EJB 컴포넌트 조립을 위한 MDA 기반 접근 방법*

최유희[O] 권오천, 신규상
한국전자통신연구원 컴퓨터소프트웨어연구소
{yhchoi[O], ockwon, gsshin}@etri.re.kr

An Approach to Composition of EJB Components Based on MDA

You-Hee Choi[O] Oh-Cheon Kwon, Gyu-Sang Shin
ETRI-Computer & Software Research Laboratory

요 약

이종의 컴포넌트 개발 플랫폼 및 다양한 구현 기술의 발달로 이들간의 상호 운용성 및 통합을 지원하기 위하여 OMG에서 채택한 MDA(Model Driven Architecture) 기반 소프트웨어 개발이 대두되고 있다. MDA 기반 소프트웨어 개발 방법에서 기본적으로 고려하고 있는 것은 새로운 컴포넌트 생성을 초점으로 하고 있어 이미 개발되어 있는 컴포넌트의 조립에 대한 고려도 요구된다. 따라서 본 논문에서는 MDA의 중심 개념인 플랫폼 독립 모델(PIM: Platform Independent Model)과 플랫폼 종속 모델(PSM: Platform Specific Model)의 분리와 매핑을 통한 변환에 대해 조립하고자 하는 EJB 컴포넌트를 고려하여 플랫폼 독립 모델을 생성하고 이를 플랫폼 종속 모델로 변환하는 방법을 제안한다.

1. 서론

현재 새롭게 대두되고 있는 OMG의 MDA[1]는 플랫폼 독립 모델(PIM)과 플랫폼 종속 모델(PSM)로 분리하여 시스템 모델을 작성하고 각 모델은 매핑을 통해 상호 변환을 하여 플랫폼 종속 모델을 코드로 자동 변환해 줌으로써 소프트웨어를 개발하는 방법을 지원한다[2]. MDA를 지원하는 도구는 ArcStyler[3], OptimalJ[4] 등 여러 도구가 있다. 이러한 도구들이 지원하는 소프트웨어 개발 절차는 기본적으로 시스템의 비즈니스 모델링을 수행하고 이를 바탕으로 PIM을 작성하고 작성된 PIM은 원하는 플랫폼에 따라 PSM으로 변환하고 코드를 생성해 주는 절차를 따른다.

그러나 현재 CORBA, .NET, J2EE 등의 다양한 컴포넌트 플랫폼의 출현과 컴포넌트 기반 개발 방법의 도입에 따라 각 플랫폼에 종속적인 컴포넌트들이 이미 다수 개발되어 있는 상황에서, 이미 존재하는 컴포넌트의 조립 또는 재사용을 통한 소프트웨어 개발에 많은 관심을 가지고 있다. 따라서 새로운 컴포넌트를 생성하여 소프트웨어를 개발하는 것을 중점으로 한 MDA 기반 소프트웨어 개발 방법을 재사용 컴포넌트의 조립을 통한 소프트웨어 개발에 적용하기에는 어려움이 있다.

따라서 본 논문에서는 MDA 기반 소프트웨어 개발 방법에서 가장 핵심이 되는 부분인 PIM을 작성하고 이를 변환하여 PSM으로 생성하는 과정과 관련하여 EJB 컴포넌트 조립을 위한 PIM을 작성하는 방법과 이를 PSM으로 변환하는 방법에 대하여 제안하고자 한다.

2. 관련 연구

2.1. MDA 기반 소프트웨어 개발 프로세스

MDA를 기반으로 한 기본적인 소프트웨어 개발 프로세스는 PIM을 생성하고 이를 원하는 플랫폼에 따라 PSM으로 변환하고 변환된 모델을 바탕으로 코드를 생성한다[5]. PIM 생성 과정에서는 시스템을 플랫폼 및 구현 기술에 독립적으로 모델링하는 작업을 수행하게 되며 플랫폼 관련 내용을 제외하고 최대한 상세하게 모델링한다. 또한 자동 코드 생성의 이점을 얻기 위해 AL(Action Language), OCL(Object Constraint Language)을 이용하여 비즈니스 로직에 대한 코드를 생성할 수 있도록 한다. PSM 생성 과정에서는 PIM에 대해 각 플랫폼에 해당되는 UML 프로파일을 바탕으로 변환 규칙을 적용하여 PSM을 얻는다. 또한 PSM에는 구현 언어, 미들웨어 종류, 컴포넌트 아키텍처 등 시스템의 실제 구현과 운영에 직접으로 관련 된 정보들이 반복적인 모델 정제 작업을 통해서 기술된다.

MDA 기반 소프트웨어 개발시 PIM에서 PSM으로 변환하는 작업과 PSM 으로부터 코드를 생성하는 작업을 ArcStyler나 OptimalJ 등의 MDA 지원 도구를 이용하여 수행하도록 하는 데에 이점이 있다. 그러나 ArcStyler나 OptimalJ 등의 도구들이 지원하는 기본적인 소프트웨어 개발 프로세스에서는 이미 개발된 컴포넌트에 대한 조립이 전체 과정에서 고려되지 않았고, 이로 인해 이미 개발된 컴포넌트를 고려한 모델링 과정에서는 컴포넌트의 내부 상세 PIM 모델링과 같은 불필요한 모델링 과정이 존재할 수 있고 초점을 두어야 되는 상호작용 부분에 대한 고려가 부족하다.

3. EJB 컴포넌트 조립을 위한 MDA 기반 접근 방법

3.1 EJB 컴포넌트 조립을 위한 PIM 모델링

*본 연구는 2002년 과학기술부 국가지정연구실 사업으로 수행되었음.

PIM 모델링 과정에서는 시스템의 기술적인 상세내용을 제거한 구조와 기능을 표현하는 것이 목적이며 이에 따라 시스템의 기능 중심으로 시스템의 아키텍쳐를 생성한다. 이 때 이미 개발된 컴포넌트의 조립이므로 컴포넌트 구현과 관련된 각 컴포넌트의 내부 모델링은 고려하지 않고 아키텍쳐 구성 요소들인 조립 대상 컴포넌트들간의 호출 관계를 기반으로 이들간의 상호작용만을 상세하게 모델링한다. 그림 1은 EJB 컴포넌트의 조립을 위한 PIM 모델을 생성하는 과정 및 관련 정보의 관계를 나타낸다.

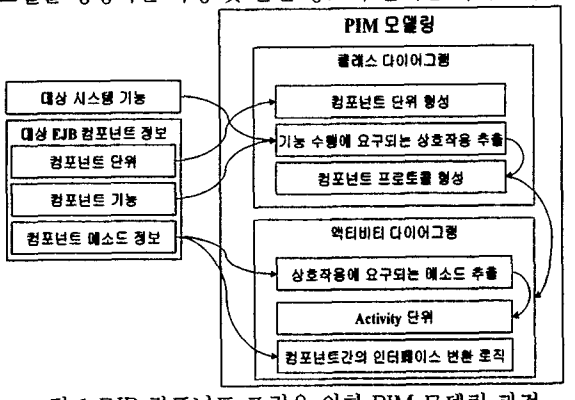

그림 1 EJB 컴포넌트 조립을 위한 PIM 모델링 과정

그림 1의 PIM 모델링 과정에서는 컴포넌트와 컴포넌트간의 상호작용 관계를 나타내기 위해 클래스 다이어그램과 액티비티 다이어그램을 이용한다.

클래스 다이어그램에는 각 컴포넌트와 컴포넌트간의 상호작용을 나타내는 프로토콜과 각 요소들간의 관계 정보가 표현된다. 이 때 조립하고자 하는 컴포넌트의 정보를 반영하여 작성하되 컴포넌트의 구현 플랫폼 및 구현 언어와 관련된 정보가 배제되어 작성된다.

따라서 EJB 컴포넌트의 조립을 위한 PIM 모델링 과정의 클래스 다이어그램에서 우선 고려할 것은 아키텍쳐를 구성할 컴포넌트와 각 컴포넌트의 상호작용이 존재하는지의 여부를 중심으로 모델링하게 된다.

또한 각 컴포넌트의 단위는 조립하고자 하는 컴포넌트의 단위에 따라 결정되고 해당 컴포넌트의 기능을 고려하여 개발하고자 하는 시스템의 기능 수행을 위해 상호작용 관계를 각 컴포넌트간의 프로토콜로 모델링한다.

그리고 각 컴포넌트의 세부 인터페이스는 컴포넌트간의 프로토콜을 상세히 모델링 한 후에 컴포넌트간의 상호작용중에 필요한 메소드들로 구성한다.

따라서 컴포넌트간의 상호작용 관계를 세부적으로 모델링하기 위해 액티비티 다이어그램을 이용한다.

즉 각 프로토콜에 대해 해당 프로토콜에서 다른 컴포넌트의 인터페이스를 사용하는 호출자에 해당되는 컴포넌트와 인터페이스에 대한 구현을 제공하는 응답자에 해당되는 컴포넌트간의 메소드 호출 및 데이터 흐름 정보를 포함하는 액티비티 다이어그램을 작성한다.

각 액티비티의 단위는 조립하고자 하는 EJB 컴포넌트 인터페이스의 메소드 단위가 되고 EJB 컴포넌트의 메소드를 참고로 하여 상호작용을 모델링한다. 이때 사용될 수 있는 메소드는 EJB 컴포넌트가 세션빈인 경우 컴포넌트의 기능과 관련된 메소드는 Remote Interface에 정의되므로 Remote Interface의 메소드 정보를 반영하되 구현 언어와 관련된 정보는 제외시켜 나타낸다. EJB 컴포넌트가 엔티티빈인 경우 Home Interface에도 컴포넌트의 주요 기능이 포함된 메소드가 존재하므로 Home Interface와 Remote Interface로부터 메소드 정보를 추출한다.

액티비티 다이어그램을 작성하면서 각 프로토콜마다 포함된 각 EJB 컴포넌트의 메소드들이 해당 컴포넌트의 인터페이스 메소드로 표현된다. 플랫폼 독립 모델이므로 각 컴포넌트의 인터페이스는 Home Interface와 Remote Interface를 구별하지 않고 하나의 인터페이스로 표현한다.

또한 흐름상에서 상호작용하는 컴포넌트간의 인터페이스가 일치하지 않을 경우 중간 변환 로직에 대한 액티비티를 추가로 모델링한다. 추가로 모델링된 액티비티와 메소드 호출 흐름 정보는 코드 생성 과정에서 각 컴포넌트간의 연결 코드로 생성될 수 있다.

3.2 PIM으로부터 PSM 변환

PSM은 PIM 모델링 단계에서 기술된 각 컴포넌트의 인터페이스 정보와 컴포넌트간의 상호 호출 정보가 플랫폼 및 구현 언어에 맞도록 표현된 다이어그램 형태로 생성된다. 그림 2는 PIM 모델을 PSM 모델로 변환하는 과정 및 관련 정보의 관계를 나타낸다.

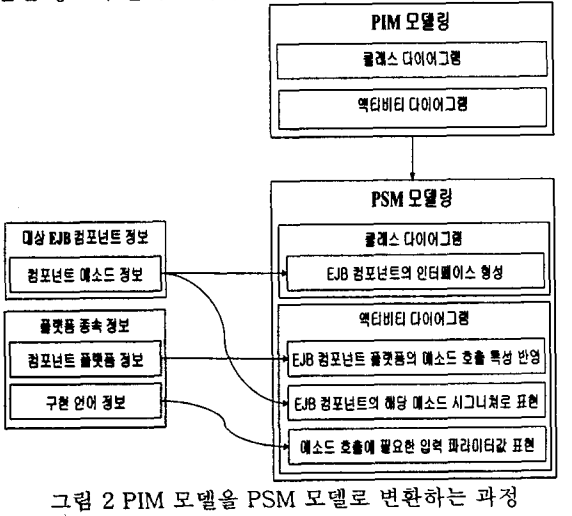

그림 2 PIM 모델을 PSM 모델로 변환하는 과정

그림 2에서 PIM 모델링 단계에서 작성된 클래스 다이어그램은 조립 대상 컴포넌트의 구현 플랫폼 관련 정보가 반영되어 생성된다. PIM의 클래스 다이어그램에서 각 컴포넌트의 인터페이스는 각 컴포넌트간의 상호작용에 의해 필요한 메소드들만 추출되어 작성된 인터페이스이므로 추출된 인터페이스 메소드가 실제 EJB 컴포넌트의 메소드들의 일부가 될 수 있다.

따라서 PSM에서는 해당 메소드들만 EJB 컴포넌트의 인터페이스 메소드로 정의한다. EJB 컴포넌트의 인터페

이스는 Home과 Remote로 나눠지므로 PIM 모델링 과정에서 고려하였듯이 맵핑되는 EJB 컴포넌트가 세션빈일 경우 PIM 모델링 과정에서 추출된 메소드들은 EJB 컴포넌트의 Remote Interface의 메소드로 반영한다. EJB 컴포넌트가 엔티티빈인 경우에는 Finder와 Create 메소드는 Home Interface의 메소드로 반영하게 되고 그 외의 메소드는 Remote Interface로 반영하여 클래스 다이어그램을 생성한다.

컴포넌트간의 상호작용 관계를 기술한 액티비티 다이어그램은 구현 플랫폼 정보가 배제된 컴포넌트간의 오퍼레이션 단위의 호출 흐름 정보이다. 따라서 작성된 메소드 호출 정보에 대해서는 조립 대상 컴포넌트의 구현 플랫폼 기술이 반영된 메소드 호출 정보로 변환한다. 이 때 각 구현 플랫폼의 컴포넌트간의 메소드 호출 특성상 필요한 추가적인 메소드 호출 정보를 반영하여 PIM 모델링 단계에서 작성된 메소드 호출 흐름 정보를 수정하여 생성한다. EJB 컴포넌트의 경우 원하는 메소드를 호출하기 위해 필요한 기본적인 과정인 Home 객체를 얻어와서 Create 메소드를 수행하여 Remote 객체를 얻어오는 과정이 액티비티 다이어그램에 반영될 수 있다. 또한 메소드 호출 순서 정보를 중심으로 작성된 모델에 대해 PSM 모델링 과정에서는 코드 생성을 고려하여 메소드 호출에 필요한 입력 파라미터값에 해당되는 정보를 상세히 표현한다. 그리고 컴포넌트간의 인터페이스 불일치 문제로 인해 추가된 액티비티에 대해 상세히 정제한다. 요구되는 입력 파라미터값에 대해 이전 액티비티의 메소드에서 제공하는 파라미터 정보가 다를 경우 이를 변환하는 로직이 추가되어 표현된다. 이러한 매핑 정보는 코드 생성 과정에서 파라미터간의 매핑 코드로 생성될 수 있다.

4. 적용

본 장에서는 간단한 쇼핑몰 예제에 적용한다. 조립하고자 하는 EJB 컴포넌트로는 상품 정보를 관리하는 Stock, 주문 정보를 관리하는 Order, 주문자 정보를 관리하는 Member 컴포넌트가 있다.

4.1 PIM 모델링

조립하고자 하는 컴포넌트간의 호출 관계를 기반으로 클래스 다이어그램과 액티비티 다이어그램을 작성한다.

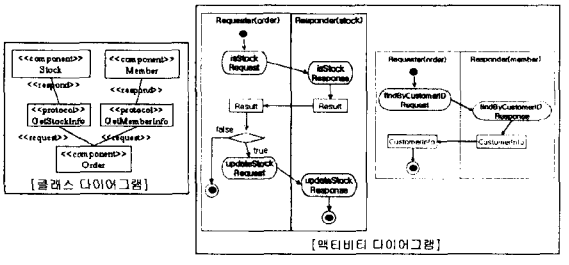

그림 3 PIM 모델링 과정

그림 3에서 클래스 다이어그램은 상품 주문 기능을 수행하기 위해 상호작용이 필요한 컴포넌트의 관계를 나타낸 것으로 Order와 Stock 컴포넌트의 관계는 주문하고자 하는 상품의 재고가 충분한지 확인하기 위한 기능을 수행하기 위해 필요하고 Order와 Member 컴포넌트의 관계는 주문자의 정보가 올바른가를 확인하고 주문자 정보를 얻어 오기 위해 필요하여 프로토콜로 연결된 것을 나타낸다. 액티비티 다이어그램은 각 프로토콜을 정제하여 나타낸 것이다.

4.2 PIM으로부터 PSM 변환

PIM 모델링 과정에서 작성된 클래스 다이어그램과 액티비티 다이어그램을 PSM으로 변환한다. 그림 4는 플랫폼 정보와 EJB 컴포넌트의 정보가 반영된 액티비티 다이어그램과 프로토콜을 바탕으로 추출된 메소드들로 구성된 컴포넌트의 인터페이스를 보여주는 클래스 다이어그램이다.

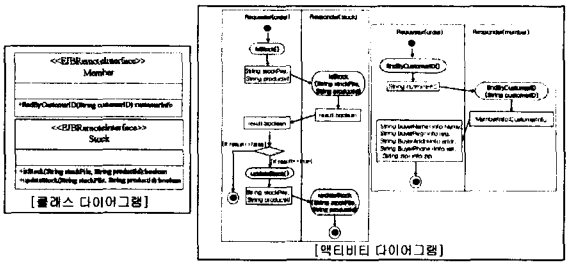

그림 4 PIM으로부터 변환된 PSM

5. 결론

본 연구에서는 EJB 컴포넌트의 조립을 위하여 작성되는 PIM의 모델링 과정과 모델링된 PIM을 조립 컴포넌트의 정보와 플랫폼 정보를 고려하여 PSM으로 변환하는 방법을 제안하였다. 본 연구를 통해 기존의 MDA 기반 소프트웨어 개발 절차에 EJB 컴포넌트의 조립을 통합하여 효율적인 소프트웨어 개발을 가능하게 한다.

향후에는 좀더 실질적인 다양한 적용을 통한 보완과 EJB 컴포넌트뿐만 아니라 다양한 플랫폼 종속 컴포넌트의 조립을 고려한 PIM 모델링 방법과 모델링된 PIM을 PSM으로 변환하는 방법에 대한 연구가 필요하다.

6. 참고 문헌

[1] OMG Architecture Board, "Model Driven Architecture", http://www.omg.org/mda.

[2] John D. Poole, "Model-Driven Architecture: Vision, Standards And Emerging Technologies" ECOOP 2001, April 2001.

[3] "ArcStyler Tools Guide Reference Manual" http://www.ArcStyler.com.

[4] Compuware, "Using OptimalJ", http://javacentral.compuware.com.

[5] Jon Siegel and the OMG Staff Strategy Group, "Developing in OMG's Model Driven Architecture", ftp://ftp.omg.org/pub/docs/omg/01-12-01.pdf, November 2001.