

Software 제품계열공학에서 온톨로지에 기반한 feature의 공통성 및 가변성 분석모델

김진우⁰ * 이순복* 이태웅** 백두권*

고려대학교 컴퓨터학과, 한국과학기술정보연구원

{pkm311,lsb0510,Baik}@software.korea.ac.kr, **minote@kisti.re.kr

An approach to analyze commonality and variability of feature based on Ontology in Software Product line Engineering

Jinwoo Kim⁰, Soonbok Lee*, Taewoong Lee**, Dookwon Baik*

*Department of Computer Science and Engineering, Korea University,

**Korea Institute of Science and Technology Information

요 약

제품계열공학에서 feature diagram(FD)은 개발자의 직관이나 도메인 전문가의 경험에 근거하여 작성되어, feature간의 공통성 및 가변성분석 기준이 불명확하며 비정형적인 feature의 공통성 및 가변성 분석으로 인한 stakeholder의 공통된 이해가 부족한 문제점을 내포하고 있다. 따라서, 본 논문에서는 이를 해결하기 위하여 공통된 feature의 이해를 위해 feature 속성리스트에 기반한 메타 feature모델과 feature간의 의미유사성관계를 이용한 온톨로지를 적용한 공통성 및 가변성 분석모델을 제안한다.

1. 서 론

오늘날 소프트웨어 재사용의 초점은 소스코드의 재사용에서 소프트웨어 설계의 재사용으로, 설계 재사용에서 다시 도메인공학에 기반을 둔 재사용으로 그 패러다임이 옮겨져 왔다. 따라서 지난 수 년간 소프트웨어 재사용 연구는 도메인 분석 및 공학(domain analysis & engineering)에 중점을 두고 수행되어 왔다 [1]. 도메인 공학에서 도메인을 설계하는 방법 중 대표적인 방법 중의 하나는 제품계열공학(PL)측면의 feature 중심 도메인 분석 방법이다[2]. 하지만 현 feature 중심의 소프트웨어 재사용 개발 방법론은 실제 공통성 및 가변성을 분석하는 기준이 비정형적인 도메인 전문가의 경험(Heuristic)이나 개발자의 직관(Intuition)에 의해 작성되어 feature간의 상호충돌 및 feature interaction 문제가 발생된다.

따라서 stakeholder간의 이해의 오류가 발생하는 문제점을 내포하고 있다. 따라서 본 논문에서는 단순히 기능, 비 기능 중심으로 표현하고 있는 애매모호한 feature를 속성기반의 메타모델을 구축하고, feature 온톨로지 그래프 안에서 의미(semantics) 중심의 개념 유사성을 비교하여 공통성 및 가변성 분석의 준거를 제시하고 명시적인 feature를 추출하는 모델을 제안한다.

2. 관련 연구

2.1 FD의 공통성 및 가변성 분석 측면의 한계점
공통성 및 가변성 분석에서 사용된 가장 현저한 접근은 FD이며, PL(product line)에서 공통 feature와 가변 feature를 그들의 관계로 표현한다. 즉 여기서 공통 feature는 하나의 PL에서 모든 product member에 포함

될 mandatory feature로 정의되고 가변 feature는 반드시 포함되지 않고 선택 되어 질 수 있는 feature이다. 하지만 도메인 전문가의 경험 및 직관에 의해 분석되어 feature가 명시적이지 못하고 공통성 및 가변성의 구분하는 준거가 비정형적이다[2]. 또한 feature의 관계는 FD에서 표현되고 정의된 것들보다 훨씬 복잡하고 예를 들어 feature간의 의존성(dependency)이 커진다면 복잡성(complexity)이 매우 증대되는 문제점이 있다. 또한 S/W PL에서 핵심자산으로 공통성과 가변성을 분석하는 연구가 진행되어 왔지만, PR(Primitive requirement)과 context의 속성을 확률적으로 결정하는 매트릭스의 정렬로 공통성과 가변성을 결정하는 방법은 실제 적용하는데 있어 명시적이지 못한 추상적인 절차로서 그 한계를 내포하고 있다 [3].

2.2 FD와 온톨로지의 상호 접목 필요성

feature는 구현, 테스트되고 배포 및 유지 되어야 하는 기능/비기능적 추상화를 말하는 것으로 요구사항이나 특징적인 기능을 말한다[2]. FD는 node와 edge로 구성된 트리 구조이다. 각 node는 하나의 feature로 표현되고 각 edge는 edge에 연결된 두 가지 feature 사이의 관계를 나타낸다. 여기에서 접목 주안은 FD의 taxonomy의 구조는 온톨로지 형태로 표현이 가능한 점이다. 온톨로지는 "하나의 지식에서 공유할 수 있는 이해를 기준한 명시적 명세(formal specification of shared understanding for a knowledge)"이다[4]. 지식을 내포하고 있는 정보를 동일의미의 용어로 표현한다는 측면과 feature와 특정분야의 지식을 표현하는 체계를 제공한다는 측면에서 taxonomy인 온톨로지와 관련성이 있다. 따라서 온톨로

톨로지 안에서 정의된 feature를 말하며, relation은 은톨로지 내의 정의된 concept간의 relation을 정의하는 이원화 관계로 정의된다. 하나의 개발 도메인에 대하여 FD에 기반한 은톨로지를 구성하면 feature간의 공통성 및 가변성 비교를 feature 간의 의미관계로 각 feature간의 유사성을 측정할 수 있는 은톨로지 유사성 그래프를 도출할 수 있다[5].

3. feature 공통성 및 가변성 분석 제안 모델
 Feature에 대한 공통성 및 가변성 분석 모델은 아래 그림과 같이 meta feature modeling 단계와 feature ontology similarity graph 단계를 통해서 이루어진다.

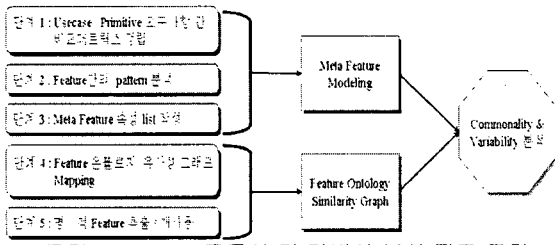


그림 1. feature 공통성 및 가변성 분석 접근 모형

첫째, meta feature modeling 단계는 Usecase/Primitive 요구사항간의 비교 매트릭스에서 1차적 feature 의미 정형성(semantic formalism)을 도출한 다음 각각의 feature pattern(sibling relation, part_of, has_part, self relation) 분석을 바탕으로 meta feature 속성리스트를 작성하여 명시적 feature로의 실체화를 하는 과정이다. 둘째, meta modeling 단계는 feature를 바탕으로 한 MFR(meta feature repository)에 저장된 feature와 개발 도메인 상에서 feature 은톨로지와의 의미 유사성 관계를 feature ontology similarity graph를 이용한 공통성과 가변성을 분석하는 단계이다. 이 단계에서 도메인 개발 시 재사용이 가능한 명시적 feature를 추출한다.

3.1 Meta Feature Modeling

아래 그림 2는 UML로 제시한 meta feature model이다.

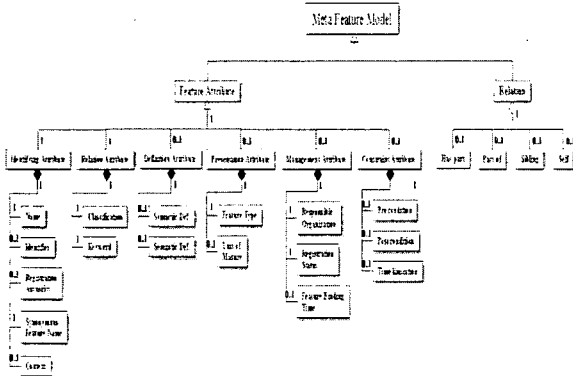


그림 2. meta feature model

지는 concept과 relation의 이원화 형태로 concept은 은 아래 표는 meta feature model에 기반한 개별 feature 속성을 정의하는 feature 속성 리스트를 보여주고 있다.

표 1. feature 속성 리스트

구분	세부 속성	속성 정의
식별형 속성 (Identifying Attribute)	이름(Name)	Feature의 이름
	식별자(Identifier)	500만 UML의 feature 고유 ID
	등록기관(Registration Authority)	Feature를 등록한 조직 그룹 또는 회사명
	동어(Сynonymous Feature Name)	동어적 의미에서 Feature와 동일하나 유사한 개념의 Feature 이름
	문맥(Context)	Feature가 사용되는 문맥적 표현에
관계형 속성 (Relation Attribute)	분류체계(Classification)	500만 UML의 상위 Feature의 이름
	핵심어(Keyword)	Feature 이름을 구성하는 단어 또는 상/하위 Feature의 Keyword
정의형 속성 (Definition Attribute)	문법적정의 (Syntactic Definition)	시각적 의미로서 Feature의 형태적 정의
	의미적 정의 (Semantic Definition)	시각적 의미로서 Feature의 기능적 정의
표현형 속성 (Presentation Attribute)	Feature 유형 (Feature type)	시각적 의미로서 Feature의 형태 및 분류
	측량단위(Unit of measure)	feature를 측정할 수 있는 단위
관리형 속성 (Management Attribute)	책임조직 (Responsible organization)	Meta Feature Repository를 관리하는 기관 또는 조직
	등록상태(Registration status)	Feature 속성 리스트 작성의 현상태 정도 (높음:high, 중간:mid, 낮음:low)
제한형 속성 (Constraint Attribute)	선행조건(Pre-Conditions)	Feature가 생성되기 위한 선행조건에 대한 조건
	후발조건(Post-Conditions)	Feature가 생성된 후 반드시 수행되어야 하는 후발 조건
	시간제한(Time limitation)	Feature의 시간적 제한

feature ontology similarity graph에서 의미유사성 정도를 측정하기 위해서 feature 속성별 세부 가중치를 적용한다. feature의 세부 속성별 가중치를 적용하는 기준은 아래와 같다.

기준 1. feature의 의미 유사성 (Similarity : s)
 feature가 의미적으로 동일하거나 유사한 정도 (범위 : 0 ≤ s ≤ 1)

구분	동일하거나 유사도 매우 높음	유사도 높음	유사도 보통	유사도 낮음	유사도 매우 낮음	유사도 없음
가중치	1.0	0.8	0.6	0.4	0.2	0

기준 2. 상/하위 Feature와의 관계성 (Relation : r, r = 0 or 1)

Taxonomy 내에서 상/하위 Feature와의 Edge는 의미적 관계에서 매우 중요하다. Sibling 관계의 Feature는 상위 Feature로부터 가변성을 가져 다른 Feature로서 구분되므로 의미적 관계 거리는 멀다고 볼 수 있다. Feature 자신과의 관계는 상/하위 Feature와 동일하게 적용하여, 수식으로 나타내면 아래와 같다.

$$f(r) = \text{Distance}_{\text{relation}}(\text{feature}_{\text{lower}} \rightarrow_r \text{feature}_{\text{upper}}) = \begin{cases} 0, & \text{for Sibling Relation, no Relation;} \\ 1, & \text{for Part-of, Has-part, Self Relation.} \end{cases}$$

→_r 기호는 상/하위 Feature간의 의미적 관계를 나타내며, 관계의 범위값은 0 또는 1을 가진다. 따라서 Feature의 의미 유사성과 상/하위 Feature와의 의미적 관계성을 적용한 세부 속성 가중치(w_v)는 아래와 같다.

$$w_v = \frac{s + f(r)}{2}$$

표 2. 가중치(weight)를 부여한 Meta Feature 속성 list

구분	세부속성	Feature 의미유사성 (0 ≤ s ≤ 1)	상/하위 Feature와의 관계 범위값 (1/2 or 1)	가중치 (0 ≤ w ≤ 1)
식별형 속성 (Identifying Attribute)	이름(Name)	1.0	1	1.0
	식별자(Identifier)	0	0	0
	등록기관(Registration Authority)	0	0	0
	동언어(Synonymous Feature Name)	0.8	1	0.9
관계형 속성 (Relation Attribute)	문맥(Context)	0.8	0	0.4
	분류체계(Classification)	0.6	1	0.8
정의형 속성 (Definition Attribute)	핵심어(Keyword)	0.8	1	0.9
	구문적 정의 (Syntactic Definition)	0.8	0	0.4
표현형 속성 (Presentation Attribute)	의미적 정의 (Semantic Definition)	0.8	0	0.4
	Feature 유형 (Feature type)	0.2	0	0.1
관리형 속성 (Management Attribute)	측정단위(Unit of measure)	0.2	0	0.1
	책임조직(Responsible organization)	0	0	0
제약형 속성 (Constraint Attribute)	등록상태(Registration status)	0	0	0
	선행조건(Pre-Conditions)	0	0	0
	후발조건(Post-Conditions)	0	0	0
	시간제한(Time limitation)	0	0	0

3.2 Feature Ontology Similarity Graph

Feature의 공통성 및 가변성 분석의 준거로 의미적 범위와 의미적 사상은 아래 그림 3의 정형적 명세로 정의된다[5].

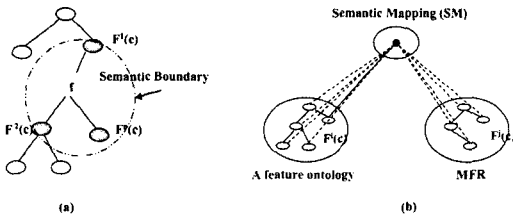


그림 3. 의미적 범위와 사상

Def. Semantic Mapping(SM)

$Onto(f) := (U, I, L)$ 은 아래 조건을 만족한다.

U : upper feature concept,

I : intermediate feature concept

L : lower feature concept, $F(c)$: feature의 concept.

1. $Onto(f)$ 상에서 하나의 feature는 무한 개념의 집합(an infinite set of Concepts)이다.

2. $E \subseteq F(c) \times F(c)$

($F(c)$ 는 feature의 개념적 의미의 집합,

E 는 $F(c)$ 간의 edge).

3. feature 의미 유사성 사상 알고리즘

· Child node에 feature 의미 유사성이 있을 경우 :

Child node와 mapping

· Leaf node에 feature 의미 유사성이 있을 경우 :

Parent node와 mapping

· leaf node에서 sibling 간 feature 의미 유사성이 있을 경우 :

Sibling node와 mapping

4. 구현 및 평가

본 논문에서는 S/W PLE 도메인 분석 시 사용되는 정형적 feature에 대한 공통성 및 가변성 분석 기준을 온톨로지를 이용하여 적용하였으며, 아래와 그림 4 같이 의미관계 온톨로지 OWL(Ontology web language)과 Java를 이용하여 의미 매핑 툴을 구현하였다

```

<owl:Class rdf:ID="NewDocument">
  <rdf:subClassOf rdf:resource="#Request"/>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="InternalApproval"/>
    <owl:Class rdf:ID="ExternalDispatch"/>
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="InternalApproval">
  <rdf:subClassOf rdf:resource="#NewDocument"/>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="Informality"/>
    <owl:Class rdf:ID="WorkAttendance"/>
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="ExternalDispatch">
  <rdf:subClassOf rdf:resource="#NewDocument"/>
</owl:Class>
  
```

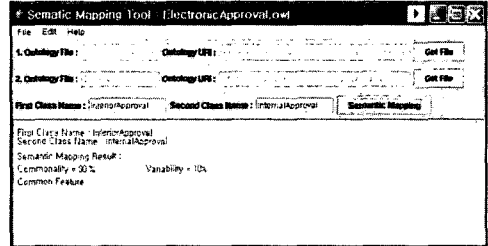


그림 4. OWL source code / UI 화면

5. 결론

본 논문의 의의는 정형적인 feature의 메타모델 체제로 feature의 실체화 와 feature ontology similarity graph를 이용하여 feature를 공통성과 가변성을 명시적으로 분석하여 feature의 차기동일 도메인에 대한 재사용성을 증진시키는 데 중요한 의의가 있다고 볼 수 있다.

참고문헌

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

[2] 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.

[3] Mikyeong Moon, Keunhyuk Yeom, " An Approach to Developing Domain Requirements as a Core Asset Based on Commonality and Variability Analysis in a Product Line, IEEE Transactions on Software Engineering, Vol. 31, NO.7, pp.551-569, July2005.

[4] S.J. Green." Building Hypertext Links by computing semantic similarity. IEEE trans. Knowledge and data Eng. Vol. 11, No.5, pp 713-730.

[5] Ju-hum Kwon, Doo-Kwon Baik, " Measuring Semantic Similarity Based on Weighting attribute of Edge Counting, LNCS 3397, 2005.