아키텍쳐 기반의 컴퍼넌트 명세 정의

김 행 곤, 차 정 은, 김 병 준
대구효성가톨릭대학교 컴퓨터공학과 소프트웨어공학연구실
e-mail : {hangkon, jecha, g00521002}@cuth.cataegu.ac.kr

Definition of Component Specification Based on Component Architecture

Haeng-Kon Kim, Jung-Eun Cha, Byung-Jun Kim
Dept. of Computer Engineering, Catholic University of Taegu Hyosung

요약

최근 활발히 연구되어 오고 있는 컴퍼넌트 기반의 소프트웨어 개발 방법론은 컴퍼넌트의 잘 정의된 인터페이스를 통해 응용 시스템을 개발함으로써 개발의 생산성과 유지보수성 그리고 신뢰성을 보장한다. 그러므로 인터페이스 명세 정의는 컴퍼넌트 기반의 소프트웨어 개발을 위한 필수적인 선행 과제이다. 컴퍼넌트 기반의 응용 시스템 생성 프로세스는 컴퍼넌트 아키텍쳐 정의와 이를 바탕으로 한 컴퍼넌트의 명세화, 그리고 컴퍼넌트 프레임워크로 적용의 단계를 통해 달성할 수 있다.

따라서 본 논문에서는 표준 하부 구조를 정의하고, 소프트웨어 전개 모델을 제공함으로써 컴퍼넌트 생성과 사용, 평가를 위한 근거를 확보할 수 있는 컴퍼넌트 아키텍쳐 모델 즉, ABCD(Architecture-Base Component- Common Component-Domain Component) **아키텍쳐**를 제안한다. 또한, 이를 바탕으로 컴퍼넌트의 개발과 이용을 위하여 기존의 컴포넌트 명세가 가지는 비효율성을 극복하고 새로운 컴퍼넌트 명세 표기법을 제시하고자 한다.

1. 서론

컴퍼넌트 기반의 소프트웨어 개발(CBD : Component Based Software Development)은 소프트웨어 개발의 최상의 경쟁적인 전략으로서 빠르게 확산되고 있다. 잘 정의된 인터페이스에 기반한 컴포넌트들의 커스터마이징과 조립에 의해 응용 개발의 생산성과 유지보수성을 보장하며, 발전된 인터넷 활용 기술의 지원을 통해 컴퍼넌트 보급과 획득의 대중화는 기반 시스템 소프트웨어에서 미들웨어, 그리고 도메인 응용에 이르기까지 보증되어진 체계로서 적용되고 있다. 이미 컴퍼넌트 시장이 형성되어 컴퍼넌트의 개발과 유통, 그리고 획득과 시스템으로의 통합을 위한 컴퍼넌트 비즈니스가 운영되고 있는 실정이다[1]. 하지만 현재까지도 컴퍼넌트 분류와 개발, 커스터마이징을 위한 명확하고 정규화된 인터페이스 명세가 표준화되어있지 않은 실정이다. 따라서 컴퍼넌트 기반의 소프트웨어 개발 환경 구축을 위해서는 반드시 표준화된 인터페이스 명세가 뒤따라야 할 것이다.

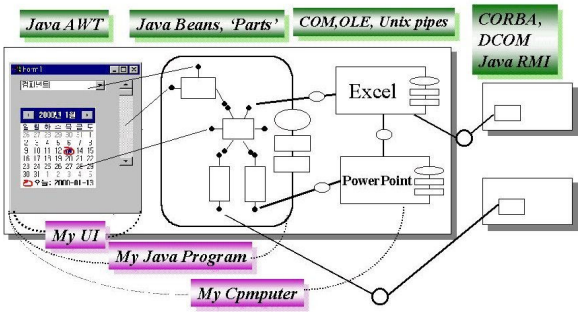
본 논문에서는 표준 하부 구조를 정의하고 소프트웨어 전개 모델을 제공함으로써 컴퍼넌트 생성과 사용, 평가를 위한 근거를 확보하기 위해 **ABCD 컴퍼넌트 아키텍쳐 모델**을 정의한다. 또한 이를 바탕으로 컴퍼넌트의 기능 및 서비스 구현시의 행위적인 정규 서술과 컴퍼넌트 조립을 위한 일반적인 메소드를 제공함으로써 컴퍼넌트를 개발하고 CBD로의 전개에 적용함에 충분한 컴퍼넌트 명세 방법을 연구, 제시한다.

2. 관련 연구

2.1 CBD의 개념과 연구 현황

CBD는 이미 생산, 배급되어진 각 부품들, 즉 컴퍼넌트들을 비즈니스 요구를 실현시키기 위해 소프트웨어 프로덕트의 군(群)으로 구축하는 것이다. 컴

韓國領 독도 Dokdo is Korean Territory



(그림 1) 다양한 형태의 구현 컴퍼넌트

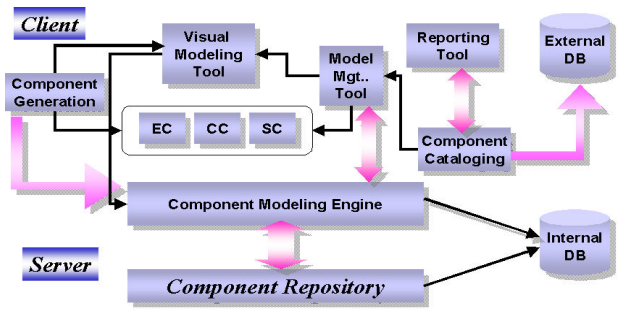
퍼넌트는 어떤 상황에서든지, 시스템의 요구 상황에 적절하게, 다양한 의미로 플러깅 되어질 수 있는 블랙박스화된 실행성의 시스템 모듈이다. 따라서 CBD에서 컴퍼넌트는 특정 비즈니스 도메인에서 목적 달성을 위해서 조합을 통해 응용의 일부로서 혹은 더 큰 의미의 또 다른 컴퍼넌트로서 서비스된다. (그림 1)은 다양한 스케일의 다양한 형태로 다양한 의미 조합, 구현되는 컴퍼넌트를 도식화 한 것이다[2].

현재는 공인된 특정 영역의 소프트웨어 컴퍼넌트 라이브러리 구축을 중심으로 컴퍼넌트 등록과 유지 보수, 탐색, 선택, 검색 및 조립을 위한 기술과 도구 개발이 실질적인 사업 이슈로 전개되고 있다. (그림 2)는 CBD를 지원하기 위한 도구 들의 상관 관계로 EC(Element Component), CC(Common Component) 그리고 SC(Scenario Component) 들의 생성과 조립, 관리를 위한 도구의 지원을 나타낸다.

2.2 컴퍼넌트 아키텍처

컴퍼넌트 아키텍처는 컴퍼넌트들의 독립성과 상호 협력을 위한 기초로서 필수적으로 컴퍼넌트와 그들의 동작 환경 사이의 상호작용을 정규화하고 인터페이스의 표준화를 규정한다.

현재의 기존 아키텍처 모델은 분산 컴퓨팅을 위한 다중 솔루션의 통합에 초점을 두고 정의되였다. 대중적인 아키텍처로 OMG의 OMA(CORBA)와 IBM의 Sanfrancisco, Sun의 EJB가 있다



(그림 2) CBD를 위한 지원 도구의 구성

<표 1> 컴퍼넌트 인터페이스 서술 방법

기법	설명
Contract	·Server_Broker_Vender 간의 약속 명시 ①기본약정 : OOP, IDL로 Operation, I/O 인자,예외 명시 ②행위약정 : 선/후/불변조건의 assertion으로 행위 검증 ③동기화약정 : 메소드 사이의 동기화 고려한 행위 기술 – 컴퍼넌트 사이의 순차성, 병행성, 혼합성 관계 서술 ④서비스 품질 약정 : 행위들의 정량화
Modeling	·UML 기반의 컴퍼넌트 정의 – 기존 모델링 기술에 스테레오 타입, 인터페이스 클래스, 실현 관계를 추가하여 표현 – 패키지, 클래스, 시퀀스, 컴퍼넌트 다이어그램
Formal Spec.	·아키텍처 정의 언어, 프레임워크 정의 언어로 작성 – Wright, COM, CORBA, Z 언어 – 시스템 통합의 불일치 제거, 명세 검증

2.3 컴퍼넌트 명세

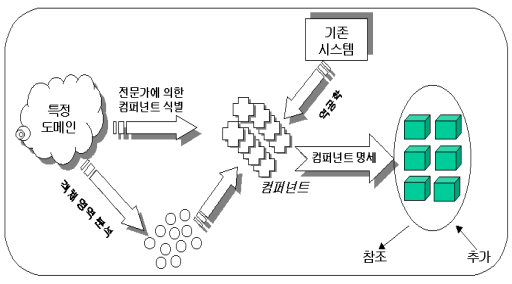
컴퍼넌트 아키텍처를 기반으로 컴퍼넌트의 개발, 관리, 사용을 위한 기초를 제공하기 위한 컴퍼넌트 명세는 CBD로의 전개를 위한 기초적인 세부 작업을 확정하며 컴퍼넌트 개발과 사용을 위한 유일한 식별 수단이다. 따라서 의미적으로 정규화된 명세를 갖는 컴퍼넌트만이 의미를 가진다.

현재 제시된 컴퍼넌트 명세 방법은 크게 인터페이스 기술 방법에 따라 <표 1>과 같이 구분된다[5, 6]. 제안한 명세를 통해 컴퍼넌트 생성을 위한 접근은 1) 특정 영역 전문가에 의한 기능적 컴퍼넌트 추출과 2) 도메인 분석을 위해 객체지향 방식의 도입에 의한 컴퍼넌트 식별, 그리고 3)기존 시스템의 역공학을 통한 식별 세 가지로 구분된다(그림 3). 어떤 접근 방법에 의한 컴퍼넌트 생성이든지 오퍼레이션의 집합인 인터페이스가 명확히 정의내려진 컴퍼넌트 명세의 정규화를 통한 컴퍼넌트의 공개와 보급이 필수적이다.

3. 컴퍼넌트 아키텍처와 컴퍼넌트 명세

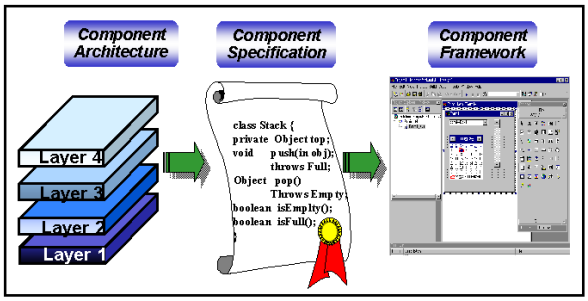
3.1 개요

본 논문에서는 컴퍼넌트 기반의 소프트웨어 개발의 궁극적인 귀결을 컴퍼넌트 프레임워크 구축을 통한 컴퍼넌트 시스템 생성의 자동화로 정의한다.



(그림 3) 컴퍼넌트 생성을 위한 접근 방법

韓國領 독도 Dokdo is Korean Territory



(그림 4) 컴퍼넌트 아키텍쳐와 명세, 프레임워크

따라서 본 논문의 동기와 의의를 (그림 4)을 통해 나타낸다. 먼저, 컴퍼넌트 아키텍쳐 모델을 정의하고, 컴퍼넌트 명세 방법을 제시한다. 그리고 향후 연구로 컴퍼넌트 프레임워크 구축으로의 접근을 제시한다.

3.2 ABCD 컴퍼넌트 아키텍쳐

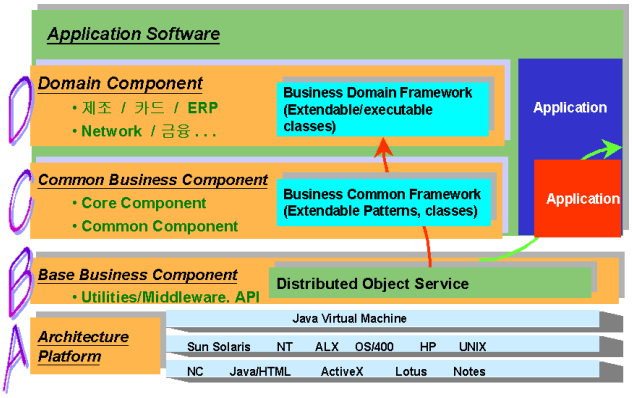
(1) 아키텍쳐 생성을 위한 요구 사항

컴퍼넌트 아키텍쳐는 관련된 다른 종류의 컴퍼넌트들을 연관시키기 위한 표준 계층으로 컴퍼넌트의 획득,이해, 조립을 위한 레이아웃을 제시함으로써 사용자들이 필요로 하는 컴퍼넌트들을 식별하고, 검색하며 커스터마이즈할 수 있는 가이드 라인을 제공해야한다. 본 논문에서는 분산 컴퓨팅 환경 하에서 비지니스 솔루션을 위한 표준 모델로 **ABCD** (Architecture-Base Component-Common Component -Domain Component) **아키텍쳐를** 제시한다. 본 아키텍쳐는 Sanfrancsico를 기초로 멀티 벤더/멀티 솔루션의 통합을 위해 컴퍼넌트의 스코프와 추상성, 입자성을 기준으로 계층적 분류를 시도했다. 다음은 ABCD 아키텍쳐 정의를 위한 원칙이다

> 비즈니스를 위한 컴퍼넌트와 일반적인 시스템 요구 컴퍼넌트로 구분하고, 기반이 되는 필수 컴퍼넌트와 부가적이고 선택적인 컴퍼넌트로 구분한다. 또한 완성되어 수정이 불가한 컴퍼넌트와 패턴 형식의 커스터마이즈가 가능한 컴퍼넌트로 분류하며 컴퍼넌트 제공 서비스의 범주에 따라 계층적 관계를 형성하는 군으로 정의한다. 기존에 정의된 분산 서비스의 컴퍼넌트를 계층에 포함시킨다.

(그림 5)은 ABCD 아키텍쳐이다. 전체 4개의 계층으로 구성되며 각 계층의 특징은 <표 2>와 같이 정의된다.

A와 B 계층은 미들웨어 역할의 API와 분산 객체 서비스 위한 기본 포맷, 하부적이고 물리적을 플랫폼을 제공한다. 따라서 계층 B에는 CORBA나 EJB와 같은 분산 객체 컴퍼넌트들이 포함된다.



(그림 5) ABCD 컴퍼넌트 아키텍쳐

계층 C는 많은 응용에서 요구되는 기본적인 기능들을 지원하기 위해 다중 도메인 상의 일반화된 메카니즘을 지원한다. 이 계층은 비즈니스 응용의 핵심 컴퍼넌트로서 공통 컴퍼넌트 영역을 두 개 계층으로 구분한다. 여기서 CBD를 위한 풍부한 조립 자원을 제공하고 비즈니스 프레임워크 구축을 위해 커스터마이즈 가능한 패턴을 포함하여 컴퍼넌트의 재사용 스코프를 확장한다. 그리고 계층 D는 수직적 도메인을 위한 특정 응용 컴퍼넌트이다. 특히, 비즈니스 영역 내에서 카테고리 되어진 컴퍼넌트 집합을 제공함으로써 조립시의 부가 비용을 축소화한다.

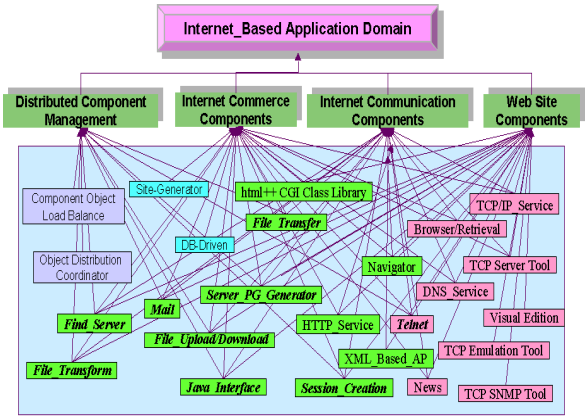
(2) 비즈니스 영역으로의 ABCD 아키텍쳐 적용

ABCD 아키텍쳐에 의해 정의된 계층에 따라 분류된 비즈니스 영역의 요구 컴퍼넌트들은 CBD로의 전개를 위한 실제적인 비즈니스 로직을 설명할 수 있다.

<표 2> 컴퍼넌트 아키텍쳐의 각 계층들

계층	이름	특징
A	Architecture platform	·분산 컴퓨팅 환경에서 멀티 벤더/응용 구축을 위한 하부적인 물리적 플랫폼
B	Base application component	·분산 컴퓨팅을 위한 미들웨어적인 통합 API 및 기존의 분산 객체 서비스 지원 ·응용에 하부적인 기능 수행에 필요한 공통적인 컴퍼넌트 ·벤더 독립적, 장비 지향적 서비스 ·기존의 분산 환경 멀티 응용 컴퍼넌트 이용
C	Common business component	·비즈니스 응용을 위한 실행성의 기능성 컴퍼넌트 ·Categorized component : 비지니스의 기초 해결군 – 확장적인 컴퍼넌트들 집합(범주화된 클래스들) – 작업 프로세스 : 컴퍼넌트 프레임워크의 부분 – 확장, 변경을 통한 응용 전개(설계 패턴 포함) ·Common component : 도메인에 공통적, 독립적 – 작은 규모로 비즈니스 영역에서 공통적으로 사용 ·Core component : 영역의 논리수행에 핵심적 프로세스 – 도메인 컴퍼넌트로 상향조정 가능 – 독립적인 작업 프로세스 단위
D	Domain component	·각 비즈니스 영역별 필요 컴퍼넌트 – 직접 활용되는 실행 모듈 형태 – 도메인 응용의 기본적인 빌딩 블록 – Categorized Component

韓國領 독도
Dokdo is Korean Territory



(그림 6) 인터넷 영역 컴퍼넌트의 계층

(그림 6)은 인터넷 기반의 응용 도메인에서 식별된 컴퍼넌트의 계층들이다. 본 논문에서는 현재 인터넷 상에서 유통되는 컴퍼넌트들을 탐색하여 ABCD 아키텍쳐에 재배치하였다.

3.3 컴퍼넌트 명세

(1) 요구되는 컴퍼넌트 명세 특징

현재 인터넷 상에 유통되고 있는 대부분의 컴퍼넌트 명세는 인터페이스 서술을 간과하며 개략적인 기능적 서술과 사용 환경상의 선택 등과 같은 항목으로 컴퍼넌트를 선택하도록 하므로써 사용자의 요구에 적합한 컴퍼넌트의 식별이 불가능하다.

<표 3> 본 논문에서 제안하는 컴퍼넌트 명세 방법

① 컴퍼넌트 명세를 상세함의 수준에 따라 두 가지로 정의.
② 각 인터페이스 항목의 의미를 명확히 하기 위해 비정규적인 문서를 이용하여 명세 항목마다 '비고'란을 작성하고 pre-,post-condition을 통해 의미를 한정
③ 컴퍼넌트 내 구성 요소들 간의 관련성 표현을 위해 Sequence Diagram과 State Transition Diagram 작성.
④ 컴퍼넌트 사용의 신뢰성 보장을 위해 사용 시나리오를 작성하고 카테고리의 내에 시나리오 컴퍼넌트 설정.
⑤ 컴퍼넌트 개발 환경과 컴퍼넌트 자체의 variant/invariant를 명시

<표 4> 본 논문에서 제시하는 컴퍼넌트 개략 명세 항목

항목	의미
Name	컴퍼넌트 식별 이름
Main Goal	(불변의)핵심적인 기능성
Description	전체적인 개요(기능, 요소, 기대 효과)
Families	컴퍼넌트의 비즈니스 카테고리
Related Component	상호 관계가 있는 컴퍼넌트
Provider	컴퍼넌트 제공자(업체)
Version	컴퍼넌트 버전
Environment	호환 가능한 플랫폼(시스템 환경)

결국 시스템으로의 적용시 개발자의 자의적인 해석에 의한 원래의 컴퍼넌트 개발 의도와 비일치가 발생하며, 적절한 행위적인 인터페이스의 의미적 확보가 미흡함에 따라 조립을 위한 쉬운 플러깅 지점을 가질 수 없다. 그러므로 비즈니스 로직을 일관적으로 수행할 수 있는 컴퍼넌트 명세가 요구된다.

(2) 제안 컴퍼넌트 명세 방법

본 논문에서는 ABCD 아키텍쳐에 기반하여 컴퍼넌트 명세에 대한 원칙을 <표 3>에, 원칙에 따른 명세 항목을 <표 4>와 같이 제시한다. 또한, 컴퍼넌트 개발을 위한 상세 명세로서 <표 3>의 특징에 따라 <표 5>과 같은 명세 항목을 결정했다. 명세시 각 항목의 비고란을 통해 상세한 부가 설명이 포함된다.

Component Diagram 항목에는 Invariant와 Variant, Exception으로 인터페이스를 명확히 한다. Usage Scenario 항목은 카테고리 내의 컴퍼넌트들의 이용 절차를 예시한 것으로 조립을 위한 확신된 가이드라인으로서 이용한다. Qualities Attribute에는 타입, 언어, 컨테이너, 데이터베이스, 미들웨어 등의 구현적 선택과 성능, 보완, 플랫폼 관점에서 준수해야 하는 요소들이 나열된다. 분류 코드에서 첫 문자 A, B, C, D는 컴포넌트가 속한 아키텍쳐 계층이며 다음의 영문자는 컴퍼넌트 도메인이다. 계속되는 숫자들은 도메인 상에서 식별된 컴퍼넌트들의 일련 번호를 의미한다.

<표 5> 본 논문에서 제시하는 컴퍼넌트 상세 명세 항목

항목	의미
카테고리	컴퍼넌트가 속한 비즈니스 도메인의 세부 기능적인 분류 따른 컴퍼넌트 군
컴퍼넌트 상관도	카테고리 내 컴퍼넌트들 간의 관계
컴퍼넌트 이름	비즈니스 관점의 서술적인 컴퍼넌트의 이름
분류 코드	ABCD 아키텍쳐에 기반한 컴퍼넌트의 계층적인 분류 코드
개요	컴퍼넌트 전반에 대한 기능, 동기, 동작 과정, 제약 사항 등에 대한 서술
용어 사전	컴퍼넌트 명세에 사용된 용어의 의미 설명
Component Context Diagram	컴퍼넌트의 주요 기능을 서술
Component Interaction Diagram	컴퍼넌트 실행시 관련된 컴퍼넌트들 간의 관계성을 표현
Component Sequence Diagram	컴퍼넌트 자체의 행위적 순서 표현
Component Diagram	컴퍼넌트가 제공하고 제공받는 인터페이스 표현
Component State Diagram	컴퍼넌트 오퍼레이션의 변화 표현
인터페이스 명세	공급하고 요구하는 인터페이스의 의미적 서술, pre/post-condition, 입력/출력 결과 명시
Usage Scenario	컴퍼넌트 사용을 위한 인증된 시나리오
Quality Attribute	비기능적인 컴퍼넌트 (품질) 특성

韓國領 독도
Dokdo is Korean Territory

4. 컴퍼넌트 명세 사례

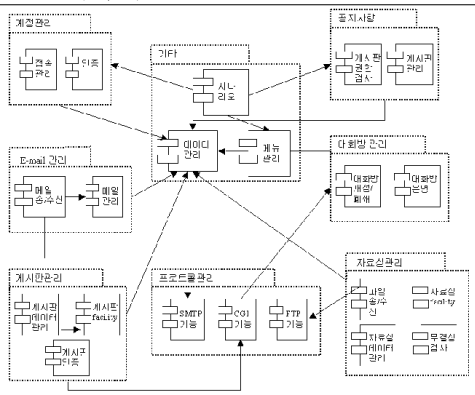
4.1 대화방 개설/폐쇄 컴퍼넌트

제안 명세의 예로서 BBS 응용 시스템 개발을 위해 필요한 컴퍼넌트의 명세를 작성하였다. BBS 관리 영역은 거의 모든 비즈니스 영역의 필수적인 영역으로 그 적용 범위가 광대함에 따라 공통 컴퍼넌트로서 큰 가치를 가진다. 컴퍼넌트 식별을 위한 도메인 분석에 의한 순공학적 접근을 시도하였다. <표 7>은 BBS 관리 카테고리 중 대화방 이용을 위해 식별된 "대화방 개설/폐쇄" 컴퍼넌트의 명세이다.

<표 7> 대화방 개설/ 폐쇄 컴퍼넌트의 명세 예

(1) BBS 관리 컴퍼넌트 카테고리(C-CO)	
High Component	Element Component
계정 관리(02)	접속 관리(01), 인증(02)
공지 사항(03)	공지사항 권한검사(01), 공지사항 관리(02)
E-Mail 관리(04)	메일 송/수신 관리(01), 메일 컴포저(02)
대화방(05)	대화방 개설/폐쇄(01), 대화방 운영(02)
게시판 관리(06)	게시판 데이터 관리(01), 게시판 퍼실리티(02), 게시판 인증(03)
자료실 관리(07)	파일송/수신(01),데이터관리(02),자료실퍼실리티(02),무결성검사(04)
프로토콜 관리(08)	SNMP 기능(01), CGI 기능(02), FTP 기능(03)
기타(01)	메뉴 관리(01), 시나리오(02), 데이터 관리(03)
"비고"	데이터 관리 : 데이터를 저장, 갱신, 수정, 삭제
	시나리오 :카테고리 컴퍼넌트의 동작 시나리오 정의

(2) BBS 관리 컴퍼넌트 상관도



(3) 이름 : 대화방 개설/ 폐쇄	(4) 분류 코드 : C-CO-05-01

(5) 개요 : 대화방 개설/ 폐쇄 컴퍼넌트는 대화방 개설자의 요청에 따라 Chat_register processing을 통해 대화방 database에 대화방을 등록시키고 등록된 대화방의 폐쇄 권한을 개설자에게 부여한다.대화상의 메시지 전송은 CGI, ASP, Perl CGI, PHP등을 사용, 대화방 운영 컴퍼넌트가 담당하며 대화방 개설/ 폐쇄는 대화방 운영 컴퍼넌트와 동등하게 대화방database를 접근하지만 개설과 폐쇄는 본 컴퍼넌트만이 접근할수 있다.

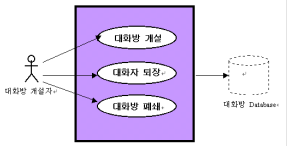
대화방 폐쇄시는 Chat_register Processing을 통해 폐쇄할려는 대화방의 data를 대화방 database에서 삭제하게 된다.

(6) 용어사전

대화방 database : 개설된 대화방의 저장 공간, 새로 생성되는 대화방을 등록시키며 개설자나 운영자가 방을 폐쇄하는 경우 해당 대화방의 data가 삭제된다.

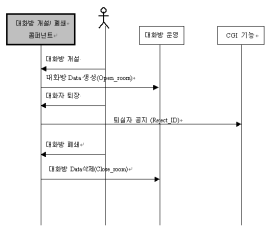
Chat_register processing : 대화방 생성(open_room)과 대화방 폐쇄(Close_room)의 메쏘드를 가지며 대화방data를 대화방 database에 등록시키는 일련의 작업을 수행하는 Process이다.

(7)Component Context Diagram



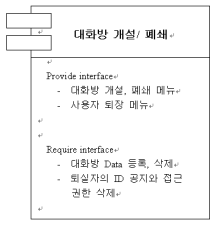
"비고" : 대화방 폐쇄는 대화방 개설자와 대화방 관리자만이 접근할 수 있는 권한이다.
대화방 Database는 개설되고 폐쇄되는 대화방들의 Data들을 관리하는 Database이다.

(8)Component Interaction Diagram



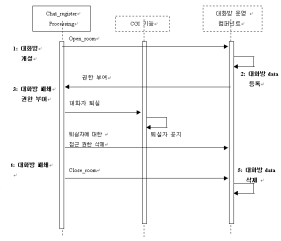
"비고" : 각각의 개설된 대화방의 data는 대화방 운영 컴퍼넌트에 의해 관리된다.

(9)Component Diagram



"비고" :
Invariant
대화방의 인원 제한, 운영자의 권한
Variability
개설자의 권한

(10)Component Sequence Diagram



(11) 인터페이스 명세

- 대화방 개설/ 폐쇄 메뉴
Provide Interface 대화방 개설/ 폐쇄 메뉴 [to] 사용자
Description : 대화방을 사용하기 위해 사용자에게 제공하는 interface이다.
[Preconditions : 정해진 최소와 최대 제한 인원,]
[Postconditions :]
Input : 방 제목, 제한 인원, 공개/ 비공개의 선택
Output : 대화방 사용에 관한 선택 사항들.

- 사용자 퇴장
Provide Interface 사용자 퇴장 [to] 사용자
Description : 불량 대화자의 해당 대화방 사용 권한을 삭제하는 interface이다.
[Preconditions :]
[Postconditions : 퇴실된 대화자의 ID가 모든 대화자에게 공지.]
Input : 대화자 ID
Output : 해당 ID의 대화방 접근 권한 삭제와 공지

- 대화방 Data 등록/ 삭제
Require Interface 대화방 Data 등록/ 삭제 [to] 대화방 운영
Description : 대화방 등록/삭제로 인한 Data관리를 위한 interface이다.
[Preconditions : 대화방 Data 스키마의 정의.]
[Postconditions : Database의 물리적인 변화.]
Input : 대화방의 속성들./ Output : 대화방 Data.

(13)Usage Scenario

[필수 기능]

대화방 개설

- 대화방을 개설하려는 사용자는 선택 사항을 선택하고 대화방을 개설하면개설자의 권한을 부여받고 운영자 또는 대화방 관리자와 함께 해당 대화방을 폐쇄시킬수 있으며 대화방을 사용시킬 수 있다.

대화방 폐쇄

- 대화방 사용을 마친 개설자의 요구에 따라 해당 대화방을 폐쇄시킨다.

[부가적 기능]

- 대화자 퇴실 기능으로 해당 대화방의 사용자중 불량 대화자에 대해 퇴실 시킬 수 있다.
-퇴실된 대화자는 해당 대화방에 다시 접근할 수 없다..

(14) Quality Attribute

- 비기능적 속성, 보완, 성능, 신뢰성에 대한 요소

Environment: Implementation Choice
Type : CGI, Java Sevelet
Language : Java, VB CGI, CGI, Perl CGI, PHP
Tool : MS Visual Studio, java Editor, HTML Editor
Container : Plug-in 될 수 있는 툴, 환경 BBS, Web site
Database : MS-SQL Server 6.5

Criterial
대화방 불량 사용자의 퇴실 기능과 대화방 관리자나 BBS운영자가 사전 통보없이 대화방을 폐쇄시킬 수 있다.

韓國領
독도
Dokdo is Korean Territory

4.2 제안된 컴퍼넌트 명세의 특징

현재, 컴퍼넌트 자체 및 컴퍼넌트 기반 소프트웨어 개발을 위한 컴퍼넌트 식별에서 표현 기호를 이용한 명세화, 저장소에서의 관리 및 응용으로의 전개를 지원하는 도구 등의 CBD 방법론에 대한 총체적인 평가 기준이 정립되지 않은 상태이다. 따라서 어떤 명세가 명확하며 어떤 컴퍼넌트의 도메인 비즈니스에 유용한지는 시스템 개발자의 주관적 판단에 의존해야 한다. 본 논문 역시 정량적인 논리적 평가 기준의 적용이 어려운 까닭에 컴퍼넌트 개발과 사용 모두의 측면에서 상세하고 체계적 수준의 컴퍼넌트 정보 제공을 원칙으로 직관적인 인식이 가능하도록 컴퍼넌트 아키텍처의 기준을 정립하고 명세 항목을 결정했다. 본 논문에서 정의한 컴퍼넌트 아키텍처 기반의 명세의 특징은 <표 8>에 요약, 제시된다.

<표 8> 제안 컴퍼넌트 명세의 특징

① 비즈니스 카테고리를 중심으로 컴퍼넌트를 분류, 식별, 명세화함으로써 프로세스 로직 수행을 위한 컴퍼넌트의 명확한 이해를 제공한다. 특히 카테고리 내에 시나리오 컴퍼넌트를 추가하여 전체적인 컴퍼넌트 전개 과정에 대한 참조 문서로서 활용한다.
② Component Sequence Diagram과 State Transition Diagram을 포함함으로써 컴퍼넌트 자체의 이해 획득을 통해 컴퍼넌트 활용의 핵심인 인터페이스의 자세히 파악한다.
③ 컴퍼넌트 분류 기호는 본 논문에서 정의한 ABCD 아키텍처를 기반하여 분류한 것으로 컴퍼넌트의 수직적인 계층 관계를 표현할뿐더러 컴퍼넌트 관리를 위한 유일한 식별 요소로 이용한다.
④ 컴퍼넌트의 개발, 실행 환경 등의 비기능적 요소들과 개발이 완성된 컴퍼넌트의 품질적 제한 조건을 명시함으로써 지원 하부구조의 표준화를 유지하고 최소한의 검증 기준으로 사용한다.
⑤ 비정규화된 문서를 이용해 컴퍼넌트의 상세한 기능성을 제시하며 UML 기반의 모델링 기호를 사용하여 다른 컴퍼넌트와의 비교 요소들을 포함하였다. 인터페이스 서술시의 제약 조건을 정규 서술 언어 형식으로 명시함으로써 차후 컴퍼넌트 검색, 분류 등의 조작을 위한 메타 정보로 이용한다.

5. 결론 및 향후 연구

컴퍼넌트 기반의 소프트웨어 개발에는 구현시의 행위적인 정규 서술과 컴퍼넌트 조립을 위한 일반적인 메소드를 제공하는 컴퍼넌트 명세 방법이 필요하다.

따라서 본 논문에서는 ABCD 컴퍼넌트 아키텍처를 정의하고 이에 기반한 새로운 컴퍼넌트 개발 명세 방법을 제시하였다. 제시된 방법은 컴퍼넌트 개발과 사용 모두의 측면에서 상세하고 체계적 수준의 컴퍼넌트 정보 제공을 원칙으로 직관적인 인식이 가능하도록 두 가지 수준으로 분리 정의하였다. 특히 비즈니스 프로세스 컴퍼넌트 실현을 위한 컴퍼넌트의 카테고리를 분리하고 카테고리 내에서의 컴퍼넌트 활용 프로세스를 시나리오 컴퍼넌트를 통해 전체적으로 정의, 참조하며 Component Interaction Diagram과 Usage Scenario를 통해 컴퍼넌트 조립을 위한 가이드라인을 제공한다. 또한 컴퍼넌트의 개발, 실행 환경 등의 비기능적 요소들과 개발이 완성된 컴퍼넌트의 품질적 제한 조건을 명시함으로써 지원 하부구조의 표준화를 유지하고 최소한의 검증 기준으로 사용한다.

현재, 컴퍼넌트 활용의 기반조성과 CBD 프로세스 구현을 위한 체계적인 지원환경은 거의 전무한 상태이다. 이는 컴퍼넌트 명세의 표준화가 여러 밴더들의 이익에 발맞추어 쉽사리 잡히지 않음과 컴퍼넌트 기반의 소프트웨어 개발 및 시험/검증 도구의 부재를 문제점으로 꼽을 수 있다.

향후 연구로는, 본 논문에서 정의된 컴퍼넌트 명세를 이용한 비즈니스 컴퍼넌트의 명세 작업을 진행하며 이를 컴퍼넌트 저장소 개발을 통해 CBD 환경을 구축함으로써 실질적인 컴퍼넌트 기반의 소프트웨어 개발 프레임워크로서 활용할 것이다.

[참고 문헌]

[1] Clemens Szyperski, Component Software : Beyond Object-Oriented Programming, Addison-Wesley, 1998

[2]. Alan C. Wills, "Designing Reusable Components", at URL : http://www.trire.com/catalysis

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

[4] IBM, Sanfrancisco Technical Report, at URL : http://www-4.ibm.com/software/ad/sanfrancisco

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

[6] Sherif Yacoub Ammar, "A Model for Classifying Component Interface", Proceedings of 1999 International Workshop on CBSE, Los Angeles, at URL:http://www.sei.cmu.edu/cbs/icse99/cbsewkshp

韓國領
독도
Dokdo is Korean Territory