

# MOF 메타모델을 이용한 EDOC 다이어그램 생성

정양재<sup>0</sup> 신규상  
한국전자통신연구원  
{yjjung<sup>0</sup>, gsshin}@etri.re.kr

## Generating Diagram for EDOC Using MOF--Meta Model

Yang Jae Jeong<sup>0</sup> Gu Sang Shin  
Dept. of Office, Electronics and Telecommunicatins Research Institute

### 요 약

모델을 중심으로 하는 MDA 개발 환경에서는 많은 모델링이 필요하다. 기본적으로 MDA의 기본 단계인 CIM, PIM, PSM 단계의 모델링이 필요하며, 각 단계에서 추상화 수준, 뷰포인트 등에 따라 다양한 다이어그램이 필요하다. 현재 MDA의 PIM과 PSM을 상대적인 개념으로 인식하기 때문에 MDA의 단계는 더욱 확대될 것이다. 본 논문에서는 분산 환경에서 컴포넌트 기반 엔터프라이즈 시스템을 개발하기 위한 EDOC 프로파일을 중심으로 MOF 기반 모델링을 수행하는 방법과 프로토타입을 기술한다. OMG의 제안한 메타모델 표준인 MOF를 이용하므로 JMI, MOF XMI, MOF IDL 매핑을 이용할 수 있는 장점을 갖게 된다.

### 1. 서 론

OMG에서는 새로운 시스템 개발 방법으로 MDA(Model Driven Architecutre)[1]를 제안하였다. MDA의 여러 개발 단계 중 핵심은 플랫폼 독립적인 모델(PIM:Platform Independent Model)과 플랫폼 종속적인 모델(PSM:Platform Specific Model)이다. PIM 단계에서 OCL, ASL 등의 추가적인 입력을 통해 구현에 필요한 최대한의 정보를 입력한 후 PSM으로 변환하는 방법을 통해 한번의 모델링을 통해 모든 플랫폼에서 실행 가능한 구현을 얻도록 한다.

MDA의 이름에서 알 수 있듯이 MDA의 가장 중심은 모델이다. OMG에서는 모델 정의를 위하여 MOF(Meata Model Facility)를 사용한다. MOF는 M0에서 M3까지 4단계의 메타모델 구조를 가지고 모델을 정의한다. 일반적으로 사용되는 UML과 같은 모델은 M1 단계의 모델이며 MOF를 이용하여 M2 단계의 메타 모델을 정의하고, 메타 모델의 정의에 사용되는 MOF를 정의한 내용이 M3 단계의 *the MOF*이다. PIM 및 PSM 단계에서 MOF를 정의한 모델들을 사용하므로 MOF XMI를 이용하여 표준 XML 스키마를 얻을 수 있고 MOF IDL 매핑을 통해 표준 코바 코드를 얻을 수 있는 장점을 갖게 된다. 현재 널리 사용되는 UML1.4는 MOF와 완벽히 호환할 수 있는 단계가 아니며 UML2.0에서 MOF와 합쳐질 예정이다. 많은 소프트웨어 개발툴에서 MDA를 지원하고 있지만, 현재는 UML 프로파일을 이용하여 PIM, PSM 단계를 정의하고 있기 때문에 MDA의 개념을 완벽히 살리지 못하고 있다. 본 논문에서는 MOF를 이용하여 메타모델을 정의하고 그에 대한 다이어그램 생성하는 것을 기술한다. 분산 환경의 컴포넌

트 기반 엔터프라이즈 시스템을 개발하기 위해 정의한 EDOC 프로파일을 이용하여 MOF 모델을 모델링 하는 다이어그램을 생성하는 프로토타입에 대해 기술한다.

### 2. 연구배경

#### 2.1 MOF(Meta Object Facility)[2]

MOF는 OMG에서 정의한 메타모델 구조로서 4단계의 레벨로 이루어졌다. MOF에 대한 메타모델은 MOF를 이용하여 정의하기 때문에 MOF 구조에서 더 이상의 레벨은 필요하지 않다. MDA의 환경에서의 다양한 모델들에 대해 MOF가 중요한 이유는 MOF를 중심으로 XML을 위한 MOF XMI, 코바 IDL을 위한 MOF IDL 매핑이 정의되어 있으며, MOF 리파지토리를 위한 JMI, IDL 등의 표준 인터페이스 등이 정의되어 있어 다른 표준들과 상호작용이 쉽게 이루어진다. 또한 CWM의 메타모델에도 MOF가 포함으로 DB와의 통합도 용이하다.

#### 2.2 EDOC 프로파일[3]

EDOC 프로파일의 목적은 분산 컴포넌트 기반의 EDOC 시스템의 모델링 프레임워크를 제공하여 EDOC 시스템의 개발을 쉽게 하는 것이다. EDOC 프로파일은 UML 1.4와 OMG의 MDA 표준을 따르며 플랫폼 독립적인 모델링 방법을 제공하고, 다양한 수준의 결합 단계를 모델링 할 수 있는 모델링 방법을 제공한다. EDOC 프로파일은 크게 플랫폼 독립적인 단계에서 아키텍처 내부의 상호관계를 보여주는 ECA(Enterprise Collaboration Architecture) 프

<sup>1</sup> 본 연구는 2002년 과학기술부 국가지정연구실 사업으로 수행되었음

로파일과 ECA 프로파일 내부에서 재사용 할 수 있는 Pattern profiles 과 EJB 등과 같은 기술 종속적인 프로파일로 이루어졌다. 특히 플랫폼 독립적인 모델의 중심이 되는 ECA가 EDOC의 중심이라고 할 수 있으며 다음 다섯 개의 서브 프로파일로 구성되어 있다.

- CCA 프로파일 : 시스템을 구성하는 컴포넌트 사이의 구조와 행위를 다양한 추상화 수준에 따라 보여준다.
- Entities 프로파일 : 문제 영역 내의 개체를 엔티티로 모델링한다.
- Events 프로파일 : 이벤트 중심의 시스템을 모델링한다.
- Business 프로파일 : 시스템이 지원하는 행위를 모델링한다.
- Relationships 프로파일 : 소프트웨어를 모델링 하는 발생하는 관계를 자세히 모델링 한다.

3. EDOC 메타 모델과 다이어그램

3.1 EDOC 모델 생성

MOF 메타모델을 이용하여 정의된 모델을 저장할 수 있는 리파지토리를 MOF 리파지토리라고 한다. 일반적으로 MOF 리파지토리는 IDL이나 JMI[4]를 표준 인터페이스로 사용한다. 본 논문에서는 그림 1과 같이 nsMDF[5]를 이용하여 모델 저장소를 생성한다. Rose를 이용하여 profile for MOF를 사용하는 다이어그램을 생성한 후 Rose의 플러그인 중의 하나인 Unisys XMI exporter를 이용하여 메타모델에 대한 XML DTD를 생성한다. nsMDF는 생성된 XML 파일에 대한 모델 저장소를 생성하며, JMI를 이용하여 모델을 접근한다.

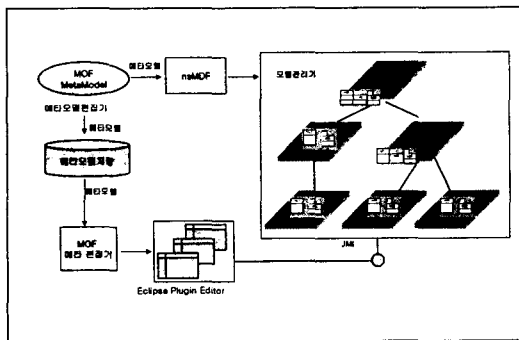


그림 1 MOF 메타모델과 편집기 생성 과정

그림 2는 CCA(Component Collaboration Architecture) 프로파일이다. MOF 전용 다이어그램이 정의되지 않았으므로 profile for MOF를 이용하여 CCA 프로파일을 정의한다. CCA 외에 Entities profile, Events profile, Business Process profile, Relationships profile을 정의한 후 XMI 파일을 생성한다.

3.2 EDOC 다이어그램 정의

EDOC에서 가장 많이 활용되는 프로파일과 다이어그램은 컴포넌트 사이의 관계를 표현하는 CCA 프로파일과 그에 관련된 다이어그램이다. CCA에 대한 다이어그램은 현재 UML 다이어그램을 확장한 형태의 다이어그램을 주로 사용하고 있지만 EDOC 프로파일 스펙에서는 CCA를 위한 표기법을 정의하고 있다. 본 논문을 위한 프로토타입에서는 MOF를 이용한 모델을 사용하고 노테이션은 UML의 노테이션을 이용한다.

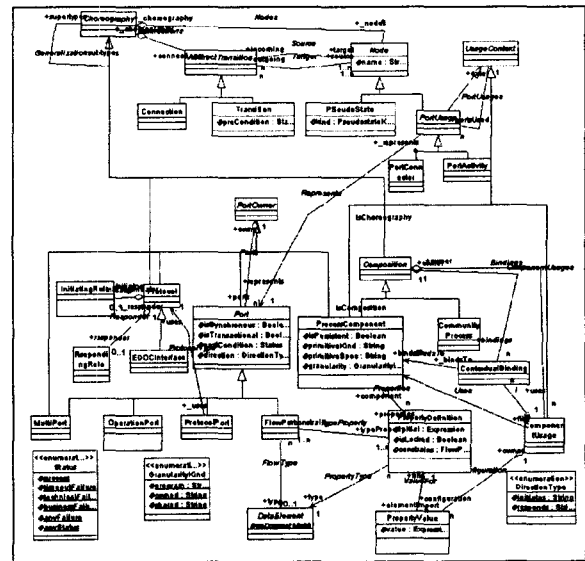


그림 2 CCA 프로파일

표 1 프로토콜의 구조를 보여주는 다이어그램

CCA element	Stereo type	UML Element	MOF Element
Protocol	Protocol	Class	Protocol
FlowPort	FlowPort	Class	FlowPort
Ports	initiates or responds	Association	Ports
ProtocolPort	ProtocolPort	Class	ProtocolPort
OperationPort	OperationPort	Class	OperationPort
InitiatingRole	InitiatingRole with relation to protocol	Class	InitiatingRole
RespondingRole	RespondingRole with relation to protocol	Class	RespondingRole
Interface	Interface	Classifier	Interface
Direction	initiates	Association	value
Direction	responds	Association	value

CCA를 이용하여 문서, 컴포넌트 구조, 컴포넌트들의 조합, 프로토콜의 구조와 상태들을 표현하는 다이어그램을 생성할 수 있다. 표 1은 프로토콜의 구조를 나타내기 위한 다이어그램의 노테이션들을 기술한 표이다. 첫번

째 컬럼은 다이어그램 내의 노테이션이며 두번째와 세번째 컬럼은 UML의 스테레오 타입을 이용하여 다이어그램을 표현할 때의 스테레오 타입이다. 네번째 컬럼은 노테이션과 관련된 MOF 메타 모델들이다. 큰 차이는 없지만 몇가지 차이점이 있다. Direction의 경우 스테레오 타입에서는 연관관계로 표시하지만 MOF에서는 Port 내부의 속성으로 정의되며 *initiates*, *response*를 값으로 갖는 열거형으로 타입이다.

### 3.3 EDOC 다이어그램 생성

그림 3은 CCA를 이용하여 프로토폴의 구조를 표현하는 다이어그램이다. 표1에서 정의된 다이어그램 내용은 XML 파일로 변환하여 다이어그램 정의 정보가 된다. MOF 메타 편집기는 다이어그램 정의에 따라 노드와 라인을 생성하여 ECLIPSE[6]의 플러그인을 생성한다. 다이어그램 편집기에서 노드와 라인을 생성하면 MOF 리파지토리에 는 그에 관련된 모델들이 생성된다. nsMDF는 JMI를 인터페이스로 갖기 때문에 JMI를 이용하여 다이어그램과 모델을 연결한다.

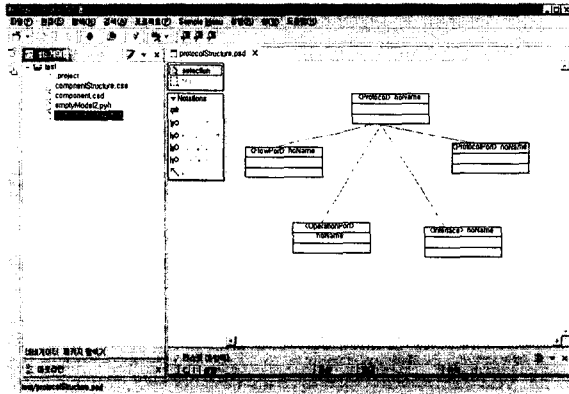


그림 3 프로파일의 구조를 나타내는 다이어그램

### 4. 결론 및 향후 일정

본 논문에서는 MDA의 가장 핵심이 되는 모델을 정의하는 방법과 그에 따른 다이어그램을 생성하는 방법을 기술했다. 현재는 간단한 노테이션만을 정의할 수 있지만 점차 사용자의 요구에 따른 복잡한 노테이션을 표현할 수 있도록 할 예정이다. MDA 개발 환경에서는 모델 뿐 아니라 메타 모델이 아주 중요한 정보로 활용된다. MDA의 기초가 모델이라면 MDA의 성공 여부는 변환에 따라 결정되고 변환할 때 모델과 메타모델이 같이 이용되기 때문이다. 본 논문에 대한 프로토타입에서는 메타 모델에 대한 저장 기능을 구현하지 않았지만, 메타 모델의 저장용 MOF 리파지토리를 그대로 이용할 수 있다. MOF에 대한 정의를 MOF를 이용하여 직접 정의하기 특성에 따라 MOF 리파지토리에 M1, M2, M3의 모든 단계의 모델을 저장할 수 있기 때문이다. 현재 개발된 EDOC 다이어그

램 뿐 아니라, MOF 메타 모델링을 수행하는 다이어그램을 메타 방식으로 생성하면 MOF 리파지토리를 이용하여 모든 단계의 모델을 저장하도록 변경 가능하다.

### 참 고 문 헌

- [1] David S. Frankel, "Model Driven Architecture Applying MDA to Enterprise Computing," Wiley Publishing, 2003.
- [2] Object Management Group, "Meta Object Facility(MOF) Specification version 1.4," April 2002.
- [3] Object Management Group, "UML Profile for Enterprise Distributed Object Computing Specification," 2002.
- [4] Java Community Process. "Java™ Metadata Interface(JMI) Specification version 1.0," 2002.
- [5] nsMDF, <http://nsuml.sourceforge.net>
- [6] Eclipse Project, <http://www.eclipse.org>