

# Rule Database를 활용한 보험상품 규칙시스템의 설계 및 구현

김도형\*, 이유호\*, 오영배\*\*

\*삼성 SDS, \*\* 수원여대 컴퓨터응용학부

## Design and Implementation of Rule-based System for Insurance Product

Kim, Do-Hyung, Lee, You-Ho, Young Bae Oh

Samsung SDS, Suwon Women's College

E-mail : [dohyung1.kim@samsung.com](mailto:dohyung1.kim@samsung.com), [yboh@suwon-c.ac.kr](mailto:yboh@suwon-c.ac.kr)

### 요 약

보험시스템은 상품 및 보험 종류에 따라서 결정되는 요소들이 많고 이에 대한 예외 사항이 많이 존재하는 특성을 가지고 있다. 기존 시스템에서의 상품속성 반영은 테이블을 통한 값 정의와 어플리케이션에서의 예외처리 로직 (if then else)을 병행하여 사용함으로써, 상품변경과 신상품 개발에 대한 비용이 증가하고 신속한 시장 대응이 어려웠다. 본 논문에서는 보험상품 속성의 비즈니스 로직을 데이터화로 가능하게 하는 Well Formed Rule Base 시스템을 제시하고 실제 프로젝트 적용을 통한 효과를 설명한다.

### 1. 서론

보험시스템은 상품 및 보험 종류에 따라서 결정되는 요소들이 많고 이에 대한 예외 사항이 많이 존재하는 특성을 가지고 있다. 기존 시스템에서의 상품속성 반영은 테이블을 통한 값 정의와 어플리케이션에서의 예외처리 로직 (if then else)을 병행하여 사용해 왔다. 신상품 추가 및 변경 시 프로그램을 일일이 확인하여 반영해야 하는 작업이 빈번히 발생하므로 인해 유지보수 및 테스트에 대한 부담과 신속한 대응에 어려움을 갖고 있었다. 이러한

특성으로 규칙시스템에 대한 필요성이 증대되어 왔다.

규칙시스템은 상품 속성에 대한 테이블화(전통적인 방법)와 예외사항을 규칙시스템에 표현(데이터화)하여 사용한다. 신상품 추가 및 변경시 어플리케이션의 수정이 아닌 규칙 데이터의 추가 및 변경 방식으로 전환 가능하여 유지보수성, 신뢰성, 신속성 등을 향상시킬 수 있다. 궁극적으로는 신상품 반영 작업을 전산실이 아닌 현업에서 시스템을 통하여 반영할 수 있는 체제로 전환하는데 도움을 줄

수 있다. 비즈니스 로직의 데이터화는 유사업종의 프로젝트 진행시 이전 고객사의 업무 노하우 유출을 줄일 수 있어 재사용 프로젝트가 가능하게 하는 한 원인이 되었고, 전체 시스템이 계층별, 기능별로 컴포넌트화 되어 있어 프로젝트에 특화된 기능을 선별하고 커스터마이징 하는 작업을 수월하게 수행할 수 있었다.

## 2. Rule 기반 상품규칙 시스템의 설계 및 구조

### 2.1 보험시스템과 상품규칙 시스템과의 관계

일반 보험 시스템의 Application은 상품관련 속성을 얻기 위하여 규칙해석엔진에서 제공하는 단일 인터페이스를 통하여 규칙해석모듈을 호출하고, 규칙해석모듈은 룰마트 DB에 정의된 정보를 이용하여 상품속성을 리턴해 주므로, 일반 업무용 Application은 상품처리를 위한 로직을 구현할 필요가 없다. <그림1>은 보험시스템과 상품규칙시스템 연관도를 나타낸다. 규칙데이터는 <그림1>의 아래부분에 위치한 규칙관리화면을 통하여 입력할 수 있고 입력된 상품규칙 데이터는 성능을 고려한 룰마트 DB로 자동 변환된다.

### 2.2 조건규칙 데이터 모델 및 구조

<그림2>는 조건규칙 데이터 모델 구조를 나타내며 모든 조건규칙 자료는 Conjunctive Normal Form 형태로 저장된다. 상품 속성이 상품코드와 보험코드의 조합으로 결정되지 못하고 조건에 따라 속성이 결정될 때 사용한다.

If a=1 and b=2 then c=3

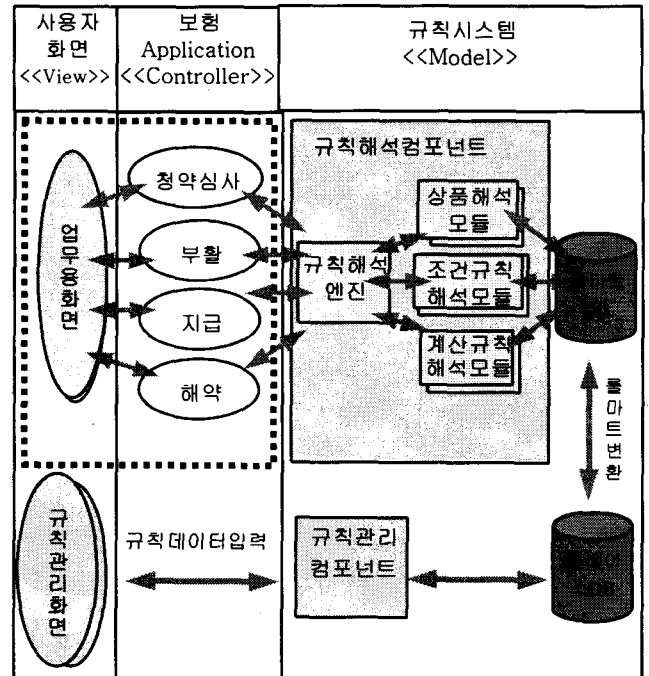
If a=2 and b=1 then c=1 과 같은 조건식을 예로 들어 보면, 앞의 조건식은 조건절(If절)과 Then 절

(Action)로 나눌수 있고 조건절은 다시 단위조건식(Atomic predicate)으로 구성된다. Atomic Predicate에는 a=1, b=2와 같은 자료가 저장되고 If Clause에는 If문을 구성하는 Atomic Predicate의 관계정보

(ApsANDed in If Clause)로 표현된다.

Then 절을 표현하는 Action은 Then 절에서 결과값으로 사용되는 항목(Action Domain)과 실제 결과값(Action Domain Value)의 형태로 저장된다.

한 개의 규칙정보(Rule Subject)는 이와 같은 If Clause와 Action의 조합정보(Rule)로 표현된다.

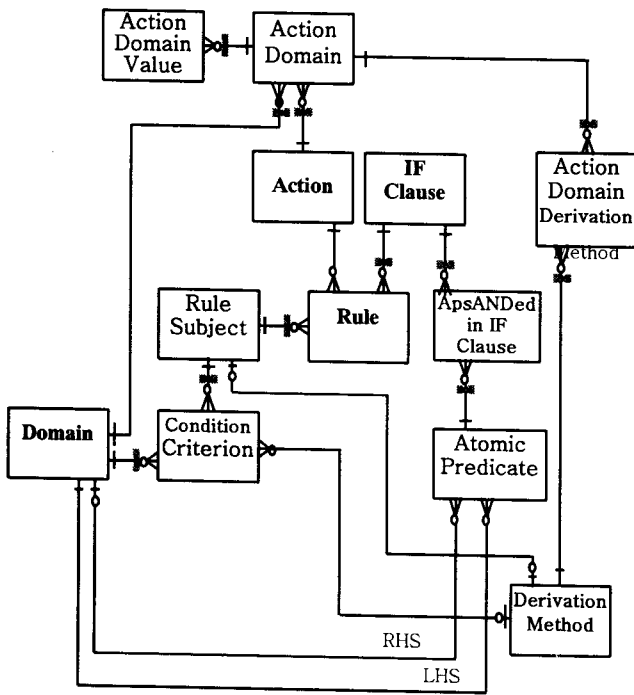


<그림1> 보험시스템과 상품규칙시스템 연관도

### 2.3 계산규칙 데이터 모델 및 구조

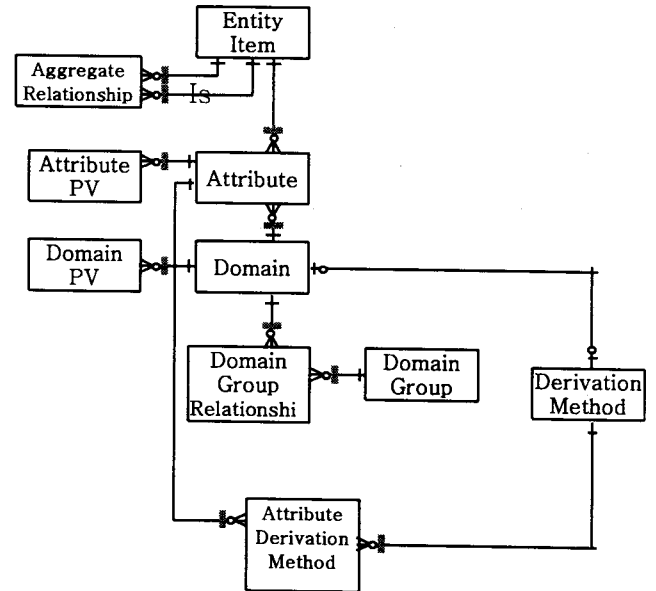
<그림3>은 계산규칙 데이터 모델을 표현한다. 계산 규칙은 계산식을 이항 연산식으로 전개한 형태로 저장된다. 단위 계산규칙을 나타내는 Calc Formula는 해당 계산규칙에서 사용하는 In/Out항목(Calc Formula Signature)와 실제 전개된 순서를 갖고 있는 이항식정보(Calc Expression)형태로 저장된다. 미상각예정 신계약비를 계산하는 예를 들면, { 미상각예정 신계약비= 예정 신계약비 \* (상각 기준년수 - 보험료 납입 경과기간) / 상각 기준년수 }는 아래와 같은 이항식으로 전개된다.

- 1 번식 : 상각 기준년수 - 보험료 납입 경과기간
- 2 번식 : 예정 신계약비 \* 1번 연산식 결과
- 3 번식 : 2번 연산식 / 상각기준년수

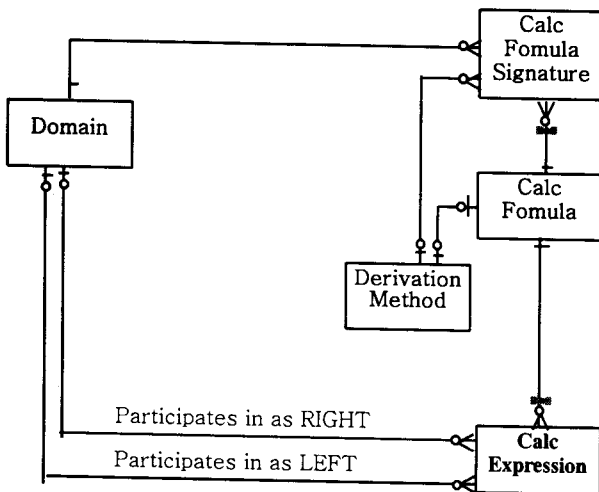


<그림2 조건규칙 데이터 모델>

있고 이에 대한 실제 값(Attribute PV)을 상수값, 구간값, 허용값, 복수값, 조건식, 계산식등의 다양한 방법으로 정의할 수 있도록 수용한 구조이다. 상품의 항목을 그룹핑하거나 순서를 정의할 수 있고 기존 테이블 단위에 매핑할 수 있다.



<그림4> 상품규칙 데이터 모델



<그림3> 계산규칙 데이터 모델

## 2.4 상품규칙 데이터 모델 및 구조

<그림4>는 상품의 속성값을 정의할 수 있는 상품규칙 데이터 모델 구조로서 개개 상품 및 보험코드 (Entity Item) 혹은 이에 대한 구조(Aggregate Relationship)에 대하여 항목(Domain)을 정의할 수

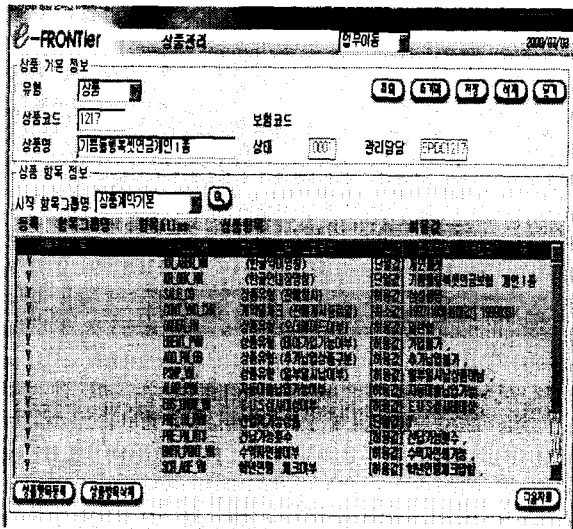
## 3. 규칙시스템

규칙시스템은 보험상품 속성의 비즈니스 로직을 정의하는 규칙관리 컴포넌트와 규칙시스템에 저장된 데이터를 해석하여 서비스를 제공하는 규칙해석 컴포넌트로 구성된다. 본 절에서는 각 컴포넌트의 중요기능을 살펴본다.

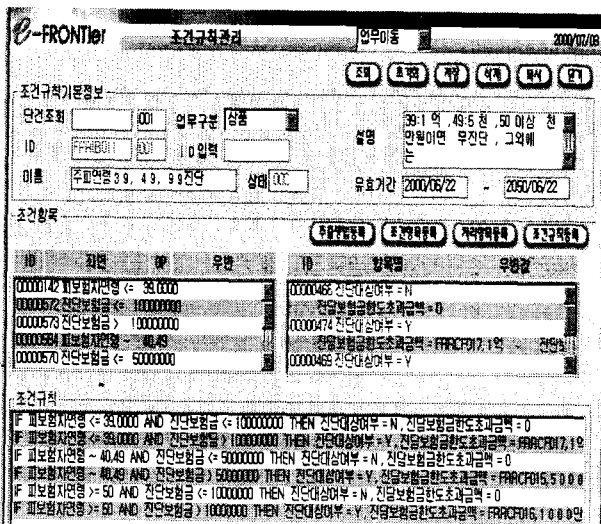
### 3.1 규칙관리 시스템

규칙관리시스템은 상품의 속성값을 정의하는 상품규칙관리 시스템, 보험상품 조건규칙을 정의하는 조건규칙관리 시스템과 상품속성에 다양한 계산식을 정의하는 계산규칙관리 시스템으로 구성되어 있다. 규칙관리시스템은 규칙시스템 데이터의 입력/수정/삭제/조회 기능과 권한관리 및 변경관리 기능을 포함하며 Web 사용자 화면을 통하여 기능을 제공한다.

<그림5>는 상품규칙관리 시스템의 샘플 화면으로서, 상품규칙관리 기능은 기존 상품 테이블과 기능이 유사하나, 테이블에 저장할 상품속성을 데이터화 하여 항목 추가 및 변경, 조합 등을 비교적 자유롭게 수행할 수 있다. 상품이나 보험코드의 조합으로 항목 값이 상수로 정해지지 않을 경우, 조건규칙이나 계산규칙으로 정의한다.

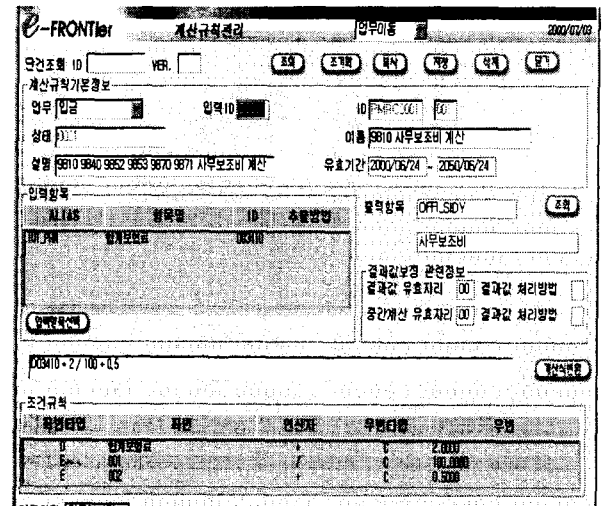


<그림6>은 조건규칙관리 시스템의 샘플 화면으로서, 조건규칙은 프로그램에 표현된 if then else 로직을 데이터로 저장하며, 해석 요청시 규칙데이터와 입력으로 들어온 데이터를 사용하여 원하는 결과를 제공한다. 조건규칙은 조건항목(if절)과 처리항목(then 절)의 조합으로 구성된 정규화 형태로 테



<그림7>은 계산규칙관리 시스템의 샘플 화면을 나타내며, 시스템에서 상품속성을 정의하는 계산식을 데이터로 저장하며, 어플리케이션에서 호출시 계산식 결과를 제공한다. 일반적으로 계산규칙 단독으로 사용하는 경우보다 조건규칙에서 계산식을 표현하는 보조수단으로 주로 사용한다.

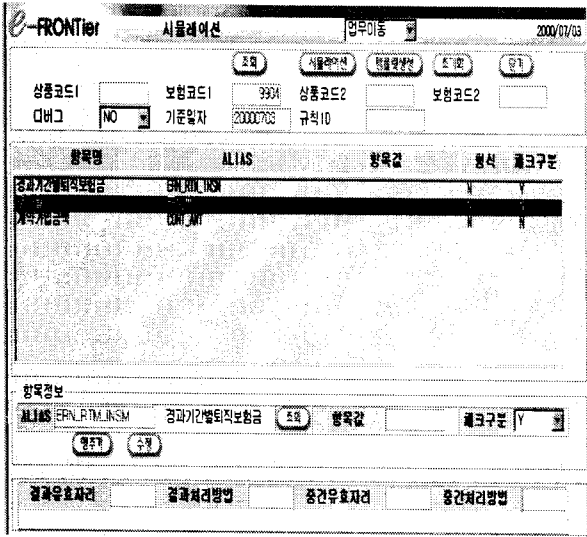
<그림6> 조건규칙 관리 화면



### 3.2 규칙해석 시스템

규칙시스템에 저장된 자료는 원칙적으로 해석 모듈을 통하여 서비스를 제공한다. 사용자는 (Application 개발자) 해석모듈에서 제공하는 단일 인터페이스를 통하여 모듈을 호출 함으로써 사용할 수 있다. 일반 업무 프로그램 작성자는 상품 속성 관련 기능은 규칙시스템에 맡기고 고유 업무로직 구현에 충실 할 수 있다. 성능향상을 위하여 해석 모듈은 비정규화 된 형태의 룰마트 DB를 사용하며 규칙관리시스템에서 자료 입력시 규칙 메타모델 형태의 정규화DB와 룰마트DB에 동시에 Update 한다. 온라인용과 배치용의 해석모듈을 제공하며, 어플리케이션에서 해석모듈을 호출할 경우 원하는 대로

결과 값을 가져오는지를 확인할 수 있는 규칙 테스트 기능을 제공한다.



<그림8> 규칙해석 시뮬레이션(테스트) 화면

### 3.4 적용 사례

규칙시스템의 운영 및 개발 환경은 IBM MainFrame WebToHost 환경을 기본환경으로 하고 있으며, Operating System은 OS/390 (MVS), Web Application Server는 CICS Transaction Server, DBMS는 DB2를 사용하고 있다. 본 절에서는 실제 적용하여 운영되고 있는 사례에 대하여 살펴본다. 아래의 적용 결과에서는 본 논문의 목적상 보험사의 실제 명칭은 언급하지 않고 A 및 B사로 표현한다.

- **A사:** 계정계 신시스템 프로젝트 (1999년-2000년, 2년) 수행시 규칙시스템을 구축, 적용하였다. 현재 운영중인 규칙데이터 건수는 약 250만건 (상품/보험코드 \* 정의항목) 이상으로 예상된다. 기존 Application의 로직을 데이터화 하였으므로 적용시 성능에 대한 우려가 높았으나, 규칙 DB의 2차에 걸친 비정규화를 통하여 계정계 시스템에 적용되었다. 보유/운영 데이터 건수와 트랜잭션 발생건수가 매우 높은 사이트에 적용되어 규칙시스템의 업계 상용화가 가능한

것으로 입증되었다.

- **B사:** 계정계 신시스템 프로젝트 (2001년-2002년, 1년) 수행시 기존 구축되어 운영하고 있는 A사의 보험시스템을 도입하기로 결정하였고, 패키지 형태의 커스터마이징 프로젝트 형태로 진행하였다. 운영중인 데이터 건수는 약 200만건 이상으로 예상된다. A사와 매출 및 보유 계약 건수 대비 예상보다 높은 비율의 규칙데이터가 운영되고 있다. A사 적용시 보다 운영 데이터와 트랜잭션 발생건수가 상대적으로 유리하였으나, A사 대비 하드웨어 성능에 차이가 존재하였다. B사의 적용으로 하드웨어 성능에 크게 의존적이지 않음이 어느정도 입증되었다.

B사의 프로젝트 수행시 A사 프로젝트를 통해 개발된 비즈니스 컴포넌트 재사용을 통하여 3배의 개발생산성 향상을 달성하였으며 개발원가 및 품질 지표에도 예상보다 많은 효과가 있음을 확인할 수 있었다. 이러한 결과는 소프트웨어 개발시, 모듈간의 재사용성과 독립성 및 유지보수성 향상을 목표로 계층별, 업무기능별 모듈화 및 컴포넌트화 된 구조를 통하여 달성할 수 있었다.

사용자 측면에서도 규칙시스템의 도입으로 초기에 목표한 상품변화에 대한 신속한 대응, 상품 유지보수 비용 감소, 현업주도의 상품관리가 실현 가능하게 되었다.

## 4. 결론

본 논문에서는 보험상품 속성의 효율적 표현 및 관리를 위해 IF/THEN 구조의 Rule Base를 제안하였다. 본 연구에서 제시한 Well Formed Rule Base를 적용함으로써 얻을 수 있는 이점으로는, 보험 신상품 추가 및 변경 시 어플리케이션의 수정이 아닌 규칙 데이터의 추가 및 변경 방식으로 전환이 가능하여 시스템의 유지보수성, 신뢰성 및 신속성 등을 획기적으로 향상시킬 수 있었다. 궁극적으로 신상품 반영 작업을 전산실이 아닌 현업에서 시스템을 통하여 처리할 수 있는 체제로 전환이 가능하고 궁극적

으로 급변하는 경영환경 변화에 신속한 대응이 가능하여 기업의 경쟁력을 제고 할 수 있다.

특히, 소프트웨어 개발업체 입장에서는 비즈니스 로직의 데이터화는 유사업종의 프로젝트 진행 시 이전 고객사의 업무 노하우 유출을 방지 할 수 있어 기 개발된 비즈니스 컴포넌트를 재사용 이 가능하고, 전체 시스템의 계층 및 기능별 컴포넌트화를 통해 프로젝트에 특화 된 기능을 선별하고 커스터마이징 하는 작업을 수월하게 수행할 수 있다.

### [참고문헌]

- [1] Aberdeen Group, “The Power of Rules-Driven Processing.”, Internal Report, Aberdeen Group, Boston, Sept.2000.
- [2] 황혜수의 3인, “룰 기반 CRM 시스템에서 시간 요소를 고려한 룰 모델 제안”, 정보과학회 2001년 추계학술대회, VOL.28, NO.2, pp.283~285, 2001.
- [3] 황혜정의 3인, “질의 완화를 이용한 지능적인 질의 응답 시스템”, 한국정보처리학회 논문지 A, VOL.7, NO.1, pp.88~98, 2000.
- [4] 김철수, “웹 고객의 개인화를 지원하는 지식기반 통합시스템”, 한국정보처리학회 논문지 B, VOL.9-B, NO.1, pp.1~6, 2002.
- [5] 최중민, “인공지능”, 사이텍미디어, pp229~323, 1998.
- [6] Khawar Zaman Ahmed, “Developing Enterprise Java Applications with J2EE and UML”, Addison-Wesley.
- [7] Stephen Stelting, “Applied Java Patterns”, Sun Microsystems.