

Rule Database를 활용한 가망수요 정보 규칙 관리

김도형^o 강명철

삼성 SDS

{dohyung1.kim^o, mc.kang }@samsung.com

Rules-Driven Processing for Sales Leads

Dohyung Kim^o Myungchul Kang
Solution Development Center, Samsung SDS

요 약

오늘날 기업의 판매형태는 Channel Partner를 통한 간접판매가 50%이상을 점유하고 점차 증가 추세에 있다. 이러한 판매형태를 지원하고 기업의 이익 극대화를 위해 Internet 기술을 이용한 가망 수요정보의 효율적 관리가 중요한 이슈로서 대두되고 있다. 본 논문은 Web을 통해 포착된 Sales Leads를 Channel Partner들을 통해 실제 판매로 연결시켜주는 기능을 담당하는 Lead Management 시스템 (LM)의 기능을 Rule기반으로 구축하여, Dynamic Market에 신속하게 대응함으로써 Vendor 와 Partner간의 상호 Win-Win 결과를 얻을 수 있는 방안을 제공하고자 한다.

1. 서 론

최근 기업의 판매 영역에서 파트너관계관리(PRM : Partner Relationship Management)의 개념이 새롭게 부각되고 있다. PRM은 ' 간접 채널 중심의 회사가 온라인 상에서 파트너들과의 상호 이익관계를 지속적으로 개발하고 유지하기 위한 전략 및 활동' 을 의미한다.

궁극적으로 고객 대응이라는 측면에서 보면 PRM은 CRM과 맥을 같이 하고는 있지만 CRM이 최종 고객을 대상으로 한 직접 판매를 수행하는 기업을 주 대상으로 하는 반면, PRM은 유통 채널을 통한 간접 판매를 수행하는 기업을 대상으로 한다는 점에서 차이가 있다. 또한 CRM이 영업, 마케팅, 콜센터등 제조기업내의 고객관련 업무를 주요 관리 대상으로 하는 반면에 PRM은 기업과 외부 파트너 간의 고객관련 업무를 관리하는 특성을 가지고 있다.

PRM이 제공하는 다양한 기능 중에서 특히 가망수요관리 기능은 다양한 경로(웹, e-mail, call-center, direct mail, trade show ...)를 통해 수집된 가망 수요정보의 수율(Sales Conversion Rate)을 향상시키기 위하여 이러한 정보들을 특정 지식 기반으로 Filtering하고, Scoring하는 작업과 가장 적합한 파트너에게 분배해 주는 기능을 필요로 한다. 본 논문은 PRM의 기능중 Web을 통해 포착된 가망 수요정보를 Channel Partner들을 통해 실제 판매로 연결시켜주는 기능을 담당하는 Lead Management 시스템 (LM)이 Dynamic Market에 신속하게 대응하여 Vendor 와 Partner간의 상호 Win-Win 결과를 얻을 수 있도록 해주는 Rule 기반의 Sales Leads Filter / Score 시스템에 대해 연구한 결과를 설명한다.

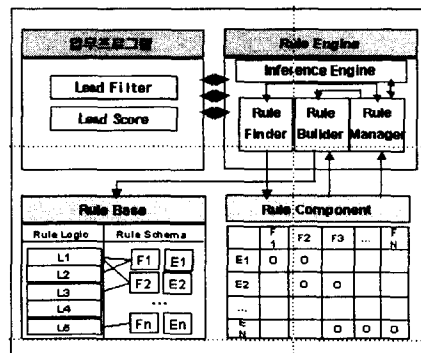
2. Rule 기반 Lead Management 시스템의 설계 및 구조

본 절에서는 Lead Management 시스템의 Sales Leads Filter / Score / Assign Rule 시스템의 개발을 위한 설계의 원칙들과 이를 바탕으로 한 설계의 장점과 컴포넌트 단위의 시스템 구조를 설명한다.

2.1 Rule 기반 Lead Management의 설계 개념

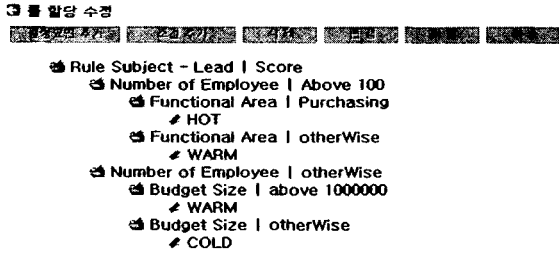
Rule Base란 경영활동에 내재된 모든 경영규칙 (Business Rule)을 체계적으로 시스템화 하여 전사적 업무의 일관성을 확보하고 경영규칙의 유지관리를 쉽고 신속하게 하는 시스템 및 사상이다. Rule Base를 적용함으로써 시스템을 쉽고 빠르게 개발하며, 시스템의 재사용성을 향상시키고, 어떤 IT 구조에서도 시스템 적용과 통합을 용이하게 할 수 있다

<그림 1> Rule Base 구조



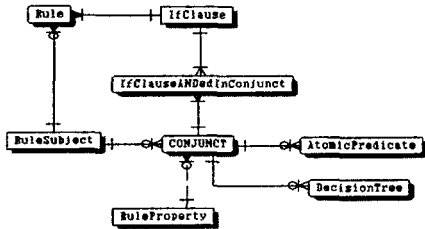
Lead Management 시스템의 구조는 필요시 간단한 품을 통해 쉽게 Business Rule을 수정하고 관리 할 수 있도록 사용자의 편의성, 효율성, 비즈니스 프로세스의 변화에 대한 적응성을 바탕으로 설계 되었다. 이러한 사용자의 편의성을 위해 Tree구조의 Rule 입력기를 제공한다.

<그림 2> Tree 기반 Rule 입력기



Lead Management 시스템은 Rule이 애플리케이션에 독립적으로 사용될 수 있도록 Rule Repository를 관리한다. If 구문안의 모든 연속적인 정규화 형태(CNF : Conjunctive Normal Form)가 단위 술어(Atomic Predicates)의 연결로 표시 된다고 가정하면, 어떤 복잡한 If 구문 풀도 일련의 CNF 룰의 집합 형태로 변환이 가능하며 정규화 형태로 변환된 Scoring Rule을 Rule Repository에 저장한다. 정규화된 Rule의 Rule Repository에 저장을 위한 data model은 <그림 3>과 같다.

<그림 3> Rule Repository Data Model



2.2 Rule 정의

본 절에서는 Lead Management 시스템에서 가장 주요 정보의 Scoring을 예로 설명한다. 가장 주요정보에 대한 Scoring Rule을 각 판단 기준과 경우의 수를 모두 표현하면 <표 1>과 같고, 이를 정규화 형태로 변환하면 <표 2>와 같이 표시된다.

<표 1> Scoring Rule

Condition	종업원수	>=100	X	X	X	X				
		<100					X	X	X	X
	부서	= '구매'	X	X			X	X		
		기타			X	X			X	X
예산금액	>=100만	X		X	X			X		
	<100만		X		X			X	X	
Act.	Hot		X	X						
	Warm				X	X	X		X	
	Cold							X	X	

<표 2> Condensed Scoring Rule

Condition	종업원수	>=100	X	X					
		<100				X	X		
	부서	= '구매'	X						
		기타		X					
예산금액	>=100만				X				
	<100만						X		
Act.	Hot		X						
	Warm			X	X				
	Cold						X		

이 정보를 <그림 3>과 같은 형태의 Repository에 저장할 때 각 요소별 저장되는 내용은 아래와 같다.

<표 3> Rule Subject

RS ID	RS Name	Type
RS1	Rule Subject- Lead	Score

<표 4> Rule

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

(RS: Rule Subject, CID: Conjunct ID, AID: Atomic Predicate ID)

실행하고자 하는 규칙의 일반적 정보를 Rule Subject <표 3>로 표현하고, Rule은 Rule Subject를 구성하는 If Clause와 각 경우가 참일 때의 Action을 표현한다.

<표 5> IfClause&Conjunct

RS 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

<표 6> Rule Property

RS ID	Property ID	Name
RS1	P1	종업원수
RS1	P2	부서
RS1	P3	예산

<표 5>는 If Clause를 구성하는 조건 구문을 표현하고 Rule Property<표 6>는 Rule Subject의 실행에 사용될 수 있는 속성집합을 표현한다

<표 7> Conjunct

RSID	CID	Property	Condition Name
RS1	1	P1	Above 100
RS1	101	P1	otherWise
RS1	2	P2	Purchasing
RS1	103	P2	otherWise
RS1	3	P3	Above 100만원
RS1	105	P3	otherWise

Conjunct<표 7>는 If Clause를 구성하는 조건 구문의 속성을 표현하고 Atomic Predicate<표 8>는 If Clause를 구성하는 조건 구문의 최소 단위 표현을 나타낸다.

<표 8> Atomic Predicate

RSID	CID	AID	Op	Scalar1	Scalar2	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

2.3 Rule Processing

Scoring Rule을 프로세싱하는 과정은 다음과 같다. 다음과 같은 값을 갖는 가망 수요정보에 대한 Scoring을 해보자.

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

Rule Database를 기반으로한 추론 Algorithm은 다음과 같다.

1. AtomicPredicate Table로부터 입력된 경우를 만족시키는 Conjunct ID를 추출 → 101, 2, 30이 선택됨
2. Conjunct ID 101, 2, 30이 True일 경우 True인 If Clause Id를 ConjunctANDedInIfclause 테이블에서 선택 → IF30이 선택됨
3. Rule테이블에서 IF30이 true일때 Action 값을 추출함 → WARM이 선택됨

하지만 이와 같은 방법은 많은 Join으로 인한 Performance의 문제를 발생시킬 수 있다. 실제 구현에서는 다음과 같은 Data Mart를 사용한다.

<표 10> Rule Data Mart

C1	C2	C3	C4	C5	C6	C7	C8	C9
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: Scalar1, C8: Scalar2, C9: Unit)

Input	Output
InputRuleSubjectID InputCriterionID[1...L] InputCriterionScalar[1...L]	Action

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

```

Select * from Datamart where RuleSubjectID = InputRuleSubjectID
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. 결론

본 논문에서는 가망 수요정보의 효율적 관리를 위해 IF/THEN 구조의 Rule Base를 제안하였다. 본 연구에서 제시한 Well Formed Rule Base를 적용함으로써 얻을 수 있는 이점으로는, 각 업무 프로그램은 업무 규칙/ 논리를 기술하지 않고 순전하게 사용자 위주의 프로세스 위주로 프로그래밍하게 된다. 필요한 업무 규칙/ 논리는 Rule Engine 시스템을 활용하여 결과를 얻게 되므로 사용자 인터페이스 또는 시스템 인터페이스를 제외한 부분은 프로그래밍 되지 않는다. 따라서 업무 프로그래밍의 유지보수가 거의 발생하지 않게 되며, 급변하는 경영환경 변화에 신속하게 대응이 가능하여, 궁극적으로 웹 기반하에서 수집된 가망 수요정보의 Sales Conversion Rate를 극대화하여 기업의 경쟁력을 제고 할 수 있다.

참고 문헌

- [1] Aberdeen Group, " The Power of Rules-Driven Processing.", Internal Report, Aberdeen Group, Boston, Sept.2000.
- [2] 황해수와 3인, " 룰 기반 CRM 시스템에서 시간요소를 고려한 룰 모델 제안", 정보과학회 2001년 추계학술대회, VOL.28, NO.2, pp.283~285, 2001.
- [3] 황해정과 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 Stelling, " Applied Java Patterns", Sun Microsystems.