2000년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

컴포넌트 개발기법 비교 연구

김현주*, 서동수
성신여자대학교 전자계산학과
e-mail : {hjkim, dseo} @cs.sungshin.ac.kr

A Comparision Study on Component Based Software Development Methods

Hyen-joo Kim*, Dongsu Seo
Dept. of Computer Science, Sungshin Women's University

요 약

컴포넌트는 소프트웨어의 재사용의 극대화를 통해 소프트웨어 제품의 생산성과 안정성을 제공해 줄 수 있는 방안으로 인식되고 있다. 컴포넌트 개발기법은 컴포넌트 개발 시 컴포넌트를 추출하고, 개발되는 각 컴포넌트간의 연결하며, 구현된 컴포넌트 어플리케이션에서 사용할 수 있도록 한다. 본 논문에서는 현재 주요한 컴포넌트 개발 방법론으로 거론되는 UML, 카타르시스 방법론, CBD96 의 내용을 비교 분석 한다.

1. 서론

컴포넌트란 하나의 독립적인 단위로 개발되어지고 배치될 수 있으며, 요구되는 시스템 구성을 위해 다른 컴포넌트와 연결 될 수 있고 그 자체는 변경되어지지 않는 응집된 소프트웨어 패키지라고 정의하고 있다.

컴포넌트 기반 개발(CBD)은 컴포넌트를 구축, 선택, 조립에 이용할 수 있는 소프트웨어 개발 방법으로서 소프트웨어 개발자들에게 잘 정의된 인터페이스를 제공하여 블랙박스 형태의 컴포넌트를 가지고 컴포넌트의 조립을 더 쉽고 빠르게 응용이 가능하도록 하게 한다. 또한 컴포넌트의 재사용은 소프트웨어의 개발에 구성요소의 모듈화를 촉진시켜 산업화와 대량화를 통한 코드의 재사용으로서의 개발비용의 절감과 기간의 단축에 도움을 줄 수 있다.

이렇게 볼 때 보다 높은 질을 가진 컴포넌트를 사용하는 것이 높은 신뢰성과 생산성 향상에 기여한다. 컴포넌트 개발 방법론은 컴포넌트의 분석 단계에서부터 설계 및 구현단계까지 사용되며 그 산출물들은 실행되는 컴포넌트와 컴포넌트에 대한 자료로 나타난다. 따라서 컴포넌트 개발자는 현재 사용하고자 하는 개발기법들에 대하여 각 방법론의 장점과 단점을 파악하고 개발하고자 하는 컴포넌트에 적합하다고 생각되는 개발 기법을 사용하는 것이 필요하다. 그러기 위해서는 다양한 방법론에 대한 비교 분석이 절실히 요청된다.

이에 본 논문에서는 각 방법론을 사용하였을 때의 컴포넌트 산출물들과 각 개발 단계에 대해 비교한다. 2 장에서는, 관련연구로 컴포넌트의 대한 기존의 연구와 분석방법에 대해 언급하고, 3 장에서는 각 방법론에 대한 분석을 하고 마지막 4 장에서는 본 논문의 결론을 맺는다.

2. 관련연구

컴포넌트의 기본적인 구성요소는 컴포넌트의 구현물, 인터페이스, 그리고 이에 대한 명세로 말할 수 있다. 일반적으로 그 구현은 블랙박스로 감추어 두고 제공되는 인터페이스를 통해서만 사용하고자 하는 컴포넌트의 서비스를 이용할 수 있다. 따라서 인터페이스에 대해서는 정확하고 상세한 명세가 필요하며 그 명세를 통해 컴포넌트끼리의 연결이 가능해진다.

따라서 인터페이스만 일치하면 다른 컴포넌트로의 대체가 가능하게 되는데 이러한 컴포넌트 개발 기술로는 OMG 의 CORBA, Microsoft 의 ActiveX/DCOM, Sun Microsystems 의 JavaBean, EJB(Enterprise JavaBeans) 등이 있다.

컴포넌트의 개발주기는 크게 요구사항 수집, 분석, 설계, 개발, 배치의 다섯 단계로 볼 수 있는데 각 방법론들은 개발 단계를 제외한 (이 단계는 실제 코딩 작업을 말한다.) 각 단계에 해당하는 모델링 기법들을 제공하고 있으며 이 단계들 마다 필요한 경우 중간 산출물들을 제공하도록 하고 있다.

3. 방법론 분석

이 장에서는 대표적인 컴포넌트 관련 방법론인 UML 과 Catalysis, Cool Software 의 CBD96 의 주요 모델링기법과 프로세스를 분석한다.

각 방법론들은 요구분석에서부터 컴포넌트의 명세, 설계, 구현단계에 이르기까지의 단계들을 정의하는 모델링 기법을 제안하고 있다. 여기서는 각 개발 단계별 모델링 기법들을 다음의 측면에서 비교해보려고 한다.

측면	내용
서비스	해당 단계의 모델링기법이 정의되어 있는가.
모델링 개념의 정의	명확하게 그 단계를 위한 것인가
산출물	산출물의 유무
재사용	재사용 관련된 명세

[표 1] 평가 측면 및 내용

각 측면에 대해서는 다음과 같이 3 가지로 표기한다.

- ● : 상세함
- ▲ : 어느 정도 지원
- Δ : 약간 관계있음
- × : 지원 안함

3.1 UML

3.1.1 소개

UML(Unified Modeling Language)은 시스템 개발 세계에서 표준으로 인정 받은 시스템으로서 시스템 분석가에게 의뢰인, 프로그래머, 그리고 시스템 개발 과정에 참여한 모든 사람들이 각자의 시점에서 이해할 수 있는 다방면의 설계도를 그릴 수 있는 표준을 제공하며, 제안하는 그래픽 요소를 조합하여 도를 그릴 수 있도록 되어있다.

UML 에서 사용되는 그림으로는 클래스도, 객체도, 사용사례(use case)도, 상태도, 순차도, 활동도, 협력(collaboration)도, 컴포넌트도, 배치도 등이 사용되며 이를 조직화 하고 확장하는 장치로 패키지와 노트, 스테레오 타입 등의 개념이 더 들어간다.

3.1.2 프로세스 개요

UML 표준안에서는 표준인 개발 프로세스를 특별히 정의하진 않는다. 그 이유는 프로세스의 접근방법이나 개발자의 기술, 문제의 성격등에 따라 매우 다양할수 있기 때문이다. 대신 대표적인 스텝으로 성공적인 프로세스를 설명하고 있다.

크게 3 스텝으로 구성되는데 첫번째 계획 과 작성으로 문제의 정의 및 프로토타입의 구성 등으로 되어 있고 2 단계인 구성단계에서는 시스템을 구성하며 세번째로는 배치로서 시스템을 완성한다.

첫번째 스텝에서는 계획을 수립하고 요구분석과 프로토타입을 구성하며 전체적인 사용사례를 정의한다.

두번째 구성에서는 필요한 만큼 개발 주기를 사용하는데 이 개발 주기는 6 단계로 되어 있으며 계획수립, 분석, 디자인, 구성, 테스트 단계를 거치며 이 개발주기마다 사용사례가 나온다. 이 사용사례를 가지고 이 단계중의 분석과 디자인 단계가 도로 표현이 된다.

분석단계에서는 사용사례의 세부 정의가 되고 다시 사용사례, 개념모델의 재정의와 시스템의 시퀀스 도, 오퍼레이션, 상태 도가 정의된다.

디자인단계에서는 실제 사용사례와 UI, 스토리보드 등이 정의되고 시스템 구성과 관계도 등이 재 정의되며 필요하다면 데이터베이스 스키마가 구성이 된다.

3.2. Catalysis

3.2.1 프로세스 개요

카타르시스는 DesMond F.D'Souza 등에 의해 제안되어 1991 년 시작된 OMT 의 정형화 작업에 그 뿌리를 두고 있다. 카타르시스는 비즈니스모델에서 소스코드에 이르기까지 전과정에 대한 추적성을 보장하고,불명확한 모델이나 문서를 최대한 명확하게 만드는 정확성과 컴포넌트 기반의 개발, 코드뿐 아니라 분석이나 디자인의 산출물들까지 확대된 재사용 개념,확장성, 그리고 유연한 프로세스를 그 주요 특징으로 하고 있다.

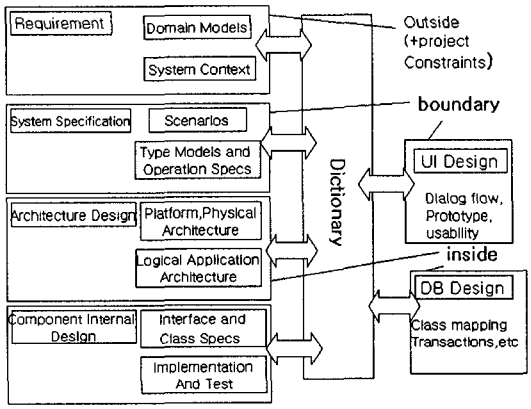
카타르시스의 가장 큰 특징이자 장점은 프로세스 패턴으로서 프로세스에 패턴의 개념을 도입한 것이다. 미리 사용이 예상되는 프로세스에 대한 패턴을 정의하고, 각 패턴별 작용 지침을 만들어 두고 있다.따라서 적정한 프로세스를 선택한다는 것은 요구사항에 대한 수집이 이루어졌다고 보아도 된다. 카타르시스가 포함하고 있는 패턴들은 개괄적인 객체 개발과 같은 패턴에서부터 모델링 패턴, 디자인 패턴에 이르는 세부패턴까지 다양하여 카타르시스가 프로젝트에 유연한 적용이 가능하게 한다. 따라서 카타르시스는 고정 프로세스가 제공되지 않고 개발 대상 시스템의 형태나 주어진 업무에 의해 그 프로세스가 변하는데 [그림 1]은 그 중 대표적인 프로세스의 구성을 보여주고 있다.

3.2.2 단계별 모델링 개념

카타르시스는 다음의 3 가지 모델링 개념에 기초 하여 구성되어 있는데 각각은 협력(collaboration), 타입(Type), 정제(Refinement) 이다.

협력은 연관된 객체간의 상호 연동으로 설명이 가능하며 특정 서비스를 제공하기위해 수행되는 작업과 그 작업에 참여하는 객체들로 이루어 진다. 이런 협력 모델을 통해 대상 시스템이나 도메인의 기능을 큰 수준에서 작은 수준으로 명세할 수 있으며 이것을 협력

도의 형태로 보여줄 수 있다.



[그림 1] 카타르시스 프로세스

타입은 한 객체의 외부에서 보여지는 행동 양식을 명세한 개념으로 도메인의 정적 모델을 모형화 하기 위해 제안되었는데 객체의 확장 형태로 표현되며 각 타입모델은 내부내의 객체모델을 가지고 구성요소로 타입이름, 속성부분, 오퍼레이션의 3 부분으로 표현된다.

정제는 동일 대상에 대해 추상적 수준에서 구체적 수준으로 정제해가면서 상하간의 맵핑과 추적성을 지원하며 크게 3 가지 수준의 모델링 단계를 명시하는데 먼저 가장 상위 수준인 도메인/비즈니스 단계, 그 다음은 시스템의 계약(Contract) 을 명세하는 컴포넌트 명세 단계, 마지막으로 내부 설계 단계는 내부 아키텍쳐와 협동의 내부를 정의하며 시스템과 컴포넌트의 내부를 설계한다.

그러므로 협력과 타입은 개발 단계 중 분석단계에 사용되는 모델링으로 볼 수 있으며 정제는 컴포넌트의 설계단계로 인식할 수 있다.

3.2.3 프로세스 패턴

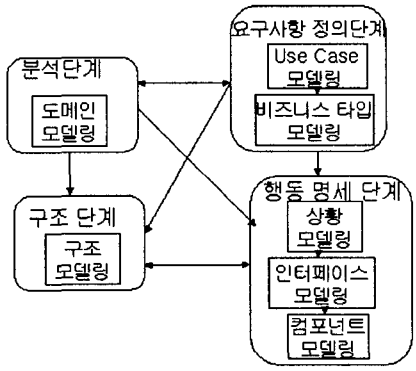
다른 컴포넌트 기반 방법론에 비해 가장 큰 카타르시스의 특징은 패턴의 사용이다. 각 패턴들은 시스템 혹은 컴포넌트 개발 시 부딪힐 수 있는 전형적인 경우들에 대해 지침을 제안하고 있다. 각 패턴들을 통해 비즈니스 모델이 객체, 속성의 수준으로 정제되는 과정을 볼 수 있으며 시스템 수준의 액션과 타입 명세에서 실제 구현단계까지 표현된다. 이를 통해 설계 단계와 배치단계에서도 패턴이 사용됨을 알 수 있다.

3.3 CBD96

CBD96 은 현재 Cool Software 사에서 자사의 COOL 시리즈 케이스 툴과 연계하기 위해 제안한 방법론으로서 프로세스 구현에 특별한 방법론을 제시 하지는 않는다. 다만 그림 3 과 같은 모델링을 기준으로 작업의 단계를 구분할 뿐이다.

Use Case 모델링은 기능적 요구사항의 주요 부분들을 Use case 로 표현하여 사용자와 시스템간의 중요한 상호작용을 표현한다.

비즈니스타입 모델링은 관리대상인 비즈니스정보의 분류와 어떻게 구성되어있는지를 표현하여 도메인 행동의 중심이 되는 타입의 식별에 도움을 준다. 초기 컴포넌트 구조 모델의 입력자료로서 사용이 가능하다. 도메인 모델링의 목적은 연관 객체간의 작용과 각각의 개별적 역할 식별을 통해 분석가가 문제영역을 잘 이해하도록 돕는 것에 있다. 따라서 카타르시스에서 정의한 협력모델이 많이 사용된다.



[그림 2] CBD96 의 컴포넌트 모델링 태스크

상황모델링은 업무를 지원하기 위한 소프트웨어시스템에 요구되는 오퍼레이션의 식별이 그 목적이다. 그러므로 하나이상의 인터페이스를 갖는 도메인 타입 또는 소프트웨어간 상호작용을 보여주는 협력모델로 나타낼 수 있는데 상황모델은 대상 소프트웨어의 주요 기능들을 정의하고 그 소프트웨어가 사용되는 상황을 묘사하여 개발자가 자료중심이 아닌 행위중심으로 기능에 대한 요구사항에 관심을 갖게 한다.

인터페이스 모델링은 인터페이스 타입모델을 통해 각 인터페이스별로 각각에 속한 오퍼레이션들을 상세히 명세하게 한다. 각 오퍼레이션들은 사전 또는 사후 조건을 사용하여 상세화 된다.

컴포넌트 명세는 컴포넌트가 지원하는 인터페이스를 식별하는 작업인데 CBD96 에서는 특별한 프로세스나 지침들을 포함하지는 않고 있다.

구조 모델링은 컴포넌트간의 연결과 그들의 인터페이스를 통해 전체 소프트웨어의 총체적이고 개략적인 정보를 제공하는데 CBD96 에서는 구조모델링에 UML 의 컴포넌트 도를 이용하고 있다.

3.4 총괄 분석

각 방법론의 단계를 컴포넌트개발 5 단계와 비교하여 각 단계마다 다음의 측면을 놓고 비교했을 때다음 표와 같이 정리한다.

개발단계	기법	활용도	산출물	산출물 사용
요구사항 수집	Use case 모델링	●	Use case , system architectue 등	

분석	개념 모델링, 시스템 행위모델링	●	use case, 시퀀스 도등..	설계단계
설계	상세 사용사례, 시스템행위모델링	●	협동도,클래스도	코딩단계
배치	배치 모델링	●	배치도.컴포넌트도	

[표 2] UML 총괄 요약

개발단계	기법	활용도	산출물	산출물 사용
요구사항 수집	프로세스패턴	▲	타겟 컴포넌드 패턴	배치단계
분석	협력,타입모델	●	협력도,타입 모델도	설계단계
설계	정제모델	●	세분화된 협력도,타입모델도	코딩단계
배치	프로세스패턴	▲		

*개발 단계는 실제 코딩 부분이므로 제외

[표 3] 카타르시스 총괄 요약

개발단계	기법	활용도	산출물	산출물 사용
요구사항 수집	Use Case 모델링, 비즈니스 타입 모델링	●	UseCase 도 ,비즈니스타입 도	설계단계
분석	도메인 모델링	●	협력도	설계단계, 배치단계
설계	상황모델링, 인터페이스모델링,컴포넌트 명세	▲	상황 도, 인터페이스도	코딩단계
배치	구조모델링	▲	UML 컴포넌트 도	

*개발 단계는 실제 코딩 부분이므로 제외

[표 4] CBD96 총괄 요약

4. 결론

UML 은 정적, 혹은 동적 특성에 대한 명확한 정의와 표준으로 시스템을 다양한 시점에서 점검, 관찰이 가능하다. 또한 그 표준 언어를 제공하여 개발 시 표현에 관한 오해가 없도록 해준다. 다만 시스템이 "무엇을" 의도하는지를 말해주며 "어떻게"를 설명하지는 않는다.

또한 각 단계별로 많은 산출물들(세부단계마다의 도들)이 나오며 대부분이 그 다음 단계에서 재사용되기보다는 재정의되는 것이 많다.

카타르시스는 타입 모델이나 협동모델같은 진화된 모델링 기법을 제안하며 프로세스의 패턴을 제안하여 새로운 시스템 구축에서 역공학에 이르기까지 다양한 분야에 적용이 가능한 유연성을 제공하고 있다. 특히 타입모델 같은 경우 자료중심적인 개발에 치우치지 않고 시스템이나 도메인의 자료와 행동에도 비중을 두고 관리할 수 있도록 지원하고 있다.

다만 카타르시스에 익숙하지 않거나 경험이 부족한 분석자와 설계자들에게는 혼란스럽게 할만큼 태스크별 상세 적용 지침들이 부족하고 특히나 프로세스 패턴의 경우 상당한 경험을 요구하고 있다. 어떤 프로세스패턴을 선택할 것인가에 대해 분석자나 설계자가 사용 가능한 지침이 부족하다. 또한 컴포넌트를 식별하고 추출하는 프로세스에 대한 구체적 지침이 누락되어 분석자의 임의의 판단을 요구하는 것이 단점이다.

CBD96 은 컴포넌트의 특징과 제품으로서의 컴포넌트 제공자가 포함해야 할 제품 구성요소들을 잘 정의하고 있다. 이러한 표준은 CBD96 뿐만 아니라 다른 컴포넌트 관련 방법론에 그대로 적용이 가능하다. 하지만 CBD96 은 구체적인 프로세스를 지원하는 것이 아니라 모델링 태스크와 그들 간의 관계만을 보여 주고 있다. 또한 비즈니스 모델에서 컴포넌트를 추출하고 설계하는 구체적인 프로세스와 지침들이 제공되고 있지는 않다. 이것은 그 자체로 하나의 방법론으로 사용되기보다는 케이스툴의 지원성격이 강하다고 볼 수 있다.

따라서 앞으로는 실제구현에서 발생하는 문제들에 대한 비교연구가 필요하다고 생각된다.

참고문헌

[1]Craig Larman, "APPLYING UML AND PATTERNS", Prentice Hall PTR

[2] D'Souza Wills, "Object, Components, and Frameworks with UML – The Catalysis Approach", Addison Wesley, 1997

[3] Clemens Szyperski, "Component Software", Addsion Wesley, 1997

[3] Joseph Schmuller, "Teach Yourself UML",SAMS