

휘처 모델의 Z 정형 명세와 검사 기법

송치양 *, 조은숙 **, 김철진 ***

A Formal Specification and Checking Technique of Feature model using Z language

Chee-Yang Song *, Eun-Sook Cho **, Chul-Jin Kim ***

요약

시각적이고 비정형적인 구조로 표현된 휘처 모델(Feature model)은 구문적 명확성을 보장할 수 없고, 자동화 툴(tool)에 의한 구문(syntax)의 검증이 어렵다. 따라서, 휘처 모델이 가진 구조물의 구문적 명확성을 입증하기 위한 정형적 명세와 모델 검사(model checking)가 필요하다. 본 논문은 Z 언어를 이용한 휘처 모델의 정형적 명세와 모델 검사를 통해서, 휘처 모델의 정확성을 검사하는 기법을 제시한다. 이를 위해, 휘처 모델과 Z 간 변환 규칙을 정의하고, 이 규칙에 의거하여 휘처 모델의 구문에 대해 Z 스키마(schema)로 명세한다. 모델 검사는 Z 스키마 명세에 대해 Z/Eves 툴을 사용하여 구문, 타입 검사(type checking), 그리고 도메인 검사(domain checking)를 수행하여 모델의 모호성을 검사한다. 이로서, 휘처 모델의 구조물을 좀더 명확하게 표현할 수 있으며, 설계된 모델의 오류를 검사할 수 있다.

▶ Keywords : 휘처모델, Z, 정형명세, 모델검사, Z/Eves

Abstract

The Feature model can not be guaranteed the syntactic accuracy of its model and be difficult the validation using automatic tool for its syntax, because this model is expressed by a graphical and informal structure in itself. Therefore, there is a need to formalize and check for the feature model, to precisely define syntax for construct of the model. This paper presents a Z formal specification and a model checking mechanism of the feature model to guarantee the correctness of the model. It first defines the translation rules between feature model and Z, and then converts

• 제1저자 : 송치양 • 교신저자 : 조은숙

• 투고일 : 2012. 9. 26. 심사일 : 2012. 11. 22. 게재확정일 : 2012. 12. 13.

* 경북대학교 컴퓨터정보학부(School of Computer and Information, Kyungpook University)

** 서울대학교 컴퓨터소프트웨어학과(Dept. of Computer Software, Seoul University)

*** 인하공업전문대학 컴퓨터시스템과(Dept of Computer System, Inha Technical College)

※ 이 논문은 2012학년도 경북대학교 학술연구비에 의해 연구되었음.

the syntax of the feature model into the Z schema specification by applying these rules. Finally, the Z schema specification is checked syntax, type, and domain errors using the Z/Eves validation tool to assure the correctness of its specification. With the use of the proposed method, we may express more precisely the construct of the feature model. Moreover the domain analyst are able to usefully verify the errors of the generated feature model.

▶ Keywords : Feature model, Z, Formal specification, Model checking, Z/Eves

I. 서 론

최근 들어 유비쿼터스, 임베디드, 모바일, IPTV 및 웹 등 다양한 시스템 혹은 소프트웨어 개발에 있어서 재사용을 고려하기에 초기 도메인 모델의 중요성은 높아가고 있다. 대표적인 시스템의 소프트웨어 개발방법으로 재사용 초점의 객체/컴포넌트(마르미 III, UP: Unified Process)와 휘처(feature)[1-3] 지향적인 개발 접근이 있다. 이들 방법에 의한 도메인 모델로서 클래스 모델과 휘처 모델이 생성된다. 그 간 개발 프로세스에 따른 추상화 수준별 모델내 혹은 모델간 일치성 및 추적성에 대해 주로 연구가 진행되어 왔다. 특히, UML 클래스 모델의 Z로의 변환, 명세 및 검증에 대해서는 체계화되어 있다[4-7]. 그러나, Z[8-9]를 이용한 휘처 모델에 대한 정형적 명세 및 검증에는 연구가 성숙되어 있지 않다. 즉, 휘처 모델의 Z로의 변환 규칙, 지칭, 명세 방법의 가이드라인이 구체적이지 못하다. 또한, 이들은 휘처간 관계 기반의 개별적 변환만을 제시하여, 복잡한 휘처 모델 전체 구조 차원의 Z 명세 방법을 제공하지 않으므로써 실제적인 적용이 어렵다.

시스템의 개발은 명확하게 도메인 모델이 분석되어야 하고, MDD(Model Driven Development) 지향의 재사용성을 고려해서 수행되어야 한다. 즉, 재사용을 위해서 모델이 정확성을 가져야 한다. 그러나, 휘처 모델은 요구 도메인에 대해 시각적이고 비정형적인 구조로 표현한다. 그래서, 이들 응용 모델의 구조물이 가진 구문(syntax), 제약사항(constraint)을 명확하게 표현하고, 모델의 정확성 검증이 필요하다. 즉, 휘처 모델을 정형적 언어인 모델/집합론(set theory) 기반의 Z를 사용한 모델의 명세와 Z/Eves[10-11] 사용의 검사가 필요하다. 특히, 실시간적이고 반응적인 품질을 요구하는 국방 방어 시스템, 원자력 제어 시스템, 고속철 운영시스템인 경우, 분석 모델에 충돌 등 모호성 문제가 내포되어서는 안된다.

본 논문에서는 휘처 모델의 구문적 구조를 명확히 표현하고, 모델의 정확성을 입증하기 위해, Z를 사용하여 휘처 모델을 정형적으로 명세하고, Z-Eves 툴(tool)을 통한 Z 명세의 정확성을 검사한다. 먼저, 휘처 모델과 Z간의 변환규칙을 정의하고, 이 변환 규칙을 적용해서 휘처 모델의 구문에 대해 Z 스키마로 명세한다. 즉, 휘처 모델의 구문에 대해 Z로 기본 타입을 선언하고, 각 휘처를 대상으로 요소 기반으로 Z 스키마를 정의하고, 또한 휘처간 관계 및 다중성에 기반하여 Z 스키마를 정의한다. 다음으로 Z에 대한 모델의 편집과 검사를 제공하는 Z/Eves 툴을 사용해서 정의된 Z 명세를 입력하고, 구문, 타입 검사(type checking), 그리고 도메인 검사(domain checking)를 수행하여 모델의 구조적 불일치를 검사한다. 이로서, 휘처 모델 구조물의 아키텍처 의미를 명확히 나타낼 수 있고, 모델의 정확성을 검사할 수 있다. 아울러, 변환의 프로파일 및 규칙이 알고리즘으로 개발되고, 자동화 툴로 구축된다면 자동적 분석이 가능해진다.

본 논문의 구성은 II장에서 휘처 모델, Z 그리고 이들 간 변환의 연구현황을 살펴본다. III장은 휘처 모델-Z 모델간 변환규칙에 대해서, IV장에서는 적용사례로서, 모바일 폰 시스템을 대상으로 Z 명세와 모델 검사를 보인다. VI장은 비교평가와 특징을 다룬다.

II. 관련 연구

2.1 휘처 모델

휘처 모델은 휘처와 그들 간의 관계에 의해 SPL(Software Product Line)의 가능한 모든 제품들의 정보를 표현하는 정보 모델이다[12]. 초기 휘처 기반의 모델링 방법으로 FODA(Feature-Oriented Domain Analysis)[1]는 시스템의 집합체 중에서 주도적이고 독특한 휘처를 식별하여, 이것들을 바탕으로 공통성과 가변성에 의해 도메인을 분석한다.

FODA는 4가지 계층인 능력(capability), 운용환경(operation environments), 도메인 기술(domain technologies), 구현기술(implementation techniques) 상에서 회처를 식별하여 모델화한다. [2]에서 회처는 “구현, 테스트되고 배포 및 유지 되어야하는 기능적, 비기능적 추상화를 말하는 것으로 요구사항이나 특징적인 기능”이라 정의한다. 회처의 재사용에 기반한 FORM (Feature-Oriented Reused Method)[3]은 도메인의 회처를 분석하고, 이들 회처를 유사 도메인 개발시 재사용하는 개발방법이다. FORM에서 회처 모델은 AND/OR 그래프로 표현되는데, 회처간 AND 노드는 회처의 필수적(mandatory)인 회처를 말하고, OR 노드는 다른 제품으로 선택 될 수 있는 양자택일(alternative)의 회처를 말한다. [13]에서 회처는 요구사항, 기능, 행위의 추상화의 개념을 가지며, 계층적(트리) 구조로 표현하며, 회처간 composed-of/generalization/specification/implemented-by의 관계 타입을 갖는다. 회처 모델의 확장 연구로서 다중성과 속성의 추가가 있다. 회처에 다중성을 확장한 [14-15]에서, 회처 다중성(feature cardinality)과 그룹 다중성(group cardinality)로 분류하여 회처간 mandatory/optional/alternative/or에 다중성을 부여했다. [12][16]에서 회처에 필수 속성으로 name/domain/value 모델링 요소를 확장했다.

이러한 확장된 회처의 식별적, 관계적 속성과 다중성이 Z로의 변환에 주요한 구조물의 요소들이 된다.

2.2 Z

Z[8-9]는 범용적 ISO(International Standard Organization) 표준 언어로서 증명(proof)을 위한 좋은 기능을 제공한다. Z는 성숙되었고, 표현력에 제한이 없으며, 추상적 언어로서[17], 이를 지원하는 다양한 툴들이 풍부하다. 특히, Z 언어는 수학적 이론에 의거해서 집합론과 first-order predicate logic에 기반을 둔 상태 기반의 정형적 명세 언어이다. 그래서, Z는 정확성이 요구되는 시스템 혹은 모델의 정형적 명세에 활용되어져 왔다. Z의 구조물은 타입과 스키마로 구성되는데, 스키마는 선언부(signature 혹은 declaration part)와 술어부(predicate part)에 의해 그 구성요소와 제약사항을 명세한다. Z를 객체지향 개념으로 확장한 Object-Z [18]가 있는데, 공리(axiomatic), 표시 시맨틱(denotative semantics), 지원 툴에 갖은 변경을 안고 있다. 이론 증명기(theorem prover)인 Z/Eves[10-11]는 Z 명세에 대해 조합하고, 검사하고, 분석을 제공하는 상호작용 시스템이다. 이 툴은 Z 명세의 증명을 위해 범용 이론(general theorem)을 지원하는데, 이론(theorem)은 모델

의 속성(property)을 표현한다. “Z를 사용하여 회처 모델을 명세한다.” 의미는 절차적 프로그램 언어처럼 일이 어떻게 수행되는 지를 표현하는 것이 아니라, 모델을 구성하는 요소들을 위한 변수(variable)와 모델에 가해지는 불변의 제약조건들을 명세하는 것이다[7]. 따라서, Z에서는 집합으로 표현되는 변수 값들과 그에 대한 제약조건을 명시한 스키마(schema)를 사용하여 타입, 관계(relationship), 그리고 제약사항 등을 명세한다. 이에, Z의 상태(state) 스키마가 회처 모델의 회처와 동일한 구조를 갖으며, 정적 속성(static property)을 갖고 있어 상호 정보의 손실없이 직접적 변환이 가능하다.

2.3 Z 기반 회처 모델의 정형 명세 및 검증

회처 모델은 시각적-트리 구조로 구조물의 정형적 시맨틱 표현에 한계를 갖고 있다. 회처의 개념과 관계를 명확히 표현하기 위해 정형적 기법 혹은 언어인 algebraic-based technique, Higher-Order logic[19], Propositional Logic/ CSP(Constraint Satisfaction Problem)[12]과 description logic(틀 미 지원), Alloy language와 Z notation 등을 통한 명세와 검증에 연구가 있어왔다. 회처 모델의 분석 툴로 SAT Solver, Alloy analyzer, BDD(Binary Decision Diagram) Solver, SMV(Symbolic Model Verifier) 등이 있다.

Z와 Alloy 언어는 각각 Z/Eves(혹은 ZTC, ZEST) 및 Alloy analyzer 검증 툴을 사용해서, 회처 모델의 정형 명세와 검증을 제시한 [20]은 회처 모델을 Z의 first-order semantics를 사용해서 정의하고, theorem proving 기법을 통해 제시한 시맨틱의 정확성을 검증했다. 이를 통해 회처 모델의 일치성을 보였다. 즉, 부모-자식 회처간 mandatory, optional, alternative, or에 대해, 그리고 회처간 제약사항을 require, excludes 관계에 의해 Z 스키마로의 변환 명세, 명세된 것을 Z/Eves 등 사용해서 불일치를 검증했다. 그러나, [20]은 회처간 composed-of, generalization의 관계, 다중성 기반 변환, 그리고 전체 회처 모델상에서 구성된 회처들을 어떤 순서 및 규칙에 의해 Z로 명세되는지에 대해서 제시하지 못했다. 이들의 후속 연구인 [21]에서, 회처 모델의 추론을 제시했다. 즉, 회처 모델을 시맨틱 웹 온톨로지(OWL)로 변환하여 FaCT++ 툴을 통해 회처 모델의 불일치를 식별하고, Alloy analyzer를 통해 불일치를 추적하고, Z/Eves 툴을 통해 복잡한 비표준 회처들의 정제화를 수행했다. 그 외, Z 관련된 주로 UML(클래스 모델)과 Z간의 변환 혹은 통합[4-7] 및 Z/Eves 통한 검증[22]에 많은 연구가 있

으며, 두 모델간 기능들에 대한 검증, 휘처모델-온톨로지(OWL)-클래스 모델간 변환[23] 등 연구가 있다.

III. 휘처 모델 - Z간 변환 규칙

휘처 모델의 Z 변환은 메소드(휘처 모델)와 정형적 언어(Z)간의 통합을 의미한다. 대상 도메인을 위한 휘처 모델은 휘처들과 그들 간의 관계로서 표현된 비정형적 정보 모델인데 비해, Z는 구성요소와 제약사항을 스키마 타입으로 나타낸 정형적 명세 모델이다. 이에, 상이한 두 모델간의 변환 규칙이 필요하다.

3.1 모델간 변환 접근의 설계 원리

휘처 모델과 Z간의 변환을 위한 접근 모델은 그림 1과 같다. 그림 1에서, 제시방법에 따른 절차는 어떤 시스템의 도메인에 대한 휘처 모델이 생성되면, 이 모델이 가진 구문에 대해 속성(식별형, 관계형)과 다중성(multiplicity)에 기반해서 정의된 변환규칙에 의해 Z로 변환 명세하고, Z 명세의 정확성을 Z/Eves를 사용해서 검사한다.

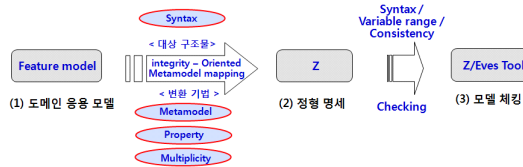


그림 1. 휘처모델 - Z간 변환의 접근 모델
Fig. 1. Approach model for the translation between feature model and Z

변환 접근을 위한 설계원리는 다음과 같다.

- 메타모델 지향(Metamodel Driven) : 메타모델 기법을 사용해서 휘처 모델과 Z 언어의 추상 구문(abstract syntax)을 정의한다. 메타모델은 상위 MOF(Meta Object Facility)의 서브 클래스로 정의된다. 메타모델은 휘처 모델과 Z 언어의 구조물이 가진 모델링 요소 혹은 명세 요소들을 추출하고, 이들을 관계화하여 정립한다. 메타모델의 모델링 요소에 기반하여 휘처 모델과 Z간 변환 프로파일을 정립한다.
- 속성 지향(property driven) : 휘처 모델과 Z간 변환은 이들 구조물이 가진 식별형(이름, 도메인, 값)과 관계형(mandatory/optional/alternative/or, requires/excludes) 속성에 의거하여 매핑을 규칙화한다.

- 다중성 기반 매핑 : 휘처 모델의 부모-자식 휘처간 생성되는 휘처 객체 수 즉, 다중성(multiplicity 혹은 cardinality)을 반영해서 Z 스키마의 술어부에 명세한다. 다음은 어떻게 휘처 응용 모델이 변환 기법을 적용해서 모델의 정확성이 검증되는 지를 보여주는 것이 그림 2이다.

3.2 휘처 모델-Z간 변환 규칙

휘처 모델의 Z로 변환을 위해 먼저, 이들의 메타모델을 정의하고, 메타모델의 모델링 요소에 기반하여 두 모델간 매핑 프로파일을 정립하고, 구체적인 변환 규칙을 기술한다. 기존 class model의 Z 변환 연구[4-5][7]를 수정, 적용하여 변환 규칙을 정의한다.

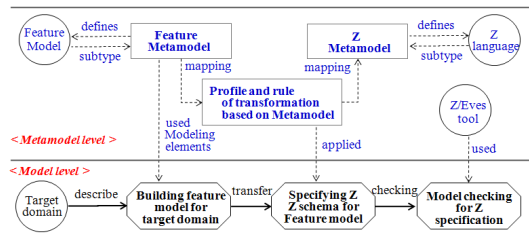


그림 2. 휘처 모델의 정확성 검사 프로세스
Fig. 2. Checking process for verifying the accuracy of feature model

3.2.1 휘처 모델과 Z의 메타모델

휘처 모델은 제품이 가진 공통적이고 가변적인 요구사항의 휘처들을 표현하는 도메인(domain) 다이어그램이다. 즉, 제품을 구성하는 휘처들과 그들 간의 관계로 표현하며, 확장 모델링 요소로서 휘처의 속성(attribute) 및 휘처간 다중성을 갖는다[14-16]. 이들의 요소들에 기반하여 휘처 모델의 메타모델을 정의한 것이 그림 3이다. 휘처 메타모델은 휘처와 관계 중심의 요소들로 정의한다. 휘처는 속성과 다수의 타입을 가지며, 휘처간 관계는 mandatory, optional, alternative 및 OR의 유형을 갖는다, 그림 3에서, 점선 박스는 확장된 모델링 요소를 나타내며, "value"는 속성이 가질 수 있는 값으로 측정될 수 있는 어떤 특징을 의미한다. "domain"은 값이 가질 수 있는 범위를 나타내는 것으로 정수, 불리언 혹은 enumerated의 유형이 있다. 여기서, mandatory, optional, alternative, OR, composed-of, generalized-by, implemented-by는 직계 부모-자식 휘처간 관계를 표현하며, requires와 excludes는 직계 형제간, 타 부모 혹은 자식들과의 관계를 나타낸다. 특별히 휘처간 종속 여부를 표현하는 "requires"와 "excludes"는 관계 유형에서 parental relationship 혹은 cross-tree constraints라 칭한다. "constraints"는 휘처와 휘처간 관계에 공통적으로 적용될 수 있는 제약사항이다.

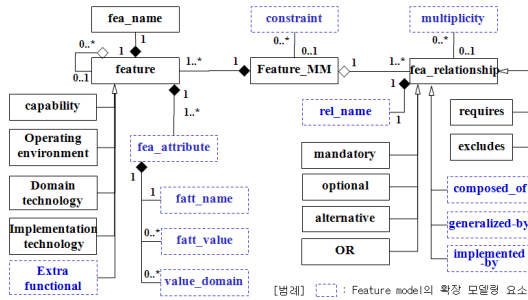


그림 3. 회처 모델의 메타모델
Fig. 3. Feature metamodel

그림 3에서, fea attribute는 회처의 속성, fatt_name은 회처의 이름, fatt_value는 회처가 갖는 값을 의미한다.

Z 모델은 도메인 시스템에 대해 상태에 기반하여 수학적으로 타입과 스키마로 표현하는 정형적 명세 언어이다. Z 명세는 타입, 공리(axiom) 그리고 스키마의 요소들로 나타내는데, 이들 요소들을 가지고 Z의 메타모델을 나타낸 것이 그림 4이다. 그림 4에서 스키마는 선언부에 명세하는 "state"와 술어부에 기술하는 "operation"으로 구성되며, "state schema/ initialization schema/ operation schema/ generic schema"의 유형을 가진다. Z의 관계 요소인 "include"는 상위에 명세된 스키마에 대해 하위 스키마들에서 포함할 수 있는 것으로서, 스키마가 스키마를 가질 수 있는 경우이다.

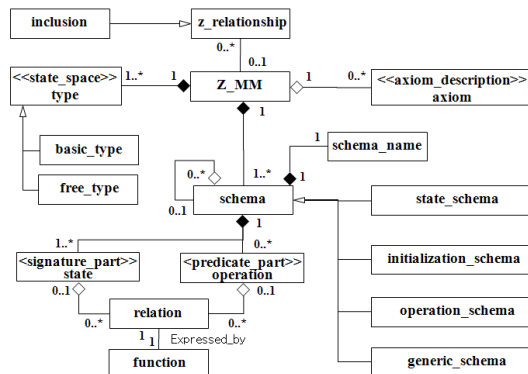


그림 4. Z 모델의 메타모델
Fig. 4. Z metamodel

3.2.2 회처 모델과 Z의 구조물간 변환 프로파일

구조물간 변환을 위해, 회처 모델이 갖고 있는 속성인 요소(회처), 관계(회처간), 다중성 및 제약사항을 고려하여 Z로의 매핑 프로파일이 필요하다. 이를 위한, 회처 모델의 추상 구분과 정적 시맨틱(static semantics)을 Z 스키마로 변환

을 위한 모형은 그림 5와 같다. 회색 박스 부분이 본 논문에서 해결하고자 영역을 나타낸다. 여기서, 회처 모델이 표현하지 못하는 정적 시맨틱은 OCL(Object Constraint Language)을 사용해서 liveness, soundness 등의 품질 특성을 추가적으로 명세할 수 있다. 공리기술, 제약사항(가령, 정적/동적 시맨틱)에 대해서는 또 다른 이슈이기에 다루지 않는다.

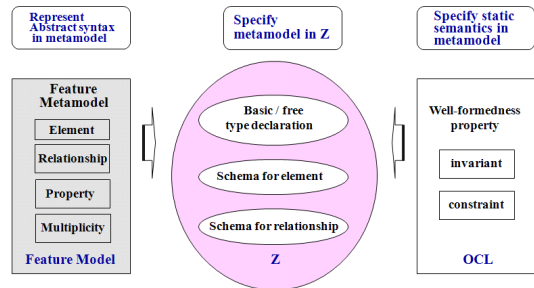


그림 5. 회처 모델의 Z 명세로의 변환 모형
Fig. 5. Translation model Z for feature model

표 1. 휘처 모델의 Z 명세 위한 프로파일
Table 1. Profile for converting feature into Z

속성 (Property)	휘처 모델 요소	클래스 모델 요소	Z 모델 요소	휘처 모델의 요소 설명
식별형 (요소 기반)	fea_name	class_name	schema_name	휘처 이름
	fea_attribute	cla_attribute	state	휘처의 세부 속성들
	fatt_name	catt_name	state	휘처 속성의 이름
	fatt_value	catt_name	state	속성의 값
	value_domain	tagged_value	state	값의 범위
관계형 (관계 기반)	mandatory	composition	schema(inclusion), state, operation	부모에 대해 자식 휘처가 필수 요소
	optional	association	schema, state	부모 휘처가 자식 휘처의 선택이 옵션
	alternative	inheritance	schema(inclusion), state	부모 휘처가 다수 자식 휘처중 한 개만 선택
	OR	inheritance	generic_schema, state	부모 휘처가 다수 자식 휘처중 한 개 이상 선택
	requires	dependency	schema(inclusion), state, operation	한개 휘처가 타 휘처를 포함 (mutual-combination)
	excludes	constraint	schema(inclusion), state, operation	한 개 휘처가 타 휘처를 배제 (mutual-exclusions)
	composed-of	composition	schema(inclusion), state, operation	자식 휘처가 부모 휘처에 포함 부모 휘처를 위한 자식휘처의 집합체
	generalized-by	inheritance	generic_schema, state	자식 휘처가 부모 휘처에 대해 한 종류 혹은 타입
implemented-by	realization	not support	부모 휘처의 구현을 위해 관계되는 자식 휘처	
다중성 (요소/관계의 함수 기반)	1..1 (mandatory, alternative)	1..1	relation (state: 함수, operation: 정의역/치역)	한 개의 휘처 인스턴스
	0..1 (optional)	0..*	relation	휘처 인스턴스의 수가 없거나 혹은 한 개
	1..* (OR)	1..*	relation	휘처 인스턴스의 수가 없거나 혹은 한 개 이상
	n..m (OR)	n..m	relation	휘처 인스턴스의 수가 특정 n개에서 m개로 제한

이러한 변환 모형과 메타모델에 근간하여, 모델링 요소에 의한 세부적인 구조물간 변환의 매핑 프로파일을 보여주는 것이 표 1이다. 즉, 휘처 메타모델에 정의된 모델링 요소들과 Z 메타모델내 명세 요소들간의 매핑 관계를 보여준다. 이 프로파일은 식별형, 관계형, 그리고 다중성에 의해 분류한다. 이 매핑 변환을 통해 휘처 모델은 Z 명세로 표현될 수 있다. 또한, 클래스 모델의 모델링 요소를 대비해서 보여준다. [7]에서 클래스 모델을 Z 모델로의 변환을 제시했다. 이에, 휘처 모델과 클래스 모델간의 변환 관계를 정립하면 휘처 모델의 Z 변환을 정의할 수 있다. 표 1에서, "concept"를 의미하는 휘처는 "정보의 조각 혹은 공통 aspect"를 나타내는 스키마와 시멘틱하게 동일하므로 상호 변환 매핑한다. 즉, "fea_name"을 "schema_name"으로 변환한다. 휘처간 관계에 대해 클래스 모델에 대비하여 그 유사성을 살펴보자. 관계형 속성에서,

mandatory는 부모-자식간 필수적인 포함관계이므로 클래스 모델의 관계 유형중 composition에 해당된다. 이때, aggregation이 아닌 이유는 휘처 모델은 트리형의 구조로 휘처간 직계 계층에서 자식 휘처가 부모 휘처하고만 관계를 가지기 때문이다. optional은 선택 휘처(0 또는 1)이기에 휘처 각각이 독립성을 갖기에 이들 휘처 간에 관계는 association에 해당한다. alternative와 OR는 부모-자식 휘처간 동등 관계 즉, 자식은 부모의 한 종류이므로 inheritance에 관계한다. requires 경우는 한 개 휘처가 다른 휘처를 요구하는 즉, 의존적인 관계이므로 의존(dependency)에 해당된다. 한편, 휘처의 다중성은 Z 스키마에서 그 다중성 관계의 유형별로 함수로서 표현된다.

3.2.3 회처 모델과 Z의 구조물간 변환 규칙

회처 모델은 다수의 회처들과 관계로서 구성된 트리형의 구조이다. 단원 3.2.2의 표 1의 식별형, 관계형 및 다중성의 속성에 기반해서, 회처 모델의 회처들 및 관계를 가지고 Z 모델로의 매핑을 정의했다. 본 단원에서는 회처의 모델링 요소들이 Z로 어떻게 표현되는지 그리고 복잡한 회처 모델 구조물에서 어떠한 순서에 의해 Z 스키마로 명세해 나가는지에 대한 구체적인 규칙이 필요하다. 이를 위해, Z 명세에 요구되는 타입 정의 및 스키마 정의(요소와 관계 기반) 순으로 변환 규칙을 정의한다. 여기서, 회처의 다중성은 관계 기반 스키마 명세에 통합되어 기술한다.

3.2.3.1 Z 타입 정의

규칙 3-1 (Z의 기본타입 선언) 기본타입은 회처간 alternative/ OR/generalized-by 즉, inheritance 관계를 갖는 단말 회처들을 제외한 모든 회처들을 Z의 기본타입으로 선언한다.

3.2.3.2 Z 스키마 정의

Z 명세는 다수의 state들과 operation 스키마들로 정의한다.

이때, 회처모델은 정적인 트리 모델(그래프형의 클래스 모델에 순환이 제거된 것)이기 때문에 operation 스키마는 정의하지 않고, state 스키마를 정의하면 된다(회처에 속성이 없는 경우로 가정). Z의 state 스키마의 구조는 선언부와 술어부로 구성되어 있다. 선언부는 변수를 선언하며, 술어부는 변수의 값에 제약사항을 기술한다. Z 스키마 정의는 회처모델 내 요소(element, feature)와 라벨(label)을 가진 관계를 대상으로 요소 식별자, 그들 간의 관계 및 불변 제약사항을 명세한다.

가. 요소 기반의 Z 스키마 변환 규칙

요소 기반의 Z 스키마 명세는 표 1의 식별형 속성에 의해 회처 요소별 스키마 정의, 스키마 내부 명세, 요소간 관계 유형별 스키마 명세로 변환 규칙을 정한다. 먼저, 회처 요소의 스키마 명세는 회처 모델내에 포함된 모든 회처들에 대해 각각 Z 스키마로 선언을 한다. 한 개의 회처에 대해 한 개의 스키마로 선언한다(4). 그림 6은 요소 기반에 의한 회처의 Z 변환을 보여준다. A 회처에 대한 Z 명세는 기본타입 선언과 A 스키마로 정의한다. A 스키마는 선언된 기본타입 [A_ID]를 사용해서 식별자(ident)로 명세함으로써, A 회처를 변환, 표현한다.

Feature	Z Description	Comment
A	[A_ID] └─ A ── ident: A_ID	// Basic Type // Schema Description

그림 6. 회처 요소의 Z 변환
Fig. 6. Z translation of an feature element

규칙 3-2 (요소 기반 Z 스키마 내부 명세) Z 스키마 내부명세는 스키마의 선언부에 그 요소의 고유 식별자를 선언하고, 이 요소와 관계(mandatory 등)하는 요소들을 변수로서 선언한다. 술어부에는 이 요소가 갖는 불변적 제약사항(well-formedness 등)을 명세한다.

규칙 3-3 (관계 유형별 요소 기반 Z 스키마 명세 순서 및 방법) 회처 모델 구조상의 회처의 스키마 정의 및 명세 방법은 다음과 같다. 변환 대상 회처의 관계 유형은 mandatory/composed-by (composition), optional (association), alternative/OR/generalized-by (inheritance) 이다.

- 요소들간에 복잡한 구조를 갖는 회처 모델의 Z 스키마 명세 순서는 상향식 접근으로 하부 회처에서 점차 상위 회처의 순으로 스키마를 정의해 해나간다.
- 회처간 optional 관계의 표현은 두 회처 요소에 대해 각각 스키마로 명세한다. 이 관계는 관계 기반 Z 스키마 정의에서 이를 표현한다.
- 회처간 mandatory 및 composed-of 관계의 표현은 두 회처 요소에 대해 각각 Z 스키마로 선언한 후, 추가로 부모(A 회처 : parent 혹은 whole) 회처 스키마의 선언부에 자식(B 회처 : child 혹은 part) 회처 스키마를 포함하여 명세한다. 그림 7에서 A 회처가 b 회처를 필수 회처로 갖는 경우, 두 회처의 기본 타입을 선언하고, 자식 B 회처의 B 스키마를 먼저 생성후, 부모 A 회처의 A 스키마 선언부에 B 스키마를 요소로서 포함해서 표현한다.

Type of Relationship of Feature model	Notation of Feature model [two feature]	Z Description
mandatory / composed-of	A ───● B	[A_ID, B_ID] └─ B ── ident: B_ID └─ A ── ident: A_ID b: B

그림 7. 회처간 "composed-of" 관계의 요소 기반 Z 변환
Fig. 7. Z Element-based translation for "composed-of" relationship between two features

- alternative/OR/generalized-by/requires 관계의 휘처 휘처 요소는 부모 휘처 요소를 먼저 선언하고, 점진적으로 후속 자식 요소들을 명세해 나간다. 이때, 부모 스키마로부터 상속을 받기에 자식 스키마에 대한 기본타입 선언에 유일(unique) 식별자 선언이 필요치 않다.

나. 관계 기반의 Z 스키마 변환 규칙

휘처간 관계 기반의 Z 스키마 변환 명세는 표 1의 관계형과 다중성의 속성에 기반해서, 두 휘처간에 관계 “라벨”을 가진 모든 관계들을 대상으로 Z 스키마로 명세한다. Z 명세로의 변환 대상 휘처 관계는 관계명이 부여되지 않은 관계(상속: alternative/OR/ generalized-by)를 제외한 mandatory/composed-by, optional, requires 이다.

규칙 3-4 (관계 기반 Z 스키마 대상 및 생성) Z 스키마 명세 대상은 휘처 모델내 휘처간 라벨이 명명된 각 관계별로 스키마를 생성한다. 관계하는 두 휘처 요소들에 대한 별도의 기본 타입(basic type) 및 자유 타입(free type)은 명세하지 않는다. 왜냐하면, 요소-기반 스키마 정의에서 사전에 명세하기 때문이다.

규칙 3-5 (관계 기반 Z 스키마 내부 명세) Z 스키마 내부는 선언부와 술어부로 명세한다. 선언부의 명세는 휘처간 관계별 특성(mandatory, optional 등)과 다중성을 반영해서, 함수로 선언한다. 술어부에는 휘처간 관계상의 다중성 및 불변 제약사항을 기술한다.

규칙 3-6 (관계 유형별 Z 스키마 명세 및 순서) 휘처간 관계 기반의 Z 스키마 명세는 다음과 같다.

- 관계별 스키마 정의는 순서에 무관하게 명세한다. 요소 기반 스키마 명세에서 휘처 모델의 구조가 반영되기 때문이다.
- 관계 유형별 스키마 명세는 관계 유형별 제약사항과 다중성에 의한 Z 변환 함수를 포함하여 정의한다.
- 휘처간 optional 관계는 두 휘처가 선택적 연관 관계이므로 관계명에 의해 Z 스키마로 정의한다. 두 휘처가 관계를 갖는다는 것만 명세하면 되기에, 스키마 내부에 별도 제약사항의 명기가 필요치 않다 (그림 8).
- mandatory 및 composed-of 관계는 휘처 관계명에 의해 Z 스키마 명세를 한다. 추가적 제약사항으로 두 휘처들간에 필수적 포함 관계를 갖도록 명세한다. 또한, 스키마의 술어부에 자식 요소가 부모 요소(parent

element)를 하나만 갖도록 제약을 가하여 명세한다.

- requires 경우, 한쪽 휘처가 다른 휘처에 영향을 주는 의미적 의존 관계이다. Z 스키마 명세는 휘처 B가 휘처 A에 의존관계를 갖는다면, 휘처 A가 변경 이전과 변경 이후가 동일하지 않다면, 휘처 B도 변경 이전과 변경 이후가 같지 않도록 정의한다.
- 휘처간 다중성을 반영한 Z 변환 명세는 다중성 기반 매핑 함수를 구하여([5] 근거), 이 함수를 Z 스키마의 선언부에 기술하며, 술어부에는 구체적인 부모-자식 휘처간 다중성의 수를 명세한다(그림 8).

그림 8에서, 관계명에 의해 기술되는 opt_ab 스키마는 선언부에 선택 속성과 다중성에 기반하여 해당 함수를 기술하고, 술어부는 정의역과 치역에 의해 각 휘처가 갖는 다중성의 수를 나타낸다. 선언부에서, 휘처 A와 휘처 B간의 다중성이 “0..* 대 0..1” 이라면, [5]의 변환 규칙에 의거하여 부분 함수(partial function)로 다중성을 표현한다. 술어부 경우, 정의역 B 휘처의 0 혹은 1을 술어 논리로 표현한다.

Type of Relationship of Feature model	Notation of Feature model [two feature]	Z Description
optional (including Multiplicity)	$A \xrightarrow{\text{opt_ab}} B$ $0..* \quad 0..1$	<p>[A_ID, B_ID]</p> <p>A ident: A_ID B ident: B_ID</p> <p>opt_ab optional</p> <p>n: N multiplicity</p> <p>opt_ab: A → B</p> <p> $\forall s: \text{ran opt_ab} \cdot \#(\text{opt_ab} \triangleright \{s\}) < n$ $\wedge \#(\text{opt_ab} \triangleright \{s\}) > 0$ $\forall t: \text{dom opt_ab} \cdot \#(\{t\} \triangleleft \text{opt_ab}) = 0$ $\vee \#(\{t\} \triangleleft \text{opt_ab}) = 1$ </p>
	<p>opt_ab ∈ 0..* → 0..1 [a, b]</p> <p>Domain Range</p>	

그림 8. 휘처간 optional 및 다중성을 갖는 Z 스키마 변환
Fig. 8. Z translation for optional and multiplicity between two features

IV. 휘처 모델의 Z 명세 및 검사

Z를 이용한 휘처 모델의 정형적 명세 및 검사를 위하여, 본 장에서는 모바일 폰 시스템을 대상으로 제 3장에서 제시된 휘처 모델 - Z간 변환 기법을 적용한 사례를 보인다. 그림 2의 휘처 모델의 검사 프로세스에 따라서, 모바일 폰 시스템의 도메인에 대한 휘처를 모델링하고, 이 휘처 모델을 가지고 변환규칙을 적용해서 Z로 정형적 명세하고, 이를 Z/Eves를 통한 모델의 검사를 수행한다. 이러한 정형적 명세와 검사를 통해서, 휘처 모델의 의미를 추가적으로 명확하게 명세할 수 있으며, 구축된 휘처 모델의 구문적 구조에 대한 정확성 및 일관성의 분석 및 검증이 가능해진다.

4.1 모바일 폰 시스템의 회처 모델링

회처 모델링을 위해 [12]에서 사례로 보여준 모바일 폰 시스템의 회처 모델에 “wireless” 회처 및 관계명이 추가된 그림 9를 대상으로 적용사례를 보인다.

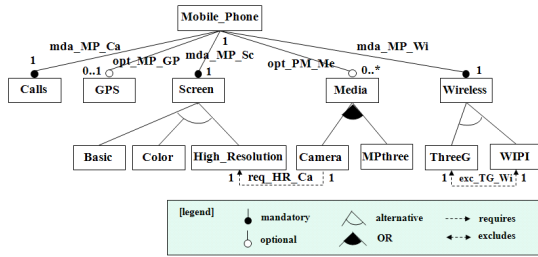


그림 9. 모바일 폰 시스템의 회처 모델
Fig. 9. Feature model of mobile phone system

그림 9에서, 모바일 폰은 calls, screen 및 wireless 회처들은 필수 기능으로, GPS와 media는 선택 기능으로, media 회처의 camera 및 MP3는 OR 기능으로, 그리고 wireless 회처는 그 종류인 3G와 WIPI를 alternative 기능으로 모델화하였다. 또한, camera는 high resolution이 요구되기에 이들 회처간 requires로 표현하며, 3G와 WIPI 회처간에는 둘 중 하나의 무선편만 존재해야 하므로 excludes로 나타내었다. 한편, 회처간 관계 명은 mandatory와 optional에 대해 부여하고, alternative, OR 및 generalized-by인 경우는 상위 스키마로부터 상속받으므로 관계명을 부여치 않았다.

4.2 모바일 폰 회처 모델의 Z 명세

모바일 폰 회처 모델의 정형적 Z 명세를 위하여, 3장에서 제시한 요소, 관계 및 다중성 기반의 변환 규칙을 적용해서, Z로 명세한다. 즉, 회처 모델내 각 회처들에 대해 기본 타입과 자유 타입을 선언하고, 정의된 변환 규칙을 적용해서 회처 모델내 회처들에 대한 스키마를 정의하고, 그리고 관계상에 존재하는 각 관계명별로 스키마를 정의한다. Z 명세의 구축 결과로서, 기본타입으로 5개 회처 타입 선언, 요소 기반 13개 스키마 명세, 관계 기반 7개 스키마를 명세했다. 구축된 Z 명세에 대해 일부를 발췌하여 기술한다.

4.2.1 회처의 기본타입 선언

타입 선언은 기본 타입과 자유 타입으로 기술된다. 기본 타입의 선언은 규칙 3-1에 의거해서, 그림 9의 회처 모델에

포함된 모든 회처들에 대해서 각 회처별로 기본 타입을 명세한다. 단, 회처간 alternative/OR/generalized의 상속 관계를 갖는 단말(leaf) 회처는 제외한다. 타입 선언시, 회처별 고유한 식별을 위해 ID를 부착했다. 기본 타입의 선언은 다음과 같다.

[Calls_ID, GPS_ID, Screen_ID, Media_ID, wireless_ID, Mobile_Phone_ID]

한편, 자유 타입의 선언은 회처간 타입(종류)의 관계에서 role 타입을 가진 경우에 선언한다. 모바일 폰 회처 모델에서는 role 타입이 존재하지 않으므로 기술하지 않는다.

4.2.2 모바일 폰 회처 모델의 Z 스키마 명세

스키마 명세는 모바일 폰 회처 모델내에 포함된 모든 요소(회처) 그리고 관계들에 대해서 Z 스키마로 명세를 한다.

4.2.2.1 Z 회처 요소 기반 Z 스키마 명세

요소 기반 Z 스키마 명세는 회처 모델내 회처별로 각각 스키마를 생성한다. 이의 일부를 그림 10에서 보여준다. 회처들간의 스키마의 선언 순서는 규칙 3-3에 따라 하부 회처에서 상위 회처 순으로 스키마를 선언해 나간다. 가령, 그림 10에서 ‘Screen’ 회처의 스키마 정의는 ‘basic’ 스키마 정의 이전에 명세되어야 한다. 왜냐하면, 자식 회처가 부모 회처로부터 상속을 받아야 하기 때문이다. 회처들간의 관계인 mandatory/ alternative/requires 등에 대해서도 요소 기반 스키마에서 부분적으로 명세된다. 각 스키마는 규칙 3-2에 따라 변수들로 구성되는 선언부 그리고 제약조건의 시멘틱을 표현하는 술어부로 명세한다. Mobile_Phone 스키마는 하부 회처로 Calls, GPS 등을 갖기에 이들 스키마를 선언부에 포함시키고, 술어부에 유일 식별자 선언과 다중성을 표현한다.

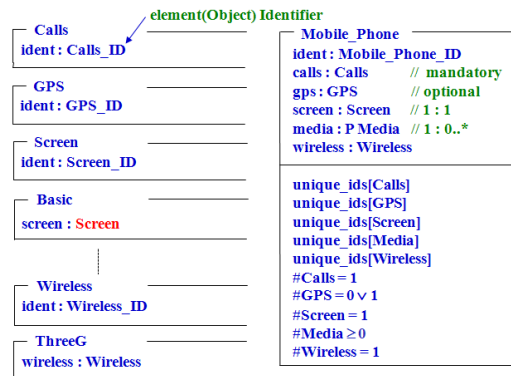


그림 10. 모바일 폰 회처 모델의 요소 기반 Z 스키마 명세
Fig. 10. Element-based schema specification for mobile phone feature model

즉, 휘처들에 대한 선언과 그들 간의 관계 표현은 선언부에 명세하고, 불변적 제약사항들은 술어부에 정의한다. 예를 들어, 'Mobile_Phone' 스키마의 술어부에 기술한 '#Media ≥ 0'은 불변적 제약사항을 표현한다. 이것은 모바일 폰에서 미디어의 휘처 수가 0개 이상이어야 함을 의미한다.

4.2.2.2 Z 휘처 관계 기반 Z 스키마 명세

관계 기반 스키마 명세는 두 개 휘처간의 관계별로 각기 스키마로 정의한다. 이 스키마는 휘처간 관계들과 다중성을 명확하게 구분하여 정의된다. 모바일 폰 휘처 모델의 관계 기반 스키마 명세의 일부를 나타낸 것이 그림 11이다.

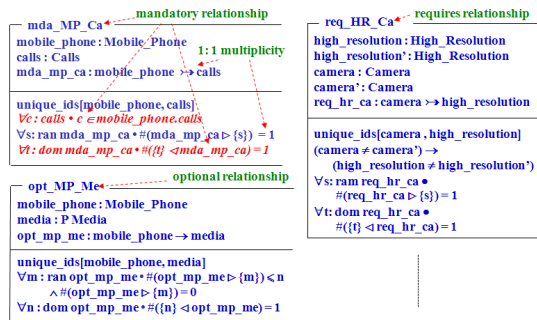


그림 11. 모바일 폰 휘처 모델의 관계 기반 Z 스키마 명세
Fig. 11. Element-based schema specification of mobile phone feature model

휘처 모델상의 관계 명칭은 그림 9에서와 같이 관계의 유형 및 관계하는 두 개 휘처의 이름을 약칭하여 부여했다. 그림 11은 규칙 3-5에 의해 관계별로 스키마를 생성하며, 규칙 3-6에 의해 관계 타입 및 다중성에 따라서 스키마 내부의 선언부와 술어부를 기술하며, 규칙 3-7에 의해서 휘처의 관계 유형별 변환 규칙을 적용해서 명세하였다.

먼저, 부모-자식 휘처간 mandatory 관계를 갖는 'mda_MP_Ca'는 'Mobile_Phone'과 'Calls'의 휘처간에 관계한다. 규칙 3-5 및 규칙 3-6에 의거, 이 두 휘처를 선언부에 기술하고, 아울러 다중성에 기반하여 선언부와 술어부에 정의한다. 선언부에서 각 휘처의 변수 선언시, 다중성에 의해 단일 set(휘처가 1개) 혹은 Power set(휘처가 2개 이상)으로 표현된다. 이 두 휘처의 스키마 생성은 요소 기반 스키마 명세(단원 4.2.2.1)에서 개별적으로 이전에 스키마로 정의된다. 또한, 두 휘처간 다중 수가 1:1이므로 bijection 함수로 기술한다. 술어부의 mandatory 표현은 글자가 기올림체로 표시된 부분이다. 즉, mandatory는 부모 휘처의 다중 수가 1인 경우를 의미한다. dom(정의역)은 부모 휘처, ran(치역)

은 자식 휘처의 다중 수를 각각 표현한다. optional 관계를 갖는 'opt_MP_GP'는 'Mobile_Phone' 휘처와 'GPS' 휘처에 관계한다. 규칙 3-6에 의거, 휘처간 참조 연관 관계로서 그 관계성을 명세한다. 아울러, 이들 휘처간 다중성인 1:0..1이기에 total injection 함수로 기술한다. 'req_HR_Ca' 관계는 휘처간 의존의 관계로서 선언부에 동일 휘처에 대해 두 개의 휘처를 선언하여(가령, camera, camera'), 술어부에서 휘처간 변경전과 변경후가 동일하다는 것으로 명기한다. 한편, 본 논문에서 다루지 않는 휘처 모델내 휘처, 이들 간의 관계들에 대한 추가적인 제약사항인 적형성 속성 (Well-formedness property)에 대해서 스키마의 술어부에 기술할 수 있다.

4.3 Z 스키마 명세의 모델 검사

모바일 폰 휘처 모델의 정확성 및 의미적 타당성을 검증하기 위해서, 먼저 이 모델은 Z로 변환, 명세하였다. 다음으로 이 Z 스키마 명세의 정확성을 보이기 위해, Z/Eves 툴을 사용한 Z 모델의 구문적 속성 등 검사가 요구된다. 즉, Z 명세의 구문 검사, 가변 범위 검사(variable range checking) 및 일치성 타입 검사(consistency type checking)를 위하여 Z/Eves 툴을 통하여 모델 검사를 해야한다.

통상, Z 명세의 검증 툴로서 Z/Eves가 널리 사용되고 있다. 사용된 Z/Eves 툴은 V2.1이다. 이 툴은 검사와 타입 검사, 스키마 확장(schema expansion), 사전 조건 계산(precondition calculation), 도메인 검사, 명세 애니메이션(specification animation) 그리고 범용 이론 증명(general theorem proving)을 지원한다. 이에, 이 툴의 Z 편집기를 사용해서 Z 명세를 입력하고, 이론 증명기를 실행하여 모바일 폰 휘처 모델의 구문, 타입 검사, 그리고 도메인 검사를 수행한다.

Z/Eves 툴의 실행 화면으로 그림 12는 기본 타입 선언의 검사 결과를 보여준다. 왼쪽 상단에 표시된 'syntax'는 구문과 타입 검사의 결과를 보여주는 것으로 'Y'로 나타나면 타입간 일치성이 있음을 나타낸다. 반면, 'proof'는 타입 검사의 결과를 보여주는 것으로 'Y'로 나타나면, 그 명세가 정확함을 의미한다.

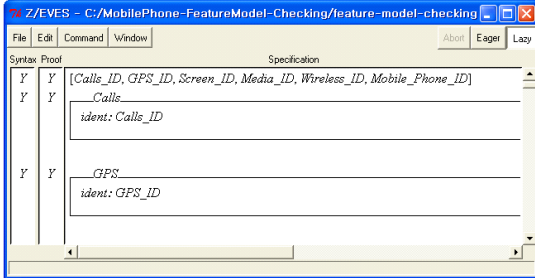
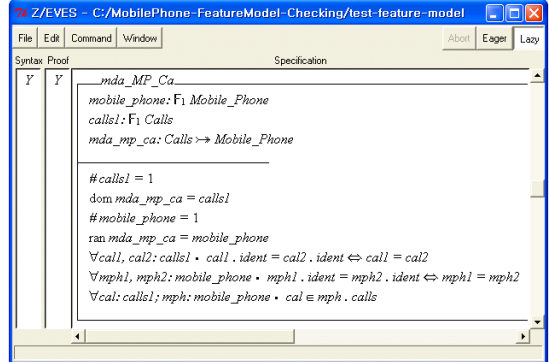


그림 12. 기본 타입 Z 명세의 구분 검사 화면
Fig. 12. Syntax checking screen for basic type of Z specification



(a) 'mandatory' 휘쳐 관계의 검사

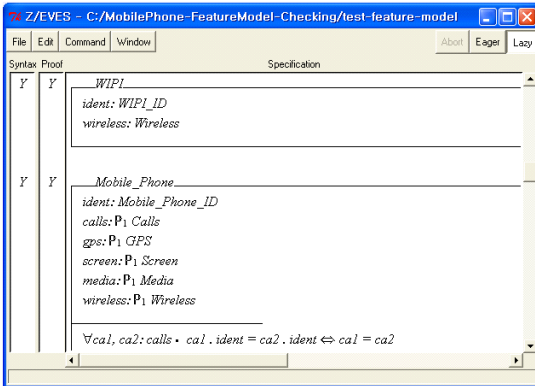
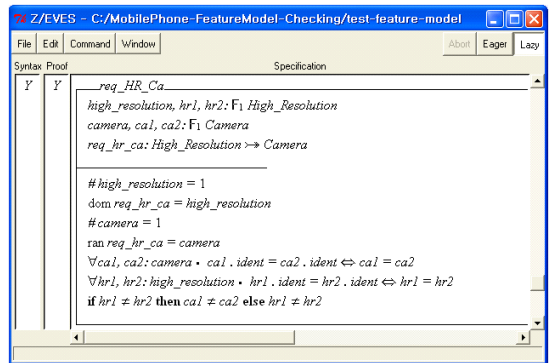


그림 13. 요소 기반 Z 스키마 명세의 구분검사 화면
Fig. 13. Syntax checking screen for an element-based schema specification



(b) 'requires' 휘쳐 관계의 검사

그림 14. 관계 기반 Z 스키마 명세의 구분검사 화면
Fig. 14. Syntax checking screen for a relationship-based schema

그림 13은 요소 기반 Z 스키마 명세에 대한 모델 검사의 결과를 보여준다. 이 검사에서 이전 Z 명세된 일부 변경하여 기술하였다. 그 이유는 Z 언어의 표현력과 이 표현력에 대해 Z/Eves에서 완전히 동일하게 지원하지 않는 부문도 있기 때문이다. 그래서, Z 스키마의 술어부에 당초에 기술된 명세의 의미를 손실없이 달리 변환, 표현하여 검사를 수행했다.

관계 기반 Z 스키마 명세의 모델 검사의 결과를 나타낸 것이 그림 14이다. 그림 14에서, (a)는 'Mobile_Phone'과 'Calls' 휘쳐간 'mandatory' 관계에 대해 'syntax'와 'proof' 부분이 "Y"의 결과를 보임으로서 Z 명세가 정확함을 알 수 있다. (b)는 'Camera' 휘쳐가 'High_Resolution' 휘쳐를 요구하는 requires 관계를 보여준다. 'proof' 부분의 검사는 reduction의 'prove by reduce'를 수행하여 도메인 에리가 없다면 proof 된다.

결과적으로, 이러한 Z 모델 검사를 통해, 모바일 폰 휘쳐 모델의 구분적 타입 및 도메인 검사에서 오류가 없음을 입증할 수 있었다.

지금까지 휘쳐 모델의 정확성 검증을 위하여, 휘쳐 모델 - Z간 변환규칙을 정의하고, 이 규칙에 의거 Z 모델로 명세하고, Z/Eves 툴을 사용해서 Z 명세를 검사하였다. 본 장은 본 연구에서 제시한 휘쳐 모델과 Z간의 변환 규칙, Z 명세와 검사 방법 측면에서 기존 방법과 대비하여 비교평가 및 제시방법의 특징을 기술한다.

V. 비교 평가

5.1 기존 방법과의 비교평가

먼저, 휘쳐 모델을 Z 명세로 변환하는 규칙 측면에서 기존 방법과 제시 방법간의 상이점을 비교분석을 나타낸 것이 표 2이다. 2장에서 살펴 보았듯이 클래스 모델 - Z 간 변환에는 많은 연구가 있으나, 휘쳐 모델 - Z 간 변환에 대해서는 연구 [20]가 미진하며 그 외 휘쳐 모델의 PL/CSP/Alloy로의 정

형 명세 및 검사 등이 있다. 제시 방법은 기존 방법 대비해서, 휘처 모델 전체 구조상에서의 변환, 요소, 관계 및 다중성의 복합적 속성에 의한 좀더 명확한 변환을 제공한다 할 수 있다. 즉, 휘처 모델의 전체 구조를 고려한 변환에 있어서, 스키마들간의 선언 순서를 관계 유형(상속 등)을 고려하여 스키마들을 선언을 함으로서 구조적 오류를 제거하였다.

표 2. 휘처 모델 - Z간 변환방법 비교
Table. 2. Comparison with legacy translation methods between feature and Z

비교 항목	FSV-FM (20)	제시 방법
휘처	- 요소 기반 Z 스키마 변환	- 요소 기반 Z 스키마 변환
Metamodel 및 모델링 요소의 매핑	- 미 지원	- 휘처 및 Z의 metamodel 정의 - 휘처 및 Z 확장 모델링 요소 반영 - 모델링 요소간 변환 프로파일 정의
휘처 모델 전체구조	- 부분 지원 관계별 개별적 변환	- 휘처 모델의 전체 구조 반영
휘처간 Z 스키마 선언 순서	- 미 지원	- 지원 : 휘처 관계 유형 고려 (규칙 3-3과 규칙3-6)
휘처 관계 기반 Z 변환	- Mandatory/Optional/Alternative/Or/Requires/Excludes	- 관계 유형별 지원 - 휘처 관계별 스키마간 상호 관계도 명세 (포함 관계 등)
다중성	- 미 지원	- 관계 기반 변환시 다중성 반영

다음은 Z 명세와 Z/Eves 틀에 의한 모델의 검사 측면에서 본 연구 대비 기존 방법간의 비교평가를 보여주는 것이 표 3이다. 표 3에서 휘처 모델의 구문적 측면에서는 제시방법이 우수하나, 시멘틱 측면에서는 alloy의 정형적 언어를 사용한 추가적 휘처 모델의 일치성을 검사한 [20]이 양호함을 알 수 있다.

표 3. Z 명세와 모델 검사에 의한 평가
Table. 3. Evaluation for Z specification and model checking

평가 매트릭스		FSV-FM (20)	제시 방법
모델 명세 표현력	구문	보통	양호
	시멘틱 (정적, 동적)	양호 (Alloy사용)	부족
모델 검사 속성	구문	보통	양호
	시멘틱	양호	부족
정확성 (모델의 명세/검사)		양호	양호

5.2 특징 및 한계점

제시 방법의 특징 및 기대효과는 다음과 같이 들 수 있다.

- 휘처 모델 구조상의 Z 변환
 - 기존 방식은 휘처간 관계 기반의 개별적 변환이나, 제시 방법은 휘처의 상향식 접근에 의한 전체 구조 및 개별(요소/관계)적 변환이 통합된 Z 명세를 지원한다.
- 메타모델 차원이 아닌 휘처 모델 수준에서의 변환 방법 제시
- 식별형, 관계형 및 다중성 기반의 통합적 모델간 변환
 - 두 개 휘처간 요소 기반, 관계 기반 및 다중성이 고려된 변환을 제공한다.
- 휘처 모델의 확장된 모델링 요소 기반의 변환
 - 휘처 모델의 feature-attribute, composed-of, generalized-by 등 변환 제시한다.

제시 방법의 한계점은 다음과 같이 들 수 있다.

- 휘처 모델의 정적 및 동적 시멘틱에 대한 변환 명세 및 검사 방법을 지원하지 못한다.
- Z 명세의 Z/Eves 틀을 통한 모델 검사시, 틀 지원의 제약에 따른 Z 명세서 대비해서 완전히 일치된 검증을 못했다.

VI. 결 론

휘처 모델은 시각적 트리형의 정보 구조를 표현하고 있어, 모델의 구문적 정확성에 그 문제를 안고 있었다. 본 논문에서 휘처 모델을 Z로 명세하고, Z/Eves 틀을 통한 모델의 정확

성을 보였다. 이를 위해, 메타모델 기반의 휘처 모델 - Z간 변환 프로파일과 식별형(요소), 관계형(관계) 및 다중성에 기반한 변환 규칙을 정립했다. 이 변환 규칙을 모바일 폰 시스템의 휘처 모델에 적용해서 Z로 명세하고, 이를 검사 틀을 사용해서 모델의 구문 및 타입 검사, 그리고 도메인 검사를 수행했다. 이로써, 휘처 모델을 좀더 명확하게 표현할 수 있으며, 모델에 내재된 구조적 불일치를 Z로 검증할 수 있으며, 또한, 추가적 속성(well-formedness 등)들이 가미된 모델의 명세 및 검사가 가능하다.

미래 연구로서, 휘처 모델의 정형적(well-formed 등) 속성과 동적 시멘틱에 대한 정형적 명세와 이의 관련된 속성들에 대한 검증이 수반되어야 한다. 또한, 업그레이드된 Z/Eves 틀의 조사와 적용, 그리고 자동적 모델의 일관성 검사를 할 수 있는 Meta-CASE 틀의 개발이 이루어져야 한다.

참고문헌

- [1] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study", TechnicalReportCMU/SEI-90-TR-21, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, Nov. 1990.
- [2] Lee, K., Kang, K., Lee, J., "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering", Software Reuse: Method, Techniques, and Tools, Springer Berlin/Heidelberg, Vol. 2319, pp. 62-77, 2002.
- [3] 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.
- [4] M. Shroff, R. France, "Towards Formalization of UML Class Structures in Z", in Proc. of the COMPSAC '97, Washington DC, pp. 11-15, Aug. 1997.
- [5] S. K. Kim and D. Carrington, "Visualization of Formal Specification", in Proc. of the APSEC 98, Taipei, Taiwan, 1998.
- [6] S. K. Kim, D. Carrington, "A Formal Mapping between UML models and Object-Z Specification", The University of Queensland, Australia, Technical report 00-03, 2000.
- [7] Chee-Yang Song, "A Metamodel-Based Modeling Mechanism for Hierarchical Design in UML", Thesis for the Degree of Doctor, 2003. 7.
- [8] J.M. Spivey. "The Z Notation: A Reference Manual," 2nd Ed. Prentice Hall, 1992.
- [9] J. Woodcock, J. Davis, "Using Z: Specification, Refinement and Proof, International Series in Computer Science," Prentice-Hall, <http://www.usingz.com/text/online>, 1996.
- [10] M. Saaltink, "The Z/EVES system", <http://link.springer.com/chapter/10.1007/BFb0027284>.
- [11] M. Saaltink. "The Z/EVES 2.0 User's Guide," Technical Report TR-99-5493-06a, ORA Canada, CANADA, Oct. 1999.
- [12] D. Benavides, S. Segura, A. Ruiz-Cortés, "Automated Analysis of Feature Models 20 Years Later: A Literature Review," Journal of Information Systems, <http://www.lsi.us.es/~dbc/en/?download=benavides10-is.pdf>, March 2010.
- [13] S. B. Lee, J. W. Kim, C. Y. Song, J. H. Kwon, T. W. Lee, H. S. Kim, D. K. Baik, "Ontology-based approach to analyzing commonality and variability of features in the Software Product Line Engineering," Korea Institute of Information Scientists, Vol. 34, pp. 196-211, 2007.
- [14] M. Riebisch, K. Böllert, D. Streitferdt, and I. Philippow. "Ex-tending feature diagrams with UML multiplicities," In 6th World Conference on Integrated Design & Process Technology (IDPT2002), June 2002.
- [15] K. Czarniecki, S. Helsen, and U. Eisenecker. "Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice," Vol. 10, No.1, pp. 7-29, 2005.
- [16] D. Benavides, A. Ruiz-Cortés, P. Trinidad. "Automated reasoning on feature models," In Advanced Information Systems Engineering:

17th International Conference, CAiSE, LNCS, Springer-Verlag, Vol. 3520, pp. 491-503, 2005.

[17] Gi-Hwon Kwon, "Object-Oriented Animation of Z Formal Specification", Korea Information Science Society, Software Engineering Society Journal, Vol. 23, No. 12, pp. 1257-1269, Dec. 1996.

[18] G. Smith, "The Object-Z Specification Language," Kluwer Academic Publishers, 1999.

[19] M. Janota, J. Kiniry, "Reasoning about Feature Models in Higher-Order Logic," 11th International Software Product Line Conference, IEEE Computer Society, 2007.

[20] J. Sun, H. Zhang, Y. F. Li, H. Wang, "Formal Semantics and Verification for Feature Modeling", Proceeding ICECCS '05 Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems, <http://elec.uq.edu.au/~liyf/papers/sun-Feature.pdf>, pp. 303-312, 2005.

[21] H. Wang, Y. F. Li, J. Sun, H. Zhang, J. Z. Pan, "Feature Reasoning - Scaling Up and Pinning Down," <http://homepages.abdn.ac.uk/jeff.z.pan/pages/pub/WFSZP07.pdf>, 2005.

[22] E. S. Cho, C. J. Kim, "A Formal Specification of Reusable Framework of Embedded System," Korea Information Processing Society," Vol. 17-D, No. 5, pp. 431-442, 2010. 12.

[23] D. L. Kim, C. Y. Song, D. S. Kang, D. K. Baik, "An Ontology-based Transformation Method from Feature Model to Class Model," The Korea Society of Computer and Information, Vol. 13, No. 5, pp. 53-67, 2008. 9.

저 자 소 개



송 치 양
 1985: 한남대학교 전자계산학과 공학사.
 1987: 중앙대학교 전자계산학과 공학석사.
 2003: 고려대학교 컴퓨터학과 이학박사
 1990~2005: KT 중앙연구소
 책임연구원
 2005~2008: 상주대학교
 소프트웨어공학과 조교수
 현 재: 경북대학교 컴퓨터정보학부 부교수
 관심분야: CBD, MDA, Meta-Model,
 Formal Specification,
 IPTV Service
 Email : cysong@knu.ac.kr



조 은 속
 1993: 동의대학교 전산통계학과 공학사.
 1996: 숭실대학교 컴퓨터학과 공학석사.
 2000: 숭실대학교 컴퓨터학과 공학박사
 2000~2005: 동덕여자대학교
 정보학부 강의전임교수
 현 재: 서일대학
 컴퓨터소프트웨어과 부교수
 관심분야: CBSE, Embedded Software,
 Web Service, SOA,
 Service-Oriented Computing
 Email : escho@seoil.ac.kr



김 철 진
 1996: 경기대학교 전자계산학과 공학사.
 1998: 숭실대학교 컴퓨터학과 공학석사.
 2004: 숭실대학교 컴퓨터학과 공학박사
 2004~2005: 가톨릭대학교
 컴퓨터정보공학부
 강의전담교수
 2004~2009: 삼성전자 책임연구원
 현 재: 인하공업전문대학
 컴퓨터시스템과 조교수
 관심분야: CBD, Component
 Customization, Embedded
 Software
 Email : cjkim@inhac.ac.kr