴포넌트 저장소 개발의 필요성과 중요성에 대한 일치된 의견이 제시되고 있다. 따라서 개별적 기술로 나타나는 CBD 연구들을 시스템 개발에 실제로 적용하기 위해서는 (그림 4)와 같은 세부적 접근 항목들의 구체적인 결과물들을 저장소를 중심으로 CBD 프로세스 환경 내에서 연계 지을 수 있는 단계별 지원 기술의 정의 및 활용이 필요하다.

3. 컴포넌트 저장소 시스템

3.1 개요

본 논문에서 초점을 맞춘 컴포넌트 저장소는 다양한 관점(도구, 사용자, 개발)에서 관리 정보의 통합과 공유를 위한 매개체로서의 저장소가 아닌, 컴포넌트 결과물들이 효과적으로 활용될 수 있는 통합 관리 시스템으로서의 컴포넌트 공용 뱅크인 저장소이다. 컴포넌트 개발자는 컴포넌트 사용자에게 꼭 필요한 컴포넌트를 생산, 공급하고, 사용자는 자신의 요구을 충족시킬 수 있는 컴포넌트를 획득할 수 있는 기능 지원이 필요하다. 이를 위해서는 재사용 자원의 생산과, 관리 그리고 활용을 위한 기술과 유사한 접근 체계가 필요하다.

따라서, 컴포넌트 저장소 구축을 위해 저장소의 관리

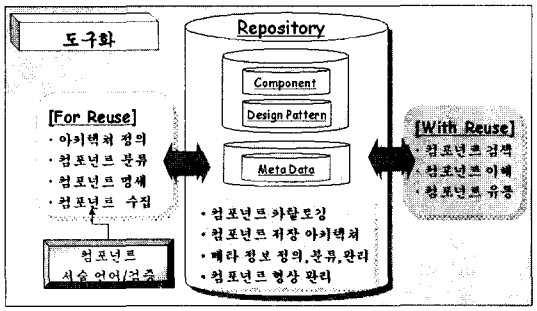

그림 4 저장소 중심의 CBD와 관련된 기반 기술

요소들의 저장 아키텍처를 정의하고 재사용을 위한 다양한 관점에서 메타 데이타를 식별, 분류하고 이에 대한 실제화된 예를 제시하고자 하였다.

3.2 컴포넌트 메타 데이타 정의 및 분류

메타 데이타는 데이타가 가지는 속성과 구조, 연결성 등의 기본 사항을 포함하는 데이타의 데이타로 대상의 저장, 검색, 관리, 유지보수 등에 결정적인 역할을 수행함으로써 정보의 높은 활용을 가능케 한다.

실행성의 이진 형식의 컴포넌트는 현재의 문서 중심의 개요적 서술로는 비즈니스 솔루션 제공을 위한 프로세스에 적용할 수 없다. 컴포넌트의 메타정보는 그 이용자의 목적과 사용 범주에 따라 달라지므로 컴포넌트의 개발과 사용에서의 정확한 요구 수용을 위해 저장, 관리, 검색 등의 다양한 관점에서 메타 데이타로 구분하여 고려되어야 한다(그림 5).

메타 데이타는 식별 정보, 자료 품질 정보, 공간 자료 구성 정보, 공간 참조 정보, 실체 및 속성 정보, 배포 정보들을 포함하고 있어야 한다. 따라서 ISO/IEE 11179[6] 및 CBOP[7] 등의 권고 사항에 의해 컴포넌트 메타 데이타의 요건은 다음과 같이 정리된다.

본 논문에서는 위에서 정리한 메타 데이타 요건에 따라 컴포넌트 메타 정보를 <표 2>와 같이 정의, 분류하였다. 정의된 메타 데이타는 저장소의 물리적인 데이타베이스 스키마 설계를 위한 핵심 데이타가 될 뿐 아니라 저장소의 컴포넌트 형상 관리를 위한 기준 값으로 적용된다. 또한 이용자들에 의한 검색 및 이해 정보 획득을 위한 표준 포맷을 제공한다.

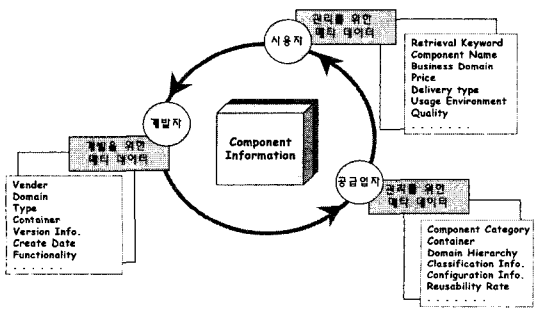

그림 5 컴포넌트 메타 데이타에 대한 관점들

· 정보의 요청과 등록, 접근과 사용을 위한 지원 데이타 정의
· 컴포넌트 관리 체계성 유지를 위한 아키텍처 구성
· 정보 이해 및 공유를 위한 표준 표기 형식으로 표현
· 컴포넌트 기능성 서술을 위한 세부 데이타 정의
· 컴포넌트 실행을 위한 환경적 제약 요소 제시
· 컴포넌트 관리 및 배포 정보 제시

표 2 컴포넌트 메타 데이타 분류(계속)

컴포넌트 프로덕트 메타 데이타	이름	관리 컴포넌트의 이름
	버젼	관리 컴포넌트의 버젼
	등록일	관리 컴포넌트의 등록일
	분류코드	인증 번호에 대응되는 분류 코드
	분류 계층	컴포넌트의 수직적, 수평적 계층
	배포 상태	관리 컴포넌트 배포 형태
저장소 메타 데이타	관리도메인 이름/수	저장소 관리 비즈니스 도메인 이름 리스트, 수
	컴포넌트 수	도메인 별 관리 컴포넌트 리스트
	DB 이름	사용하는 데이타베이스 이름
	저장소 구현환경	저장소구축 환경에 대한 서술
	접속 횟수	저장소 접속 횟수
	오류 회수/내력	저장소 운영 오류에 대한 서술/획수
이용자 메타 데이타	사용자 ID	시스템 관리용 이용자 ID
	사용자 이름	이용자 이름
	패스워드	이용자 패스워드
	전자우편	이용자 전자 우편 주소
	전화번호	이용자 전화 번호
	방문 횟수	이용자 방문 횟수
	구매 ID	시스템 관리용 이용자의 구매 ID
등록을 위한 메타 데이타	벤더명	제조 벤더 이름
	주소	벤더 소재지
	협력 업체	기술적 지원 업체
	전화번호	담당 부서 전화번호
	메일 주소	문의 어드레스
	URL	지원 페이지
	이름	컴포넌트 이름
	분류 번호	관리자에 의해 부여된 아키텍쳐 기반 분류 번호
	도메인	비즈니스 도메인
	버전 번호	컴포넌트의 버전 번호
	제공 방법	컴포넌트 제공 방법
	제공매체/장소	저장 매체,다운로드 사이트
	기능 소개	컴포넌트의 개요
	인증 번호	컴포넌트 등록 인증 번호
관리를 위한 메타 데이타	분류 정보	추상계층/비즈니스 계층
	이용률	컴포넌트 다운로드 횟수
	구매율	컴포넌트 판매 횟수
	버전갱신내역	컴포넌트의 버전 변경 내용

표 2 컴포넌트 메타 데이타 분류

범위	메타데이타	설명
관리를 위한 메타 데이타	품질평가기준	컴포넌트 평가 기준 항목
	품질평가결과	컴포넌트 평가 결과
	등록 연월일	컴포넌트 등록일자
	가격	컴포넌트의 가격
	소유/사용권	소유에 대한 법적인 권한
구성에 관한 메타 데이타	구성 컴포넌트 분류 번호	구성 컴포넌트 분류 번호
	구성 컴포넌트 이름	컴포넌트를 구성하는 컴포넌트 이름
	관련 설명본	컴포넌트간 관계,기능분담
	유스 케이스	컴포넌트 이용사례도
인터페이스/ 기능에 관한 메타 데이타	인터페이스	인터페이스 이름
	기능 설명	인터페이스가 제공한 기능
	패러미터	패러미터의 설명
	제약 조건	인터페이스 이용시의 사전/사후 조건
	기능명	기능의 이름
	기능 설명	기능에 관한 상세 설명
	이용 시나리오	이용 시나리오
	시스템 요구	프로그램,메모리 크기/ 실행시간
	임계값	실행시의 기능적 제약
환경적인 메타 데이타	플랫폼	실행용 OS 이름 및 버전
	미들웨어	동작을 위한 미들웨어/버전
	데이타베이스	사용 가능한 DB/버전
	개발 툴	개발 툴 이름
	기술 언어	컴포넌트를 기술한 언어
	참조 모델	개발/실행의 참조 모델

3.3 컴포넌트 저장 아키텍쳐

CBD에 의한 컴포넌트 조립 및 계층적 카탈로깅을 통한 컴포넌트의 저장, 분류, 검색의 가이드라인으로서 저장소는 컴포넌트 아키텍쳐에 기반하여 개발되어져야 한다. 컴포넌트 저장소 또한 컴포넌트 아키텍쳐를 바탕으로 정보 식별 및 공간적 구성 정보의 논리적 접근을 위한 저장 아키텍쳐가 요구된다.

본 논문에서는 이미 정의한 ABCD 컴포넌트 참조 아키텍쳐[8]를 바탕으로 이를 컴포넌트의 물리적인 저장을 논리적 관점의 메타 정보로서 확보하기 위해 두 가지 관점에서의 저장 아키텍쳐를 제시한다. 컴포넌트 기능성 영역의 수직적 관점과 라이프사이클 관점의 수평적 관점이 혼합된 아키텍쳐를 정의하고 이를 도식화하였다<표 3>(그림 6).

표 3 컴포넌트 저장 아키텍쳐

아키텍쳐 / 수직적 관점	물리적 / 수평적 관점
컴포넌트 아키텍쳐"참조 모델 기반	컴포넌트의 검색/유통의 결론적 관점
비즈니스 영역 규모에 따라 분류,저장 -의미적(기능적) 관점의 분류 -도메인에 따른 브라우징 -컴포넌트 조합의 기본 시나리오 -컴포넌트 메타 정보로 이용	컴포넌트의 생산의 abstraction과 realization 사이의 수준에 따라 구분 -분석/설계 정보의 모델링 파일 -컴포넌트 메타 정보 -인터페이스 위한 텍스트 문서 -원시코드/실행파일

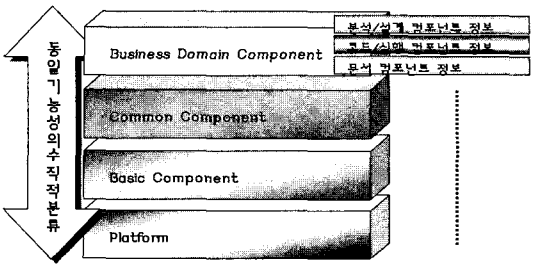

그림 6 컴포넌트 저장 아키텍쳐

또한 아키텍쳐에 따른 컴포넌트의 저장 분류 코드를 앞 절에서 분류한 메타 데이타를 바탕으로 4개 항목의 저장 메타데이타를 정의하였다(그림 7)<표 4>. 이를 통해 일관된 정규화 코드를 부여함으로써 상품화된 컴포넌트의 단일화된 명세화, 저장, 검색을 위한 컴포넌트의 생산, 등록, 검색 등에 대한 관점을 확정하고 유통 체제 관점에서 그 이용자의 목적과 사용 범주의 차별화를 하고자 하였다[9]. 따라서, (그림 7)의 공통(**C**) 비즈니스 관리(**BM**) 컴포넌트 인덱싱(**02**)의 분석정보(**01**) 코드는 (그림 8)에서 표시되는 저장소 아키텍쳐의 논리적/물리적 위치로서 "**CBM0201**" 분류 정보로 표현되는 곳에 저장된다.

표 4 컴포넌트 저장을 위한 메타 데이타

항목	설명
도메인	컴포넌트 적용되는 총체적인 비즈니스 영역
비즈니스계층	실질적인 기능성을 포함하는 업무를 식별
추상계층	분석 · 설계 · 구현에 따른 컴포넌트 저장형태
구성 컴포넌트	컴포넌트가 조립될 수 있는 비즈니스 프로세스

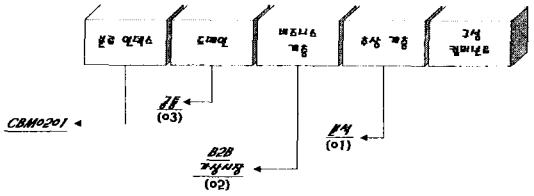

그림 7 컴포넌트 저장 메타 정보

3.4 컴포넌트 저장소 형상 관리 정보

컴포넌트 개발 작업은 무형의 산출물들을 생산하는 것이므로 개발과 변화 과정 및 유지보수에 대한 통제가

어렵다. 이러한 컴포넌트 개발 상의 올바른 정보 제공의 결핍은 재사용 가능한 빌딩 블록으로서의 기능성에 대한 신뢰성과 개발 시간과 비용의 중복으로 전체적인 효율성을 떨어뜨린다.

컴포넌트 형상관리는 전체 컴포넌트 개발 및 유지보수 과정에서 컴포넌트의 변화 사항을 적절하게 조절할 뿐 아니라 요구사항을 체계적으로 적용시키기 위한 것이다. 따라서 컴포넌트의 공용 뱅크로서 저장소가 제공해야만 하는 컴포넌트의 형상 관리 서비스들은 아래와 같이 정리될 수 있다[10].

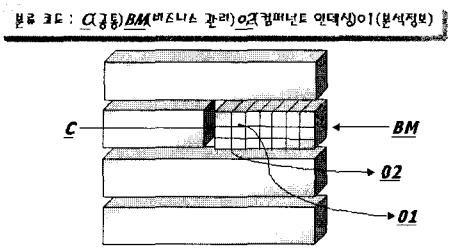

그림 8 분류코드 및 메타 데이타의 적용

- 컴포넌트 재사용율 및 판매율 측정
- 컴포넌트 다운로딩 횟수 측정
- 컴포넌트 사용자의 요구 분석 및 측정, 요구에 대한 응답
- 컴포넌트 버전 관리 : 컴포넌트 버전 변경 히스토리
- 전체 컴포넌트의 변경 사항 관리
  - -관리 카테고리 및 해당 카테고리에서 관리하는 컴포넌트의 수
  - -DB 트랜잭션에 따른 형상 정보 변경
- 저장소 변경 사항에 대한 히스토리
  - -관리 DB 교체에 따른 물리적 변화
  - -저장소 아키텍쳐 및 분류 원칙 변화에 따른 구조적인 변화
- 유통 벤더 및 파트너 변동 사항

본 논문에서 구축한 저장소 프로토타이핑 시스템에서는 컴포넌트의 메타 정보 관리를 포함한다. 특히, 컴포넌트 버전 관리를 위해 정보 패키지를 설정, 관리한다. 하나의 컴포넌트에 관련된 메타 정보를 포함한 모든 프로덕트들을 논리적 패키지 단위로 작성, 관리함으로써 저장소의 형상 관리를 위한 컴포넌트의 정보 단위로 이용한다. 컴포넌트의 분류 코드는 저장소에서 유일한 식별 정보로 간주되므로 컴포넌트 하나에는 하나의 버전이 존재한다. 따라서 새로운 버전이 부여된다면 새로운 컴포넌트의 등록으로 간주되어 컴포넌트에 관한 모든 정보들은 구별되어 개별적으로 관리된다. 동일 컴포넌트 이름을 가진 다른 버전의 독립적인 컴포넌트를 의미하는 것이며 단지 "패밀리 컴포넌트", "컴포넌트 카테고리" 등의 컴포넌트 정보 항목 등을 통해 연계되어질 수 있다. 이는 컴포넌트의 변경 정보를 원시 코드 중심으로 관리하고 확장한다면 소프트웨어 시스템의 대체 형상(Alternative Configuration)을 제공하기 위해 동일 컴포넌트에 대한 버전의 유지가 필요하지만, 본 논문에서는 컴포넌트 개발자와 다른 관점을 가질 수 있는 사용자가 실행성의 컴포넌트들을 이해하고 사용하는데 초점을 두기 때문이다. (그림 9)는 CRMS에서 정의한 컴포넌트 프로덕트의 정보 패키지이다.

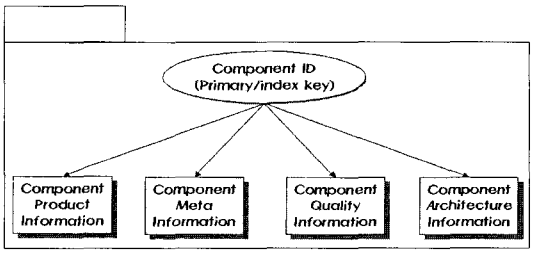

그림 9 컴포넌트 프로덕트 정보 패키지

3.5 컴포넌트의 검색

컴포넌트 검색은 컴포넌트의 특성을 사용자 측면에서 세분화함으로써 사용자의 요구를 정확히 전달하고, 컴포넌트 획득의 정확도를 높일 수 있다. 본 논문에서는 등록, 인터페이스와 기능, 환경 그리고 관리를 위한 메타 정보들을 통합, 세분화함으로써 컴포넌트 검색을 위한 메타 데이타로 정의하였다. 다음은 검색 메타 데이타 정의를 위해 고려한 사항이다.

표 5 컴포넌트 검색을 위한 메타 데이타

일반적 속성	등록 메타정보	Name	Component name
		Business Domain	Problem which can be applied
		Description	Brief Features of component
기능적 속성	인터페이스 메타정보	Interface	interface of component
		Method	user defined method
환경적 속성		Container	Reference architecture/ Language
	환경 메타정보	Family	Related Component name
		Platform	OS, Middleware
유통을 위한 속성	관리 메타정보	Delivery Status	Delivery method
		Creation date	last modification
		Price	cost
		Vender	developer or provider, Company

① 시스템 모듈로서 가지는 일반적인 형식적인 속성
② 실행성의 구현 객체로서 가지는 기능적 속성
③ 비즈니스 로직 요소로서 전체 시스템 관점의 환경적 요소
④ 컴포넌트 유통을 위한 요소

<표 5>은 본 논문에서 컴포넌트의 검색을 위해 정의한 메타 데이타들이다. 컴포넌트만의 기능적인 속성들을 위한 항목으로서 "method"는 개발자 정의 메소드와 구현 언어 종속적인 메소드를 참조하며 "interface"는 다른 컴포넌트와의 연계성을 표현한다. 이들 항목은 컴포넌트의 기능성과 적용 가능성의 쉬운 이해를 지원할 수 있다. 유통 속성으로 컴포넌트의 배포 형태와 작성일, 가격 및 생산 업체를 선정함으로써 최신 개발 컴포넌트를 확보하도록 했다. 일반적 속성으로 이름, 비즈니스 도메인, 간략한 기능적 서술을 설정하였으며 환경적 속성으로 컴포넌트 개발 및 이용의 컨테이너와 플랫폼, 관련 컴포넌트 군을 설정하였다. 많은 컴포넌트의 검색 항목을 설정함으로써 사용자의 수고스러움을 요구하고 있으나 사용자 지향의 인터페이스를 간소화시킴으로써 해결한다.

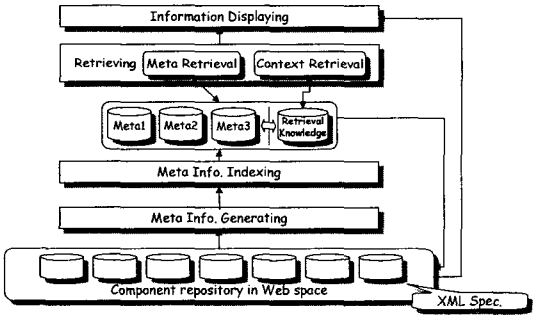

그림 10 컴포넌트 검색 프로세스

(그림 10)은 본 논문에서 제시하는 컴포넌트 검색 프로세스이다. 웹 상에 분산된 컴포넌트 저장소로부터 XML로 작성된 컴포넌트 명세를 수집하고 이로부터 정의된 메타 정보 형식에 따르는 정보를 추출, 생성한다. 다음, 생성된 메타 정보를 인덱싱하여 검색을 위한 정보 클러스터링을 수행하여 비즈니스 도메인별로 관련성이 높은 데이타들끼리 분류되어 메타 정보 저장소에 저장한다. 사용자 검색에 대해 적절한 결과 값들을 반환한다. 또한 "Context" 검색에서 올 수 있는 애매함과 불필요하게 발생할 수 있는 재현율에 정확도를 강화하기 위해 이전 검색의 히스토리를 "Retrieval Knowledge"에 기록하여 동일 질의에 대한 검색 결과를 재사용 함으로써 검색의 올바른 진행과 효율성을 높인다. 프로토타이핑[11]를 참조하였다.

메타 데이타는 컴포넌트 개발자가 새로운 컴포넌트 개발, 확장시 정해진 양식으로 메타데이타 레지스터에 등록하거나, 메타데이타 레지스터가 웹 상의 컴포넌트 저장소를 주기적으로 수집해온다. 수집된 메타 데이타는 메타 데이타 생성기에 의해 검색용 메타 데이타로 생성되고 카테고리 되어 저장된다. 메타데이타는 컴포넌트 정보의 업데이트에 대비, 메타데이타 레지스터가 이미 등록된 URL을 기준으로 주기적으로 체크하여 메타데이타 생성기에 의해 메타데이타를 재작성하며 변경사항이 비즈니스 도메인과 관계가 있을 시 재분류되어 검색을 위한 최상의 데이타를 보유하게 된다.

4. 컴포넌트 저장소 프로토타이핑 시스템

4.1 시스템 요구 분석

본 논문에서 개발하는 프로토타이핑 시스템인 CRMS는 컴포넌트의 공용 저장소이다. 즉, 컴포넌트의 각 생명 주기에 따른 각 결과물들과 이와 관련된 메타 정보들을 체계적으로 등록, 저장, 관리하고 사용자가 웹 상에서 적절한 컴포넌트를 검색, 이해하고 구매할 수 있는 기능을 제공한다. 그러므로 사용자는 컴포넌트를 공급하는 업체 및 등록된 컴포넌트를 구매하기 위한 어플리케이션 개발자가 된다.

CRMS의 기능성은 1) 컴포넌트 결과물과 관련 메타 정보의 관리와 2) 컴포넌트 정보 패키지의 검색, 3) 컴포넌트 프로덕트의 배포로 분류된다. <표 6>은 CRMS의 기능을 분류 범주 하에 정리한 것이다.

CRMS를 개발하기 위해 Windows NT 하의 MS-SQL을 컴포넌트 관리 데이타베이스로 사용한다. 그리고 클라이언트를 위한 인터페이스 시스템 및 관리 로직은 웹 서버 상의 ASP를 이용하여 구축하였다.

표 6 CRMS의 기능

기능	설명
보안	· 모든 사용자에게 접근은 허용하나 컴포넌트의 판매는 시스템으로부터 인증 받은 사람에 한함
관리	· 컴포넌트 정보 패키지의 등록/검색/삭제/갱신
형상관리	· 컴포넌트 버전 관리를 수행 · 저장소 관리 컴포넌트의 히스토리 감지, 보고
다운로딩 /업로딩	· 검색/이해된 사용 확정된 컴포넌트를 다운로딩 · 사용자가 작성한 컴포넌트의 업로드
검색	· 웹 상에서 등록된 컴포넌트를 사용자가 검색할 수 있도록 다양한 접근 방식을 제공 · 사용자가 선택한 컴포넌트에 대한 자세한 정보를 제공하여 컴포넌트에 대한 이해를 증가

4.2 시스템 분석/설계

CRMS는 웹 상에서의 컴포넌트 사용자를 위한 컴포넌트 검색, 이해, 구매 시스템과 컴포넌트 및 관련 메타 정보의 관리 및 형상 관리를 위한 컴포넌트 등록, 저장 시스템으로 구성된다. 그러므로 CMRS는 컴포넌트 프로덕트들의 물리적인 저장 매체인 데이타베이스를 중심으로 사용자 측면에서 검색 및 관련 메타 정보의 이해, 그리고 이해되어진 필요한 컴포넌트의 구매 및 획득을 위한 서비스와 서버의 관리 측면에서 컴포넌트의 등록, 삭제, 갱신과 버전 제어, 컴포넌트와 저장소의 히스토리 관리 및 보고 등의 형상 관리 서비스로 분류될 수 있다.

(그림 11)은 CRMS의 개략적인 시스템 개략도이다. 기능적 관점에서 저장소 프로세스를 패키지 단위로 표현하면 (그림 12)와 같다. 개별 패키지는 독립적인 서브 모듈로 개발, 전개 가능하다. 컴포넌트 사용 관리를 위한 부분과 저장소 관리를 위한 부분으로 구분되며 효과적인 컴포넌트 관리를 위해서는 컴포넌트 유지 관리와 형상 관리 기능과 컴포넌트 분류 및 카테고리 기능을 포함하고 있다. 또한 기술정보 관리 및 컴포넌트 검색과 이해의 부분에서 사용자 관점에서의 저장소 이용의 편의를 위한 기능을 지원한다.

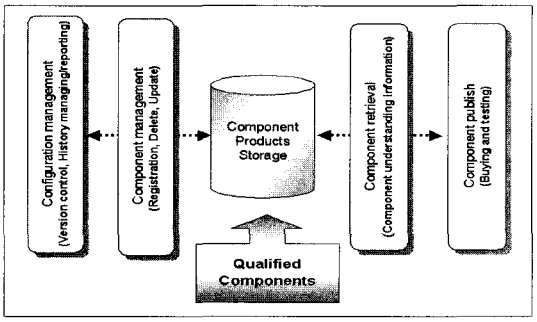

그림 11 CRMS 시스템 개략도

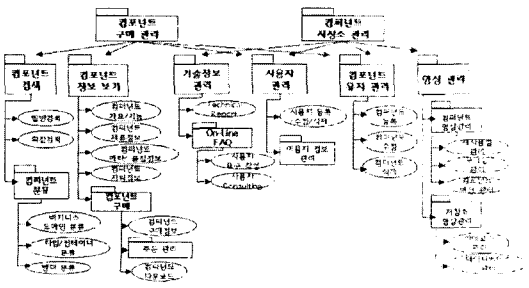

그림 12 CRMS 프로토타이핑 시스템의 개념도

(1) 분석 모델링

본 논문에서는 시스템을 시각화하고, 기술하고, 구축하며 산출물들을 문서화하기 위한 모델링 언어로서 UML (Unified Modeling Language)을 사용하였다.먼저, 시스템에 대한 이용자(클라이언트 사용자와 관리자)의 요구 사항을 파악하고 이를 바탕으로 Use Case와 Actor를 추출하여 Use Case Diagram과 각 경우의 시나리오를 중심으로 Use Case Model을 작성하고 각각의 이벤트 흐름에 대한 Activity Diagram과 Interaction Diagram을 작성하였다.

(2) 시스템 설계

작성된 분석 모델과 실제 구현될 아키텍처를 바탕으로 서브시스템들을 구분하고 이들 각각의 기능성을 클래스 다이어그램 작성을 통해 정의한다 그리고 서버의 데이타베이스 스키마를 작성하고 클라이언트 시스템을 위한 사용자 인터페이스를 설계, 구현한다. 재사용 지원을 위한 분류, 인덱스된 컴포넌트의 검색 서비스가 다양한 방식으로 제공되며 아울러 컴포넌트의 획득을 위한 이해 정보의 제공 서비스와 다운로딩 서비스가 지원된다. 형상 관리를 위해 컴포넌트의 버전 제어나 컴포넌트 변경에 대한 히스토리를 보고한다.

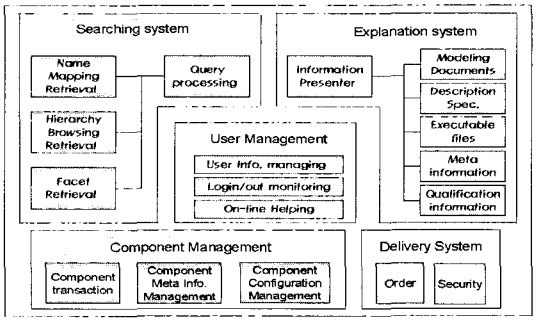

그림 13 기능적인 서브시스템

CRMS는 (그림 13)과 같이 크게 5개의 기능적 서브시스템으로 분할, 개발된다. 컴포넌트 저장소 관리는 아키텍쳐 모델을 기반으로 한 분류 계층에 따라 범주화되고 사용자는 컴포넌트의 구매를 위한 웹 프로그래밍의 구현 기술을 포함한다. (그림 14)는 CRMS개발 위해 작성된 패키지 다이어그램이다.

또한, 사용자가 자신의 프로젝트에 적용할 수 있는 최적의 컴포넌트를 선택할 수 있도록 컴포넌트에 대한 풍부한 정보 제공은 필수적이다. 따라서 CRMS에서 사용자에게 제공되는 컴포넌트의 이해 정보를 각 특징에 따라 범주화하면 <표 7>과 같다.

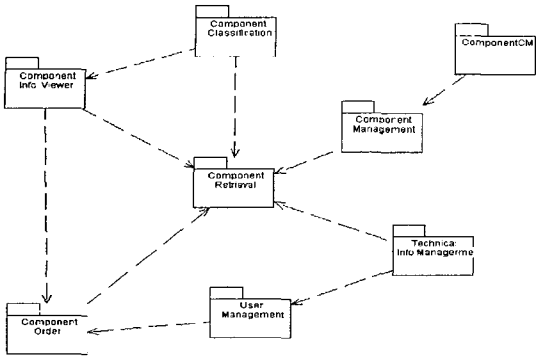

그림 14 CRMS의 패키지 다이어그램

표 7 CRMS에서 제공하는 컴포넌트 이해 정보

분류		항목 및 설명
Brief Info.		검색 경로에 의해 접근한 하나의 컴포넌트 선택시 자동적으로 제공되는 정보 벤더, 간단한 기능성, 컨테이너, 패밀리 컴포넌트
Detail Info.	Overview	개발 동기, 전체 기능성, 실행에 필요한 시스템 요구 사양, 이용 효과
	Products	Functionality Spec. Interface, Analysis/Design Mode, Screen Shots
	Demo	컴포넌트의 실행 가능한 데모 파일/예제 파일
	Buy & Support	구매를 위한 참조 정보 및 구입 시 지원 사항
	Quality	검증 시스템에 의해 평가된 결과
	Meta Inf.	형상 관리 정보 및 메타 정보를 제시 · 개발 회사, 개발자, 등록일, 버전 · 타입, 참조 모델, 이용 가능한 도구 · 이용율(구매성, 재사용성) · 도메인 분류 계층, 관련 컴포넌트들

4.3 시스템 실행

CRMS는 현재 운영되는 상업적인 컴포넌트 유통 사이트를 모델로 하였다[12][13]. 저장소의 모든 사용자 인터페이스는 영문으로 표시하였으며 컴포넌트 등록과 검색, 이에 따르는 정보 제공 및 사용자 관점의 서비스 지원을 위한 저장소 기능 구현에 초점을 두었다. (그림 18)은 CRMS의 메인 화면이다. 상단에 Java Script로 구현한 트리 메뉴는 CRMS의 주요 기능들을 분류한 메뉴로서 'Repository News'는 CBD의 전반적인 연구 및 유통 정보를 제공하며 "Component" 항목은 컴포넌트 브라우징과 검색을 제공한다. "Customer" 항목은 CRMS를 이용하는 사용자 정보 영역으로 컴포넌트 이용 기록과 개인 정보를 제공한다. "Admin."항목은 관리 컴포넌트 및 이와 관련된 저장소의 형상관리 정보를 제공한다.

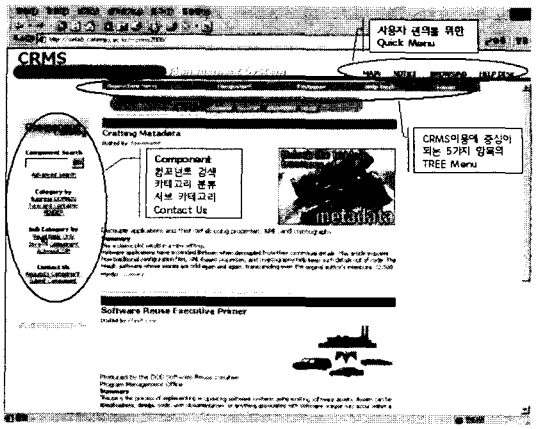

그림 15 CRMS의 메인 화면

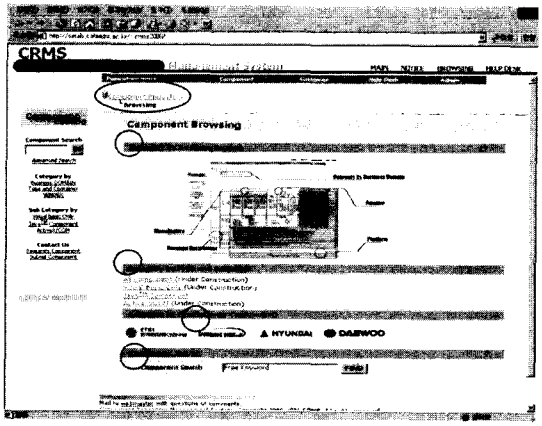

그림 16 컴포넌트 분류 계층에 따른 브라우징

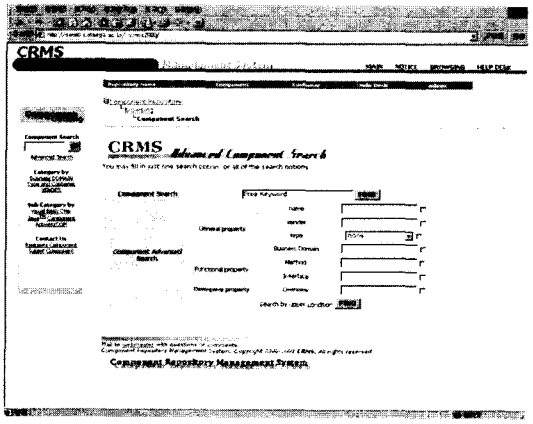

그림 17 메타 정보에 기반한 컴포넌트 검색

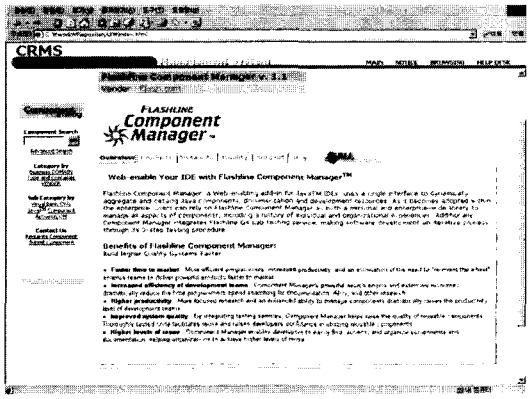

그림 18 컴포넌트의 이해 정보 제공

(그림 19)는 저장소 관리 컴포넌트의 브라우징 화면이다. 먼저, 참조하는 비즈니스 도메인에 따른 컴포넌트 분류 계층을 그래픽 이미지 맵을 이용, 직관적으로 컴포넌트 기능의 이해가 가능하도록 하였다. 다음으로, 컴포넌트의 타입과 컨테이너 그리고 컴포넌트 개발 벤더에 따른 브라우징 방법을 제공함으로써 사용자가 선호하는 특정 타입, 벤더에 의한 컴포넌트의 검색이 가능하도록 하였다. (그림 20)은 컴포넌트 검색용 메타 데이타에 기반한 CRMS의 컴포넌트 패싯 검색 화면이다. 사용자가 원하는 비즈니스 도메인의 컴포넌트부터 결과 내 검색을 사용, 반복적인 검색을 수행하며 사용자에 의한 검색을 위한 부가적인 옵션 즉, 필수 조건과 선택 조건 등을 결정하도록 하여 검색의 정확율과 재현율 사이의 상반관계를 조정한다. 검색된 컴포넌트에 대한 정보를 CRMS는 6개의 정보 범주로 구분하여 제공한다. "Overview" 항목은 컴포넌트의 전반적인 기능과 환경 등의 개요가 제시되고 "Products"에서는 컴포넌트의 제품군, 버전 정보, 데모 정보가 보여진다. "Meta Info." 항목은 컴포넌트의 이용률 및 컴포넌트의 메타 정보가 제시되고 "Buy & Support"에서는 컴포넌트 획득 및 이에 따른 지원사항들을 제공한다(그림 21).

4.4 시스템 비교 및 제약점

컴포넌트 저장소는 컴포넌트의 모든 라이프 사이클에 연관된 정보를 체계적으로 관리하고 사용자들에 의해 효율적으로 이용할 수 있는 라이브러리 시스템으로서의 역할이 기본이 된다. 하지만, 현재에 컴포넌트 저장소로 사용되는 대부분의 시스템들은 단지 컴포넌트의 상업적인 유통 매체로서의 역할에 가장 큰 목적을 둠으로써 아키텍쳐에 기반한 의미있는 컴포넌트의 분류 체계나 진보적인 컴포넌트 검색 편의를 제공하지 않을뿐더러, 저장소 관리 컴포넌트의 메타 및 형상 정보의 관리에 대한 접근은 거의 이루어지지 않고 있다. 다음은 컴포넌트 저장소 평가를 위한 고려 항목들을 정리, 제시한 것이다.

- CBD 프로세스 지원 단계
- 명확한 아키텍쳐 및 명세 지원 여부
- 관리 및 제공 컴포넌트 타입
- 관리 컴포넌트의 신뢰성
- 검색의 편리성
- 비실행성 도메인 정보의 지원 여부
- 기술적인 사용자 지원 서비스

<표 9>는 현재 가장 활발히 운영, 참조되고 있는 웹 상의 컴포넌트 유통 사이트 및 저장소들 간의 특징들을 비교한 것이다. 특히, CRMS는 컴포넌트 라이프 사이클 전반에 걸친 컴포넌트 프로덕트 정보를 관리하며 나아가 메타 정보 및 형상 관리 정보 역시 중요한 재사용 요소로 포함한다. 그리고 검색된 컴포넌트에 대한 상세한 정보를 제공함으로써 컴포넌트 사용을 위한 최종 결정을 내리기 전에 최상의 비즈니스 솔루션 컴포넌트를 획득할 수 있도록 하였다.

본 논문에서는 현재까지 표준적인 컴포넌트 저장소의 기능과 특성에 대한 정의가 아직 내려지지 않았지만, 기존 컴포넌트 저장소 및 상업적 유통 시스템을 조사, 분석하였다. 그리고 시스템의 평가를 위해 현재 활용되고 있는 웹 상에서의 컴포넌트 유통 사이트[15, 16]를 조사, 비교함으로써 CRMS의 특징을 제시하였다. 이는 본 논문에서 정의한 메타 정보에 따른 컴포넌트 전체 라이프 사이클의 프로덕트를 갖는 컴포넌트의 확보가 어려웠으며, 그로 인해 컴포넌트 획득의 성능을 평가할 수 있는 정확도와 재현율의 평가를 제대로 할 수 없었다. 이에 대한 보완으로 보다 진보적인 검색 편의를 제공하기 위해 컴포넌트 메타 데이타에 기반한 검색용 메타 데이타를 정의, 시스템에 적용하고 사용자의 선택에 따라 이용하도록 함으로써 컴포넌트 검색의 정확성을 이용자의 주관에 따라 결정하도록 하였다.

표 9 기존 컴포넌트 저장소들의 특징 비교

연동 저장소	주요 기능	형상 관리	아키텍처/명세 지원	컴포넌트 타입	검색 방법	비실행 속성
FlashLine	검색/판매/평가	형상 관리 도구 지원	Not Define	Execution File	브라우징 키워드매핑	NO
Component Source	브라우징/판매	NO	Not Define (분류체계)	Execution File	브라우징 이름매핑	NO
Java Repository	브라우징/판매	NO	EJB	Execution File	브라우징 이름매핑	NO
CRMS	검색/이해/사용자피드백	재사용성/버전 관리	공용 아키텍처	All Component Products (분석/설계/실행)	브라우징 키워드매핑	YES

5. 결 론

소프트웨어 개발과 운영의 새로운 패러다임으로 현재 컴포넌트 기반의 소프트웨어 개발은 소프트웨어 개발을 위한 최상의 경쟁 전략으로 빠르게 확산되고 있다. CBD를 위해서는 컴포넌트 저장소를 중심으로 컴포넌트의 추출과 생산, 배포 및 관리 등에 대한 일련의 Reuse For Component를 위한 프로세스와 생산되어진 컴포넌트의 검색, 이해, 구매 및 새로운 어플리케이션으로의 적용 등의 Reuse With Component를 설명할 수 있는 통합된 접근 방식의 정립 및 이에 대한 실질적인 적용 기술이 필수적으로 요구된다.

따라서 본 논문에서는 CBD 프로세스의 효과적인 지원을 위해 컴포넌트 저장소 구축에 필요한 이론적 접근 기술을 제시하고 이를 실제적인 저장소 프로토타이핑 도구의 개발에 적용하였다. 컴포넌트 저장소 관리 시스템(CRMS : Component Repository Management System)은 UML 기반의 CBD 방법에 의해 분석, 설계, 구축된 클라이언트/서버 시스템으로 컴포넌트 생산에 관련된 모든 결과물 및 이와 관련된 메타 정보들을 형상관리의 측면에서 관리하는 서버 시스템과 웹 상에서 사용자가 편리하고 효과적으로 컴포넌트를 검색, 이해, 획득 등에 관한 총체적인 유통 체제를 담당하는 클라이언트 시스템으로 구성된다.

본 논문은 CBD 프로세스의 실제화를 위한 접근 체계로 저장소를 구축함으로써 컴포넌트 생산과 관리 그리고 유통을 위한 실질적인 공용 컴포넌트 저장에 대한 선행 연구로서 이용할 수 있을 것이다.

향후, 본 논문에서 구축한 저장소 프로토타이핑 시스템을 경제성이 포함된 컴포넌트 유통 시스템으로 확장, 개발하고 컴포넌트의 보다 진보적인 검색 효율 및 컴포넌트의 품질에 대한 신뢰성 보장에 대한 접근을 연구할 것이다.

참 고 문 헌

[1] Mikio Aoyama, "New Age of SoftwareDevelopment : New CBS Changes the Way of Software Development", 1998 International Workshop on CBSE, ICSE, p.124~128, 1998. http://www.sei.cmu.edu/cbs/icse98/papers/p14.html

[2] Peter Herzum, Oliver Sims, Business Component Factory : A Comprehensive Overview of CBD for the Enterprise, OMG press, December, 1999

[3] Robert C. Seacord, "Software Engineering Component Repository", Proceedings of 1999 Workshop on CBSE, Los Angeles, at URL, http://wwww.sei.cmu.edu/cbs/icse99/cbsewkshp.html

[9] 김행곤 외, "CBD 기반의 컴포넌트 저장소 프로토타이핑 시스템", 한국정보처리학회 추계 학술발표회, Vol 7. No 2. 2000년 10월

[4] Select Component Manager, http://www.aonix.com/

[5] MS Repository 2.0, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/reposit/html

[6] Metadata Resources, http://www.mnhs.org/preserve/records/meta.html

[7] CBOP, 메타데이타 구조 표준 V.01, CBOP 리파지토리메타 데이타 구조 정의서, 1999

[8] 김행곤 외, 컴포넌트 저장소 형상관리 시스템 최종 보고서, 한국전자통신연구원, 2000

[10] Dart, S., "Concepts in Configuration Management Systems," Proceedings of Third International Conference on SCM, Trondheim, Norway, June 12-14, 1991, pp. 18.

[11] http://www.htdig.org/

[12] ComponentSpurce, http://www.Component Spurce.com)

[13] FlasjLine URL at www.FlasjLine.com

차 정 은

1995년 효성여대 전자계산학과 이학사. 1997년 대구효성가톨릭대학교 전자계산학전공 이학석사. 2001년 대구효성가톨릭대학교 전자계산학전공 이학박사. 2001년 ~ 현재 ETRI 컴퓨터소프트웨어기술연구소 S/W공학부 선임연구원. 관심분야는 컴포넌트기반 개발, 재사용, 아키텍쳐, 설계 패턴, 재공학

김 행 곤

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 한국전기통신공사 전임연구원. 1988년-1989년 AT&T 객원 연구원. 1990년 ~ 현재 대구가톨릭대학교 컴퓨터공학과 교수. 2001년 Central Michigan University 교환교수. 관심분야는 CBSE, 소프트웨어 재공학, CASE, 도메인 공학, 요구공학