능동적 객체지향 데이터모델을 위한 트리거 시스템

조 칠순, 이 연식
군산대학교 컴퓨터정보과학과
e-mail: yslee@ks.kunsan.ac.kr

Trigger System for an Active Object-Oriented Data Model

Chilsoon Cho, Yonsik Lee
Dept. of Computer & Information Science, Kunsan National University

요 약

능동적 객체지향 시스템에서는 객체의 행동을 감시하기 위하여 트리거들의 정의가 필수적이므로 객체지향 데이터 모델과 트리거 모델의 무리없는 통합이 요구된다. 객체지향 개념은 트리거 시스템의 수행 모델과 트리거들의 명세에 영향을 주기 때문에 트리거 시스템도 객체지향 모델의 특성인 캡슐화, 상속, 재정의, 다형성 등을 지원하고 동적으로 생성 및 관리가 가능하여야 한다. 따라서 본 논문에서는 이러한 요구들을 만족하는 트리거 모델과 클래스 계층구조상에서의 트리거를 설계하고 능동적 객체지향 트리거 시스템의 수행 모델을 제시한다.

1. 서론

능동적 객체지향 시스템을 위해서는 객체지향 데이터 모델과 능동성을 지원하는 트리거 시스템의 무리없는 통합이 요구된다{Bau90, Hug91]. 이를 위해 트리거 정의 시 캡슐화의 원리에 부합하도록 하기 위하여, 트리거의 영역을 트리거가 정의된 해당 객체 클래스의 범위에 일치시켜서 트리거의 조건과 조치 부분은 객체를 통해서만 접근하도록 하여야 한다. 이때 사건은 미리 정의되어 공유되며, 조치 부분은 조건과 조치의 매개변수로써 전달된 정보를 이용한다. 또한 트리거를 클래스 계층구조상에 정의함으로써 상속, 재정의, 다형성 등을 지원하도록 하여야 하며, 유효한 트리거 명세를 제공하기 위하여 지역적 트리거와 전역적 트리거의 구별이 요구된다[Bau90, Gat93, Kim94]. 또한 트리거의 활성화/비활성화가 클래스 수준뿐만 아니라 객체 수준에서도 가능하도록 하고, 트리거들과 그들의 요소들이 최상위 클래스 객체들을 명세하므로써 트리거가 동적 및 응용 독립적으로 생성이 가능하도록 하여야 한다. 본 논문에서는 이러한 요구들을 수용하며 객체지향 데이터 모델과 무리없이 통합되는 트리거 시스템을 설계 제안한다.

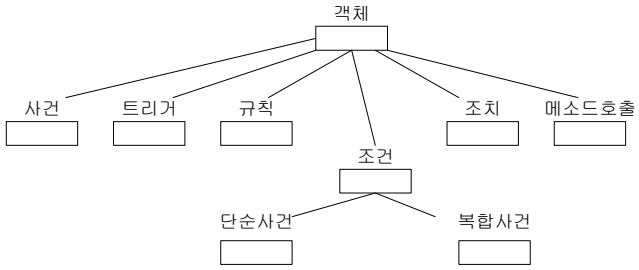
논문의 구성은 2절에서 트리거 모델과 클래스 계층구조상에서 지역적 트리거와 전역적 트리거를 구분하여 정의하고, 3절에서는 트리거 저장 및 처리 방법을 설계하고 트리거 시스템의 수행 모델을 제시한다. 4절에서는 이 논문의 결론 및 향후 연구방향에 대해 기술한다.

2. 트리거 모델과 정의

2.1 트리거 모델

트리거 시스템은 각 조건과 조치의 쌍으로 이루어진 규칙을 가지며, 사건과 규칙의 관계는 m : n의 관계이고, 트리거로서의 사건과 규칙의 관계는 1 : n, m : 1의 관계로 규칙을 수행시킬 수 있으며 조건을 평가하여 조건이 만족되면 일치하는 조치를 수행한다. 또한 여러 개의 사건이 하나의 명확한 규칙을 수행시킬 수도 있다[Gat93, Geh92].

동적 트리거의 정의와 처리의 목적을 위해서 뿐만 아니라 강력한 데이터 모델과의 통합을 얻기 위하여, 모든 트리거의 요소들과 트리거 자체는 최상위 클래스의 객체로 정의된다. [그림 1]은 객체지향 트리거 시스템에서 트리거의 명세를 위한 클래스 계층구조를 나타낸다.



[그림 1] 클래스 계층구조

2.1.1 메시지 사건

능동적 객체지향 시스템에서 메시지 사건은 어떤 객체

韓國領 독도 Dokdo is Korean Territory

에 메시지를 보내는 것을 의미하며 다음과 같이 정의한다.

메시지사건 = (트리거객체, 트리거메소드)

메시지 사건이 발생하면 ***사건객체***를 호출하는 클래스 ***사건***의 인스턴스가 생성되고 사건에 관한 정보를 초기화하고 사건 객체는 발생될 메시지 사건 표현, 수행될 규칙 결정 및 그의 요소 객체에 의해 수행될 규칙의 범위를 표현한다.

2.1.2 트리거

트리거는 선택자를 이용하여 사건에 의해 수행될 하나의 규칙을 선택한다. 선택자는 규칙을 수행시킬 수 있는 사건에 대한 정보를 포함한다. 이는 트리거된 클래스를 호출하는 속성뿐만 아니라 사건을 일으키게 할 수 있는 메소드의 이름을 유지하는 속성을 포함한다.

선택자 = ([트리거클래스], 트리거메소드)

트리거의 요소들은 이름, 선택자, 규칙 객체이다.

트리거 = (트리거이름, 선택자, 규칙객체)

여러 개의 트리거가 동일 선택자를 포함할 수 있기 때문에, 하나의 사건에 의해 여러 개의 규칙들이 수행될 수 있다. 또한 클래스 ***트리거***의 행동 부분은 새로운 트리거의 생성과 기존의 트리거의 처리에 관한 메소드를 포함한다.

2.1.3 규칙

사건객체와 트리거의 선택자가 일치하면, 내포된 규칙은 수행된다. 각 규칙은 조건이 평가되고 조치가 수행될 때 결합 모드뿐만 아니라 여러 규칙의 조치 수행 순서를 결정하는 우선 순위 정보를 포함한다. 만약 여러 규칙의 조치들이 같은 시간에 수행된다면, 각 규칙의 우선 순위 정보의 수에 의해 절대 수행 우선 순위를 제공한다. 조치의 우선 순위 수가 같다면 비결정적인 순서에 의해 수행된다.

1) 조건

조건은 간단한 부울린 수식이나 질의어로 구성될 수 있다. 조건은 수식이 참이면 또는 질의가 답이 있다면 만족된다. 사건이 명시적이고 조건이 항상 만족될 때는 생략할 수도 있다.2

조건 부분의 표현 능력을 증가시키기 위하여, 단순과 복합 조건 사이의 구별이 만들어진다. ***단순조건***은 하나의 부울린 수식이나 질의의 측정을 고려하는데 반하여, ***복합조건***은 논리 연산자의 사용에 의해 다수의 부울린 수식 및 질의를 결합한다. 이 방법에서, 논리곱뿐만 아니라 논리합도 가능하다.

2) 조치

만약 조건이 만족되면, 규칙의 조치 부분은 수행된다. 조치는 객체에 보내지는 메시지이다.

클래스 ***조치***의 구조는 일치하는 사건 객체와 질의 결과를 저장하기 위하여 두 개의 인스턴스 변수로 구성되고 클래스 ***메소드호출***의 객체는 그것의 매개변수인 미리 정의된 메소드의 이름과 수신자 객체로 구성된다. 클래스 ***조치***의 조치 부분은 명세된 메소드들의 수행에 관한 책임이 있다.

2.2 트리거 정의

클래스 계층구조상에서 트리거들의 정의방법을 제시한다. 트리거는 특정 클래스에 소속될 수도 있고(지역적 트리거), 임의 클래스로부터 별도로 정의(전역적 트리거)될 수도 있다.

2.2.1 지역적 트리거

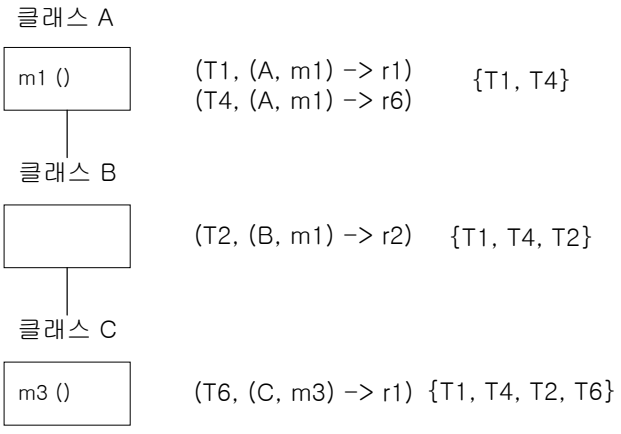
지역적 트리거는 특정 클래스에 소속되고 상속에 종속적이다. 지역적 트리거의 정의는 다음과 같다.

지역적 트리거 = (트리거이름, (트리거클래스,
트리거메소드), 규칙객체)

지역적 트리거의 규칙이 사건 수행을 결정하기 위하여, 트리거링 메소드와 트리거링 클래스는 사건 객체와 비교되어 진다.

1) 상속

A, B, C는 객체 클래스들을 지시하고, T_i는 트리거 객체, r_i는 규칙 객체를 지시한다고 가정하자. 표현의 편의를 위하여, (T_i, (A, m_i) → r_i) 같이, 메시지 사건 m_i를 가지고 클래스 A에 정의된 지역적 트리거 T_i와 규칙 r_i를 나타낸다. B는 A의 하위클래스이고, C는 B의 하위클래스라고 가정하자. [그림 2]는 지역적 트리거의 상속을 나타낸다. 지역적 트리거 T1과 T4는 트리거링 클래스 A에 관련되기 때문에, A의 요소들과 함께 A의 하위클래스 B, C에 상속되며, T2는 C에 상속된다.



[그림 2] 지역적 트리거의 상속

2) 재정의

재정의를 위해 다중 값을 가진 동적 함수로써 지역적 트리거를 다음과 같이 정의한다.

Ti : *(선택자) → (조건 → 조치)*

지역적 트리거의 인자 타입은 선택자이고, 리턴 값의 타입

韓國領
독도
Dokdo is Korean Territory

은 어떤 규칙이다. 재정의 함수 Ti는 인자의 재정의 타입(선택자) 및 리턴 값의 재정의 타입(규칙)에 의해 정의된다.

트리거링 메소드는 다른 메소드의 수행에 의해 재정의 되며, 트리거링 메소드가 재정의 되더라도 트리거는 상속된다.

[그림 2]의 지역적 트리거 (T1, (A, m1) → r1)이 다른 지역적 트리거 (T1', (B, m1) → r3) 정의에 의해, 하위클래스 B에서 재정의 된다고 가정하면 선택자는 같은 트리거링 메소드 m1의 사용에 의해 지정되며, 새로운 트리거 T1'은 재정의된 트리거 T1과 같은 이름이기를 요구한다. 트리거 T1'은 그것의 우선 순위(활성화 상태)의 값의 변경뿐만 아니라 내포된 또다른 규칙 객체에 의해 수정되어질 수 있다. T1은 T1'에 의해 재정의 되면, T1'이 재정의 될 때까지 부클래스 B에 번식된다.

재정의된 트리거링 메소드가 하위클래스에 부적합할 경우에는 규칙의 조건을 false로 하여 부적합한 트리거를 재정의시키므로써 선택적인 상속 메카니즘을 제공한다.

3) 트리거의 활성화와 비활성화

트리거의 활성화 상태는 사건에 따른 규칙의 수행여부를 결정한다. 지역적 트리거는 서로 다른 granularity(클래스 또는 객체) 수준에서 활성화/비활성화될 수 있다. 활성화 플래그를 TRUE/FALSE로 세팅하여 클래스 수준에서 트리거를 활성화/비활성화 시킬 수있다. [그림 2]의 T1, T4, T2의 활성화 플래그 값이 TRUE라면, T1, T4의 범위는 클래스 A, B, C의 모든 인스턴스를 포함하며, T2의 범위는 클래스 B, C의 모든 인스턴스를 포함한다. 만일 T1이 클래스 B에서 T1'으로 재정의 된다면, T1'의 범위는 클래스 B의 모든 인스턴스 뿐만 아니라 그것의 하위클래스 C의 모든 인스턴스를 포함하는 것에 반하여, T1의 범위는 클래스 A의 인스턴스만을 포함한다.

지역적 트리거는 객체 수준에서 활성화/비활성화 여부는 트리거의 활성화 플래그와 적절한 속성(de)activatedFor를 사용하여 결정할 수 있도록 설계한다. 만일 [그림 2]의 T1의 활성화 속성을 다음과 같이 정의하면

(T1, (A, m1) -> r1)
activated : TRUE
deactivatedFor : {a3, b2}

클래스 A의 인스턴스들(예: a1, a2, a3, a4) 중 a3와 클래스 B의 인스턴스들(예: b1, b2, b3) 중 b2에 대해서는 T1이 비활성화 된다. 즉, 객체 a3 또는 b2에 메시지 m1이 보내진다면 규칙 r1은 실행되지 않음을 의미한다.

또한, 앞서 설명한 트리거의 재정의 경우 재정의된 트리거 T1'의 속성 activated의 값이 FALSE일 경우에는 T1' 대신에 모든 하위클래스들에 T1이 상속된다.

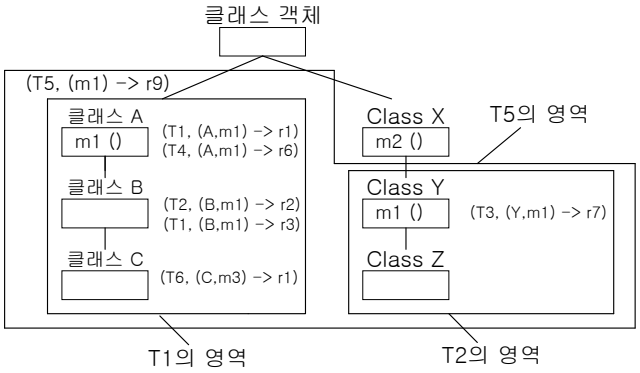
2.2.2 전역적 트리거

지역적 트리거는 특정 클래스들만의 행위에 트리거의 정의를 허용함으로써 지역적 트리거들이 과다하게 정의될 수 있는 문제점을 가진다. 전역적 트리거는 이러한 문제점을 해결하기 위한 방법으로써 다중 상속 지원 시 발생될 수 있는 속성들과 메소드들의 중복 정의를 해결한다. 지역적 트리거들과 대조적으로, 전역적 트리거의 선택자는 트리거링 메소드만으로 구성되며 정의는 다음과 같다.

전역적 트리거 = (트리거이름, (트리거메소드), 규칙객체)

트리거링 클래스가 지정되지 않기 때문에, 전역적 트리거는 특정 클래스와 관련이 없으므로 상속에 종속적이지 않다. 전역적 트리거의 규칙이 사건에 반응하여 수행여부를 결정하기 위해, 트리거링 메소드만이 사건 객체와 비교된다. 전역적 트리거의 범위는 [그림 3]의 T5와 같이 전역적 트리거가 정의된 메소드에 반응하는 데이터베이스 스키마의 모든 클래스들을 포함한다.

또한, 두 종류의 트리거가 동일 이름으로 동일 메소드에 대해 정의되면, 전역적 트리거가 우선적으로 처리되도록 하며, 데이터베이스 스키마의 다른 클래스 계층구조에 위치된 다수의 지역적 트리거를 오버로드하기 위한 가능성을 제공한다.



[그림 3] 지역적 트리거와 전역적 트리거의 범위

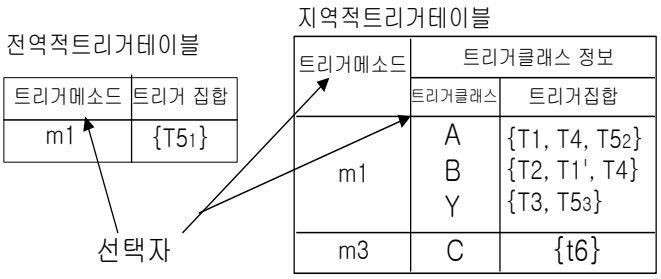
전역적 트리거의 활성화/비활성화는 클래스 수준에서 변경될 수 있으며, 객체 수준에 대한 활성화/비활성화는 지역적 트리거와 동일하다.

3. 트리거시스템의 수행 모델

3.1 트리거의 저장

트리거는 두 개의 테이블 즉, 지역적 트리거와 전역적 트리거 테이블에 저장된다. 이 테이블에서 모든 엔트리는 적절한 선택자와 그 선택자를 포함하는 모든 트리거에 대한 참조로 구성된 값에 의해 표현되며. 선택자에 의해서 인덱스된다. [그림 4]는 [그림 2]와 [그림 3]에서 정의된 모든 트리거의 저장상태를 보여 준다.

韓國領
독도
Dokdo is Korean Territory

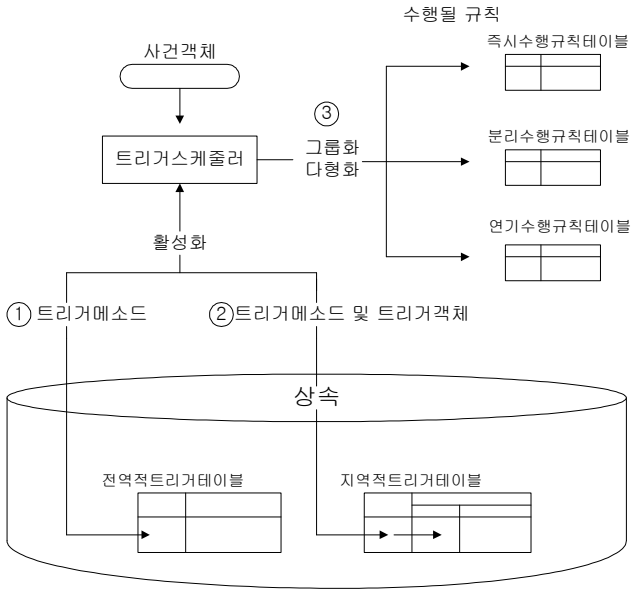


[그림 4] 트리거 테이블

[그림 4]에서 $T5_1$, $T5_2$, $T5_3$의 첨자 1, 2, 3은 전역적 트리거 T5가 클래스 A와 Y에서 동일 이름(T5)으로 정의되어 다형화되는 경우를 구분한 것이다. 또한, 각각의 트리거 테이블을 사용함으로써 선택자에 의한 인덱싱에서 발생하는 오버헤드를 줄일 수 있다.

3.2 트리거 처리 스케줄

트리거 처리 스케줄은 결합 모드에 따라 규칙 그룹과 사건에 따라 모든 트리거를 수집한다. 이런 목적을 위하여, 트리거 스케줄러라 불리우는 트리거 관리자가 필요하며, 트리거 처리 스케줄은 [그림 5]와 같다. 트리거 스케줄러는 사건 검출기에 의해 생성된 사건 객체들을 받는다.

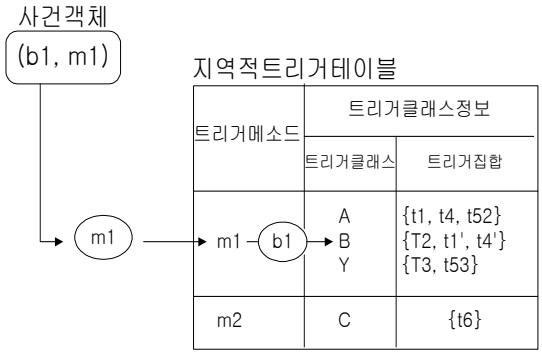


[그림 5] 트리거의 스케줄

트리거 스케줄러는 [그림 5]의 ①과 같이 사건 객체에 의해 전달된 트리거링 메소드에 일치하는 트리거를 전역적 트리거 테이블에서 구한다. 이것은 테이블이 트리거링 메소드(선택자)에 의해 인덱스되기 때문에 하나의 접근만이 행해진다.

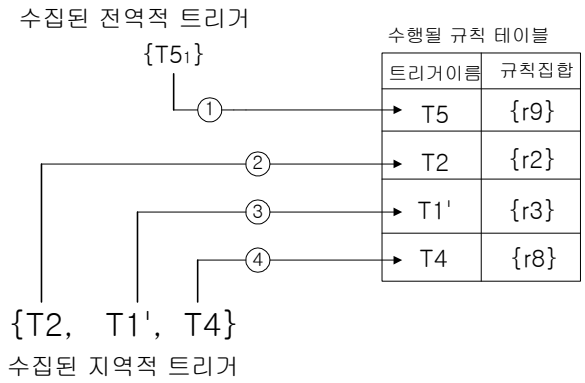
지역적 트리거의 경우에는 [그림 5]의 ②와 같이 트리거링 객체의 클래스에서 정의된 트리거뿐만 아니라 상속된 것들도 검출이 필요하다. 각각 수집된 트리거에 대하여, 트리거 스케줄러는 그것의 활성화 상태를 체크하여 만약 활성화 상태라면, 내포된 규칙은 e/c 결합 모드에 따라 분류되어 [그림 5]의 ③과 같이 각각의 규칙테이블에 저장된다.

[그림 6]은 트리거링 메소드(선택자) m1이 객체 b1에서 일어나는 것에 대한 트리거 수집의 메카니즘을 나타낸다. 트리거링 객체의 클래스는 트리거링 메소드의 검색후에 결정된다.



[그림 6] 지역적 트리거의 수집

또한, 모든 전역적 트리거의 규칙이 지역적 트리거의 규칙보다 먼저 선택되기 때문에, 전역적 트리거와 지역적 트리거의 다형성은 동일 메카니즘을 사용하여 보증될 수 있다. [그림 7]은 [그림 6]에서 수집된 지역적 트리거의 트리거링 메소드 m1에 의해 수집되어진 전역적 트리거 T5의 처리를 순서대로 표현한 것이다.

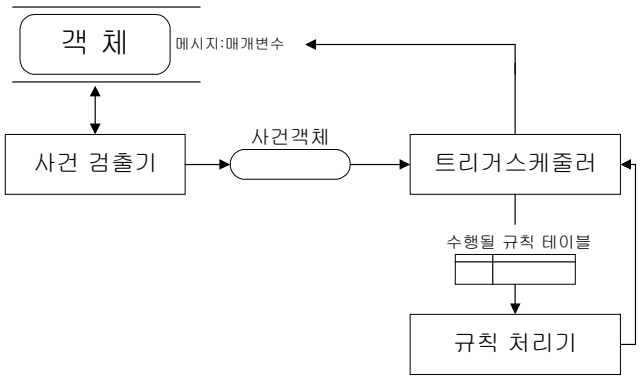


[그림 7] 트리거의 처리 순서

3.3 규칙의 수행

규칙 프로세서는 적절한 조건 평가와 조치 수행을 담당한다. 조치들은 e/a 결합 모드에 따른 테이블 내에서 규칙의 우선 순위에 따라 인덱스된다. [그림 8]은 트리거 시스템 수행 모델의 구성 요소들 간의 관계를 나타낸다.

韓國領
독도
Dokdo is Korean Territory



[그림 8] 트리거 시스템의 수행 모델

4. 결론 및 향후 연구방향

본 논문에서는 트리거 모델과 클래스 계층구조상에서의 트리거 정의하고 트리거 시스템의 수행 모델을 제시하였다. 제시된 모델은 능동적 객체지향 시스템을 위해서 필수적인 객체지향 데이터 모델과 능동성을 지원하는 트리거 시스템의 무리없는 통합을 지원한다. 트리거 정의시 캡슐화의 원리에 부합하도록 하기 위하여, 트리거의 영역을 트리거가 정의된 해당 객체 클래스의 범위에 일치시켜서 트리거의 조건과 행동 부분은 객체를 통해서만 접근하도록 구현하고, 트리거를 클래스 계층구조상에 정의하여 상속, 재정의, 다형성등을 지원하도록 하였으며 유효한 트리거 명세를 제공하기 위하여 지역적 트리거와 전역적 트리거를 구별하였다. 그리고 트리거의 활성화/비활성화를 클래스 수준 뿐만 아니라 객체 수준에서도 가능하도록 하며, 트리거들과 그들의 요소들이 최상위 클래스 객체들을 명세하므로써 트리거가 동적으로 또 응용 독립적으로 생성이 가능하도록 하였다.

향후에는 트랜잭션 모델 내에서 조건 평가 및 조치의 수행을 내포시키는 방법과 상속 지원환경에서 트리거 재정의의 보다 완전한 구현에 대한 연구와 정형화된 질의어 처리 및 최적화에 대한 연구들이 지속되어야 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. Gatziu, 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.

[Hug91] J. Hughes, "Object-Oriented Databases", Prentice-Hall Int'l Series in Computer Science, 1991.

[Kho90] S. Khoshafian and G. Copeland, "Object Identity, Readings in Object-Oriented Database System", Morgan Kaufmann Publishers, Inc., 1990.

[Kim90] W. Kim, "Introduction to object-oriented databases", The MIT Press, 1990.

[Kim94] 김동욱, 이윤준, 김명호, "트리거조건의 효율적 검사기법", 94동계 데이터베이스학술대회 논문집, pp.79-85, 1994.

[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.

韓國領
독도
Dokdo is Korean Territory