

# 에너지 관리 시스템을 위한 능동 규칙 설계

김광중, 조상규, 이연식  
군산대학교 컴퓨터정보과학과  
e-mail: ysllee@ks.kunsan.ac.kr

## Design of Active Rules for an Energy Management System

Kwangjong Kim, Sangkyu Joe, Yonsik Lee  
Dept. of Computer & Information Science, Kunsan National University

### 요약

능동 데이터베이스 시스템은 일반적인 데이터베이스 시스템과 달리 어떤 사건이 발생하면 조건의 성립 여부에 따라 적절한 반응을 자동적으로 실행하는 시스템이며, 능동성은 사건, 조건 그리고 조치로 이루어진 능동규칙에 의해 수행된다. 그러므로 능동 데이터베이스는 규칙의 개념이 중심이 된다. 규칙은 사용자와 응용프로그램, 또는 데이터베이스 관리자에 의해 정의되며, 대부분 일반적인 형태인 ECA로 이루어져 있다. 따라서 본 논문에서는 에너지 관리 시스템에 적용하기 위한 능동 규칙 시스템의 구조에 따른 능동규칙을 일반적인 규칙언어 형태를 바탕으로 설계하고, 다양한 응용시스템에서의 확장성을 제시한다.

### 1. 서론

능동 데이터베이스 시스템은 일반적인 데이터베이스 시스템과 달리 어떤 사건이 발생하면 조건의 성립 여부에 따라 적절한 반응을 자동적으로 실행하는 시스템이다. 이 시스템은 ECA규칙을 사용하여 요구처리, 공정제어, 주식시장과 재정상태, 항공 교통관리 같은 실시간 응용 프로그램이나 무결성 제약 규칙, 버전 관리, 보안 등에 사용된다. 이러한 능동적 형태의 데이터베이스 시스템은 여러 가지 유용한 기능들을 효과적으로 지원할 수 있다[Wid96]. 발전소 감시 시스템과 같은 경우 냉각수 파이프의 압력이 일정상태 이상이면 이에 대응한 적절한 조치를 취해야 할 것이다. 이 경우 기존의 능동 기능이 없는 수동 데이터베이스는 냉각 파이프의 압력을 감시하는 프로세스를 데이터베이스와 별도로 갖고 있어야 하며, 적당한 조치를 수행하는 프로세스도 별도로 갖고 있어야 한다. 그리고 이러한 모든 부가적 프로세스는 데이터베이스와는 무관하게 응용프로그램에서 프로그래밍 되어야 하며, 이 응용프로그램이 아닌 다른 응용프로그램에는 적용할 수 없는 문제점을 갖는다. 반면 능동 데이터베이스일 경우 사건으로 제한된 파이프 입력값을 설정하고 이에 대응한 적절한 SQL조치문 또는 외부 프로시저 만들 작성하면 문제를 해결할 수 있다. 이처럼 능동기능을 발전소의 일부분인 감시 시스템뿐만 아니라 발전소에서 생산된 전기를 안정되고 경제적으로 공급하기 위한 전력계통에 적용함으로써 사용자는 다양한 사건, 조건, 조치를 사용자 스스로 정의함으로써 데이터베이스의 응용성과 가용성을 배가시킬 수 있다.

이에 본 논문에서는 능동 개념을 효율적으로 지원하는

규칙 개념 및 규칙 처리를 위한 능동 규칙 시스템에 따른 능동기능을 고려하여 에너지 관리 시스템을 위한 능동 규칙을 설계 제시한다.

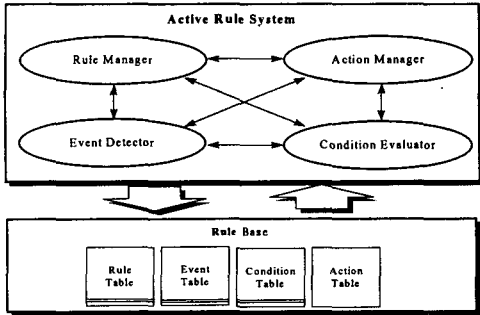
논문의 구성은 2장에서 능동 규칙 시스템의 구성 요소와 구조, 규칙언어의 일반적인 형태를 제시하고, 3장에서는 이를 바탕으로 에너지 관리 시스템을 위한 능동 규칙 시스템의 구조에 따른 능동 규칙을 설계하고, 4장에서는 이 논문의 결론 및 향후의 연구방향에 대해서 기술한다.

### 2. 능동 규칙 시스템과 규칙언어

규칙 시스템은 규칙을 정의하고 관리하며 규칙 요소들을 처리하기 위한 처리기 시스템으로서 규칙언어, 사건 검출기, 조건 평가기, 조치 실행기, 규칙 관리자로 구성된다. 또한 이러한 구성 요소들과 규칙베이스 및 데이터베이스가 결합하여 능동 데이터베이스 시스템으로 확장된다. 규칙 시스템이 데이터베이스와 결합하는 형태에 따라 세 가지 유형의 구조를 갖는데 각각의 구조를 결합의 강도에서 보면, 밀 결합 구조인 통합형 구조, 소결합 구조인 계층형 구조, 변형 결합구조인 컴파일 구조로 구분할 수 있다[Wid91, 96].

규칙시스템의 또 다른 중요한 요소로 규칙베이스를 들 수 있는데, 규칙베이스는 규칙을 저장하고 관리하는 부분으로서 규칙은 데이터베이스에서 메타데이터로 취급되며, 일반 데이터와는 달리 규칙을 위한 전용데이터베이스인 규칙베이스에 저장되고 관리된다. 규칙은 컴파일 되어 규칙베이스에 저장되며, 필요한 경우 각 처리기에 적재되었다가 처리가 완료되면 다시 데이터베이스에 적재된다. 또한 규칙데이터베이스에 저장된 규칙은 규칙명령을 이용하여 검색, 수정

및 삭제 될 수 있다. [그림 1]은 규칙시스템과 규칙베이스의 관계를 보여준다.



[그림 1] 규칙 시스템과 규칙베이스

또한 능동 규칙언어는 능동 데이터베이스 시스템의 사용자 접속도구로서 규칙을 정의하고 관리하는 기능을 갖고며, 사용자에게 규칙을 정의, 수정, 삭제할 수 있는 기능을 부여한다. 일반적으로 규칙언어는 해당 데이터베이스의 질의어를 확장하여 구성하며 규칙 선언부, 사건절, 조건절, 조치절로 구성된다[Day88]. 이러한 능동 규칙 언어는 규칙 시스템에 따라 다양한 형태로 제시되고 있으나 대부분 능동 데이터베이스 시스템에 장착되어 있는 질의어와 유사한 문법을 따르도록 구성하는 것이 일반적이다. 따라서 능동 규칙 언어는 규칙의 생성, 변경, 삭제 및 활성화/비활성화 등을 정의하고 관리할 수 있는 기능을 제공한다.

규칙 선언부는 규칙을 명시하는 예약어와 규칙 이름으로 구성하며, 질의어의 CREATE 또는 DEFINE 문과 같은 데이터 정의어의 확장을 통하여 구성하며, 사건절은 사건을 발생시키는 연산과 이 연산의 대상이 되는 릴레이션 또는 릴레이션 내 속성을 함께 명시한다. 조건절은 이 규칙의 조치가 수행되기 위해 만족해야 할 데이터베이스의 상태를 평가하는 것으로 데이터 검색문을 포함하는 술어로 구성되고, 조치는 원칙적으로 아무 제약도 가하지 않은 일련의 데이터베이스 질의어로 구성된다.

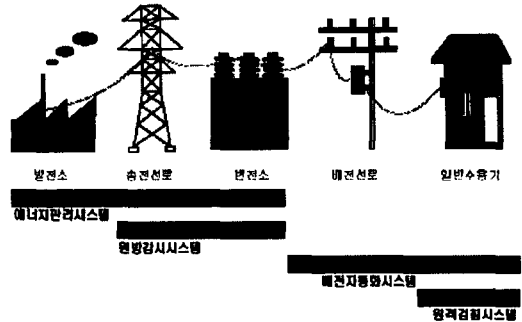
예를들어, 규칙을 생성하기 위한 구문은 다음과 같다.

```

Create rule name
on table
when triggering-operations
[ if condition ]
then action-list
[ precedes rule-list ]
[ follows rule-list ]
    
```

### 3. 에너지 관리 능동 규칙

현재 전력계통 시스템은 네가지로 구분할 수 있다. 에너지관리시스템, 원방감시시스템, 배전자동화시스템, 원격검침시스템 등이다. 위와 같은 각각의 시스템의 영역을 살펴보면 아래 [그림 2]와 같다.



[그림 2] 전력 계통 시스템의 각 영역

전압 망은 연결 사이트의 쌍인 브랜치와 함께 사이트로 이루어져 있다. 사이트는 전압이 발생한 어떤 곳일 수도 있다. 발전소는 배전소가 될 수 있으며 반면 배전 전압을 사용하는 곳을 수신지라 한다. 그러한 전압을 수신 전압이라 하고 중간 노드들은 배전과 수신역할을 한다. 브랜치는 끝에 위치한 두 개의 스위치 기어를 가진다. 각 브랜치를 열고 닫는 것은 두 스위치 기어의 상태에 의해 결정된다. 브랜치는 두 스위치 기어가 꺼진다면 켜지게 된다. 망의 동적 동작은 모두 켜진 브랜치와 현재 전류의 흐름으로 연관되어진 방향과 스위치 기어들의 모든 상태에 의해 묘사된다. 망의 동작 조건들은 매일 감시되며 가끔 재구성된다. 토폴로지는 대략 달에 한번 정도 수정된다.

토폴로지 규칙은 토폴로지가 불완전하게 명세된 네트워크의 설계를 평가하거나 완성시키는 설계자를 도와주며, 전압 배전 규칙은 사용자 요구 변화에 따라 전압을 재분배한다. 이러한 규칙들은 네트워크 토폴로지에 변화를 수행을 하지 않는다.

배전 규칙은 배선 설계시 검사된다. 각 브랜치는 운반책이며 각 선은 전류와 전압과 연관되어진다. 선의 전류와 전압은 선의 종류에 의해 최대 전류와 전압을 초과할 수 없다. 선은 적당한 튜브로 둘러 쌓여 있고 브랜치로 결합되어 있다.

#### 3.1 토폴로지 규칙

다음 규칙 R11, R12, R13은 브랜치들이 삽입되거나 삭제 될 때 자동적으로 스위치들의 삽입 삭제로 전,유도되며 그들은 또한 사이트들이 삭제 될 때 자동적으로 스위치들이 삭제된다.

```

CREATE RULE R11 For branch
ON Insert
WHEN Inserted(branch(B))
THEN Insert switch(site :B.site1,
                  branch:B, status:open)
      Insert switch(site:B.site2,
                  branch:B, status:open)
    
```

```

CREATE RULE R12 For branch
ON Delete
WHEN Deleted(branch(B))
THEN Delete switch S
      Where S.branch = B
    
```

```

CREATE RULE R13 For site
ON Delete
WHEN Deleted(site(S))
THEN Delete branch B
      Where B.site1 = S ∨ B.site2 = S
    
```

다음 규칙 R14, R15, R16, R17은 브랜치에서의 현재 전류 흐름을 제어하고 감시하는 것을 표현한다. 스위치가 꺼질 때 관련된 브랜치의 한쪽 끝만은 0이 아닌 입력 파워를 가져야 한다. 만일, 이 조건이 위반 될 때 에러가 기록된다. 그렇지 않으면 브랜치의 방향을 현재 전류 흐름에 따라 이를 반영하도록 한다.

```

CREATE RULE R14 For switch
ON Update to status
WHEN Updated(switch(S)), NEW S.status = open
THEN Update branch B Set B.passingPower = 0
      Where S.branch = B
    
```

```

CREATE RULE R15 For switch
ON Update to status
WHEN Updated(switch(S)), NEW S.status = closed
      ^ (the other switch of the branch has
          status = closed)
      ^ (exactly one site S1 of the two end sites S1,
          S2 has powering > 0)
THEN Update branch B
      Set B.orientation.from = S1,
          B.orientation.to = S2
      Where S.branch = B
    
```

```

CREATE RULE R16 For switch
ON Update to status
WHEN Updated(switch(S)), NEW S.status = closed
    
```

```

      ^ (the other switch of the branch has
          status = closed)
      ^ (powering = 0 for both end sites S1, S2)
THEN report-error(branch S1-S2 has no current flow)
    
```

```

CREATE RULE R17 For switch
ON Update to status
WHEN Updated(switch(S)), NEW S.status = closed
      ^ (the other switch of the branch has
          status = closed)
      ^ (powering > 0 for both end sites S1, S2)
THEN report-error(branch S1-S2 has double incoming
          flow)
    
```

### 3.2 전압 분산 규칙

기본적으로 다음 세가지 규칙 R21, R22, R23은 사용자로부터의 파워 요청에 대한 변화에 반응하며, 또한 사용자를 발전소에 연결하여 변화를 위임해서 파워를 재분배한다. 규칙 R21은 입력 브랜치에 변화를 위임하여 사용자 요구에 따른 변화를 반응한다.

```

CREATE RULE R21 For user
ON Insert, Update to absorbedPower
WHEN Updated(user(U)), NEW U.absorbedPower
      != U.powering
THEN Update U
      Set U.powerIn = U.absorbedPower
          ;Update branch B
      Set B.passingPower = U.powerIn
      Where U.branchIn = B
    
```

규칙 R22는 중간 노드에서의 요구의 변화에 반응한다.

```

CREATE RULE R22 For node
ON Insert, Update to powerOut, absorbedPower
WHEN Updated(node(N)),
NEW N.absorbedPower + NEW N.powerOut
      != powerIn
THEN Update N
      Set N.powerIn = N.absorbedPower
          + N.powerOut;Update branch B
      Set B.passingPower = N.powerIn
      Where N.branchIn = B
    
```

규칙 R23은 브랜치에서 송전소의 변화에 반응한다.

```

CREATE RULE R23 For branch
ON Insert, Update to passingPower
WHEN Inserted(B) ∨ Updated(B),
    
```

```

S = B.orientation.from,
T = (total of passingPowers of branches output
from S), T != S.powerOut
THEN Update S
Set S.powerOut = T
    
```

위 규칙들은 브랜치에서 입력 사이트에 의해 고려된다. 브랜치에서 전체 파워를 다시 계산하도록 하고 해당 사이트의 배전 파워에 따라 값을 할당한다. 이러한 규칙들은 또한 새로운 사용자, 브랜치 삽입, 중간 노드에서 파워에 대한 변화를 다룰수 있게 한다.

다음 두 가지 규칙 R24와 R25는 현재 트랜잭션을 롤백함으로써 파워, 노드, 그리고 브랜치에 의해 제공되는 최대 출력 전압의 위반에 반응한다. 규칙 R24와 R25는 상대적으로 규칙 R22와 R23보다 높은 우선 순위를 갖도록 BEFORE로 운영되도록 명세 한다. 그 때 만일 트랜잭션이 롤백된다면 규칙 R22 또는 R23은 수행할 이유가 없다.

```

CREATE RULE R24 For distributor
ON Insert, Update to powerOut
WHEN Inserted(D) ∨ Updated(D),,
NEW D.powerOut > D.maxPower
THEN Rollback
BEFORE R22
    
```

```

CREATE RULE R25 For branch
ON Insert, Update to passingPower
WHEN Inserted(B) ∨ Updated(B), NEW B.passingPower
> B.maxPower
THEN Rollback
BEFORE R23
    
```

위에서 설계한 에너지 관리에 대한 능동 규칙은 비록 직관적으로 의미가 명백하더라도 정확한 능동 규칙의 어의가 명세 되기 힘들다. 그러나 전압 분배 규칙이 튜플-지향적이든, 집합-지향적이든 규칙 수행은 정확하며, 튜플-지향적 수행은 사용자로부터의 각각의 변화를 발전소에 위임하고, 집합-지향적은 발전소로부터 같은 거리에 있는 것들이 효과적으로 함께 변화를 수행한다.

#### 4. 결론 및 향후 연구방향

본 논문에서는 능동 데이터베이스 시스템의 중심인 능동 규칙 시스템의 구조와 일반적 규칙언어의 형태를 제시하고 이를 바탕으로 에너지 관리 시스템을 위한 능동 규칙을 설계하였다. 능동 규칙은 EMS에 여러 다른 목적으로 사용될 수 있지만, 설계된 규칙은 전체적인 에너지 관리의 부분은 아니며, 토폴로지와 전압 분배 규칙의 응용성이나 가용성의 몇 가지만을 설계하여, 이를 토대로 능동규칙의 확장성을 제시하였다.

그러나 능동 개념을 효율적으로 지원할 수 있도록 규칙 개념 및 규칙 처리를 위한 규칙 시스템의 이론적인 근거가 되는 규칙 모델과 수행 모델에 관한 연구와 규칙 모델의 적합성과 효율성 등을 분석하는 기법에 대한 깊이 있는 연구가 필요하며, 규칙을 정의하고 명세 하는 규칙언어와 규칙언어에 의하여 명세된 규칙을 실행하기 위한 규칙의 실행의 미에 대한 전반적인 기술확보와 실질적인 능동 데이터베이스의 구현 및 객체 지향적 개념의 도입에 의한 확장 등이 향후 지속적으로 연구되어야 한다.

#### 참고문헌

- [Bau90] C. Bauzer Medeiros And P. Pfeffer, "A mechanism for managing rules in an object-oriented database," Altair Technical Report 1990.
- [Gat93] S. Gatzju, K.R. Ditrich, "Events an Active Object Oriented database system," in Proceedings of 1st International Workshop on Rules in Database Systems, 1993.
- [Geh92] N. Gehani et al, "Event Specification in an Active Object-Oriented Database," Proc. of the ACM-SIGMOD 1992.
- [Day88]U. Dayal, B. Blaustein, A. Buchman, U. Chakravarthy, M. Hsu, R. Ledin, D. McCarthy, A. Rosenthal, And S. sarin, "The HiPAC Project : Combining Active Databases And Timing Constraints", *SIGMOD Record, Vol. 17, No. 1*, 1988
- [Hug91] J. Hughes, "Object-Oriented Databases", Prentice-Hall Int'l Series in Computer Science, 1991.
- [Sim92] E. Simon, J. Kiernan, And C. de Maindreville, "Implementing High Level Active Rules on top of a Relational DBMS", *In Proceedings of the 18th Conference on VLDB*, 1992.
- [Wid90]J. Widom, And R.J. Finkelstein, "Set-oriented Production rules in relational database systems", *In Proceedings of the ACM SIGMOD*, 1990
- [Wid91]J. Widom, R.J. Cochran, B.G. Lindsay, "Implementing Set-Oriented Production Rules as an Extension to Starburst", *In Proceedings of the 17th International Conference on VLDB*, Barcelona, September, 1991
- [Wid96]Jenifer Widom, Stefano Ceri, Introduction to Active Database Systems Triggers And Rules For Advanced Database Processing, *Morgan Kaufman Publishing Inc.* 1996