

목차

- 연구 목적
- 연구 범위
- 관련 연구
- 컴포넌트 측정 모델 구축
- 결론 및 향후 연구과제
- 참조문헌

연구 목적

□ 컴포넌트 기반 개발(CBD :Component Based Development)의 확산에 따른 컴포넌트 측정 모델의 구축

- ▷ 컴포넌트 기반 개발과 기존 객체 지향 기반 개발 간의 연관성과 상호보완성 파악
- ▷ 컴포넌트의 어셈블리를 위한 컴포넌트 측정 모델의 구축

연구 범위

- 객체 지향 측정 모델 연구
- 객체 지향 설계분석과 컴포넌트 기반 개발 간의 관계 연구
- 컴포넌트 측정 모델 구축

응집도

□ 응집도(Cohesion)의 일반적 정의

- ▷ 클래스가 가지고 있는 특성들의 집합이 문제 영역 또는 설계 영역에서 차지하는 정도
- ▷ 소프트웨어 모듈과 내부적 요소들 간의 연관 혹은 결합 등의 상호작용의 정도
- ▷ 디버깅(Debugging), 유지보수, 수정 등의 구현 사명을 예측하는데 사용되어지는 특성

결합도

□ 결합도(Coupling)의 일반적 정의

- ▷ 모듈간의 상호 의존성을 측정하는 방법
- ▷ 클래스들 간에 메시지 호출이나 상호 의존성과 같은 여러 요소(factor)들을 고려하여 두 클래스 간에 강한 결합력을 가지고 있는 클래스들을 하나로 묶기 위한 설계 측정 메트릭
- ▷ 가능한 낮은 결합도가 요구되어지며, 낮은 결합도는 재사용, 유지보수의 용이성 증대, 파급 효과(Ripple Effect)의 최소화 등의 이점을 줄 수 있음

□ OOAD의 응집도 와 결합도 측정 모델

연구자	응집도	결합도	특징
Chidamber, Kemerer	크기, 응집도, 결합도, 재사용	WMC, DIT, NOC, CBO, RFC, LCOM	클래스 중심
Lorenz, Kidd	크기, 응집도, 결합도, 재사용	CS, NOO, NOA, SI	객체지향
Harrison, Counsell, Heiss	재사용, 결합도, 재사용	MIF, CF, PF	
Binder	사양도	LCOM, PAP, PAD, NOR, FIN, NOD, DIT	
Lorenz, Kidd	객집도, 결합도	OS _{obj} , OC, NP _{obj}	동적 응집도

관련 연구

□ 컴포넌트(Component)의 정의

- > 잘 정의된 인터페이스를 통하여 서비스를 제공하고 받는 것으로서 캡슐화 되어지고(Encapsulated), 분배 가능하며(Distributable), 실행 가능한(Executable) 소프트웨어의 조각
- > 잘 정의된 어택택처의 문맥에서 분명한 기능을 수행하는 것으로서 서스어지 않고, 각의 독립적이며, 교체 가능한 시스템의 일부분
- > 잘 정의된 어택택처 상에 명확히 정의된 기능을 충족시키는 서스어지 않고, 극히 독립적이며 대체 가능한 시스템의 부분

□ 컴포넌트 기반 개발(CBD:Component Based Development)의 정의

- > 컴포넌트의 개발 및 획득 뿐만이 아닌 그러한 컴포넌트를 조합하여 소프트웨어 집약적인 시스템(software-intensive systems)을 만들고 배치하는 것
- > 컴포넌트를 근간으로 한 개발에 있어서의 모든 요소들, 즉 어택택처링, 전체 개발 생명주기, 기술적 인프라스트럭처, 심지어 프로젝트 관리 등을 포함한 소프트웨어 개발 접근법

관련 연구

□ 컴포넌트와 객체의 비교

구분	컴포넌트	클래스
목적	다중 인터페이스	단일 인터페이스
캡슐화	캡슐화	다중 오버라이선
독립적 언어 가능	독립적 언어 가능	캡슐화
서비스의 재사용	서비스의 재사용	구현의 재사용
언어 비종속성	언어 비종속성	언어의 종속성

□ 컴포넌트의 재사용

- > 조합(Composition)에 의한 재사용
- > 블랙 박스 방식의 재사용
- > 서비스의 재사용
- > 각 컴포넌트는 상이한 기술로 구현 가능
- > 인터페이스에 의해 연동을 컴포넌트로부터 분리



관련 연구

□ 클래스의 재사용

- > 상속(Inheritance)를 통한 재사용
- > 화이트 박스 방식의 재사용
- > 클래스의 내부 동작에 관한 지식을 요구
- > 구현의 재사용
- > 동일 프로그래밍 언어를 요구
- > 특정 구조 상의 변동의 영향을 예측하기 어려움

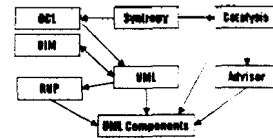
OOAD와 CBD 간의 연관성

- "CBD는 새로운 기술이 아니라 객체 지향 개발이나 데이터 모델링, 분산 시스템 등을 포함한 전통적인 개발 접근법, 기술들의 집합체이다."
- 실제 Catalysis Methodology, CBD96, RUP, UML Components 등의 대표적인 컴포넌트 기반 개발 방법론들은 객체 지향과의 강한 연계성을 지닌다.
- 여전히 CBD는 전혀 독립적인 방법론으로서 발전해 온 것이 아니라 기존의 OOP와 소프트웨어 공학 관련 기술의 통합을 통해 진보된 통합 방법론이라 할 수 있다.

관련 연구

UML Components

- Syntropy, OMT, Catalysis, Advisor, RUP 등 여러 방법론에 영향을 받은 컴포넌트 기반 개발 방법론
- 컴포넌트 시스템의 아키텍처와 의존성을 명세화하는 방법론



CK metrics

- OO 소프트웨어 매트릭스 중 가장 널리 참조되는 집합
- 1994년 Chidamber, Kemerer에 의해 발표
- 화이트박스 재사용성(White-Box Reusability) 측정 모델
- 6개지의 매트릭스를 통해서 객체 지향 시스템의 응집도와 결함도 등을 측정
- WMC, DIT, NOC, CBO, RFC, LCOM,

컴포넌트 측정 모델 구축

컴포넌트 측정 모델 구축의 두 가지 접근방법

- 블랙박스 방식
 - 수정 없이 부품을 재사용하는 방식의 특징
 - 컴포넌트의 독립성, 정복순서, 인터페이스를 통한 기능 구현 등의 특성을 바탕으로 한 측정 방식
 - 높은 응집도, 낮은 결함도, 적은 메개 변수 사용도
- 화이트박스 방식
 - 요구사항에 맞게 내부의 구현을 수정하는 수정 가능성을 측정
 - 컴포넌트 상세 단계의 컴포넌트 구성과 메개 변수 관계성을 바탕으로 한 측정 방식
 - 컴포넌트 내부의 구현은 언어 독립적이지만 구성 요소가 어려움

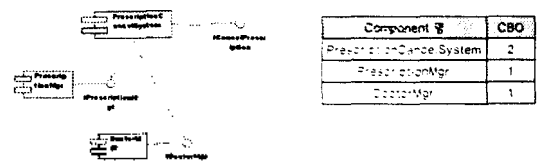
컴포넌트 측정 모델 구축 시 고려 사항

- 컴포넌트의 독립성
 - 한 컴포넌트와 구현은 다른 컴포넌트와 구현에 종속되어서는 안됨
- 컴포넌트의 상속 가능성
 - 컴포넌트의 독립성을 보장하기 위해 컴포넌트 간의 상속은 불가
 - 실제 컴포넌트 모델인 CORBA, EJB와 경우 컴포넌트의 상속 관계를 부분적으로 이해하고 있음
- CK Metrics는 실제에 바탕을 둔 측정을 수행하므로 본 컴포넌트 모델 구축에 있어서는 컴포넌트의 상속을 불가능하다고 가정

컴포넌트 측정 모델 구축

CK Metrics를 바탕으로 한 컴포넌트 측정 모델

- CBO(Coupling Between Objects)
 - 하나의 클래스의 다른 클래스와의 결합도
- m_CBO(modified_CBO)
 - 하나의 컴포넌트와 다른 컴포넌트와의 결합도
 - 결합도가 낮을수록 재사용성은 높음



- 컴포넌트 기반 개발과 객체 지향 개발 간의 연계성을 바탕으로 객체 지향 측정 모델인 CK Metrics를 모델 구축에 기본으로 함
 - ▶ 컴포넌트의 특성에 맞게 CK Metrics의 측정 인자를 변경하여 재구성하는 방법이 가능함을 확인
- 컴포넌트의 재사용성 측정을 위해 블랙박스 방식의 모델을 구축함
 - ▶ 컴포넌트 간의 응집도와 결합도를 측정함으로써 재사용성을 측정할 수 있음을 확인
 - ▶ 컴포넌트의 응집도와 결합도 측정에는 m-CBO, m-LCOM, m-RFC가 적합함
 - ▶ 컴포넌트의 내부 구현은 개발언어에 종속적이지 않으므로 화이트 박스 방식의 모델 구축은 CK Metrics 기반 모델 구축에 적합치 않음

참고 문헌

- Whitlira, S., *Object-Oriented Design Measurement*, Wiley 1997
- John Cheesman, John Dannels, *UML Components*, Addison-Wesley Publishing, 2000
- Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 5th Edition, McGrawHill, 2001
- Peter Herzum, Oliver Sims, *Business Component Factory*, Wiley 2005
- Chidamber, S.R., and C.F. Kemerer, "A Metrics Suite for Object-Oriented Design," *IEEE Trans. Software Engineering*, vol 20, no 6, June 1994, pp. 629-639
- James M. Bieman, Byung-Kyoo Kang, *Measuring Design-Level Cohesion*, *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, VOL. 24 NO. 2, FEBRUARY 1998
- Chidamber, S.R., D.P. Darcy, and C.F. Kemerer, "Management Use of Metrics for Object-Oriented Software: An Exploratory Analysis," *IEEE Trans. Software Engineering*, vol. SE-24, no. 6, August 1998, PP 629-639