

Rule 기반 상품규칙 시스템의 설계

김도형* · 오영배**

Design of Rule-based System for Insurance Product

Do-Hyung Kim* · Young-Bae Oh**

■ Abstract ■

Insurance system has a lot of decision factors which are affected by the kinds of insurance products, and has the features of many exceptions. Since in applying the product attributes to the current system the value definition through tables and the exception treatment logic (if then else) are used in parallel, the cost of a product change and a new product development becomes increased and the prompt market reaction is difficult. In this paper, we propose the well formed rule base system which makes data for the business logic of insurance attributes and discuss the benefit of application of this system to the real project.

Keyword : Rule Base, Well Formed System, Conjunctive Normal Form, Rule Data Mart

1. 서론

보험 시스템은 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 애플리케이션에서의 예외처리 로직(if then else)을 병행하여 사용해 왔다. 신상품 추가 및 변경 시 프로그

램을 일일이 확인하여 반영해야 하는 작업이 빈번히 발생하므로 인해 유지보수 및 테스트에 대한 부담과 신속한 대응에 어려움을 갖고 있었다. 이러한 특성으로 규칙 시스템에 대한 필요성이 증대되어 왔다.

규칙 시스템은 상품 속성에 대한 테이블화(전통적인 방법)와 예외사항을 규칙 시스템에 표현(데이터화)하여 사용한다. 신상품 추가 및 변경시 애플리케이션의 수정이 아닌 규칙 데이터의 추가 및 변

* 삼성 SDS

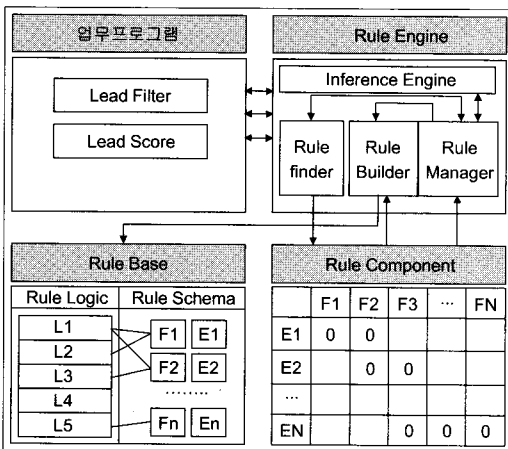
** 수원여대 컴퓨터응용학부

경 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 규칙 시스템의 설계 개념

2.1 Rule Base 구조

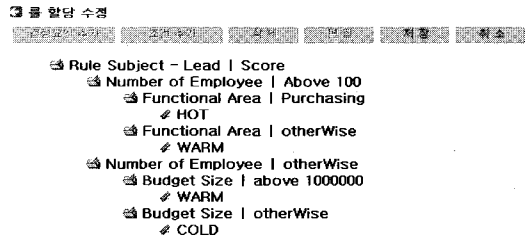
Rule Base란 경영활동에 내재된 모든 경영규칙(Business Rule)을 체계적으로 시스템화하여 전사적 업무의 일관성을 확보하고 경영규칙의 유지관리를 쉽고 신속하게 하는 시스템 및 사상이다. 빈번히 변경되는 중요 응용 업무 프로그램(Mission Critical Business Application)에 적용 함으로서 시스템 개발/유지보수를 신속히 하여 경영환경에 능동적으로 대응하는 살아있는 시스템을 만드는 기술이다.



[그림 1] Rule Base 구조

Rule Base를 적용함으로써 시스템을 쉽고 빠르게 개발하며, 시스템의 재사용성을 향상시키고 어떤 IT 구조에서도 시스템 적용과 통합을 용이하게 할 수 있다. Rule 기반 규칙 시스템의 구조는 필요시 간단한 폼을 통해 쉽게 Business Rule을 수정하고 관리 할 수 있도록 사용자의 편의성, 효율성, 비즈니스 프로세스의 변화에 대한 적응성을 바탕으로 설계한다.

Rule은 비즈니스 프로세스 전문가가 기존 프로그램의 변경이나 신규 개발이 없이 적용할 수 있어야 한다. 그래야만 비즈니스 프로세스가 변할 때마다 그 변화를 표현할 Rule을 제때에 적용할 수 있다. 규칙 시스템은 이러한 사용자의 편의성을 위해 Tree 구조와 같은 Rule 입력 기능이 필요하다. [그림 2]는 Tree 기반 Rule 입력기의 실례를 나타낸다.



[그림 2] Tree 기반 Rule 입력기

Rule 기반 규칙 시스템은 Rule이 애플리케이션에 독립적으로 사용될 수 있도록 Rule Repository를 관리한다. Rule Repository는 Rule의 개발, 실행 그리고 유지보수 관점에서 다음과 같은 장점을 가진다. 첫째, 애플리케이션의 융통성을 높이고, Rule의 개발과 배치속도를 향상시켜주며, Rule을 능률적으로 관리할 수 있도록 한다. 둘째, 애플리케이션을 BOB (Best of Breed) 방식으로 Rule을 개발할 수 있도록 한다. 마지막으로, Rule이 다른 애플리케이션에서 재사용될 수 있도록 한다.

2.2 Rule 정의

본 절에서는 다양한 경로(웹, 전자메일, 콜센터 등)를 통해 수집된 가망 수요정보를 특정 지식 기

반으로 Filtering 하고, Scoring하는 작업을 예로 설명한다. 가망 수요정보에 대한 Scoring Rule을 각 판단 기준과 경우의 수로 모두 표현하면 <표 1>과 같다.

<표 1> Filter Rule

C o n d i t i o n	종업원 수	>=100	×	×	×	×				
		<100					×	×	×	×
	부서	=‘구매’	×	×			×	×		
		기타			×	×			×	×
	예상금액	>=100만	×		×		×		×	
		<100만		×		×		×		×
A c t i o n	Hot		×	×						
	Warm				×	×	×		×	
	Cold							×	×	

<표 2> 정규화 된 Filter Rule

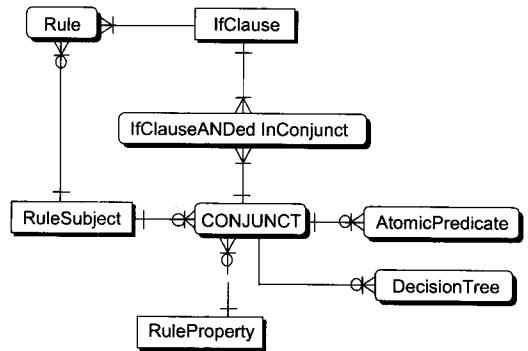
C o n d i t i o n	종업원 수	>=100	×	×					
		<100			×	×			
	부서	=‘구매’	×						
		기타		×					
	예상금액	>=100만			×				
		<100만				×			
A c t i o n	Hot	×							
	Warm		×	×					
	Cold				×				

<표 1>의 Scoring Rule을 정규화 형태로 표현하면 <표 2>와 같으며, 아래와 같이 4개의 정규화된 규칙으로 정리된다.

- 규칙 1 : 종업원수가 100명 이상이고, 가망 수요자의 부서가 ‘구매부’이면 Score는 Hot
- 규칙 2 : 종업원수가 100명 이상이고, 가망 수요자의 부서가 ‘구매부’가 아니면 Score는 Warm
- 규칙 3 : 종업원수가 100명 이하이고, 가망 수요자

- 의 예상금액이 100만원 이상이면 Score는 Warm
- 규칙 4 : 종업원수가 100명 이하이고, 가망 수요자의 예상금액이 100만원 이하이면 Score는 Cold

정규화 된 규칙들은 Rule Repository에 저장되어 애플리케이션과 독립적으로 사용 될 수 있는 환경을 제공한다. [그림 3]의 Data Model에서 If 구문 안의 모든 연속적인 정규화 형태(CNF : Conjunctive Normal Form)가 단위 술어(Atomic Predicates)의 연결로 표시 된다고 가정하면, 어떤 복잡한 If 구문 불도 일련의 CNF 룰의 집합 형태로 변환이 가능하다.



[그림 3] Data Model

<표 2>의 Scoring Rule을 Rule Repository에 저장하면, <표 3>~<표 9>와 같이 표현된다.

<표 3> Rule Subject

RuleSubject ID	Rule Name	Rule Type
RS1	Rule Subject-Lead	Score

Rule Subject는 실행하려고 하는 규칙의 일반적 정보를 표현한다.

<표 4> Rule

RuleSubject ID	IFClauseID	Action
RS1	IF1	HOT
RS1	IF2	WARM
RS1	IF3	WARM
RS1	IF4	COLD

Rule은 Rule Subject를 구성하는 If Clause와 각 경우가 참일 때의 Action을 표현한다.

<표 5> IfClauseANDedInConjunct

RuleSubject ID	IFClauseID	CID
RS1	IF1	1
RS1	IF1	2
RS1	IF2	1
RS1	IF2	103
RS1	IF3	101
RS1	IF3	3
RS1	IF4	101
RS1	IF4	105

IfClauseANDedInConjunct는 If Clause를 구성하는 조건 구문을 표현한다.

<표 6> Rule Property

RuleSubject ID	Rule Property ID	Property Name
RS1	P1	종업원수
RS1	P2	부 서
RS1	P3	예 산

Rule Property는 Rule Subject의 실행에 사용될 수 있는 속성집합을 표현한다.

<표 7> Conjunct

RSID	CID	Property	Leaf	Condition Name
RS1	1	P1	N	Above 100
RS1	101	P1	N	otherWise
RS1	2	P2	N	Purchasing
RS1	4	Score	Y	HOT
RS1	103	P2	N	otherWise
RS1	5	Score	Y	WARM
RS1	3	P3	N	Above 100만원
RS1	6	Score	Y	WARM
RS1	105	P3	N	otherWise
RS1	7	Score	Y	COLD

Conjunct는 If Clause를 구성하는 조건 구문의

속성을 표현한다.

<표 8> Conjunct

Child RSID	Child CID	Link ID	Parent CID
RS1	105	1	101
RS1	103	2	1
RS1	101	3	0
RS1	7	4	105
RS1	6	5	3
RS1	5	6	103
RS1	4	7	2
RS1	3	8	101
RS1	2	9	1
RS1	1	10	0

Decision Tree는 Tree 구조의 Visual한 표현을 위한 속성을 표현한다.

<표 9> Atomic Predicate

RSID	CID	AID	Op	Scalar 1	Scalar 2	Unit
RS1	1	1	>=	100		명
RS1	101	1	<	100		명
RS1	2	1	=	구매부		
RS1	103	1	!=	구매부		
RS1	3	1	>=	100		만원
RS1	105	1	<	100		만원

Atomic Predicate는 If Clause를 구성하는 조건 구문의 최소 단위 표현을 나타낸다.

2.3 Rule Processing

앞 절에서 저장된 Scoring Rule을 프로세싱하는 과정은 다음과 같다. 예를 들어, 종업원수가 70명 이고, 부서가 '구매부'이며, 예산이 2000만원인 가 망수요정보에 대한 Scoring을 해보자.

Input	Output
Rule Subject = RS1 종업원수 = 70명 부서 = 구매부 예산 = 2000만원	Score = WARM

Rule Database를 기반으로 한 추론 Algorithm은 아래와 같다.

- 1) AtomicPredicate Table로부터 입력된 경우를 만족시키는 Conjunct ID를 추출
→ 101, 2, 3이 선택됨
- 2) Conjunct ID 101, 2, 3이 True일 경우 True인 If Clause Id를
- 3) ConjunctANDEDInIfclause 테이블에서 선택
→ IF3이 선택됨
- 4) Rule 테이블에서 IF3이 true일때 Action 값을 추출함
→ WARM이 선택됨

● **Implementation :**

Process Step 1

Set1(AID) = AIDs of atomic predicates evaluated to true by an inference algorithm

Process Step 2

Query2(IfClauseID) = ConjunctANDEDInIfclause divided-by Set1

Process Step 3

Query3(Action) = Select Action from rule where IfClauseID = Query2

이와 같은 방법은 많은 Join으로 인한 Performance의 문제를 발생시킬 수 있으므로, 실제 구현에서는 <표 10>과 같은 Data Mart를 사용한다.

<표 10> Rule Data Mart

C1	C2	C3	C4	C5	C6	C7	C8
RS1	IF1	HOT	1	P1	>=	100	명
RS1	IF1	HOT	2	P2	=	구매	
RS1	IF2	WARM	1	P1	>=	100	명
RS1	IF2	WARM	103	P2	!=	구매	
RS1	IF3	WARM	101	P1	<	100	명
RS1	IF3	WARM	3	P3	>=	100	만원
RS1	IF4	COLD	101	P1	<	100	명
RS1	IF4	COLD	105	P3	<	100	만원

- C1 : RuleSubjectID
- C2 : IfClauseID
- C3 : Action
- C4 : ConjunctID
- C5 : RulePropertyID
- C6 : Operator
- C7 : ScalarI
- C8 : Unit

Rule Data Mart를 기반으로 한 추론 Algorithm은 아래와 같으며, 비정규화된 Data Mart와 Clustered Index를 사용함으로써 Rule을 수행하는 속도를 향상시킬 수 있다.

● **Input**

- InputRuleSubjectID
- InputCriterionID[1...L]
- InputCriterionScalar[1...L]

● **Output**

- Action

● **Algorithm**

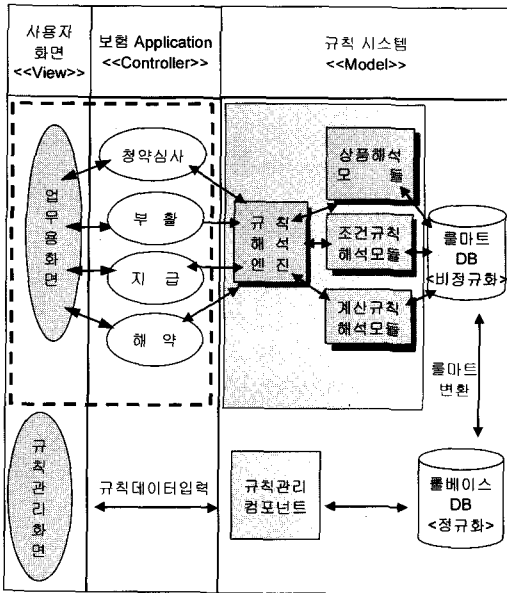
```

Select * from Datamart where RuleSubjectID = Input
RuleSubjectID into M arrays with size N+1(with
each array corresponding to each column)
K = 1
Repeat
If InputCriterionScalar[J] for
InputCriterionID[J] = RulePropertyID [K]
violates the condition of ConjunctID[K] then
ConjunctValue = False
If(K = N) then
ExitState = "No matching rules can be found"
K = K+1
Else
Repeat K = K+1
until(IfClauseID[K] <> IfClauseID[K-1])
Else
K = K+1
If((K > N) or (IfClauseID[K] <> IfClauseID[K-1]))
then
ExitState = "No matching rules can be found"
Action = Action[K-1]
Stop
Until(K > N)
    
```

3. Rule 기반 상품규칙 시스템의 설계 및 구조

3.1 보험 시스템과 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. [그림 4]는 보험 시스템과 상품규칙시스템 연관도를 나타낸다. 규칙데이터는 [그림 4]의 아래부분에 위치한 규칙관리화면을 통하여 입력할 수 있고 입력된 상품규칙 데이터는 성능을 고려한 룰마트 DB로 자동 변환된다.



[그림 4] 보험 시스템과 상품규칙 시스템 연관도

3.2 조건규칙 데이터 모델 및 구조

[그림 5]는 조건규칙 데이터 모델 구조를 나타내며 모든 조건규칙 자료는 Conjunctive Normal Form 형태로 저장된다. 상품 속성이 상품코드와 보험코

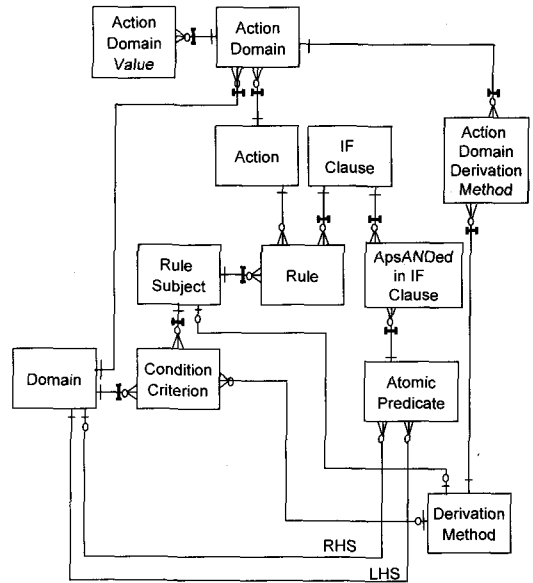
드의 조합으로 결정되지 못하고 조건에 따라 속성이 결정될 때 사용한다.

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

If a = 2 and b = 1 then c = 1과 같은 조건식을 예로 들어보면, 앞의 조건식은 조건절(If절)과 Then절(Action)로 나눌 수 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.

Then절을 표현하는 Action은 Then절에서 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.



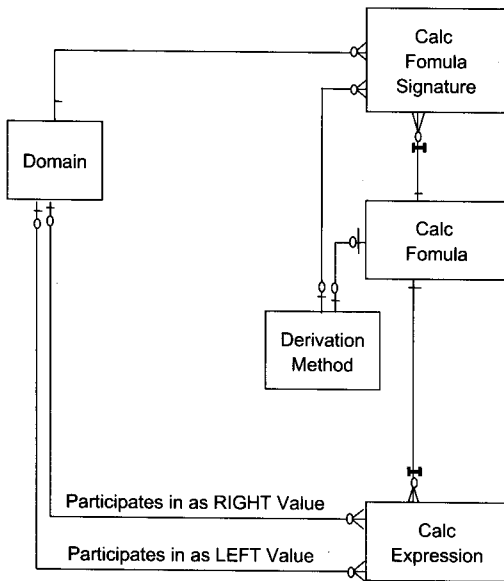
[그림 5] 조건규칙 데이터 모델

3.3 계산규칙 데이터 모델 및 구조

[그림 6]은 계산규칙 데이터 모델을 표현한다. 계산 규칙은 계산식을 이항 연산식으로 전개한 형태로 저장된다. 단위 계산규칙을 나타내는 Calc Formula는 해당 계산규칙에서 사용하는 In/Out 항목

(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.

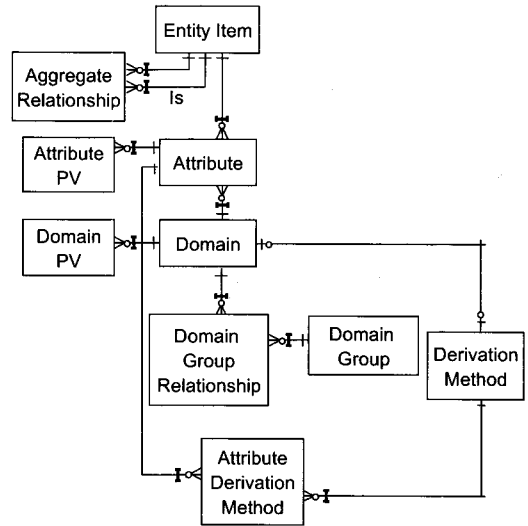
- 1 번식 : 상각 기준년수 - 보험료 납입 경과기간
- 2 번식 : 예정 신계약비 × 1번 연산식 결과
- 3 번식 : 2번 연산식 / 상각기준년수



[그림 6] 계산규칙 데이터 모델

3.4 상품규칙 데이터 모델 및 구조

[그림 7]은 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 있고 이에 대한 실제 값(Attribute PV)을 상수값, 구간값, 허용값, 복수값, 조건식, 계산식 등의 다양한 방법으로 정의할 수 있도록 수용한 구조이다. 상품의 항목을 그룹핑하거나 순서를 정의할 수 있고 기존 테이블 단위에 매핑할 수 있다.



[그림 7] 상품규칙 데이터 모델

4. 규칙 시스템

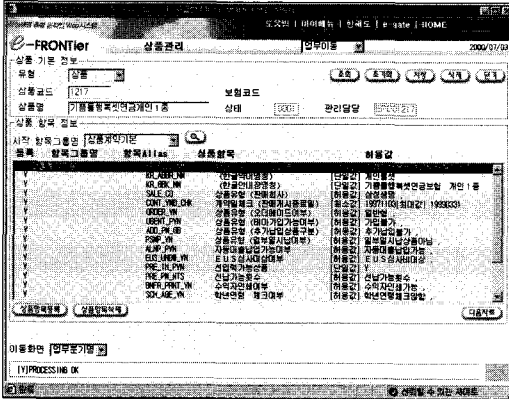
규칙 시스템은 보험상품 속성의 비즈니스 로직을 정의하는 규칙관리 컴포넌트와 규칙 시스템에 저장된 데이터를 해석하여 서비스를 제공하는 규칙해석 컴포넌트로 구성된다. 본 절에서는 각 컴포넌트의 중요기능을 살펴본다.

4.1 규칙관리 시스템

규칙관리 시스템은 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.

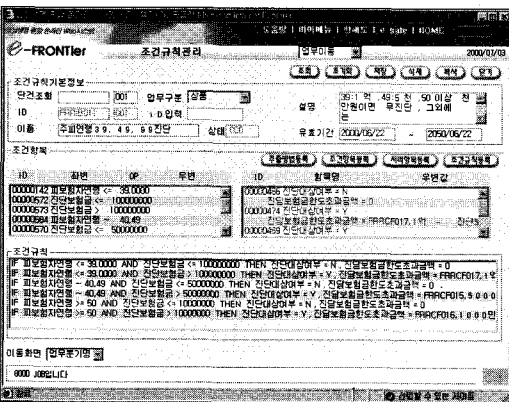
[그림 8]은 상품규칙관리 시스템의 샘플 화면으로서, 상품규칙관리 기능은 기존 상품 테이블과 기능이 유사하나, 테이블에 저장할 상품속성을 데이터화하여 항목 추가 및 변경, 조합 등을 비교적 자유롭게 수행할 수 있다. 상품이나 보험코드의 조합

으로 항목 값이 상수로 정해지지 않을 경우, 조건 규칙이나 계산규칙으로 정의한다.



[그림 8] 상품규칙 관리 화면

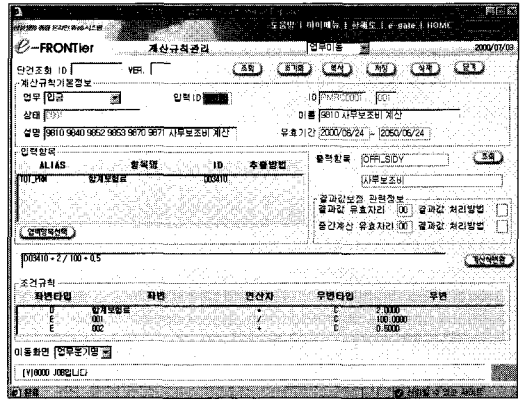
[그림 9]는 조건규칙관리 시스템의 샘플화면으로서, 조건규칙은 프로그램에 표현된 if then else 로직을 데이터로 저장하며, 해석 요청시 규칙데이터와 입력으로 들어온 데이터를 사용하여 원하는 결과를 제공한다. 조건규칙은 조건항목(if절)과 처리항목(then절)의 조합으로 구성된 정규화 형태로 테이블에 저장된다.



[그림 9] 조건규칙 관리 화면

[그림 10]은 계산규칙관리 시스템의 샘플화면을 나타내며, 시스템에서 상품속성을 정의하는 계산식을 데이터로 저장하며, 애플리케이션에서 호출시 계산식 결과를 제공한다. 일반적으로 계산규칙 단독

으로 사용하는 경우보다 조건규칙에서 계산식을 표현하는 보조수단으로 주로 사용한다.



[그림 10] 계산규칙 관리 화면

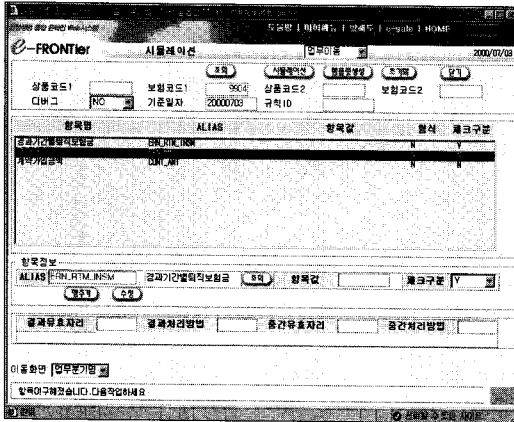
4.2 규칙해석 시스템

규칙 시스템에 저장된 자료는 원칙적으로 해석 모듈을 통하여 서비스를 제공한다. 사용자는(Application 개발자) 해석모듈에서 제공하는 단일 인터페이스를 통하여 모듈을 호출함으로써 사용할 수 있다. 일반 업무 프로그램 작성자는 상품 속성 관련 기능은 규칙 시스템에 맡기고 고유 업무로직 구현에 충실 할 수 있다. 성능향상을 위하여 해석모듈들은 비정규화 된 형태의 몰마트 DB를 사용하며 규칙관리 시스템에서 자료 입력시 규칙 메타모델 형태의 정규화 DB와 몰마트 DB에 동시에 Update 한다. 온라인용과 배치용의 해석모듈을 제공하며, 애플리케이션에서 해석모듈을 호출할 경우 원하는 대로 결과 값을 가져오는지를 확인할 수 있는 규칙 테스트 기능을 제공한다.

4.3 적용 사례

규칙 시스템의 운영 및 개발 환경은 IBM Main-Frame 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 A사 및 B사로 표현한다.



[그림 11] 규칙해석 시뮬레이션(테스트) 화면

- A사 : 계정계 신시스템 프로젝트(1999년~2000년, 2년) 수행시 규칙 시스템을 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 규칙 시스템의 업계 상용화가 가능한 것으로 입증되었다.
- B사 : 계정계 신시스템 프로젝트(2001년~2002년, 1년) 수행시 기존 구축되어 운영하고 있는 A사의 보험 시스템을 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 어느 정도 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 규칙 시스템의 도입으로 초기에 목표한 상품변화에 대한 신속한 대응, 상품 유지보수 비용 감소, 현업주도의 상품관리가 실현 가능하게 되었다.

5. 결 론

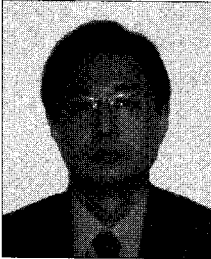
본 논문에서는 보험상품 속성의 효율적 표현 및 관리를 위해 IF/THEN 구조의 Rule Base를 제안하였다. 본 연구에서 제시한 Well Formed Rule Base를 적용함으로써 얻을 수 있는 이점으로는, 보험 신상품 추가 및 변경 시 애플리케이션의 수정이 아닌 규칙 데이터의 추가 및 변경 방식으로 전환이 가능하여 시스템의 유지보수성, 신뢰성 및 신속성 등을 획기적으로 향상시킬 수 있었다. 궁극적으로 신상품 반영 작업을 전산실이 아닌 현업에서 시스템을 통하여 처리할 수 있는 체제로 전환이 가능하고 궁극적으로 급변하는 경영환경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

특히, 소프트웨어 개발업체 입장에서는 비즈니스 로직의 데이터화는 유사업종의 프로젝트 진행시 이전 고객사의 업무 노하우 유출을 방지 할 수 있어 개발된 비즈니스 컴포넌트를 재사용이 가능하고, 전체 시스템의 계층 및 기능별 컴포넌트화를 통해 프로젝트에 특화된 기능을 선별하고 커스터마이징 하는 작업을 수월하게 수행할 수 있다.

참 고 문 헌

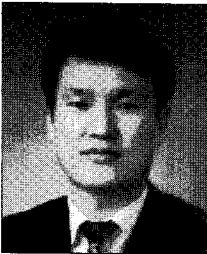
- [1] Aberdeen Group, *The Power of Rules-Driven Processing*, Internal Report, Aberdeen Group, Boston, Sept. 2000.
- [2] 황혜수 외 3인, “룰 기반 CRM 시스템에서 시간요소를 고려한 룰 모델 제안”, 『정보과학회 2001년 추계학술대회』, Vol.28, No.2(2001), pp. 283-285.
- [3] 황혜정 외 3인, “질의 완화를 이용한 지능적인 질의 응답 시스템”, 『한국정보처리학회논문지』 A, Vol.7, No.1(2000), pp.88-98.
- [4] 김철수, “웹 고객의 개인화를 지원하는 지식기반 통합 시스템”, 『한국정보처리학회논문지』 B, Vol. 9-B, No.1(2000), pp.1-6.
- [5] 최중민, 『인공지능』, 사이텍미디어, pp.229-323, 1998.
- [6] Khawar Zaman Ahmed, *Developing Enterprise Java Applications with J2EE and UML*, Addison-Wesley.
- [7] Stephen Stelting, *Applied Java Patterns*, Sun Microsystems.

◆ 저 자 소 개 ◆



김도형 (dohyung1.kim@samsung.com)

인하대학교 공과대학에서 학사, 콜로라도 주립대학에서 석사, 아리조나 주립대학에서 산업공학전공으로 박사학위를 취득하였으며, 현재 삼성SDS(주)에서 재직하고 있다. 주요 관심분야로는 e-Business, ERP, CRM, Data-warehouse, 소프트웨어공학 등이다.



오영배 (yboh@suwon-c.ac.kr)

고려대학교 공학사, 인하대학교에서 공학석사, 고려대학교 컴퓨터학과에서 박사수료 하였다. 그리고 정보처리기술사 자격증을 보유하고 있다. 캘리포니아대학(UCI)에서 객원연구원으로 근무한 바 있고, 한국전자통신연구원(ETRI)에서 책임연구원(실장)으로 근무하였고, 현재 수원여대 컴퓨터응용학부 교수로 재직 중이다. 현 주요 관심분야는 소프트웨어 아키텍처, 컴포넌트공학(CBSE) 및 전자상거래이다.