

J2EE플랫폼을 위한 소프트웨어 아키텍처 설계

배정미^o, 박재년

대경대학 컴퓨터정보과, 숙명여자대학교 정보과학부
jmbae@tk.ac.kr^o, jypark@sookmyung.ac.kr

Software Architecture Modeling for J2EE Platform

Jeong-Mi Bae^o Jae-yeon Park
Dept of Computer Science Dae-kyeung College,
Dept of Computer Science Sook-myung woman's university

요 약

최근 소프트웨어공학에 있어서 소프트웨어 아키텍처와 컴포넌트에 관한 연구가 활발히 진행되고 있다. 소프트웨어 아키텍처란 컴퓨팅 시스템의 소프트웨어적인 구조이며 아키텍처를 이루는 주요 구성요소는 컴포넌트가 된다. 소프트웨어 아키텍처는 시스템 관련자들 간의 이해를 높이는 수단이 되며 개발시스템의 분석, 설계, 개발, 테스트, 유지보수 프로세스의 품질을 좌우하는 주요한 요소로서 전 개발공정을 추적가능하게 하는 장점을 지니고 있다. 또한 향후 재사용가능한 시스템의 산출물로서 이용하게 된다. 시스템 전체 구조관점에서의 중요성을 인식하여 J2EE 플랫폼에서의 아키텍처프로세스와 EJB 컴포넌트의 추출, 모델링방법을 제안한다.

1. 서 론

아키텍처는 잘 정의된 컴포넌트의 특징을 보여주어야 하며 각 컴포넌트는 인터페이스를 가져야 한다. 즉 아키텍처는 구성요소가 되는 컴포넌트를 추출, 명세하고 인터페이스 그리고 컴포넌트간의 상호작용에 대한 정보를 효율적으로 표현하여야 한다.[5] 기업형 시스템구축에 있어서 아키텍처의 결정은 가장 근본이 되며 아키텍처는 주요설계결정, 규칙과 패턴을 포함하는 것으로 시스템의 설계와 구현을 수행할 프레임워크를 정의한다. 아키텍처 중심적인 VIEW측면의 프로세싱을 고려한 J2EE플랫폼기반의 시스템구축이 요구되어진다.[9]

J2EE 플랫폼은 멀티터기업 시스템의 개발과 배치를 위한 최상의 기술이다. 컨테이너 개념은 J2EE플랫폼의 핵심기술이며 EJB 컨테이너는 EJB 서버안에서 실행이 된다. 기업형 EJB컴포넌트는 비즈니스 티어에 위치하며 EJB 컴포넌트의 모델링은 초기 아키텍처정의 단계에서 추출, 모델링되어야 한다.

초기 분석과정에서 추출대상이 되는 컴포넌트는 업무의 기능적인 요구사항과 시스템 품질에 관련된 비 기능적인 요구사항을 효과적으로 만족시키는 아키텍처의 요소가 된다. 비 기능적인 요구사항이란 시스템의 성능, 상호운용성, 신뢰성, 사용자인편의성 등 품질에 관련된 요구사항을 의미한다. 즉 소프트웨어 아키텍처는 일정한 서비스를 사용자에게 제공하는 기능을 수행하는 컴포넌트와 컴포넌트간의 결합관계, 컴포넌트의 결합 및 동작을 제어하는 제약조건으로서 이루어지는 시스템 구조의 전체적인 골격구조를 표현한다.[9] 이러한 재사

용가능한 독립적인 EJB 컴포넌트의 명세는 컴포넌트기반 시스템개발에 있어서 중요한 요소로서 작용한다. 그러나 기 제안된 CBD(Component Based Development)방법들에 있어서의 컴포넌트 추출과 명세기법은 개발자의 직관에 의지하여 정형화된 형태를 갖추지 못하였거나 혹은 정형화된 방법을 제시한 또 다른 기법들에서는 상세 설계단계에서의 결함력, 응집력을 분석하는 매트릭스방법을 적용하는 복잡한 과정을 거치게 되어 초기 분석단계에서 컴포넌트 명세를 지원하기에는 부적절하다 그러나 이 방법들은 분석단계에서 추출된 컴포넌트의 응집력을 평가하는 검증 방법이 될 수는 있다.

2. 연구범위

2.1 관련연구

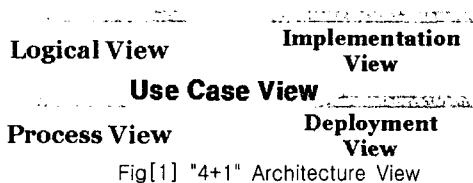
2.1.1 RUP 아키텍처[10]

RUP프로세스는 9개의 공정으로 이루어져 있다.

1. Business modeling 2.Requirments 3.Analysis & Design 4. Implementation Test 5. Deployment 6.Configuration & Change Management 7. Project Management Environment 로 나누어져 있다. 프레임워크에서 아키텍처 뷰는 시스템의 구성요소와 관계를 표현하기 위한 특정 관점이며 각각의 다른뷰가 존재한다. 특별한 관점에서의 소프트웨어 아키텍처를 표현하기 위해서 목적과 사용에 따라 뷰타입과 뷰스타일을 선택한다. 뷰타입과 스타일의선택은 복잡도를 현격하게 줄이고 시스템의 이해를 훨씬 쉽게 해주므로 아키텍처 뷰의 도입은 복잡한 시스템의 이해와 구축을 위해서는 필수적이다 이렇게 시스템을 아키텍처뷰라는 관점에서 해부하고 설계하는 것은 좀 더 세밀하고 안정적인 시스템을 구축하는 기반이 된

다. RUP에서의 Architectures 프레임워크는 4+1Views를 제시하고 있다. 기반이 되는 View는 Usecases이며 Conceptual/Logical View, Component/Process View, Implementation/Code View, Allocation/Deployment View로 구성된다. 5개뷰 중에서 +1로 표기하는 뷰는 유즈케이스뷰로서 아키텍처의 중심임을 나타내고 있다.[Fig 1]

유즈케이스가 다른 모든 뷰에 영향을 미치며 RUP의 첫 단계이고 시작점이며 시스템 외부에서 액터가 시스템을 보는 관점에서 시스템이 갖추어야 할 모든 기능들을 집약한 것으로 사용자 요구사항이 모든 뷰의 토대, 다른 뷰의 근거, 다른 뷰들을 검증하는 기준이 된다는것을 강조하는 것으로 해석된다. 다른 4개의 뷰는 유즈케이스 뷰를 각 영역별로 상세화하고 특화하여 진화시킨것이라고도 할수 있다.



Fig[1] "4+1" Architecture View

2.1.2 UML Components Method[2][4][11]

객체지향 개념기반의 Syntropy 방법론을 원조로 하여 Catalysis방법론과 스텔링의 CBD방법론이 Advisor방법론을 발전 시킨 것으로 설계,구축을 제외하고는 RUP의 비즈니스 모델링-요구사항분석,테스트,전이 단계가 동일하다. 비즈니스 타입 다이어그램의 독립적인 Core클래스를 식별하여 인터페이스를 정의하고 비즈니스 컴포넌트 후보를 식별한다. 유즈케이스 하나당 하나의 인터페이스를 식별하여 후보 시스템 컴포넌트를 식별하는 개념을 제시하고 있다.

2.2 연구방법

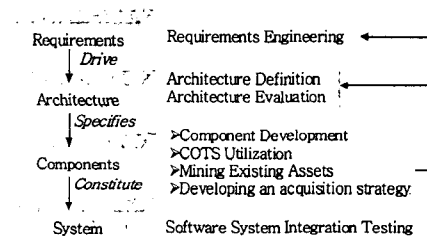
래셔널사의 소프트웨어 아키텍처인 RUP 4+1 VIEW와 ADD[8]를 기반으로 테일러링한 J2EE플랫폼 아키텍처 프로세스를 설계한다. 4+1 VIEWS의 Component/Process view 관점의 J2EE플랫폼의 시스템구축을 위하여 1,2,3,4,단계의 각 공정을 분석,테일러링 하고자 한다. 상세 프로세스가 되는 컴포넌트 분석과 설계단계에서는 유즈케이스 다이어그램과 클래스 다이어그램의 연관관계를 분석 추적하여 주요 대상 컴포넌트를 식별, 모델링하는 방법을 제안한다. 추출된 후보 컴포넌트를 분석하여 해당 아키텍처 패턴스타일과 프레임워크를 설계한다. 제안된 방법은 e-비즈니스도메인영역에서의 분석초기단계의 컴포넌트 추출과 명세를 위한 효율적이고 표준화된 방법으로 적용할 수 있다. 실례로서 e-비즈니스환경의 서적쇼핑몰에 적용한 아키텍처중심 컴포넌트 모델링을 설계하였다.

3. Architecture Process

Len Bass[5]는 아키텍처 비즈니스 사이클(Len 12) 을 다음과 같이 제안하였으며 이들은 피드백관계를 형성하여야 한다

1. 시스템에 대한 비즈니스 케이스를 생성한다.
2. 요구사항을 이해한다. - scenarios, usecases, quality attribute scenarios
3. 시스템의 기능적요구사항과 품질속성을 만족시키기 위한 아키텍처를 식별선택하거나 새로이 생성한다.
4. stakeholder의 의사소통이 중요하며 문서화 되어야 한다. 프로젝트설계의 백본역할로서 효과적인 아키텍처가 되기 위해서는 stakeholder의사소통이 중요하며 문서화되어야 한다. 문서는 개발자 테스터, 유지보수 관리자료의 의사연결의 수단 이 된다.
5. 아키텍처의 분석과 평가
설계단계에서는 여러개의 후보 아키텍처가 존재할수 있다.
6. 아키텍처의 평가
아키텍처의 평가는 stakeholder의 요구사항 만족을 충족시키는 아키텍처로부터 시스템이 구축되었나를 보장해야 한다.
7. 아키텍처기반의 시스템구현
8. 구현된시스템이 아키텍처에 맞게 형성되었는가의 적합성을 보증해야 한다.

Architecture Centric System Development Process



Fig[2] 아키텍처중심의 개발프로세스

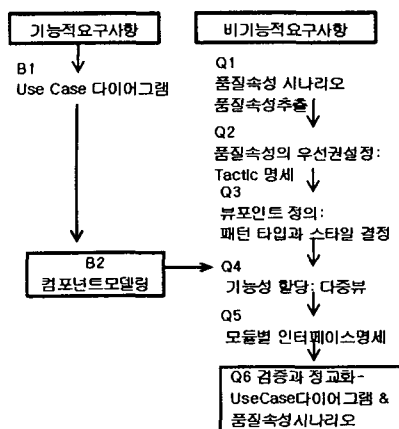
본 연구에서는 기능적 요구사항과 품질속성 요구사항을 결합한 아키텍처 프로세스를 설계하고 프로세스과정 중 컴포넌트 모델링과 추출의 방법을 살펴본다.

RUP는 초기단계의 아키텍처를 결과로서 제시하지만 상세디자인이나 구현단계로 진행시키고 있지는 못하다. RUP프로세스와 ADD방법[Fig 2]과 결합하여 일부단계를 수정하여 상위단계의 아키텍처를 설계한다.[Fig 3]

4. Component Extraction

본 논문에서 컴포넌트의 추출은 유즈케이스로부터 시작되며 추출된 컴포넌트가 동일 비즈니스 도메인의 유사 업무영역이라는 개념으로부터 시작된다. 즉 객체지향 개발에서 사용하는 일종의 서브 패키지 개념도 후보 컴포넌트로 식별될 수 있다. 이는 시스템의 설계 및 개발 시 시스템의 업무 및 기능적 구분이 되기 때문이다. 컴포넌트식별의 단계는 제안된 프로세스의 B2 컴포넌트 모델링 프로세스에 해당된다.

우선 Use Case Diagram으로부터 공통적 업무영역을 추출하고 다음단계에서는 각 Use Case마다 작성되는 클래스다이어그램의 포함관계, 연관관계를 분석하여 인터페이스를 분류하며



Fig[3] J2EE플랫폼을위한 테일링링 개발프로세스

이 인터페이스를 통하여 상호작용하는 컴포넌트를 EJB컴포넌트의 세션빈과 엔터티빈으로 식별한다.

상세절차

STEP 1) Use Case Diagram 작성

- 공용 컴포넌트 추출

Use Case Diagram에서 <<include>>관계는 또 다른 Use Case을 사용할 수 있다. 이<<include>>Use Case에서 제공하는 공통서비스를 하나의 컴포넌트로 제공할 수 있다. 또한 Use Case Diagram에서<<extend>> 유즈케이스가 제공하는 optional 서비스를 하나의 컴포넌트로 제공할 수 있다. <<Extend>>관계는 기본 유즈케이스에 따른 유즈케이스의 행동이 선택적으로 포함되어 있는 경우이다.[3]

STEP 2) 시스템컴포넌트의 식별

(1)외부 인터페이스 식별

요구분석단계에서 생성된 유즈케이스는 사용자 관점에서 시스템이 사용자에게 제공하는 기능을 이미 독립적으로 분류하여 정의한 것이다. 따라서 하나의 유즈케이스마다 서비스 가능한 하나의 외부 인터페이스를 식별할 수 있다. 컴포넌트는 하나이상의 인터페이스로 구성되어 있으며 사용자는 이 인터페이스를 통해 서비스를 제공받게 된다.

(2) 유즈케이스의 정련

식별된 외부인터페이스의 정련을 위하여 유즈케이스 시나리오, 즉 이벤트의 흐름에 따라 분석클래스를 식별하고 유사기능의 유즈케이스를 그룹화한다. 유즈케이스와 추출된 분석클래스의 관계테이블을 구성하여 클래스의 공유관계를 파악하여 유즈케이스를 재 그룹화 한다. 그룹화된 유즈케이스를 시스템 컴포넌트로 식별하며 J2EE플랫폼에서 세션빈으로 모델링된다.

STEP 3) 비즈니스 컴포넌트식별: 타입다이어그램 단계

(1) 공통 <<Entity>> Class 추출: 각 유즈케이스마다 Class Type Diagram을 작성하고 공통 <<Entity>> Class를 추출한다. 이 Class는 외부 UI와 연관이 없는 내부 인터페이스로 작용하는 하나의 Entity Bean Component로 식별된다.

STEP 4) 컴포넌트 다이어그램 작성

STEP 5) 상호작용 다이어그램 작성: 인터페이스의 정련 추출된 컴포넌트 상의 오퍼레이션과 오퍼레이션간의 흐름을 추적하여 상호작용 분석을 통한 의존성을 제거하는 컴포넌트를 통합, 분할하는 정제 과정을 거친다.

5. 결론

기존방법들은 각 작업간의 관계와 추출방법의 명확한 지침이 부족하며 컴포넌트를 추출하기 위한 추상적인 프로세스를 제시하고 있다. 본 연구에서는 J2EE플랫폼을 기반으로 하는 분석 초기단계에서 EJB 컴포넌트 추출을 지원하는 실용적이고 구체적인 방법을 제시하였다. 일반적으로 추출된 컴포넌트 내부 클래스들간의 결합도(coupling)는 높아야 하며 컴포넌트간의 결합도는 낮아야 한다. 내부 클래스들간의 결합도가 높다는 것은 컴포넌트를 구성하는 클래스들간의 응집도(Cohesion)를 측정하는것과 같은 의미가 된다. 아울러 추출된 컴포넌트의 품질을 평가하기위하여 기존에 제시된 컴포넌트 추출 매트릭스 방법[1][7]에 적용하여 역공학적인 측면이 상호 보완적 방법이 될 수 있으리라 제안하며 향후연구 과제로서 식별된 컴포넌트 품질을 평가하는 주요한 기준으로서 향후 추출된 컴포넌트 모델에 대한 품질평가 매트릭스에 관한 연구를 할 예정이다.

참고문헌

[1]Byung-Sun Ko, Jai-Nyun Park. Measuring Cohesion of a component, The KIPS Transactions D August 2002
 [2]Desmond Francis D'souza. Objects, Components, and Frameworks with UML, Addison-Wesley, 1998
 [3]Geri Schneider, Applying Use Cases, Addison-Wesley, 1999
 [4]John Cheesman. UML Components, Addison-Wesley,2000
 [5]Len Bass. Software Architecture in Practice, Addison-Wesley, 2003
 [6]Martin Fowler, "Patterns of Enterprise Application Architecture", Addison-Wesley ,2003
 [7]Misook Choi, Jai-Nyun Park Study about Component Identification Method Based on RUP, The KIPS Transactions D vol 9 ,Korea. 2002
 [8]Paul Clements. Documenting Software Architectures, Addison-Wesley, Feb 2004
 [9]Peter Eeles, " Building J2EE Applications with the rational Unified Process, 2003
 [10]Rational Unified Process, <http://www.rational.com>
 [11]Sterling Software, The CBD96 Standard Version 2.1: Standards for Specifying & Delivering Software Components Using COOL, Sterling Software Ind., 1998

부록

본연구의 결과물로 J2EE플랫폼의 IT Shopping Mall을 구축하였으며 실사례의 모델링은 발표슬라이드상에 발표할예정이다.