

영역별 분류체계에 기반 한 EJB 컴포넌트 관리 시스템

염연희⁰ 서동수
성신여자대학교 컴퓨터공학과
{yhyeom, dseo}@cs.sungshin.ac.kr

EJB component management systems based on the domain classification schema

Yeonhee Yeom⁰ Dongsu Seo
Dept. of Computer Science, Sungshin Women's University

요 약

컴포넌트 기반 개발 방법이 보급됨에 따라 개발자들은 사용 가능한 컴포넌트들로부터 필요한 컴포넌트를 선별하는데 많은 노력과 시간을 투자한다. 컴포넌트 집합을 보다 효율적으로 관리하기 위해서는 영역별로 분류하는 작업과 컴포넌트를 등록, 관리, 검색하는 작업이 필요하다. 본 논문에서는 영역별 분류를 기반으로 EJB 컴포넌트를 명세하며, 이를 등록, 검색, 관리할 수 있는 컴포넌트 관리 시스템을 구현하였다. 컴포넌트 명세는 현재 구현되어진 EJB 컴포넌트를 수용할 수 있도록 UML 기반의 명세방식으로 사용자가 필요로 하는 정보를 제공할 수 있도록 정의하였다.

1. 서론

컴포넌트의 기반의 개발 방법이 소프트웨어 위기를 극복할 만한 소프트웨어의 생산성을 높일 수 있는 방법으로 주목되면서 컴포넌트 관련 연구들이 활발히 진행되고 있다. [1] 생산성을 높이게 되는 컴포넌트의 가장 큰 특징은 바로 재사용성이라 하겠다. 이러한 재사용을 가능케 하기 위해서는 무엇보다도 컴포넌트 자체가 유지보수성과 신뢰성을 보장해야 함은 물론 표준화된 인터페이스와 명세가 컴포넌트를 개발, 관리, 사용가능하도록 지원되어야 한다. 컴포넌트 기반 개발은 컴포넌트 생산, 선택, 평가 및 통합으로 구성되는 소프트웨어 개발방법의 가장 새로운 패러다임이다. CBD(Component Based Development)는 명세, 구축 및 조립지향적이며 컴포넌트의 오퍼레이션과 상호 오퍼레이션을 정의 하는 명세의 개발, 객체나 컴포넌트들로부터 컴포넌트의 구축, 컴포넌트를 이용해 애플리케이션을 조립하는 등의 활동이라 하겠다. [2]

개발 패러다임의 변환으로 개발자들의 업무가 자신이 프로그램의 전부를 다 개발하는 것이 아니라 이미 개발된 컴포넌트를 조합하거나 개발 환경에 맞게 수정하는 형태로 개발 방법 자체가 변경됨에 따라 이를 위해 재사용성을 보장하는 컴포넌트들을 관리하여 개발자들에게 그에 대한 정보를 제공하는 일은 중요한 연구가 될 수 있겠다. 많은 컴포넌트들에 대해 개발자가 필요한 정보를 찾는 일은 쉬운 일이 아닐 것이다. 본 논문에서는 특히 컴포넌트 개발자들 사이에 많이 쓰이고 있는 EJB 컴포넌트를 대상으로 컴포넌트 등록 및 검색을 위한 관리 시스템 및 저장소를 구현한다.

본 논문에서 정의하는 컴포넌트 명세를 단위로 하여 컴포넌트 저장소를 설계하였으며 분류체계를 한눈에 볼 수 있도록 계층 검색과 단어를 이용한 검색, 패킷에 의한 검색을 제공한다.

2. 관련연구

2.1 소프트웨어 객체 분류체계

본 분류체계는 객체가 갖는 주제 혹은 개발된 용도에 따라 나열될 수 있는 모든 영역에 대한 고유한 분류번호를 부여한 분류번호 집단이다. 따라서 분류번호에 객체의 응용영역적 특성과 운영특성을 반영하여 표기하는 방법을 취하였다. 분류대상은 OMG의 CORBA 표준을 지원하는 소프트웨어, 마이크로소프트사의 COM, OCX, ActiveX 를 지원하는 소프트웨어, 썬 마이크로 시스템즈 사의 JVM에서 작동되는 소프트웨어, JavaBeans, EJB 기반의 소프트웨어와 이외의 제3자가 개발한 재사용 가능한 객체를 그 대상으로 포함한다. 분류영역은 <표1>과 같이 구분한다.

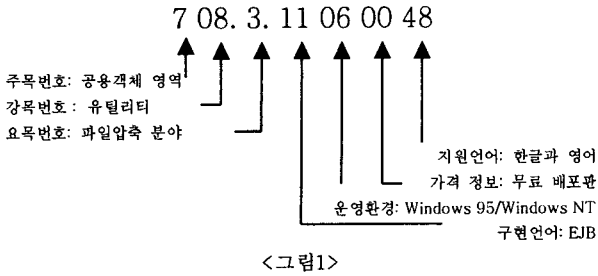
분류번호는 객체가 활용되는 응용 영역과 소프트웨어로서 가지는 구현 및 운영특성의 정보를 포함한다.

응용영역을 단순히 영역을 나열하는 개념이 아니라 어떤 요소에 대한 상위, 차상위 개념이 모두 나타나도록 하는 3 단계의 계층 구조를 가진다 [3]. 가장 상위개념을 주류(class), 중간 개념을 강목(division), 최하위 개념을 요목(section)이라 칭한다. 구분 정의의 표현을 빌어 표기하면 다음과 같으며 <그림 1>은 이에 대한 예시이다.

객체 분류번호 := {응용영역}.+(구현 및 운영환경)

응용영역 := {주류}+{강목}+.+(요목)

구현 및 운영특성 := {구현언어}+{운영체제}+{가격}+{지원언어}



<그림1>

영역	설명
100 객체 일반영역	다른 영역에 정의되지 못했거나 영역으로의 분류가 아직 이루어지지 않은 객체를 위한 영역
200 시스템 객체영역	각종 유형별 운영체제, 통신관리, 시스템관리, 보안, 통신망관리, 트랜잭션 관리 등의 서비스를 제공하는 객체 영역
300 개발용 객체영역	프로그래밍과 테스트를 위한 지원 소프트웨어와 프로그래밍 도구 개발시 사용되는 객체를 위한 영역
400 응용 객체영역	회계, 금융, 전자상거래, 공학, 과학 등 다양한 응용영역을 대상으로 하고 있다.
500 멀티미디어 객체영역	멀티미디어 콘텐츠 개발, 멀티미디어 기술이 강조되는 분야 지원 객체와 멀티미디어 자체 처리에 관한 객체 영역
700 공용 객체영역	응용소프트웨어가 제공하는 기능 중 공통적으로 쓰이는 기능 담당객체로 사용자 인터페이스, 정보관리, 태스크관리 등의 객체영역
800 데이터베이스 객체영역	데이터베이스의 접속관련 소프트웨어 및 파일 처리관련 객체 영역
900 인터넷, 인트라넷 영역	인터넷 인트라넷을 지원 하기위해 개발된 객체 영역

<표 1>

2.2 컴포넌트 명세

EJB 컴포넌트는 JavaBeans 기술을 바탕으로 애플리케이션 서버의 기술을 결합한 컴포넌트 프레임워크이다[4]. EJB 컴포넌트는 홈 인터페이스와 리모트 인터페이스를 통해 내부 Beans의 접근을 통제 함으로서 컴포넌트 서비스를 제공한다. 따라서 내부 Beans의 적절한 기능을 이해하기 위해서는 Beans 자체의 명세 이외에 이들간의 동적인 관계까지도 이해할 필요가 있다. 따라서 컴포넌트 명세는 계사유저에게 정보를 제공하기 위해서는 효율적이고 정확한 정보를 담고 있어야 한다.

컴포넌트의 명세 방법으로는 텍스트기반의 자연어를 이용한 명세, 정형명세[5], IDL 명세[6], UML을 이용 도식화한 명세[7] 등이 있다. 대다수를 차지하고 있는 텍스트 기반의 명세는 이해하기는 다소 쉬울 수 있으나 정확한 의미론이 부족하다. 이에 반해 정형명세는 이해와 명세가 어렵지만 분석모델의 정확성과 일관성을 알 수 있다. IDL 명세는 컴포넌트의 인터페이스, 오퍼레이션, 파라미터에 대한 상세한 내용을 명세할 수 있으나 컴포넌트의 의미 해석이 어렵다.

본 논문은 컴포넌트가 갖는 정적정보와 동적정보를 클래스 다이어그램과, Use Case 다이어그램, 컴포넌트 다이어그램을 사용하여 명세하며 또한 신뢰성을 유지하기 위해 정형명세기법을 오퍼레이션 명세에 일부 도입하는 명세 방법을 제시한다.

3.CMS(Component Management System)

CMS는 컴포넌트를 분류체계에 맞게 등록하여 검색할 수 있도록 설계한 컴포넌트 저장소이다.

분류체계관리와 컴포넌트 관리, 사용자 관리 등이 이루어진다. 분류체계관리는 현재 분류되어져 있지 않지만 따로 분류하여 정의 내릴 수 있는 영역이 발견되어지면 추가, 변경한다. 또한 사용자들의 요구사항을 수집하여 반영한다. 컴포넌트 관리는 등록된 컴포넌트를 관리해야 하는 것과 사용자들에게 등록된 컴포넌트를 제공해야 하는 두 가지 측면이 있다. 사용자 관리는 사용자 등록 절차를 두어 컴포넌트 등록자와 컴포넌트 검색을 위한 사용자 부류로 나누어 접근 권한을 적용할 수 있게 한다.

3.1 CMS 컴포넌트 명세

컴포넌트에 관한 정보를 서술하는 형식은 개발자에게 제공되어져 하는 가장 중요한 부분이다. 따라서 어떤 정보를 개발자들에게 제공하는가 하는 것이 중요한 문제가 된다. 컴포넌트의 정보로는 크게 비기능적인 정보와 기능적인 정보 두 가지로 나눌 수 있다. 비기능적인 정보 명세는 <표 2>와 같다. 보다 중요한 정보인 기능적인 컴포넌트 명세는 상세하고 체계적인 서술을 해주어야 한다. 이를 위해 UML을 이용 상세한 기능성과 컴포넌트간의 상호 관계를 서술해 줄 수 있는 명세방법이 필요하다. 본 논문에서 제안하는 명세 양식은 <표 3>과 같다.

항목	내용
분류코드	분류체계에 기반한 표기
분류영역명칭	패킷검색에 사용되기 위해 설정함
컴포넌트명	컴포넌트 이름
구현언어	컴포넌트 개발시 사용된 프로그램언어
지원언어	Application에서 지원하는 언어
동작 플랫폼	동작 운영환경 - 운영체제
배포방법	라이선스 형태(가격정보)
배포업체	제공업체에 대한 정보와 개발자정보

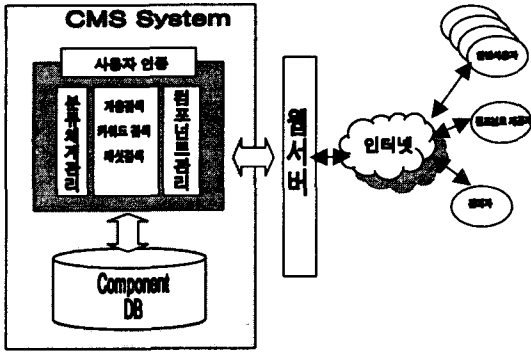
<표 2 컴포넌트의 비기능적 요소 명세>

항목	내용
기능서술	클래스 다이어그램 및 Use Case 다이어그램을 제공하여 컴포넌트의 상호동작과정을 시각적으로 표현 전체적인 시나리오를 제공한다.
Interface 간접명세	UML의 컴포넌트 다이어그램으로 컴포넌트간의 인터페이스를 간접 명세함. pre/post condition 입출력 결과 명시
Interface 직접명세	인터페이스 함수들에 대해 구현한 내용에 대해 접근할 수 있도록 public, private 함수 등의 내용을 표시함
Example	컴포넌트를 사용하는 방법에 대한 예시제공

<표3 컴포넌트의 기능적 요소 명세>

3.2 CMS구성

CMS는 <그림 2>와 같이 사용자에 대한 권한을 인증하는 사용자 인증처리 부분과, 분류 항목을 관리하는 분류체계 관리, 컴포넌트를 명세에 따라 등록, 변경, 삭제하는 컴포



<그림 2>

먼트 관리, 분류체계를 통한 계층검색, 검색어를 설정하여 단어 검색, 컴포넌트의 간단한 패킷을 통한 검색으로 이루어졌다. 구현 환경으로는 윈도우 NT 4.0 상의 MS SQL7.0, IIS 3.0 웹 서버를 사용하였다.

3.2.1 사용자 인증

사용자 인증은 회원 관리 형식으로 이루어지고 컴포넌트 제공자는 제공한 컴포넌트에 대해 추가, 변경, 삭제에 대한 권한이 주어진다. 일반 사용자들의 권한은 컴포넌트 관리 즉 추가, 삭제, 변경에 대한 권한이 없고, 검색하는 권한만 주어진다. 관리자는 모든 권한을 갖는다.

3.2.2 분류체계관리

관련논문에서 서술한 분류체계를 그대로 시스템에 반영하였다. 체계를 표현하는데 있어 Tree 객체를 사용하여 표현하였다. 체계 내에서의 분류 영역의 변화를 관리하는 역할을 담당하는 부분이다. 분류체계에 관한 사용자의 요구를 수렴하여 빠르게 반영할 수 있게 한다.

- 추가
 - 객체 사이에서 유사성이 많아져 독립적으로 영역 할당이 가능할 만큼 집단이 생길 경우 영역을 확장관리
 - 영역 구분의 범위가 다른 영역에 비하여 확장되어 있을 경우 재정의 하여 분할
- 수정, 삭제
 - 영역 범위가 협소할 경우 유사성 있는 영역과 통합

3.2.3 컴포넌트 검색기

컴포넌트 검색은 본 시스템에서는 3가지 방법으로 제공된다. 계층 검색, 키워드를 통한 검색, 컴포넌트의 특정 정보를 통한 패킷 검색이다.

계층 검색은 분류체계의 계층 구조를 사용자에게 표시해 주고, 사용자가 그 계층 구조를 네비게이션하면서 자신이 원하는 컴포넌트를 찾는 형식으로 구성되어 있다. 패킷 검색은 컴포넌트 명세요소 중 1차원적으로 분류하기에 적합한 요소로 분류 영역명, 구현언어, 지원언어, 동작 플랫폼(OS), 배포방법으로 구성된다.

3.2.4 컴포넌트 관리

컴포넌트 관리기는 컴포넌트DB를 관리한다. 컴포넌트 등록, 수정, 삭제하는 기능으로 구성된다.

- 등록 : 컴포넌트 명세 요소를 입력받아 데이터베이스에 저장한다. 등록시 분류코드를 할당 받게 되는데 이로 인해 입력의 중복을 막는다. 코드 할당은 주류번호를 먼저 선정하고 그 후 주류항목내의 강목을 설정, 강목내에서 요목과 구현언어, 운영체제, 가격, 지원언어 등의 정보에 의해 번호를 할당하게 된다. 컴포넌트 제공자에 의해 부여된 분류번호는 임시적인 것으로 최종적으로는 관리자에 의해 부여 받게 된다. 분류체계에 대한 이해가 다를 수 있기 때문에 일관성 있는 시각으로 분류되어야 하므로 관리자에 의한 분류코드를 할당 받도록 하는 방법을 취했다.
- 수정 삭제: 컴포넌트를 제공한 등록자와 시스템관리자만이 수정권한을 갖게 되어 명세내역을 수정할 수 있게 된다.

4. 결론 및 향후 연구과제

개발 방법의 전환에 의해 프로그래머들은 개발된 제품들 주 자신에 요구에 맞는 컴포넌트를 찾는데 많은 시간을 보내게 된다. 본 논문에서는 이러한 개발 환경에 맞춘 사용자에게 보다 올바르고 충분한 정보를 제공하기 위해 구성하였다. 컴포넌트들을 영역별로 분류하여 구성하였으며, 사용자가 필요로 하는 정보들로 명세를 구성하여 검색하게 하는 시스템을 구성하였다. 웹상에서 서비스를 지원하여 이미 개발되어진, 앞으로 개발될 컴포넌트들의 재사용을 증진시키며 컴포넌트 기반 개발 업무에 사용될 것이다.

향후 연구로는 컴포넌트의 재사용성에 관한 검증과, 버전 관리 기법을 하는 부분을 보다 체계적으로 정립하여 저장소에 있는 컴포넌트에 관한 신뢰도를 높일 수 있는 컴포넌트 형상관리 시스템에 관한 내용이 포함된다. 또한 컴포넌트에 관한 명세에서 사용자들이 요구하는 정보를 합리적으로 표현하는 연구가 되어집과 동시에 그에 대한 표준화작업이 이루어져 재사용율을 증진시켜 나아가야 하겠다.

5. 참고 문헌

[1] C. Szyperski, "Component Software: Beyond Object - Oriented Programming", Addison-Wesley, 1998
 [2] S.D.Hendrick, "Components, Objects, and Development Environment:Specification-Based Approach to Component -Based Development", Bulletin, IDC, April . 1998
 [3] 소프트웨어 객체 분류 표준, 한국전자통신연구원, 1999
 [4] T. Valesky, Enterprise JavaBeans, Addison Wesley, 1999
 [5] Ian Sommerville, "Software Engineering : Fifth Edition", Addison Wesley, 1996
 [6] '99 ICSE CBSE workshop <http://www.sei.cmu.edu/cbs/icse99/strawman.html>
 [7] OMG Unified Modeling Language Specification, Ver. 1.3. June.1999