

EDOC 기반 PIM 변환기의 설계 및 프로토타입 개발*

김민정⁰ 윤석진 신규상
한국전자통신연구원 컴퓨터소프트웨어연구소
{minjkim⁰, sjyoon, gsshin}@etri.re.kr

Design and Development of PIM Transformer Based on EDOC

Minjeong Kim⁰ Seokjin Yoon Gysang Shin
ETRI-Computer & Software Technology Laboratory

요 약

OMG에서 제안한 MDA는 시스템의 설계와 명세를 기술 플랫폼 독립적인 모델인 PIM으로 작성하고, 매핑을 통해서 실제 구현과 관련된 모델인 PSM을 만들어냄으로써 시스템을 보다 효율적으로 유지 및 통합할 수 있다. 이를 지원하기 위해 본 논문에서는 EDOC 프로파일 기반의 PIM 변환기를 제안하여 메타 모델 단위의 매핑을 통한 PSM으로 변환 과정의 자동화를 지원하고자 한다. 변환기의 입력인 PIM은 분산 컴퓨팅 환경에 대해 OMG에서 채택한 EDOC 프로파일을 적용하였으며, 매핑 결과인 PSM은 UML Profile for EJB를 적용하였다. 또한 자동화된 PIM 모델 변환을 위해 EDOC to J2EE/EJB 매핑 규칙을 정의하였으며, 이를 지원하는 도구를 구현하여 회의실 예약 시스템 예제에 적용시켜 검증했다.

1. 서 론

OMG는 지금까지 추진한 여러 표준화 작업을 바탕으로 모델 중심의 시스템 명세와 개발 방법인 MDA(Model Driven Architecture)[1]를 제안하여 특정 기술에 기반한 시스템을 기술 변화에 맞게 통합, 변화, 유지 가능하도록 지원하고자 하였다. MDA 접근 방법은 시스템의 설계와 명세를 기술 플랫폼 독립적인 모델인 PIM(Platform Independent Model)으로 작성하고, 매핑을 통해서 실제 구현과 관련된 모델 PSM(Platform Specific Model)을 만들어냄으로써 시스템을 보다 효율적으로 유지 및 통합할 수 있다[2].

따라서, MDA를 지원하는 도구는 PIM 수준에서 특정 도메인에 일반적인 요소들을 표준화 된 방법으로 표현할 수 있어야 하며, 검증이 완료된 PIM을 플랫폼에 따른 PSM으로 변환하기 위한 기능을 제공해야 한다.

본 논문에서는 MDA를 지원하기 위한 PIM 변환기를 제안하여 PIM의 PSM으로 변환(PIM2PSM) 과정의 자동화를 지원하고자 한다. 변환기의 입력인 PIM은 분산 컴퓨팅 환경에 대해 OMG에서 채택한 UML Profile for EDOC(Enterprise Distributed Object Computing)[3]을 적용하였으며, 매핑 결과인 PSM은 UML Profile for EJB를 적용하였다. 또한 자동화된 PIM 변환을 위해 EDOC to J2EE/EJB 매핑 규칙을 정의하였으며 메타모델 단위의 모델 변환을 지원하는 도구를 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 PIM 변환기 관련연구에 대해 살펴봄, 3장에서는 PIM 변환기의 전체 구조를 설명한다. 변환을 위한 매핑 규칙은 4장에서 기술되며, 개발된 프로토타입을 이용하여 사례에 적용한

결과는 5장에서 기술되고, 6장에서 결론 및 향후 연구에 대해 기술 한다.

2. 관련 연구

MDA 기반 모델 변환을 지원하는 대표적인 도구로는 Interactive Objects Software사의 ArcStyler와 Compuware사의 OptimalJ가 있다. ArcStyler의 경우 분석된 비즈니스 모델 정보를 기반으로 PIM을 생성하고 PIM 단계에서 플랫폼 정보 및 전개에 필요한 정보 등을 입력하여 골격 코드를 자동으로 생성시키는 기능을 제공한다. OptimalJ의 경우 클래스와 서비스로 구성된 도메인 모델을 PIM으로 간주하고 이로부터 패턴 기반 접근방식을 통해 J2EE 플랫폼 기반의 PSM을 자동으로 생성한다. 그러나 이들은 PIM 단계에서 컴포넌트 기반의 분산 환경에 대해 고려하고 있지 않으며, PIM2PSM 변환과정에서 메타 모델 매핑에 대한 고려가 생략되어 일반적으로 적용하기엔 한계가 있다.

현재 OMG의 MDA를 기반으로 RM-ODP(Reference Model of Open Distributed Processing)와 UML for EDOC을 채택한 COMBINE(COMponent-Based INteroperable Enterprise system development) 프로젝트가 Opengroup을 중심으로 수행되고 있으며, 이에 따른 결과가 2003년에 발표될 것이다.

3. PIM 변환기의 구조

MDA를 기반으로 한 개발 프로세스는 다음과 같은 단계를 통해서 이루어 진다.

* 본 연구는 2002년 과학기술부 국가지정연구실 사업으로 수행되었음.

- PIM 생성 단계: 요구사항 분석과 비즈니스 모델링 작업이 완료된 이후에 비즈니스 기능과 시스템의 기능들이 구현 기술 독립적으로 표현 된다. 반복적인 모델 정제 작업과 OCL(Object Constraint Language) 기술을 통해서 최대한의 정합성과 일치성을 가지는 모델을 얻는다.
- PSM 생성 단계: 완성된 PIM에 정해진 변환규칙을 적용함으로써 PSM을 얻는다. PSM은 구현 환경을 고려한 모델이기 때문에 이 단계에서는 구현 언어, 미들웨어 종류, 컴포넌트 아키텍처 등 시스템의 실제 구현과 운영에 직접적으로 관련된 정보들이 반복적인 모델 정제 작업을 통해서 기술된다.
- 코드 생성 단계: PSM으로부터 코드를 자동으로 생성한다. PSM 레벨에서 이미 구현에 관련된 대부분의 정보가 포함되어 있기 때문에 이 단계는 PIM, PSM 단계에 비해서 많은 작업이 필요하지 않다.

이러한 개발 프로세스를 기반으로 PIM을 초기 PSM으로 변환시키는 기능을 하는 PIM 변환기의 전체 구조는 (그림 1)과 같다. 모델 변환은 메타 모델 단위로 이루어지게 되며 메타 모델들은 MOF 기반 Repository에 관리된다. 또한 메타 모델들 사이의 매핑 규칙은 입/출력 프로파일에 따라 분리되어 모델 매핑 관리기에 의해 저장된다.

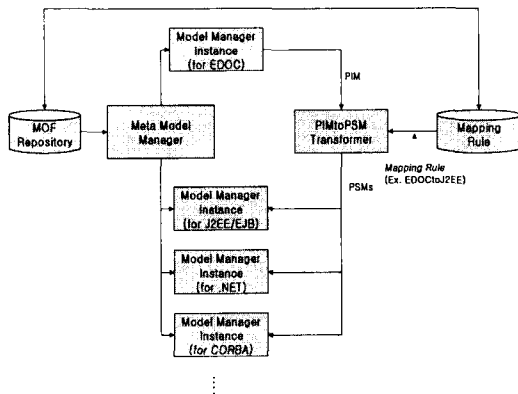


그림 1. PIM 변환기의 구조

PIM 변환기를 구성하는 주요 블록의 기능은 다음과 같다.

- 메타 모델 관리기(Meta Model Manager): MOF에 기반한 메타 모델을 생성하고 메타 모델의 인스턴스 모델을 관리할 수 있는 모델 관리기를 생성한다.
- 모델 정보 관리기(Model Manager)는 메타 모델 관리기에서 정의된 메타 모델의 인스턴스인 모델들을 생성, 관리, 참조한다. 프로파일의 종류에 따라 다양한 모델 정보 관리기의 인스턴스가

생성된다.

- PIM2PSM 변환기(PIM2PSM Transformer): 입/출력 프로파일에 따라 메타 모델들 간에 정의된 매핑 규칙을 기반으로 모델 매핑 엔진을 생성한다. 모델 매핑 엔진은 정의된 매핑 규칙을 기반으로 PIM을 입력으로 받아 메타 모델 레벨의 매핑을 수행한 결과인 PSM으로 변환한다.

4. PIM 변환 방법

현재 개발되는 대부분의 시스템이 분산환경에 기반을 두고 있으며 컴포넌트의 Collaboration 뿐만 아니라 Entity 등을 효율적으로 나타낼 수 있기 때문에 본 논문의 PIM으로 EDOC을 채택하였다. EDOC의 Information Viewpoint에서 사용된 Entity, Relationship Profile과, Computational Viewpoint에서 사용된 CCA를 EJB 매핑하는 방법에 대해 살펴보도록 하겠다.

4.1 EDOC CCA Profile의 J2EE/EJB 매핑

EDOC CCA Profile의 J2EE/EJB 매핑과 컴포넌트 간에 흐르는 정보를 나타내는 Document Model의 매핑은 (표 1)을 기반으로 수행된다.

표 1. Process Component와 Protocol, Document Model의 매핑

Metamodel Element Name	EJB
ProcessComponent	EJBSessionBean 또는 EJBEntityBean으로 매핑될 수 있다. 상위 개념인 Business Process나 하위 레벨 개념인 Object로도 매핑될 수 있다.
Port	External view의 경우 <<EJBRemoteInterface>> Internal view의 경우 <<EJBRealizeRemote>>와 <<EJBRealizeHome>>을 가진 <<EJBImplementation>>으로 매핑된다.
Direction (Port의 속성)	Initiates: 보내는 메시지로 매핑된다. Direction Responds: 받는 메시지로 매핑된다. Synchronous의 경우 인터페이스 내에 리턴값이 없는 operation으로 매핑되며, asynchronous의 경우 받는 event로 매핑된다.
DataType	MOF 1.4에 의해 정의된 규칙에 따라 매핑된다.

ProcessComponent의 경우 일반 클라이언트 컴포넌트로 변환될지 <<EnterpriseBean>>으로 변환될지 자동적인 구분이 어려우므로 Tier 별로 구분하여 매핑되어야 한다. 예를 들어, 사용자와의 interaction을 담당하는 Client

인 경우 일반 java model로 변환시키고, Server측인 경우 <<EJBSessionBean>>으로 매핑시킨다.

4.2 Entities Profile의 J2EE/EJB 매핑

Entity Profile에 기술된 메타모델은 (표2)와 같이 매핑될 수 있다.

표2. Entity Profile의 매핑

Metamodel Element Name	EJB
Data Manager	EJBEntityBean 또는 stateful EJBSessionBean
Sharable	True일 경우 DataManager를 EJBEntityBean으로 매핑한다.
Entity Data	EJB EntityBean의 data representation 부분으로 매핑된다.
Key	EJBPrimaryKey

5. 사례 적용

기술된 매핑 규칙을 기반으로 PIM 변환기인 COBAL-Transformer의 프로토타입을 개발하였으며, EDOC PartII[4]의 예제인 MRS(Meeting Reservation System)에 적용시켜 보았다. 이 시스템은 특정 시간에 이루어지는 회의에 방이나 장비 같은 자원을 할당하고, 참석자에게 참석을 요청하기 위한 시스템이다.

EDOC Profile이 적용된 PIM의 Component Structure에는 MRS에서 식별된 <<ProcessComponent>>인 *MeetingServationTool*, *MeetingResponseTool*, *ResourceAdministrationTool*, *ResourceManager*, *ReservationManager* 컴포넌트들을 중심으로 하는 구조가 외부와의 interaction을 위해 정의된 <<Protocol>>과 함께 표현된다. Entity View에는 <<Entity>>로 식별된 *ResourceComponent*와 *ReservationComponent*가 이들이 사용하는 데이터인 <<EntityData>>, <<Key>>와 함께 표현된다.

PIM에서 모델링된 Component Structure와 Entity View를 앞서 기술된 매핑룰을 구현한 PIM 변환기에 의해 PSM으로 변환되었으며, 그 결과는 (그림 2)와 같다.

PIM을 매핑한 결과를 살펴보면 Client-Tier에 속하는 <<ProcessComponent>>인 *MeetingServationTool*, *MeetingResponseTool*, *ResourceAdministrationTool*이 일반 Java로 변환되었으며, Server-Tier에 속하는 *ResourceManager*, *ReservationManager*는 SessionBean으로 변환되고 <<responds>> 관계에 있는 <<ProtocolPort>>와 <<Protocol>>의 내용은 RemoteInterface에 반영되었다. 또한 Entity View에서 모델링된 <<Entity>>인 *ResourceComponent*와 *ReservationComponent*의 경우 EntityBean으로 변환이 되었으며 <<EntityData>>는 내부 데이터로 <<Key>>는 PrimaryKey 클래스로 매핑되었다.

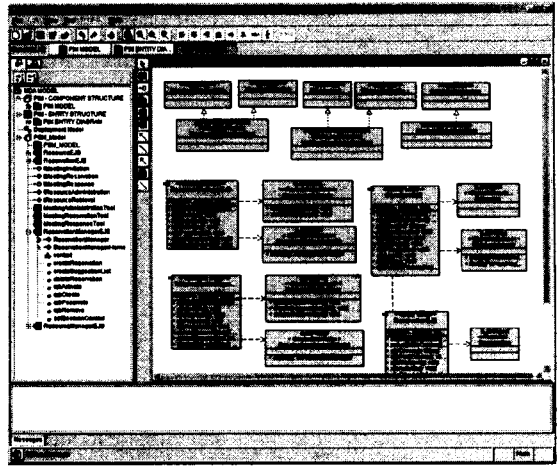


그림 2. COBAL-Transformer에 의한 PIM 변환 결과 - MRS의 예

6. 결론 및 향후 연구

지금까지 메타모델 단위의 모델 변환을 지원하는 EDOC 기반 PIM 변환기에 대한 내용을 살펴보았다. 본 논문에서는 PIM 모델 변환기를 제안하여 PSM으로 변환 과정의 자동화를 지원하였다. 변환기의 입력인 PIM은 UML Profile for EDOC을 적용하였으며, 매핑 결과인 PSM은 UML Profile for EJB를 적용하였다. 또한 자동화된 PIM2PSM 변환을 위해 EDOC to J2EE/EJB 매핑 규칙을 정의하였으며 이를 지원하는 도구를 구현하였다.

향후 연구로는 다양한 예제 적용을 통해 매핑룰을 검증해야 하며, 구조적인 측면의 매핑뿐만 아니라 Activity Spec과 같은 동적인 측면에 대한 변환의 자동화를 추가해야 하고, CORBA나 .NET과 같은 다양한 플랫폼으로 확장시켜 각종 메타 모델 간의 매핑을 정형화 하여 지원해야 한다.

- [1] Object Management Group, "Model Driven Architecture", OMG document ormsc/01-07-01.
- [2] 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
- [3] Object Management Group, UML Profile for Enterprise Distributed Object Computing Specification, OMG Adopted Specification ptc/02-02-05
- [4] Object Management Group, UML for Enterprise Distributed Object Computing Joint Final Submission Part II Supporting Annexes v0.1, July 2001