

## PIM에서 EJB기반의 PSM으로 변환에 대한 일관성 검증 규칙

김동규<sup>0</sup> 이현정 정기원

송실대학교 대학원 컴퓨터학과

topazz19@hanmail.net<sup>0</sup>, kiwili@empal.com, chong@comp.ssu.ac.kr

### Rules for verifying the consistency of transformation of PIM to EJB based PSM

Dongkyu Kim<sup>0</sup> Hyunjeong Lee, Kiwon Chong

Dept. of Computing, Graduate School, Soongsil University

#### 요 약

MDA 기반 개발을 지원하는 도구를 사용하여 모델 변환을 수행하거나 또는 수동으로 모델을 변환하는 경우 두 모델 간에 변환의 일관성 검증은 반드시 필요하다. 본 논문에서는 현재 개발 지원 도구에서 자동으로 수행되고 있는 MDA 핵심 개념인 모델 변환 가운데 한 가지인 PIM에서 PSM으로의 변환이 일관성을 유지하면서 정확하게 수행되었는가를 검증할 수 있는 기준으로 구성 요소 간의 추적성과 데이터 타입 변환의 정확성을 도출하였다. 이에 따라 모델 간의 일관성을 검증하기 위한 규칙을 추적성과 정확성으로 구분하여 제시한다.

#### 1. 서론

분산 컴퓨팅 환경이 보편화 되고, 사용하는 프로그래밍 언어, 미들웨어 등이 다양해지면서 어플리케이션의 개발뿐 아니라 다양한 어플리케이션 간의 통합 및 상호 연동에 대한 관심이 높아지고 있다. 2001년 OMG(Object Management Group)에서 국제 표준으로 채택한 MDA(Model-Driven Architecture)는 모델단위의 재사용을 통해 어플리케이션의 개발 효율성을 증대시키고, 이종 플랫폼 간의 통합 및 연동을 제공할 수 있는 방법이다.

모델변환은 MDA의 핵심 개념으로 Arcstyler, OptimaJ 등 MDA 기반 개발을 지원하는 여러 가지 도구를 사용할 경우 도구 자체에서 자동으로 수행되지만, 두 모델 사이의 변환이 일관성을 유지하면서 정확하게 수행되었는지 검증하는 기능은 지원되지 않고 있다. 그런데 만약 변환된 PSM(Platform Specific Model)에 오류가 존재할 경우, 그것이 그대로 코드에 반영되기 때문에 부정확한 PSM은 개발 시간 및 비용 낭비에 직접적인 영향을 끼치게 되므로, PIM(Platform Independent Model)에서 PSM으로의 모델 변환 시 두 모델 간 일관성을 유지하면서 정확한 변환이 수행되었는가를 검증하는 것은 반드시 필요하다. 또한 도구를 사용하지 않고 변환을 수행할 경우에는 검증이 더욱 필요하다.

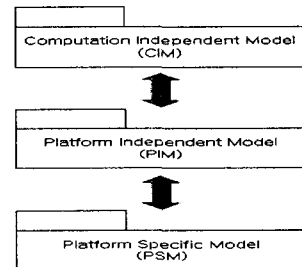
따라서 본 논문에서는 PIM에서 PSM으로의 모델 변환 시 모델의 구성 요소간의 추적성 (traceability), 정확성 (correctness) 등을 검증할 수 있는 기법을 제시한다

#### 2. 관련연구

##### 2.1 MDA (Model Driven Architecture)

MDA는 2001년 OMG에서 발표한 새로운 표준으로 모델

중심의 시스템 구현 방법을 정의한 것으로, 구현 기술에 독립적인 모델 (Computation Independent Business Model)로 시스템을 표현하고 이를 특정 플랫폼에 적합한 모델로 단계적으로 변환하여 최종적으로 코드를 생성하여 운영환경에 맞게 배치하는 것이 핵심개념이다. 플랫폼에 독립적인 모델에서 특정 플랫폼에 종속적인 모델을 얻어낼 수 있기 때문에 시스템의 통합과 유지에 유용하게 적용할 수 있는 방법론이며, 이러한 개념은 "Design once, build it on any platform"이란 MDA의 모토에서도 잘 드러나 있다 [1][2][3][4].



[그림 1] MDA의 핵심 모델

##### 2.2 PIM에서 PSM으로의 변환에 대한 기존 연구

기존의 PIM에서 EJB기반의 PSM으로의 변환 규칙으로는 다음과 같은 것이 있다[6].

**변환규칙**

1. 각 PIM 클래스에 대하여, EJB key 클래스를 생성한다.
2. 다른 PIM 클래스에 포함되지 않은 PIM 클래스는 EJB 컴포넌트와 EJB 데이터 스키마로 변환된다.
3. PIM 클래스는 변환된 PIM 클래스의 최 외각 구성(outermost composition)인 PIM 클래스로부터 생성된 EJB 데이터 스키마 내의 EJB 데이터 클래스로 변환된다.
4. PIM 연관관계(association)은 변환된 PIM 연결관계의 최 외각 구성(outermost composition)인 PIM 클래스로부터 생성된 EJB 데이터 스키마 내의 EJB 연관관계로 변환된다.
5. PIM 연결관계 클래스는 2개의 EJB 연결관계와 EJB 데이터 클래스로 변환된다.
6. 클래스의 PIM 속성(attribute)는 대응되는 EJB 데이터 클래스의 EJB 속성으로 변환된다.
7. PIM 연산(operation)은 변환된 PIM 동작의 최 외각 구성(outermost composition)인 PIM 클래스로부터 생성된 EJB 컴포넌트의 EJB 연산으로 변환된다.

기존의 변환 규칙에 대한 연구는 단순히 규칙을 제시하는 것에 그치고 있기 때문에 본 논문에서는 이러한 규칙을 통한 변환이 PIM과 PSM 사이의 일관성을 유지하면서 수행되는가를 검증하는데 중점을 두었다.

**3. PIM 에서 EJB 기반의 PSM으로의 변환에서 일관성 검증 규칙**

플랫폼 독립적인 모델 (PIM)을 EJB 기반의 플랫폼 종속적인 모델 (PSM)으로의 변환 시 두 모델 사이의 변환이 일관성을 유지하면서 적절하게 수행되었는가를 판단하기 위해서는 구성 요소의 누락이 없으며 플랫폼에 적합하게 정보를 표현하는 것을 검증해야 한다. 이에 따라 다음의 속성을 검증해야 한다.

- 추적성이란 원본모델로부터 생성된 타겟 모델에 존재하는 구성 요소를 역으로 원본모델에서 추적할 수 있음을 의미하는 것이다.
- 정확성이란 플랫폼에 독립적으로 작성된 모델의 데이터 타입이 특정 플랫폼에 맞게 적절한 데이터 타입으로 변환되었는가를 의미하는 것이다.

따라서 두 모델 사이에서 구성 요소들을 추적할 수 있고 변환된 구성 요소들이 적절한 기준에 맞게 정확하게 변환이 되었다면 두 모델 사이의 변환이 일관성을 가지고 있다고 볼 수 있기 때문에 추적성과 정확성을 검증 규칙으로 선택하였다[6].

**3.1 추적성**

본 논문에서는 여러 형태의 모델 가운데 코드화에 직접적인 영향을 미치는 클래스 다이어그램을 기준으로 추적성에 대한 규칙을 정의한다. 앞 서 기술한 추적성의 정의를 기반으로 PIM과 PSM 사이에서 추적성을 고려해야 할 구성 요소에는 클래스 다이어그램의 주요 구성 요소인 class, attribute, operation, relationship, note의 5 가지가 있으며 각각의 구성 요소 별로 추적성 검증 규칙을 정의한다. 규칙 번호는 구성 요소\_일련번호로 나타내었다. (예. 클래스에 대한 첫 번째 규칙인 경우, C\_1로 표현함.)

• class

C\_1. PIM에 표현된 class와 동일한 이름을 가진 class가 PSM에 존재하는지 체크한다.

• attribute

- A\_1. PIM에 정의된 attribute와 동일한 이름을 가진 attribute가 PSM에 존재하는지 체크한다.
- A\_2. PIM에 정의된 attribute의 타입과 PSM에 표현된 동일한 이름의 attribute의 타입이 동일한지 체크한다.
- A\_3. PIM에 정의된 attribute의 접근 제한자와 PSM에 표현된 동일한 이름의 attribute의 접근 제한자가 동일한지 체크한다.
- A\_4. PIM에 정의된 attribute에 대한 get\_(), set\_() method가 PSM에 정의되어있는지 체크한다.

• operation

- O\_1. PIM에 정의된 operation과 동일한 이름을 가진 operation이 PSM에 존재하는지 체크한다.
- O\_2. PIM에 정의된 operation의 전달인자의 개수와 PSM에 정의된 동일한 이름의 operation의 전달인자의 개수가 동일한지 체크한다.
- O\_3. PIM에 정의된 operation의 전달인자 타입과 PSM에 정의된 동일한 이름의 operation의 전달인자 타입이 동일한지 체크한다.
- O\_4. PIM에 정의된 operation의 반환값의 개수와 PSM에 정의된 동일한 이름의 operation의 반환값 개수가 동일한지 체크한다.
- O\_5. PIM에 정의된 operation의 반환값의 타입과 PSM에 정의된 동일한 이름의 operation의 반환값의 타입이 동일한지 체크한다.
- O\_6. PIM에 정의된 operation의 접근 제한자가 PSM에 정의된 동일한 이름의 operation의 접근 제한자와 동일한지 체크한다.

• relationship

- R\_1. PIM에 정의되어 있는 association, dependency, generalization, aggregation이 PSM에도 동일하게 존재하는지 체크한다.
- R\_2. PIM에 정의되어있는 role이 PSM에서도 동일하게 정의되어있는지 체크한다.

• note

N\_1. 주석 또는 제약사항을 표현하고 있는 note가 PSM에서도 표현되고 있는지 체크한다.

**3.2 정확성**

3장에서 기술한 정확성 정의에 근거하여 검증할 대상은 데이터 타입이다. 데이터 타입은 기본타입과 참조타입으로 구분할 수 있으므로 두 가지에 대하여 정확성 검증 규칙을 제시한다. 아래의 [표 1]에는 기본타입에 대한 규칙을 제시하였고, [표 2]에는 참조타입에 대한 규칙을 제시하였다. 이 규칙은 OMG의 IDL to Java Language Mapping Specification을 참조하였다[5].

[표 1] OMG IDL에서 Java로의 기본 타입의 변환

OMG IDL	JAVA	Description
short	short	Signed integer with a range of $-2^{15} \dots 2^{15}-1$
long	int	Signed integer with a range of $-2^{31} \dots 2^{31}-1$
unsigned	short	Unsigned integer with a range

short		of $0 \dots 2^{16} - 1$
unsigned long	long	Unsigned integer with a range of $0 \dots 2^{32} - 1$
unsigned long long	long	Unsigned integer with a range of $0 \dots 2^{64} - 1$
float	float	IEEE single-precision floating point number
double	double	IEEE double-precision floating point number
char	char	8-bit quantity limited to the ISO Latin-1 character set
wchar	char	wide character
boolean	boolean	8-bit quantity that is limited to 1 and 0
octet	byte	8-bit opaque data 타입, guaranteed to not undergo any conversion during transfer between systems

[표 2] OMG IDL에서 Java로의 참조 타입의 변환

OMG IDL	JAVA	Exception
string	java.lang.String	omg.org.CORBA.MARSHAL omg.org.CORBA.DATA_CONVERSION
wstring	java.lang.String	omg.org.CORBA.MARSHAL omg.org.CORBA.DATA_CONVERSION
fixed	java.math.BigDecimal	omg.org.CORBA.DATA_CONVERSION
IDL 인터페이스 내에서 선언된 상수	<ul style="list-style-type: none"> <li>비 추상화 자바 오퍼레이션 인터페이스의 필드</li> <li>유일한 추상화 자바 인터페이스</li> </ul>	
IDL 인터페이스 내에서 선언되지 않은 상수	필드, 상수값을 가진 지정값을 포함하고 있는 동일한 이름을 가진 public 인터페이스	
enum	동일한 이름을 가진 IDL개체를 구현한 자바 클래스	omg.org.CORBA.BAD_PARAMETER
struct	동일한 이름을 가진 final 클래스	
union	동일한	omg.org.CORBA.BAD_OPERATION

	이름을 가진 final 클래스	ATION omg.org.CORBA.BAD_PARAMETER
sequence	동일한 이름을 가진 자바 배열	omg.org.CORBA.MARSHAL
array	동일한 이름을 가진 자바 배열	omg.org.CORBA.MARSHAL
interface	시그니처 인터페이스와 오퍼레이션 인터페이스	omg.org.CORBA.BAD_PARAMETER omg.org.CORBA.BAD_OPERATION omg.org.CORBA.MARSHAL
exception	예외와 생성자에 대한 인스턴스 변수를 제공하는 자바 클래스	java.lang.Exception
nested 타입s	패키지에 나타나야 하는 인터페이스 범위 안에 선언된 자바 클래스	
any	org.omg.CORBA.Any	omg.org.CORBA.BAD_OPERATION

4. 결론 및 향후 연구

본 논문에서는 PIM에서 PSM으로 변환 시 두 모델 간의 일관성을 검증할 수 있는 기법을 구성 요소 간의 추적성과 정확성으로 구분하여 규칙을 제시하였다. 제시한 기법을 MDA 기반 개발을 지원하는 도구에 적용할 경우, 여러 가지 도구의 변환 성능을 측정할 수 있는 벤치마크로 활용될 수 있을 것으로 기대한다.

향후 연구로는 다양한 사례연구를 통하여 제시한 변환 규칙을 검증 및 정제해야 할 필요가 있으며, Arcstyler, OptimaJ와 같은 MDA 기반 개발 지원 도구에 규칙을 실제로 적용하여 각 도구를 비교 평가할 필요가 있다. 또한 제시한 검증 기법을 적용하여 실제 변환을 검증할 수 있는 도구의 개발이 필요하다.

5. 참고문헌

- [1] OMG, Model Driven Architecture (MDA), 2001.
- [2] OMG, OMG Unified Modeling Language (UML), version 1.5, 2003.
- [3] OMG, Meta-Object Facility (MOF™), version 1.4, 2002.
- [4] OMG, Common Warehouse Metamodel™ (CWM™) Specification, v1.1, 2003.
- [5] OMG, OMG IDL to Java Language Mapping, version 1.2, 2002.
- [6] Jos Warmer, Anneke Kleppe, Wim Bast, MDA Explained, Addison-Wesley, 2003