온톨로지를 이용한 Feature - OWL 모델 변환기법

김동리*, 송치양**, 백두권*
*고려대학교 정보통신대학 컴퓨터· 전파통신공학과
**경북대학교 컴퓨터공학과
e-mail : {c13635, baikdk}@korea.ac.kr, cysong@knu.ac.kr

A method of Feature – OWL Transformation using Ontology

Dong-Ri Kim*, Chee-Yang Song**, Doo-Kwon Baik*
*Dept. of Computer and Radio Communications Engineering, Korea University
**Dept. of Software Engineering, Kyungpook National University

요 약

소프트웨어 제품 개발에 있어서 생산성 증가와 비용 절감을 위해 기존 생성된 산출물의 재사용이 중요시 되고 있다. 이 재사용의 초점은 소스 코드의 재사용에서, 설계의 재사용, 도메인 공학에 초점을 둔 재사용으로 발전 되어 왔고, 재사용 자원을 만들기 위한 도메인 분석방법에 대한 연구가 이루어지고 있다. 현재 유사한 도메인에 대한 온톨로지 기반 feature 공통성과 가변성 분석 기법에 대한 연구가 있으나, feature 와 온톨로지에 대한 메타모델 차원의 명확한 분석과 모델들간의 매핑 프로파일이 없어서 일관성 있는 변환을 저해하고 있다. 본 논문에서는 메타모델 차원에서 온톨로지를 이용한 feature 모델과 OWL 간의 변환 방법을 제시한다. 이를 위해 feature 와 OWL 에 대한 메타모델을 정의하고, 이 속성들에 기반하여 feature 모델과 OWL 간 변환 프로파일과 알고리즘을 생성한다. 그리고 제시한 변환 규칙을 이용하여 전자결재 시스템을 통해 실제 적용함으로써 일관성 있는 모델 변환을 보여준다.

1. 서론

현재 S/W 제품 개발에 있어서 생산성 증가와 비용 절감을 위해 기존 생성된 산출물의 재사용성이 중요시 되고 있다. 이런 재사용의 초점은 소스 코드의 재사용에서, 설계의 재사용, 도메인 공학에 초점을 둔 재사용으로 발전되어 왔다. 이러한 변화는 소스코드의 재사용을 위해서는 설계의 재사용이 선행되어야 했고, 또한 설계를 재사용하기 위해서는 도메인 공학을 통해서 어떤 도메인에서 공통적으로 쓰이는 자산을 먼저 개발해야 되는지 알아야 하기 때문이다. 따라서 소프트웨어 재사용 연구는 도메인 분석 및 공학에 중점을 두고 수행되어 왔다[1][2] .

제품계열 공학에서 공통자산을 만들고 재사용하기 위한 연구로 FODA 와 FORM 같은 feature 기반 연구가 이루어 지고 있으며, 현재는 온톨로지를 이용한 feature 공통성과 가변성 분석 기법에 대한 연구가 있으나, 변환되는 모델링 요소가 충분하지 못하며, feature 와 온톨로지에 대한 메타모델 차원의 명확한 분석과 모델들간의 매핑 프로파일이 없어서 일관성 있는 변환을 저해하고 있다[3-5].

이에 본 논문에서는 메타모델 차원에서 feature 모델과 온톨로지간의 일관성 있는 변환 방법을 제시한다. 이를 위해 feature 와 OWL 에 대한 메타모델을 수정하여 재정립하고, 각 메타모델별 모델링 요소에 대한 속성을 정의한다. 그리고 이 속성들에 기반하여 feature 와 OWL 간의 변환 프로파일을 생성하고, 일관성 있는 변환을 위해 모델 변환 알고리즘 제시한다.

본 논문의 구성은 다음과 같다. 제 2 장은 관련연구에 대해 설명을 하고, 제 3 장은 본 논문의 Feature–OWL 모델 변환 접근방법을 제시한다. 제 4 장은 메타모델상의 feature 와 OWL 에 대한 모델을 분석하여, Feature-OWL 변환 규칙에 대해 설명하고, 제 5 장은 본 논문에서 제시한 변환 프로파일을 이용하여 전자결제 시스템에 대한 case study 와 본 논문에서 제시한 방법과 기존 연구방법과의 비교 평가를 한다.

2. 관련연구

2.1 feature 중심의 도메인 분석방법

제품 계열공학에서 feature 는 도메인 분석에 있어서 중요한 요소이고, 이 feature 를 명시적으로 표현하기 위한 feature 모델링에 관한 연구가 진행되고 있다.

feature 의 대표적인 분석 방법에는 FODA(Feature-Oriented Domain Analysis)가 있다. 이 방법은 feature 를 중심으로 도메인에서 주도적이고 독특한 feature 를 식별하여 도메인을 분석한다. feature 는 “구현, 테스트되고 배포 및 유지되어야 하는 기능적/비기능적 추상화를 말하는 것으로 요구 사항이나 특징적인 기능”을 말한다. 그리고 이런 feature 들은 서로 alternative, optional, mandatory 타입의 특별한 관계로써 연결 되고, 이러한 관계는 AND/OR 그래프를 통하여 계층형으로 나타내어 진다[3].

이 방법은 고객과 개발자간의 원활한 의사소통을 지

원하여, 여러 어플리케이션들 간의 공통성과 가변성을 발견해 낼 수 있다는 장점이 있지만, 분석이 부족한 시스템에 적용하는 데에는 부적합하고, 정형화에 대한 연구가 미흡하여 이에 대한 연구가 진행 중이다.

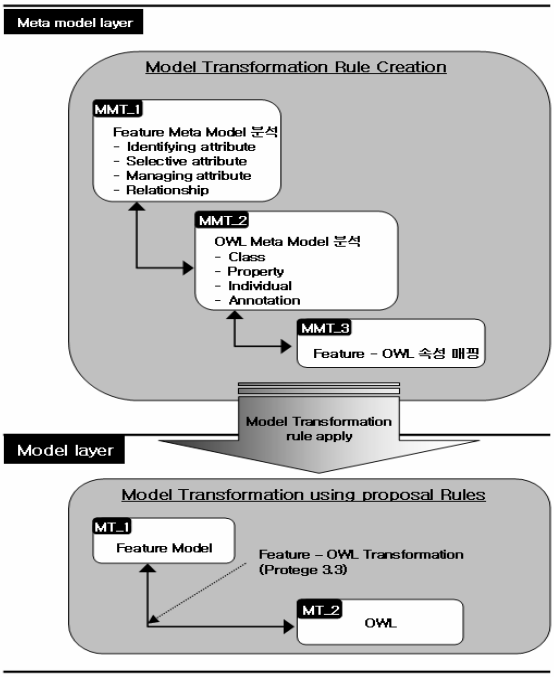
2.2 온톨로지를 이용한 feature 공통성 가변성 분석

feature 의 공통성과 가변성을 분석하는 기존 연구는 개발자의 직관이나 도메인 전문가의 경험을 중심으로 하기에, 분석 기준이 객관적이지 못하고, 비 정형적인 feature 분석으로 인해 이해 당사자들간의 이해의 오류나 불명확한 feature 를 추출하는 문제를 야기시켰다. 이에 온톨로지를 이용하여 의미 기반 유사성 분석 기준에 의해 feature 의 공통성과 가변성을 추출하는 기법이 연구 되었다[5].

이 방법은 기존 설계된 모델들간의 유사성을 온톨로지를 이용한 유사성 비교 기법을 이용하여 유사의 정도를 수치로 표현 할 수 있다는 장점이 있지만, feature 모델을 온톨로지로 변환하는데 있어 모델 변환 요소가 충분하지 못하고, 메타모델 차원의 모델 변환 알고리즘을 제시하지 않아 일관성 있는 모델 변환을 저해한다.

3. 접근방법

feature 모델을 온톨로지 모델로의 변환을 위해서는 그림 1 과 같은 단계를 따른다.



(그림 1) Feature-OWL 변환 프로세스

첫째, 메타모델 차원에서 변환 규칙을 생성하는 단계로, MMT_1 단계에서는 feature 메타모델을 분석하여 속성을 정의한다. MMT_2 단계에서는 OWL 의 메타모델을 분석하여 속성을 정의하고, MMT_3 단계에서는 분석된 두 모델 속성을 이용하여 feature – OWL 모델 간 변환 규칙을 생성한다.

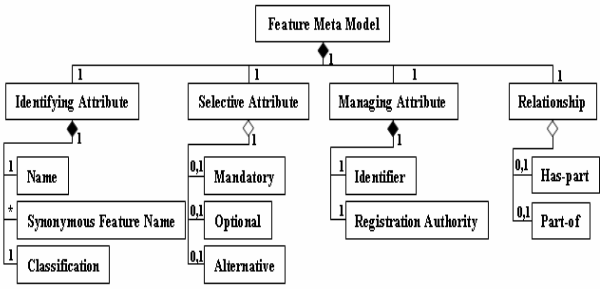
둘째, 모델 차원에서의 변환 단계이다. 기존 생성된 feature 모델을 모델 변환 규칙을 이용하여 변환한다. 각 단계의 세부사항은 4 장에서 설명한다.

4. Feature – OWL 모델변환

모델변환을 위해서는 먼저 각 모델에 대한 메타모델을 분석하고. 변환 규칙을 생성해야 한다. 이장에서는 그림 1 의 메타모델 차원의 분석 및 변환을 설명한다.

4.1 feature 메타모델 분석

이 단락은 그림 1 의 MMT_1 단계로 feature 메타모델을 분석하고 정의한다. 그림 2 는 feature 메타모델을 분석하여 UML class 모델로 표현한 것이다.



(그림 2) feature 메타모델

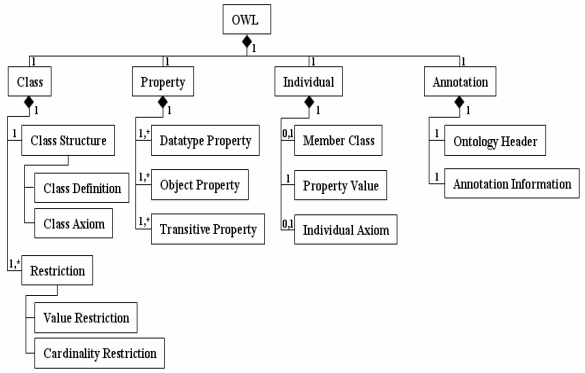
feature 메타모델은 식별형 속성, 선택형 속성, 관리형 속성, 관계형 속성으로 나누어지며, 모델변환 규칙 프로파일을 만들기 위해 <표 1> 과 같이 feature 메타모델 속성을 정의 한다.

<표 1> feature 메타모델 속성 정의

구 분		설 명
식별형속성	명칭(Name)	식별된 휘처의 이름
	분류(Classification)	도메인 내에서 부모 휘처와의 관계
	휘처 형태(FType)	휘처의 형태(Action, Function, Entity, Condition, Constraints)
	유사 휘처명 (Synset Name)	일반적 의미에서 휘처와 동일하거나 유사한 개념의 휘처
	중요도(Priority)	휘처의 중요도(0~1 값)
	상태(Condition)	necessity 나 exclusion 같은 휘처의 종속성을 나타냄
	명세(Definition)	자연어로 휘처의 의미 기술
선택형속성	필수(Mandatory)	도메인 내의 일련의 휘처들 중, 반드시 존재해야 하는 휘처
	선택(Optional)	도메인 내에서 일련의 휘처들 중, 필수적 이 아닌 선택 가능한 휘처
	양자택일 (Alternative)	도메인 내의 휘처들 중, 양자택일로 선택이 가능한 휘처
관리형속성	식별자(Identifier)	도메인에서 고유한 ID number
	등록기관 (Registration Authority)	휘처를 등록한 조직, 그룹 또는 회사명
관계형속성	집합 (Composed Of)	부모 휘처가 자식 휘처를 가지는 관계
	일반화 (Generalized by)	자식 휘처가 부모 휘처에 의해 일반화 되는 관계
	구현 (Implemented by)	구현 기술 휘처와의 관계

4.2 OWL 메타모델 분석

이 단락은 그림 1 의 MMT_2 단계 OWL 의 메타모델을 분석하고 정의한다. 그림 3 은 [6]에서 제시한 OWL 의 메타모델 속성을 UML class 모델로 표현한 것이다.



(그림 3) OWL 메타모델

OWL 의 메타속성은 Class, Property, Individual, Annotation 으로 구분되고, 정의는 <표 2>와 같다.

<표 2> OWL 메타모델 속성 정의

구 분		설 명
Class	Class Structure	온톨로지 클래스 구조적 표현
	Restriction	온톨로지의 제약조건
Property	Datatype Property	데이터 형식 속성(속성의 값이 XML 스키마 형식이나 리터럴)
	Object Property	개체 형식 속성(속성의 값이 클래스 또는 인스턴스)
	Transitive Property	추이속성, P(x,y) and P(y,z) 이면 P(x,z) 성립
Individual	Member Class	클래스로 표현되는 각각 인스턴트들의 그룹
	Property Value	클래스로 표현되는 각각 인스턴트들의 값
	Individual Axiom	개체들 사이에 사용되는 공리
Annotation	Ontology Header	온톨로지 자체에 대한 정보
	Annotation Information	주석에 대한 정보 제공

4.3 Feature – OWL 속성 매핑

4.1-2 에서 분석된 feature 와 OWL 의 속성을 비교 분석하여 <표 3>과 같은 변환 프로파일을 만들었다.

<표 3> feature – OWL 변환 프로파일

Feature		OWL
식별형 속성	명칭(Name)	OWL:class
	분류(Classification)	OWL:Property
	휘처 형태(FType)	OWL:Property
	유사 휘처명(Synset Name)	OWL:Property
	중요도(Priority)	OWL:Property
	상태(Condition)	OWL:Property
	명세(Definition)	OWL:Property
선택형 속성	필수(Mandatory)	OWL:Property
	선택(Optional)	OWL:Property
	양자택일(Alternative)	OWL:Property
관리형 속성	식별자(Identifier)	OWL:Property
	등록기관(Registration)	OWL:Property
관계형 속성	집합(Composed Of)	OWL:SubclassOf
	일반화(Generalized by)	OWL:UnionOf
	구현(Implemented by)	OWL:Property

4.4 변환 알고리즘

4.1-3 에서 분석하고 정의한 내용을 정리하면, feature 의 속성은 식별형 속성, 선택형 속성, 관리형 속성, 관계형 속성으로 나타낼 수 있고, 이것은 다시 OWL 의 속성으로 나타낼 수 있다. 이 관계를 수식으로 표현하면 아래와 같다.

$$\forall FP = \bigcup_{i=1}^{n} FP_i = \bigcup_{i=1}^{n} I_i + S_i + M_i + R_i$$

$$\bigcup_{i=1}^{n} FP_i \rightarrow \bigcup_{j=1}^{m} OP_j$$

그림 4 는 4.3 의 모델 변환 프로파일을 이용하여 Feature - OWL 모델변환 알고리즘을 Pseudo Code 형식으로 작성한 것이다.

```
Function name : TransformaitonFeatureToOntology

In.  : FP // FP(Feature Porperty)
Out. : OP // OP(Ontology Property)

Step 1 // Changing Feature to Ontology
  Switch (FP){
    if (FP = identifyingProperty){
    // identifying property transformation
          case 'name' : OP='OWL:class';
             case 'classification' : OP='OWL:DataProperty';
             case ' FType ' : OP='OWL:DataProperty';
             case 'featureSynet' : OP='OWL:DataProperty';
             case ' Priority ' : OP='OWL:DataProperty';
             case ' Condition ' : OP='OWL:DataProperty';
             case ' Definition ' : OP='OWL:DataProperty';
    }else if (FP = selectiveProperty){
     // selective property transformation
             case 'mandatory' : OP='OWL:Dataproperty';
             case 'optional' : OP='OWL:DataProperty';
             case 'alternative' : OP='OWL:DataProperty';
    }else if (FP = managingPorperty){
     // managing property transformation
             case 'identifier' : OP='OWL:Dataproperty';
             case 'registationAuthority' : OP='OWL:DataProperty';
    }else if (FP = relationPorperty){
     // relation property transformation
             case ' Composed Of ' : OP='OWL:SubclassOf';
             case ' Generalized By ' : OP='OWL:UnionOf';
             case ' Implemented By ' : OP='OWL:Property';
    }
}
```

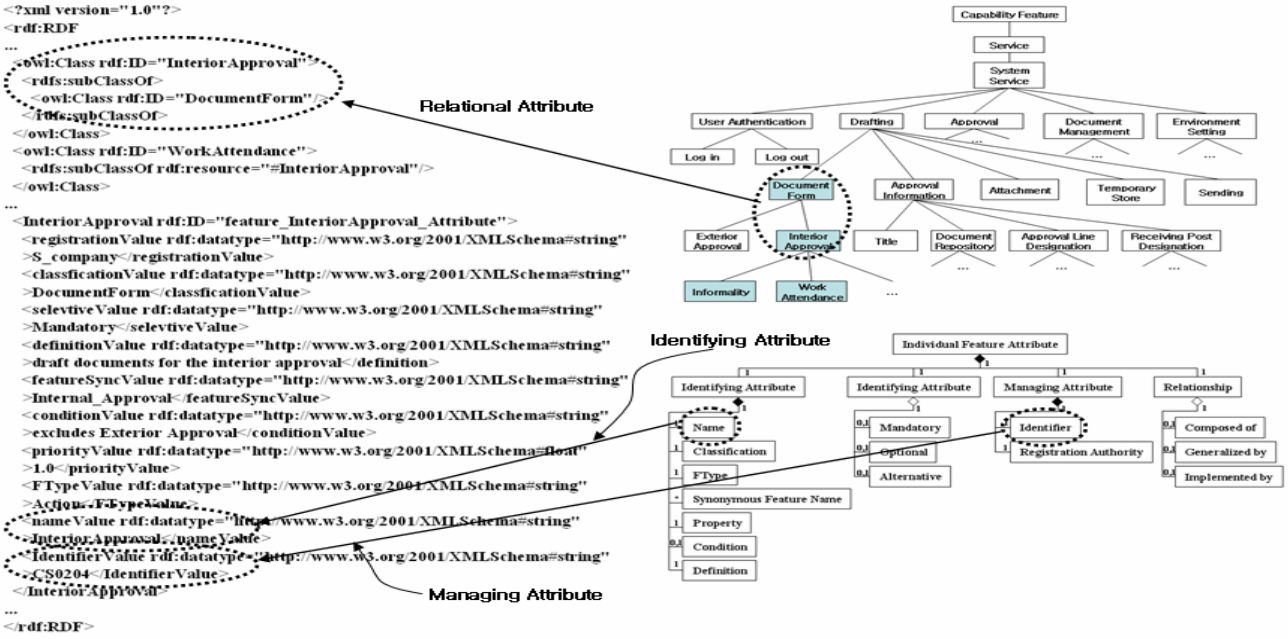
(그림 4) feature -OWL 모델변환 알고리즘

5. 사례연구 / 평가

이 장에서는 3 장 그림 1 의 MT_1,2 단계를 전자결재 시스템에 적용하여 case study 를 한다.

그림 5 는 전자결재 시스템을 feature - OWL 모델변환 규칙을 이용하여 OWL 로 변환한 그림이다.

전자결재 시스템 feature 모델의 관계형 속성과 식별형속성, 관리형 속성 등이 OWL 파일과 매핑 됨을 볼 수 있다.



(그림 5) 전자결재 시스템 OWL 변환

그리고, 기존 연구 LSB 기법[5]과 제안 기법을 비교해 보면 <표 4>와 같다.

<표 4> 변환모델 비교

구 분	LSB 기법	제안 기법
Feature 메타모델 제시	Δ	O
OWL 메타모델 제시	X	O
표현된 Feature 속성 수	8 개	15 개
변환 프로파일/ 알고리즘 제시	X	O

범례) O: 양호, Δ : 보통, X: 미흡

기존 변환기법과 비교해서 제안된 기법은 feature 와 OWL 에 대한 확장된 메타모델을 제시하였고, feature 모델의 속성을 추가로 정의하였다. 그리고, 변환 프로파일과 알고리즘을 제시하여 일관성 있는 변환을 지원한다.

6. 결론

본 논문에서는 온톨로지를 이용한 feature 의 공통성과 가변성 분석 기법에서 사용된 모델변환의 문제점을 해결하였다. 정의되지 않은 feature 와 OWL 모델에 대한 메타모델을 정의하였고, 변환 알고리즘을 만들어, 모델변환의 일관성을 유지 할 수 있는 기법을 제시하였다.

기대 효과로서, 메타모델 차원의 상세한 feature 모델의 표현과 변환 프로파일을 이용한 온톨로지로의 변환을 통해, 기존 설계된 모델들간의 유사성을 분석하는데 있어서 정확성과, 모델 변환에 있어서 일관성을 증가 시킬 수 있다.

향후 연구 과제로는 온톨로지로 표현된 feature 모델을 재활용하기 위한 방법으로 feature – OWL – UML 간 모델 변환 방법을 연구할 계획이다. 즉, 모델간의 상호 변환을 통해 기존 설계된 모델들을 재사용하는 방법에 관한 연구이다.

참고문헌

[1] Clements, P. and Northrop, L., Software Product Lines: Practices and Patterns, Addison-Wesley, Upper Saddle River, NJ, 2002

[2] Bosch, J., Florijn, G., Greefhorst, D., Kuusela, J., Obbink, J.H., Pohl, K., "Variability Issues in Software Product Lines", In: van der Linden, F. (eds.): Software Product Family Engineering. Lecture Notes in Computer Science, Vol. 2290, pp.13-21, Springer-Verlag, Berlin Heidelberg, 2002.

[3] Lee, K., Kang, K., Lee, J., "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering", In: Gacek, C.(eds): Software Reuse: Method, Techniques, and Tools. Lecture Notes in Computer Science, Vol.2319, pp.62-77, Springer-Verlag, Berlin Heidelberg, 2002.

[4] Kang, K. C., Kim, S., Lee, J., et al., "FORM: A Feature-Oriented Reuse Method with Domain Specific Reference Architecture," Annals of Software Engineering, Vol. 5, pp.143-168, 1998.

[5] 이순복, 백두권, et al., " 소프트웨어 제품계열공학의 Ontology 기반 휘처 공통성 및 가변성 분석 기법" , 정보과학회논문지, Vol. 34, pp.196-211, 3. 2007.

[6] Michael K. Smith, Chris Welty, Deborah L. McGuinness, "OWL Web Ontology Guide", W3C Recommendation 10, Feb. 2004.