# Z 언어를 이용한 보안시스템의 명세 규칙

김명재*, 이형효**, 노봉남***
전남대학교 정보보호협동과정*
원광대학교 정보·전자상거래학부**
전남대학교 전산학과***
e-mail:markkim@athena.chonnam.ac.kr

# A Study on the Specificaton Rules for Security Systems in Z

Myong-Jae Kim* , Hyong-Hyo Lee** , Bong-Nam Noh***
Dept. of Information Security, Chonnam National University*
Division of Information and Electronic Commerce, Wonkwang University**
Dept. of Computer Science, Chonnam National University***

요 약

전통적으로 소프트웨어 공학분야에서는 소프트웨어 설계 및 개발을 위한 자동화도구와 정형화기법이 적용되었으며, 안전한 보안시스템의 개발을 위해서도 수학적인 명세와 검증도구를 사용하는 경우, 보다 안전하고 완전한 보안시스템을 구축할 수 있다. 본 연구는 정형화 명세언어인 Z를 이용하여 보안시스템을 기술하고, 기술된 명세를 소프트웨어 검증도구를 이용하여 보안시스템이 제공하는 보안특성 만족여부 점검을 목적으로 한다. 이를 위해 본 논문에서는 수학적으로 기술된 시스템에 대한 명세를 Z스키마로 변환하는 변환규칙을 기술하고, 이와 함께 Z언어 검증도구인 Z/EVES를 이용한 검증결과를 제시한다.

## 1. 서 론

정보보호제품 시장이 민간기업에 까지 확대됨에 따라 정보보호시스템의 안정성이 강조되면서 수학적 논리학과 이산 수학을 기반으로 된 정형화된 명세언어가 필요하게 되었다. 여러 선진국에서는 신뢰성을 갖춘 안전한 시스템을 개발하도록 유도하고 사용자에게도 시스템의 보호 수준에 적합한 시스템을 구매할 수 있도록 하고 있다[10].

시스템을 개발하는데 있어서 개발을 의뢰한 고객 또는 실제 사용할 사용자와의 만남은 매우 중요하다. 완성된 시스템이 고객의 요구와 다르게 만들어지는 경우를 종종 볼 수 있다. 이러한 요인들 중에는 시스템의 개발단계에서 볼 때 요구사항의 부정확성(imprecision), 모호성(ambiguity), 불완전성(incompleteness), 이해 오류(misunderstanding) 등이 있다[4].

이러한 문제에 대한 방안으로 정형화된 명세언어들이 출현했고, 정형 명세로 구현된 시스템이 가장 정확한 명세가 이루어져 신뢰성이 높다는 결론을 내렸다. Z 언어와 같은 명세언어들은 시스템이 무엇을 해야 하는 지를 기술할 수 있는 장점 때문에 초기에 발생하는 문제를 줄일 수 있다. 명세언어는 의미가 명확한 수학적 기호를 이용하고, 집합(Sets),관계(Relations),함수 (Functions)등을 가지는 집합론에 기초를 두고 있다[5].

이처럼 수학적 기법인 정형논리, 이산수학 등을 이용하여 시스템을 개발하는 과정에서 필요한 요구명세

의 제시 및 이를 확인시키는 정형 기법은 수학에서 다루는 논리 및 증명체계와 밀접한 관련이 있다. 이러한 장점에도 불구하고 여전히 명세언어 자체만으로는 검증 도구에 적용하는 데 어려 움이 있다. 그 이유는 수학적인 기호나 자연어를 명세언어로 변환하는데 한계가 있기 때문이다. 이를 극복하기 위해 본 논문에서는 명세언어를 툴에 적용할 때 일어나는 몇 가지 변환 규칙들을 알아보겠다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 Z 언어와 Z/EVES[2]에 대한 특징에 대해서 알아보고 3장에서 CoreRBAC[3]의 한 예를 Z언어로 명세했을 때 적용되는 변환 규칙에 대해 알아보고 4장에서는 3장의 규칙들을 실제로 Z/EVES 도구를 통해 보여주고 마지막으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

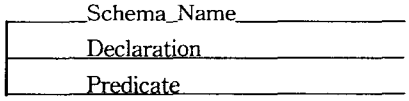
## 2. Z언어와 Z/EVES의 특징

이 장에서는 Z언어 와 Z/EVES의 특징에 대해서 알아보고 Z언어로 시스템 모델링할 때 시스템의 상태 및 동작 특성을 나타내는 스키마에 대해 기술한다[2].

### 2.1 Z 언어

Z 명세언어는 자료형으로 이루어진 객체를 갖는 정보시스템의 수학적 모델이다. 따라서 Z 명세에 나타난 모든 표현식(expression)은 자료형을 갖게 된다[8]. Z 언어는 1970년 후반에서 1980년대 초반에 영국 옥스퍼드 대학에서 개발된 것으로 현재는 학문적인 연구범위를 벗어나, 산업환경에서 대형 소프트웨어 시스템인 IBM의 고객정보 제어 시스템 (CICS)을 재명세하는 등 크게 성장하였다[7].

Z 언어에서 사용되는 자료형은 기본 자료형, 단순 자료형, 공리, 스키마 등이 있다[11]. 특히 Z 언어에 의해 모델링되는 시스템의 상태와 동작 특성을 기술한다. 스키마는 Z 언어의 중추적 역할을 담당하고 있으며 시스템이 체계적이고 모듈화된 설계 기능을 제공한다. 스키마는 그림 1과 같이 두 가지 표현할 수 있다[5,11].




SchemaName ≙ [declaration | predicate]

(그림 1) 스키마 표현

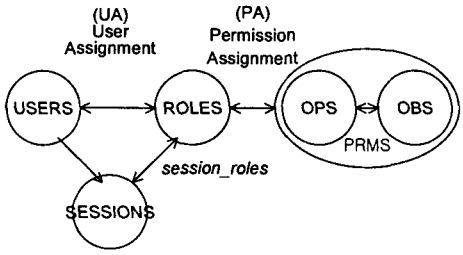
### 2.2 Z/EVES

본 논문에서는 명세된 스키마를 검사하기 위해서 캐나다 ORA사가 개발한 Z 언어명세 분석 도구인 Z/EVES를 사용하였다. Z/EVES 도구는 Z 자료형이나 스키마의 타입 체킹(type checking), 도메인 체킹(domain checking) 그리고 시스템의 특성에 대한 내용을 정리(theorem)를 만들어서 만족 여부를 검증할 수 있다[2,9].

본 논문에서는 Z/EVES를 Z언어로 쓰여진 스키마의 타입 체킹(type checking)을 하는 데 사용하였다.

## 3. 명세규칙

이 장에서는 현재 연구가 진행중인 일반적인 수식이나 형태(Form)를 Z로 명세할 때 일어나는 규칙들을 보여준다. 이러한 규칙들에서 특히 CoreRBAC 시스템을 예로 들어 기술한다[3].




(그림 2) CoreRBAC 구성요소

**[규칙①] 기본 자료형 정의 규칙**

시스템명세에서 사용되는 자료형은 Z에서 기본 자료형으로 정의한다.

예) [NAME]

Z에서 기본 자료형은 bracket내에 자료형의 이름을 나열함으로써 정의된다. 이때 NAME은 Z 명세에서 사용될 수 있는 전역(global)기본 자료형으로 선언된다. 그러나 Z 언어에서는 기본 자료형으로 선언되는 자료형의 실제적인 구조나 구현방법과는 무관하다[6].

**[규칙②] 변수, 상수 정의 규칙**

Z 명세에 쓰여지는 모든 변수(variables) 와 상수(constants)는 이미 정의되어있는 기본 자료형 타입으로 정의한다[2].

예) USERS: ℙNAME
new_name: NAME

또한 검증 도구내에 이미 정의되어있는 자료형은 정의 할 필요없이 사용한다.

예) max_number : N

**[규칙③]** 카티션 프로덕트(Cartesian Product)규칙
카티션 프로덕트 Y × Z 는 type(Y) ↔ type(Z)로 변환한다.

예) UA ⊆ USERS × ROLES ⇒

UA_schema

| |
|---|
| USERS:ℙNAME |
| ROLES:ℙNAME |
| UA:NAME ↔ NAME |
| dom UA ⊆ USERS |
| ran UA ⊆ ROLES |

X ⊆ Y × Z ⇒

스키마 이름

| |
|---|
| X : type(Y) ↔ type(Z) |
| dom X ⊆ Y |
| ran X ⊆ Z |

**[규칙④]** 멱집합(Power Set)에서의 카티션 프러덕트(Cartesian Product)의 규칙
$2^{(Y \times Z)}$ 는 ℙ(type(Y) × type(Z))로 변환한다.

예) PRMS = $2^{(OPS \times OBS)}$ ⇒

PRMS_schema

| |
|---|
| OPS : ℙNAME |
| OBS : ℙNAME |
| PRMS : ℙ(NAME × NAME) |
| dom PRMS ⊆ OPS |
| ran PRMS ⊆ OBS |

X = $2^{(Y \times Z)}$ ⇒

스키마이름

| |
|---|
| X : ℙ(type(Y) × type(Z)) |
| dom X ⊆ Y |
| ran X ⊆ Z |

**[규칙⑤]** 집합과 멱집합(Power Set)에서 함수 관계의 규칙
$2^Z$ 는 멱집합(Power Set)으로 변환한다.

예) assigned_users: (r:ROLES)→$2^{USERS}$ ⇒

assigned_users_schema

| |
|---|
| ROLES : ℙNAME |
| USERS : ℙNAME |
| ssigned_users : NAME → ℙNAME |
| dom assigned_users ⊆ ROLES |
| ran assigned_users ⊆ ℙUSERS |

X : Y → $2^Z$ ⇒

스키마이름

| |
|---|
| X : type(Y) → ℙtype(Z) |
| dom X ⊆ Y |
| ran X ⊆ ℙZ |

표 1은 3장과 4장에서 활용된 여러 용어정리를 한 것이다.

<표 1> 주요 용어 정리

| 용 어 | 내용정리 |
|---|---|
| ℙ | 멱집합(Power Set) |
| × | 카티션 프로덕트 |
| dom | 정의역(domain) |
| ran | 치역(range) |
| [NAME] | 기본 자료형 |
| USERS | 사용자 |
| ROLES | 역할 |
| OPS | 연산(operation) |
| OBS | 객체(object) |
| UA | User Assignment |
| PRMS | Permission |
| assigned_users | maps a role onto a set of users |
| assigned_permission | maps a role onto a set of permission |

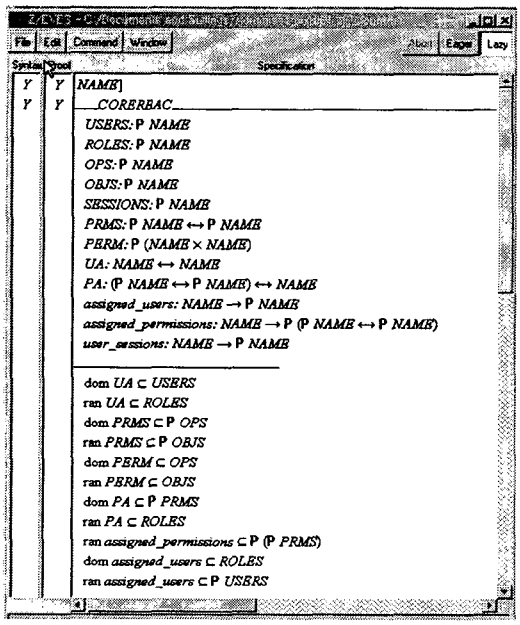
## 4. Z/EVES를 이용한 검사

본 논문의 3장에서 CoreRBAC 모델을 Z/EVES를 이용하여 명세할 때 필요한 규칙들을 기술하였다.

이러한 규칙들을 토대로 이번 장에서는 CoreRBAC 모델의 상태 스키마(state schema)의 부분을 Z언어로 명세 한 다음 Z/EVES 도구를 이용해 타입 검사와 변수의 타당성 검사를 수행한 후 결과를 제시한다. 이러한 과정을 통하여 우리가 명세하고자

하는 타입이 맞는지 그리고 다른 예를 통해서도 이러한 규칙이 적용되는지 알아보았다.

그림 3은 직접 Z/EVES를 통해서 CoreRBAC 모델을 명세하고 검증한 내용이다.




(그림 3) CoreRBAC을 Z/EVES로 구현

## 5. 향후연구

서론에서 언급했듯이, 시스템을 설계할 때 정형명세를 이용하는 것은 매우 중요하다. 개발단계에서부터 해당 시스템에 대한 정확한 명세를 하여 시스템에서 발생할 수 있는 오류를 줄일 수 있고 신뢰성있는 시스템을 개발할 수 있다. 보안시스템을 설계하는 설계자의 입장에서는 정형명세 뿐만 아니라 명세한 내용을 검증 도구에 어떻게 적용하는지가 중요한 문제로 대두되었다.

본 논문에서는 명세언어에 대한 접근의 어려움 때문에 힘들어하는 설계자를 위해 수학적으로 기술된 시스템의 특성을 Z 명세언어를 이용하여 기술하는데 필요한 변환 규칙들을 제시하였다. 앞으로의 연구방향으로는 사용자 측면에서 이용하기 쉽고 보다 완전한(complete), 안전한(secure) 보안시스템을 개발하는데 필요한 규칙들을 좀더 정리하고 체계화를 할 수 있도록 해야되고, 검증방법에 있어서 수학적 정리를 이용한 정리 증명(theorem proving)을 계속 연구 중에 있다.

명세언어가 보다 쉽고 명확하게 검증 도구에 사용될 수 있도록 여러 변환규칙이나 검증 방법들이 연구되어야 할 것이다.

<참고문헌>

[1] J. M. Spivey, "The Z Notation: A Reference Manual Second Edition", Prentice Hall, 1992

[2] Mark Saaltink. "The Z/EVES User's Guide", TR-97-5493-06, ORA Canada,1997

[3] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, and D. Richard Kuhn, "Proposed NIST Standard for Role-Based Access Control", ACM Transactions, pp.230-238. 2001

[4] 유희준, "보안기능의 정형화 설계", KIISC REVIEW, 2000

[5] David Rann, John Turner, and Jenny Whitworth, "Z: A Beginner's Guide", CHAPMAN & HALL, 1994

[6] Ben Potter, Jane Sinclair, and David Till, "An Introduction to Formal Specification and Z", Prentice Hall, 1991

[7] Iain S. C. Houston and Steve King, "CICS project report", Springer-Verlag, 1991

[8] James C.P Woodcock and Jim Davis, "Using Z: specification, proof and refinement", Prentice Hall, 1995

[9] http://www.afm.sbu.ac.uk/, "Formal method"

[10] 최진영, 서동수, "정보보호 시스템 정형 화 명세서", KISA 정보보호뉴스, 1999

[11] 노봉남 외, "BLP, Biba등 보안모델을 적용한 보안정책 평가방법 연구", 한국정보보호센터, 1999