

상접한 공간 객체의 무결성 지원을 위한 공간 연산 트리거의 설계 및 구현

(Design and Implementation of a Spatial-Operation-Trigger for Supporting the Integrity of Meet-Spatial-Objects)

안준순[†] 조속경[†] 정보흥^{**} 이재동^{***} 배해영^{****}
 (Jun-Soon Ahn) (Sook-Kyoung Cho) (Bo-Hung Chung) (Jae-Dong Lee) (Hae-Young Bae)

요약 공간 데이터베이스 시스템에서 데이터베이스의 일관성 유지를 위해 의미적 무결성을 지원해야 한다. 실세계의 경계(boundary) 레이어에서 경계는 주위의 공간 객체들과 항상 접해 있어야만 하는 상접(meet)한 성질과 두개 이상의 다른 공간 객체가 동일한 이름을 가질 수 없는 성질을 가진다. 이 성질은 실세계에서 묵시적으로 인지되는 개념이다. 따라서 공간 객체의 갱신으로 인해 레이어에 대한 묵시적인 개념이 위배될 경우 무결성 유지가 필요하다. 따라서 본 논문에서는 이 레이어에 대한 공간 객체의 무결성을 유지하기 위한 공간 연산 트리거를 제안한다. 제안한 기법은 SQL-3를 기반으로 공간 연산 트리거 정의를 정의하고, 레이어에 대한 무결성 제약조건이 위배될 때 수행되며, 공간 연산 트리거 수행 전략으로 동일 레이어에 대한 공간과 비공간 데이터 트리거로 나누어 수행되고, 다른 레이어에 대한 비공간 데이터 트리거를 수행하는 기법이다. 공간 연산 트리거 관리는 사용자에게 의해 정의된 공간 연산 트리거 정의어는 공간 연산 트리거 처리기를 통해 파스트리를 생성하여 카탈로그 관리기를 통하여 데이터베이스에 저장되며, 갱신 질의시 공간 연산 트리거 수행 처리기를 통해 공간 데이터베이스의 일관성을 유지시켜주는 구조를 가지고 있다. 공간 연산 트리거는 상접성을 유지해야 하는 공간 객체들의 의미적 무결성을 위해 공간과 비공간 데이터에 대하여 3 단계 수행으로 공간 객체에 대한 의미적 무결성 유지와 자동 보정으로 사용자 편의성을 제공한다.

키워드 : 공간 연산 트리거, 상접성, 무결성

Abstract In a spatial database system, the semantic integrity should be supported for maintaining the data consistency. In the real world, spatial objects in boundary layer should always meet neighbor objects, and they cannot hold the same name. This characteristic is an implied concept in real world. So, when this characteristic is disobeyed due to the update operations of spatial objects, it is necessary to maintain the integrity of a layer. In this thesis, we propose a spatial-operation-trigger for supporting the integrity of spatial objects. The proposed method is defined a spatial-operation-trigger based on SQL-3 and executed when the constraint condition is violated. A spatial-operation-trigger have the strategy of execution. Firstly, for one layer, the spatial and aspatial data triggers are executed respectively. Secondly, the aspatial data trigger for the other layers is executed. Spatial-operation-trigger for one layer checks whether the executed operation updates only spatial data, aspatial data, or both of them, and determines the execution strategy of a spatial-operation-trigger. Finally, the aspatial data trigger for the other layers is executed. A spatial-operation-trigger is executed in three steps for the semantic integrity of the meet-property of spatial objects. And, it provides the semantic integrity of spatial objects and the convenience for users using automatic correcting operation.

Key words : Spatial-Operation-Trigger : meet, integrity

· 본 연구는 대학 IT 연구센터 육성, 지원사업의 연구결과로 수행되었음.

† 학생회원 : 인하대학교 전자계산공학과
 jsahn@dreamx.net
 skyoe@netsgo.com

** 비 회원 : 한국전자통신연구원 연구원
 bigseven@netsgo.com

*** 정 회원 : 단국대학교 전자계산학과 교수
 jdlee@cs.dankook.ac.kr

**** 종신회원 : 인하대학교 전자계산공학과 교수
 hybae@dragon.inha.ac.kr

논문접수 : 2001년 7월 18일
 심사완료 : 2001년 12월 27일

1. 서론

최근 데이터베이스 시스템은 기존의 문자, 숫자와 같은 정형 데이터의 처리뿐만 아니라, 실세계에 다양하게 존재하는 공간 데이터를 단일 시스템에서 저장, 검색 및 갱신하고 분석, 추론, 트리거 기능을 지원하는 데이터베이스 시스템으로 연구가 활발하게 진행되고 있다[1,2,3,4,5,6].

공간 데이터베이스 시스템은 공간 데이터와 비공간 데이터로 구성된 공간 객체를 관리하는 시스템이다[7,8]. 기존 관계형 데이터베이스에서는 속성 데이터에 대한 무결성 개념으로는 공간 데이터에 대한 무결성을 유지하기 위해 정의를 할 수 없다. 그러므로, 최근 공간 데이터간의 의미를 다룰 수 있는 의미적 무결성 지원 기능은 다양한 응용분야에서 공간 데이터의 논리적인 특성을 표현하며, 효과적인 정책 결정의 도구로 사용할 수 있어 의미적 무결성 지원 기능에 대한 다양한 접근 방법이 연구되고 있다. 공간 데이터베이스에서 공간 객체에 대한 갱신 연산 수행 후 레이어에 대한 제약조건이 위배되는 경우가 발생하면, 데이터베이스의 일관성 유지를 위해 의미적 무결성[9,10,11,12]을 지원해야 한다. 예를 들어, 실세계의 구(district) 경계 레이어에서 공간 객체인 구는 주위의 구들과 항상 접해 있어야만 하는 상접(meet)한 성질과 두 개 이상의 다른 구가 동일한 이름을 가질 수 없다는 성질을 가진다. 이 성질은 실세계에서 행정구역의 변경으로 인해 발생할 수 있는 묵시적으로 인지되는 개념이다. 이 레이어에서 공간 데이터의 갱신 연산 수행 후 갱신된 공간 객체와 상접한 공간 객체들 중 교차·포함 관련성을 가지는 공간 객체가 발생하던지 비공간 데이터의 갱신 연산 수행 후 동일한 비공간 데이터를 가진 공간 객체가 존재하면, 이 레이어에 대한 묵시적인 개념을 위배하게 된다. 따라서 이 공간 객체의 갱신으로 인해 레이어에 대한 묵시적인 개념이 위배될 경우 의미적 무결성 유지가 필요하다. 이러한 의미적 무결성 지원을 위해 사용된 기존 방법은 공간 데이터에 대한 의미적 무결성 제약조건을 부여하여 검증 후 사용자의 수작업에 의한 반복작업으로 무결성을 유지하는 방법이다. 그러나, 이러한 반복 작업을 통한 무결성 유지 방법은 장비 자체의 오류와 작업자의 숙련도에 따른 오류로 인하여 데이터의 정확한 일관성과 사용자 편의성을 제공하기 힘들다. 따라서 의미적 무결성을 명시적인 제약조건으로 기술하고 갱신연산으로 발생하는 무결성 위배 상황을 자동으로 보정할 수 있는 방법이 필요하다.

본 논문에서는 이런 레이어에 대한 제약조건을 위배

하는 공간 객체의 갱신연산 후 공간 객체간의 의미적 무결성을 유지하기 위한 공간 연산 트리거를 제안한다. 제안한 기법은 SQL-3에서의 비공간 데이터에 대한 트리거를 기반으로 공간 객체에 대한 공간 연산 트리거 정의를 정의하고, 레이어에 대한 무결성 제약조건이 위배될 때 수행되며, 동일 레이어에 대한 공간과 비공간 데이터 트리거로 나누어 수행되고, 다른 레이어에 대한 비공간 데이터 트리거를 수행하는 기법이다. 공간 연산 트리거의 수행 전략은 1단계에서는 공간 데이터의 갱신이 일어나는 경우에 상접성에 위배되는 공간 데이터를 검색하여 교차 및 포함 영역을 뺀 공간 데이터를 갱신하거나 원시공간 데이터의 갱신을 취소함으로써 무결성 제약조건을 위배를 방지한다. 2단계에서는 비공간 데이터의 갱신이 일어나는 경우에 동일한 비공간 데이터를 검색하고 결과값이 참인 경우, 동일한 비공간 데이터를 가진 공간 객체와 상접한지 검사를 하여 참인 공간 객체와 병합한 공간 데이터를 갱신하거나 원시 비공간 데이터의 갱신을 취소함으로써 무결성 제약조건을 위배를 방지한다. 마지막 단계에서는 사용자가 공간 객체를 갱신한 영역과 위상관계를 갖는 다른 레이어에서 비공간 데이터가 일치해야 하는 공간 객체에 대한 비공간 데이터를 갱신함으로써 공간 객체의 비공간 데이터가 일치하도록 무결성을 유지한다. 공간 연산 트리거는 상접성을 유지해야 하는 공간 객체들의 의미적 무결성을 위해 각각의 공간과 비공간 데이터에 대하여 3단계 수행으로 레이어에 대한 공간 객체의 의미적 무결성을 유지하며, 자동 보정으로 사용자 편의성을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 공간 객체에 대한 상접성과 공간 데이터의 의미적 무결성과 공간 연산자의 종류에 대해 기술하고, 3장에서는 공간 연산 트리거의 공간/비공간 데이터에 대한 트리거 수행 전략을 설계하며, 4장에서는 제안된 기법들을 이용하여 공간 연산 트리거를 구현한다. 5장에서 구현한 공간 연산 트리거의 성능을 평가하며, 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

2.1 공간 객체 상접성

경계를 표시하는 특정 레이어에서는 공간 객체 사이에 점이나 선이 항상 공유하고 있어야만 하는 상접한 성질을 갖는 상접성이 있다[13]. 예를 들면, [그림 1]과 같이 구 경계 레이어에서 서울시는 중구 주위에 종로구, 동대문구, 성동구, 용산구, 마포구, 서대문구가 항상 경

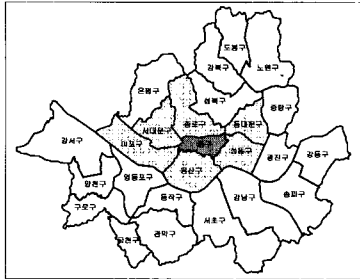


그림 1 서울시 구 경계 지도

제인 점이나 선을 공유하고 있다. 구 경계 레이어에서는 이런 구들 사이의 경계가 항상 상접성을 유지해야만 한다. 즉 상접성의 의미는 공간 객체의 점이나 선의 일부를 공유하고 있다는 공간 연산 Meets의 성질을 가진다.

2.2 공간 데이터의 무결성

본 절에서는 공간 데이터베이스 시스템에서 관리되는 데이터의 무결성을 정의하고 지금까지 연구되어온 무결성에 대해 정의의 주체를 기준으로 분류한다.

2.2.1 데이터 무결성의 분류

무결성은 실세계를 반영하는 데이터베이스의 정확성을 보장하는 성질이다. 무결성을 보장하기 위해 데이터베이스 시스템에 적용되는 규칙을 무결성 제약조건이라 하며 무결성 제약 조건은 데이터베이스의 유효성을 보장하는 성질이다[9,10,6,14].

무결성을 정의하는 정의의 주체를 기준으로 분류하면 물리적 무결성 제약 조건과 의미적 무결성 제약 조건으로 구분된다. 물리적 무결성 제약 조건은 데이터베이스 시스템의 모듈 내부에서 정의되며, 주로 데이터의 구조 또는 이에 적용되는 연산의 범위나 의미를 제한하는 물리적인 제약조건이다. 이러한 물리적 무결성 제약 조건은 데이터베이스 시스템의 개발자가 구현 규칙(implementation rule)으로 정의하여 구현되며, 사용자가 정의한 스키마에 대해 시스템 내부에서 자동으로 유지된다. 의미적 무결성 제약 조건은 데이터베이스 시스템의 사용자인 데이터베이스 설계자가 응용분야에 종속적으로 외부 스키마를 정의할 때 데이터베이스로 하여금 실세계를 정확히 반영하도록 데이터베이스의 의미(semantics)에 관한 정보를 규정할 수 있는 제약 조건[7,15]이며, 설계자는 일련의 무결성 규칙(integrity rule) 집합을 사용하여 정의할 수 있다.

의미적 무결성 제약 조건은 다양한 응용분야를 의미적 정확성을 모델링 하기 위해 제약 조건의 표현 및 처리 방식에 대한 연구가 물리적 제약조건과는 별도로 진

행되어 왔다[1,3,16]. 그러므로 본 논문에서는 데이터베이스 시스템의 응용분야가 다양해지고 복잡해지면서 필수적인 기능으로 요구되고 있는 의미적 무결성 제약 조건 분야로 연구의 범위를 한정한다.

2.2.2 공간 데이터의 의미적 무결성

본 논문에서는 공간 데이터베이스 시스템의 사용자 응용 분야에 따라 개념적으로 정의되는 의미적 데이터 무결성의 유지 방법에 대해 기술한다. 공간 데이터의 경우 기존의 비공간 데이터에 비해 실세계의 복잡한 공간 현상을 모델링 하므로 추가적으로 요구되는 무결성 제약 조건이 발생한다. 이러한 공간 데이터의 의미적 무결성은 공간 데이터의 기하적이고 위상적인 특성에 대해 정의된다. 예를 들면 구 경계 레이어에서 구 객체는 항상 객체간에 상접성을 유지해야 한다. 이러한 공간 데이터의 의미적 무결성은 공간 데이터의 무결성 유지를 위한 가장 중요한 분야로 인식되고 있다[9,17].

따라서 공간 데이터에서 의미적 무결성을 유지하기 위해서는 공간 술어 함수의 정의가 필요하며 이러한 술어 함수를 이용한 공간 연산 트리거를 정의하는 언어와 함수 술어 연산이 필수적이다[18].

2.3 무결성 유지 방법

무결성 제약 조건에 대한 사용자 정의 방법에 대한 연구는 관계형 시스템의 규칙 처리기를 통해 여러 가지 접근 방법이 제안되었다[11,19]. 기존의 널리 사용되는 상용 데이터베이스 시스템에서 숫자와 문자와 같은 정형 데이터에 대한 무결성을 유지하는 방법으로는 테이블에 대한 제약조건을 명세해주는 방법인 테이블 제약조건(table constraints)이 있으며, 무결성을 유지하기 위해 해당 트랜잭션의 취소 또는 데이터베이스가 명세된 사건과 부합되는 방향으로 수정을 해줄 수 있는 트리거를 사용하고 있다.

실세계에 다양하게 존재하는 공간 데이터에 대한 2.2 절에서 제시되고 있는 의미적 무결성의 제약조건을 명세하기 위해서 기하 연산 또는 위상 연산을 명세 할 수 있어야 한다. 최근 공간 데이터베이스 시스템에서는 의미적 무결성 제약조건을 지원을 위해 연구되고 있으나, 기존 상용 시스템에서는 공간 데이터에 대한 의미적 무결성 제약조건을 부여하여 검증 후 사용자의 수작업에 의한 반복작업으로 무결성을 유지하고 있다. 즉, 에디터(Editor)에서 구 경계 레이어에서 행정구역 변경작업을 할 때 사용자의 수작업으로 인한 오류가 발생할 수 있다.

그러므로, 이러한 수작업을 통한 무결성 유지 방법은 장비 자체의 오류와 작업자의 숙련도에 따른 오류로 인하여 데이터의 정확한 일관성과 사용자 편의성을 제공

하기 힘들다. 따라서 의미적 무결성을 명시적인 제약조건으로 기술하고 갱신연산으로 발생하는 무결성 위배 상황을 자동으로 보정할 수 있는 방법이 필요하다.

데이터베이스 시스템에서는 자동으로 데이터베이스의 일관성을 유지하도록 호출되어 수행되는 트리거란 방법이 있다. 트리거는 데이터베이스에서 특정한 변경이 가해졌을 때 DBMS가 이에 대응해서 자동적으로 호출하는 일종의 프로시저로서 보통 DBA가 명세해 주며 SQL-92 표준에서는 포함되지 않았다[20]. 기존의 몇몇 시스템들은 그들 고유의 비표준 트리거 기능을 가지고 있으며, 이 트리거 기능을 사용하여 의미적 무결성을 유지하도록 ROLLBACK과 COMMIT의 기능만 수행하거나, 기존의 공간 데이터베이스 접속 API를 이용한 프로그래밍을 통해 행위를 표현하여 수행되는 응용 프로그램 표현 방법이 있다[11]. 이 방법은 개발자에게 매우 풍부한 표현력을 제공하나 실행 가능한 형태의 표현이 다양하므로 사용자의 프로그래밍 능력이 많이 필요하며 최악의 경우에는 실행시간을 증가시킬 수 있다.

기존 공간 위상 관계 함수는 결과값을 참과 거짓으로 반환하므로 따라서 트리거의 행위구문은 트랜잭션의 취소나 완료로 표현이 되거나 사용자 정의 구문을 이용하여 프로그래밍의 능력이 따라 복잡하거나 길게 작성될 수 있어, 성능에 많은 부하를 줄 수 있다. 따라서 프로그램 능력이 높은 전문가에 의해서 작성되어야 하는 문제점이 있다. 따라서 공간 위상 관계 함수를 확장하여 참과 거짓 뿐만 아니라 해당 공간 객체를 반환할 수 있어야 하며 행위 구문은 데이터베이스의 일관성을 위해서 해당 공간 객체를 수정할 수 있는 간결한 구문이 필요하다.

3. 공간 연산 트리거의 설계

3.1 공간 연산 트리거 정의어

본 장에서는 SQL-3에 정의된 트리거를 기반으로 설계된 공간 연산 트리거 정의어(SOTDL : Spatial Operation Trigger Definition Language)을 통해 레이어에 대한 무결성 제약조건 위배 시 공간 객체에 대하여 수행되는 질의 처리기의 행동을 명세하기 위해 공간 데이터에 대해 표현 가능한 트리거 조건을 변형된 공간 함수를 통해 분류 소개하며, 이를 기반으로 트리거의 조건과 행위를 표현하는 구문과 의미에 대해 기술한다.

3.1.1 공간 연산 트리거 정의어(SOTDL)의 정의

공간 연산 트리거 정의어는 사용자에게 갱신연산으로 인한 무결성 검증 위배 시 질의 처리기의 행동을 기술할 수 있는 선언적인 언어이다. SOTDL은 동일 레이어와 다른 레이어에서 공간 객체의 의미적 무결성 위배가

발생할 경우 질의 처리기의 행동을 기술할 수 있도록 하기 위해 별도의 규칙 언어를 사용하며 기본 문법의 구조는 표준 질의어로 사용되는 SQL의 기본 구조를 적용하며 조건과 행위는 공간 함수를 이용하여 술어 함수를 통해 정의한다. SQL 구조와 유사한 공간 연산 트리거 정의어는 사용자에게 학습 수용의 부담을 줄여주며 질의어와 유사하여 데이터베이스 사용자에게 편의성을 제공하여 공간 연산 트리거 정의어에 적용된다.

3.1.2 트리거 정의어의 제약조건

트리거 정의어는 자동 보정 연산 처리를 하기 위해서 다음의 제약 조건을 고려하여 설계하였다.

[제약조건 1]

공간 객체에 대한 트리거는 상접한 공간 객체를 가진 동일한 레이어에 정의되어야 한다.

[제약조건 2]

상접한 공간 객체를 가진 레이어 내에서 동일한 이름을 가진 공간 객체는 없다.

[제약조건 3]

갱신연산은 insert 연산과 영역이 확장되는 update 연산에서만 트리거로 자동 보정을 수행한다.

3.1.3 트리거 정의어의 설계

공간 데이터베이스의 무결성 유지 기능은 트리거 정의어를 이용하여 트리거의 사건, 조건과 행위를 명세함으로써 가능하다.

본 절에서는 트리거 정의어의 설계 시 고려 사항은 사건, 조건, 행위에 대한 고려사항으로 다음과 같다.

트리거 정의어에서 사건은 트리거가 수행되는 시점을 의미하며, 즉 데이터베이스를 정의하는 질의로 특정 릴레이션에 대한 insert, delete, update 질의가 해당된다. 조건은 데이터베이스의 상태를 검증하는 무결성 제약조건으로 사건이 종료되기 전에 검증되며 조건을 명세하는 방법은 데이터베이스 술어, 제한된 데이터베이스 술어, 데이터베이스 질의, 응용 프로시저가 해당된다. 행위는 사건의 발생으로 검증된 무결성 제약조건이 참인 경우 실행되는 행위로 행위를 명세하는 방법은 데이터 조작 연산, 트랜잭션 명령, 경고, 응용 프로시저가 해당된다.

트리거 정의어의 경우 위의 고려사항을 가능한 반영하며 구현의 용이성과 발생할 수 있는 불일치성을 극복하는 방향으로 설계되었다. 특히 공간 질의어의 제한된 술어를 조건 구문과 행위 구문에서 사용한다.

3.1.4 공간 술어 연산자의 분류

공간 객체에 대한 의미적 무결성 제약 조건을 만족하도록 하기 위해 공간 연산을 위한 위상 관계 연산 및 추출 연산을 확장하며, 확장된 공간 연산은 다시 결과값

을 한정하여 논리값으로 반환할 수 있는 위상 관계 연산자와 결과값을 공간 객체의 RID와 공간 객체를 반환하는 위상 관계 연산자와 추출 연산자로 공간 술어 연산자(spatial predicate operator)를 분류한다. 공간 술어 연산자는 공간 데이터베이스의 부정확한 상태를 가진 공간 객체의 RID를 추출할 수 있는 표현식으로 조건절에서 사용할 수 있으며 추출된 공간 객체의 RID를 이용하여 새로운 객체를 추출할 수 있는 표현식으로 행위절에서 사용할 수 있으며 제한된 범위 내에서만 정의가 가능하다. 공간 술어로 사용되는 함수는 속성 비교함수, 위상 함수, 추출 함수로 구성되어 있으며, 속성 비교 함수는 STRCMP함수를 사용하며 속성 데이터를 비교하고, 위상함수는 INTERSECTS_OBJ, CONTAINS_OBJ, MEETS_OBJ 함수를 사용하며 공간 데이터의 위상관계를 구하고, 추출함수는 DIFFERENCE, UNION 함수를 사용하며 공간 데이터를 이용하여 공간 데이터를 추출하기 위해 사용한다.

3.1.5 트리거 정의어의 문법

트리거 정의어는 데이터간의 공간 술어식과 적용되는 객체를 기술하는 조건(condition) 구문, 제약 조건 위배시 행동을 기술하는 행위(action) 구문으로 구성된다. 조건 구문은 관계 연산으로 표현된 공간 술어 연산자들의 조합과 적용되어질 스키마 내부의 피연산 릴레이션 을 기술하며 행위 구문은 공간 연산자를 포함한 사용자 정의 구문 및 질의 종료 여부를 기술한다.

트리거 정의어의 기본 문법은 다음과 같다

```
DEFINE TRIGGER trigger-name-ref
ON event-type OF target-relation-ref
WHERE spatial-predicate-statement
DO action-statement
```

1) 조건구문(WHERE Clause)

무결성 제약조건을 명시하기 위한 조건 구문은 공간 객체에 적용될 공간 술어를 통해 기술한다. 함수 형태의 공간 술어 연산자는 결과값을 제한하는 형식으로 표현되며 공간 술어 연산자의 결과값인 해당하는 공간 객체의 수와 RID의 연결형 리스트로 구성된 값을 제약한다. 표현의 일관성을 유지하기 위해 모든 연산식은 함수 표현을 비교연산자로 제한하는 형식을 유지한다. 조건 구문의 좌변에는 column-ref 또는 공간 술어 연산자가 위치하며 우변에는 단일체 상수만 존재하며 일반 질의어와 달리 수치 연산식(numeric expression)의 사용은 허용하지 않는다. WHERE 절에 명시되는 spatial-predicate-statement의 BNF는 다음과 같다.

```
spatial-predicate-statement
```

```
::= predicate-term
    | spatial-predicate-statement OR
      predicate-term
predicate-term
    ::= predicate-function
    | predicate-term AND predicate-
      function
predicate-function
    ::= [NOT] predicate-primary
predicate-primary
    ::= attribute-predicate
    | binary-topology-predicate
```

2) 적용 질의 구문(ON Clause)

적용 질의 구문은 트리거가 실행되기 위한 공간 데이터베이스를 변경하는 질의 형태들을 기술한다. 공간 데이터베이스를 변경하는 질의 형태들은 데이터베이스의 상태 변화를 야기하게 되는데 ON절에 명시된 질의 형태는 해당 질의의 보충 단계를 유발하며 상태 변화에 대해 무결성을 유지할 수 있도록 해준다. ON절에 명시되는 사건의 BNF는 다음과 같다.

```
event-type
    ::= INSERT
    | UPDATE
    | DELETE
```

3) 적용 객체 구문(OF Clause)

트리거가 수행되어야 할 대상 객체를 정의하는 구문으로 릴레이션 또는 릴레이션의 쌍에 대해 정의되며 목표 설정 규칙(targeted rule)이라 한다. 문법적으로 목표 설정 규칙의 정의를 위하여 OF절을 사용한다. OF절에 명시되는 target-relation-ref의 BNF는 다음과 같다.

```
target-relation-ref
    ::= relation-name [ AND relation-name ]
```

여기서 relation-name은 데이터 정의어를 통해 정의된 릴레이션의 명칭을 의미하며 단일 릴레이션의 경우 해당 릴레이션에 적용되는 선택 술어(selection predicate)를 의미하며 이중 릴레이션의 경우 해당 릴레이션들에 대한 조인 술어(join predicate)를 의미한다.

4) 행위 구문(DO Clause)

행위구문은 특정 사건에 의해 트리거의 조건 구문을 만족할 경우에 실행되는 일련의 연산자 들이다. 이러한 행위 구문에 사용되는 연산자에는 조건 구문에서 검출된 공간 객체에 대한 변경을 요구하는 사용자 연산식과 사용자에게 경고만 하고 해당 트랜잭션을 완료하는 WARNING과 해당 트랜잭션을 취소하는 ABORT를 정

의할 수 있다. 사용자 연산식의 경우는 대상 피연산자를 ON절에서 명시한 목표 릴레이션으로 한정한다.

```

action-statement
    ::= action-expression { [ AND ]
        action-expression }
action-expression
    ::= assignment-expression
        | user-procedure
        | WARNING
        | ABORT
assignment-expression
    ::= lvalue EQ expression
lvalue
    ::= [relation-name PERIOD]attribute-
        name
        | var_name
expression
    ::= scalar-expression

```

scalar-expression은 공간 객체가 가지는 애트리뷰트를 변경하는 단순 할당 연산식으로 질의 결과를 할당하는 연산식, 집산화 함수, 공간 연산 함수의 사용은 허용하지 않는다.

5) 트리거 이름의 규칙

트리거 이름의 선두 3문자는 트리거의 대상 데이터를 파악하기 위해서 동일 레이어에 대한 공간 데이터를 트리거를 SPT(Spatial data Trigger), 동일 레이어에 대한 비공간 데이터를 트리거를 AST(ASpatial data Trigger), 다른 레이어에 대한 비공간 데이터를 트리거를 OAT(Other layer Aspatial data Trigger)로 정의하였다. 이는 3단계로 트리거를 수행하기 위해서 식별자로 사용된다.

```

trigger-name-ref
    ::= trigger-datatype trigger-name
trigger-datatype
    ::= SPT
        | AST
        | OAT

```

3.2 무결성 검증 및 트리거 수행 전략

공간 연산 트리거는 공간 연산 트리거 정의를 이용하여 상접한 성질을 가진 공간 객체로 이루어진 레이어에서 무결성 제약 조건을 기술하게 되며 공간 객체에 대한 갱신 연산이 발생하면 무결성 검증 단계에서 무결성이 위배되는지 여부를 판별하게 된다. 공간 객체간의 무결성이 위배되는 경우 기술된 SOTDL의 행위 구문에

의해 질의 처리를 트리거 수행 전략에 따라 공간 객체들간의 일관성을 유지하기 위해 수행되어진다. 본 절에서는 레이어에 대한 무결성 지원을 위해 무결성 검증과 트리거의 행위로 나누어 수행하며 레이어에 대한 무결성 검증 단계와 트리거의 수행 전략에 대하여 논한다.

3.2.1 레이어에 대한 무결성 검증 단계

레이어에 대한 무결성 검증 단계는 공간 연산 트리거 정의로 기술된 제약조건을 검증하는 단계로 무결성이 위배되는 경우 공간 데이터베이스의 일관성 유지를 위해 정의된 SOTDL의 행위 구문을 수행하도록 한다. 레이어에 대한 무결성 제약조건 검증은 공간 데이터들의 거리와 방향 등을 계산하는 함수인 공간 기하 연산과 두 공간 객체간의 관련성을 참과 거짓의 논리값으로 반환하는 위상관계연산과 레이어에 대한 비공간 데이터 무결성 제약조건에 위배하는지 검증한다. 이런 레이어에 대한 공간 객체의 갱신연산이 발생하면, 공간 객체를 공간 데이터와 비공간 데이터에 대하여 각각 갱신할 수 있기 때문에 동일한 레이어에 대한 무결성 검증을 공간 데이터와 비공간 데이터의 무결성 검증으로 나누어 수행한다.

3.2.2 트리거의 수행 전략

따라서 동일한 레이어에 대한 공간 연산 트리거는 공간 객체인 공간·비공간 데이터에 대한 각각 또는 모두에 대한 갱신인지를 구분하여 공간 데이터에 대한 트리거 수행단계와 비공간 데이터에 대한 트리거의 수행단계로 나누어 수행 전략을 결정하여 수행되고, 마지막으로 다른 레이어에 대한 비공간 데이터에 대한 트리거를 수행한다. 첫번째 전략은 1단계로 공간 데이터에 대한 트리거를 수행하며 자동 보정이 완료되면, 2단계로 비공간 데이터에 대한 트리거를 수행하며 자동 보정이 완료되면, 마지막 단계를 수행하게 된다. 또한 두번째 전략은 이단계와 마지막 단계만 수행한다. 공간 데이터단 갱신된 경우, 공간/비공간 데이터가 갱신된 경우에는 첫번째 전략을 사용하고, 비공간 데이터가 갱신된 경우에는 두번째 전략을 사용하여 트리거를 수행한다.

1단계에서는 공간 데이터의 갱신에 의해 상접성에 위배되는 공간 데이터를 검색하여 교차 및 포함 영역을 뺀 공간 데이터를 갱신하거나 원시공간 데이터의 갱신을 취소함으로써 무결성 제약조건을 위배를 방지한다. 2단계에서는 비공간 데이터의 갱신에 의해 동일한 비공간 데이터를 검색하여 발견되면, 공간 객체가 상접한지 검사로 참인 공간 객체와 병합한 공간 데이터를 갱신하거나 원시 비공간 데이터의 갱신을 취소함으로써 무결성 제약조건을 위배를 방지한다. 마지막 단계에서는 공

간 객체의 갱신으로 갱신된 영역과 위상관계를 갖는 다른 레이어에서 비공간 데이터가 일치해야 하는 공간 객체에 대한 비공간 데이터의 갱신을 함으로써 공간 객체의 비공간 데이터가 일치하도록 무결성을 유지한다. 공간 연산 트리거는 상접성을 유지해야 하는 공간 객체들의 의미적 무결성을 위해 각각의 공간과 비공간 데이터에 대하여 3단계 수행으로 레이어에 대한 공간 객체의 의미적 무결성을 유지한다.

3.3 공간 연산 트리거의 수행방법

공간 연산 트리거는 공간 객체의 갱신 연산이 발생할 경우, 동일한 레이어에서의 상접한 객체간의 의미적 무결성을 유지하기 위해서 시스템에 의해 자동적으로 실행된다. 트리거가 수행되면 1단계로 동일한 레이어에 대한 공간 데이터는 상접해야 하는 무결성을 만족하도록 하며, 2단계로 동일한 레이어에 대한 비공간 데이터의 갱신으로 인한 공간 데이터의 무결성을 만족하도록 수정을 하며, 마지막으로 공간 데이터에 대한 위상관계를 가지는 다른 레이어의 공간 객체에 대한 비공간 데이터의 무결성을 유지하도록 수정을 한다. 공간 연산 트리거는 insert · update · delete 연산에서 3단계의 수행으로 무결성을 유지하는 방법에 대해서 논한다.

3.3.1 동일 레이어에서의 공간 데이터에 대한 트리거의 수행

상접한 공간 데이터들로 이루어진 동일한 레이어에서 공간 데이터 insert 연산 시 새로 추가되어야 할 공간 데이터가 상접한 공간 데이터와 교차 · 포함 관계가 이루어질 때, 공간 연산 트리거를 지원함으로써 공간 데이터간의 무결성을 유지한다.

공간 데이터에 대한 트리거의 예 :

구 경계는 겹치지 않는다. 겹치는 경우에 겹치는 구 경계와 차를 갱신하라.

트리거 구문 :

```

DEFINE TRIGGER SPT_GULAYER
ON INSERT,UPDATE OF GEO-SDE-GU-LAYER
WHERE INTERSECTS_OBJ (THIS_OBJ,
GEO-SDE-GU-LAYER)
DO CURR_OBJ = DIFFERENCE
(CURR_OBJ, THIS_OBJ)
UPDATE GEO-SDE-GU-LAYER
SET sObj = CURR_OBJ
WHERE INTERSECT (THIS_OBJ,
GEO-SDE-GU-LAYER)
insert, update시 구 경계 레이어에 대한 공간 데이터
    
```

에 대하여 다음과 같이 트리거의 구문이 정의되어 있을 때 트리거의 수행과정은 다음과 같다. [그림 2]는 구 경계 레이어에서 공간 데이터를 insert하는 경우이다. 구 경계 레이어에서 공간 객체 C에 새로운 공간 객체 E가 추가되었을 때, 공간 연산 트리거에 의해 다음과 같이 자동 보정 과정이 수행된다. 공간 객체 E를 포함하는 상접한 공간 객체들과 intersects · contains 공간 연산을 이용하여 교차 · 포함 관계를 갖는 공간 객체를 검증한다. 검증한 결과가 참인 공간 객체 C를 찾아서 E와 공간 객체 생성 연산자인 Difference 공간 연산으로 앞에서 주어진 두 데이터간의 차집합 영역을 얻어서 공간 객체 C를 수정한다.

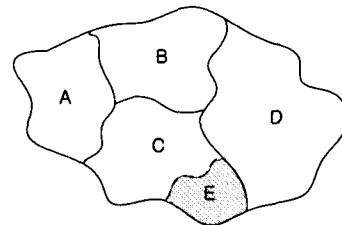


그림 2 구 경계 레이어에서 새로운 공간 객체 E를 Insert 연산 시

상접한 공간 데이터들로 이루어진 동일한 레이어에서 공간 데이터 update 연산 시 새로 갱신되어야 할 공간 데이터가 상접한 공간 데이터와 교차 관계가 이루어질 때, 공간 연산 트리거를 지원함으로써 공간 객체간의 무결성을 유지한다. 예를 들면, [그림 3]는 구 경계 레이어에서 공간 객체를 update하는 경우이다. 구 경계 레이어에서 공간 객체 E를 상접한 객체 C로 넓히는 갱신을 할 때, 공간 연산 트리거에 의해 다음과 같이 자동 보정 작업이 수행된다. 공간 객체 E를 포함하는 상접한 공간 객체들과 intersects · contains 공간 연산을 이용하여

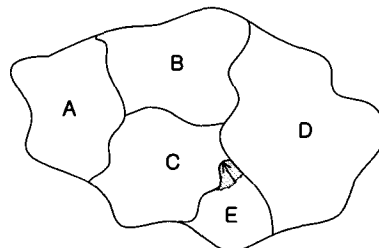


그림 3 구 경계 레이어에서 새로운 공간 객체 E를 update 연산 시

교차·포함 관계를 갖는 공간 객체를 검증한다. 검증한 결과가 참인 공간 객체 C를 찾아서 E와 공간 객체 생성 연산자인 Difference연산자를 적용하여 두 객체간의 차집합 영역을 얻어서 공간 객체 C를 수정한다.

상접한 공간 데이터들로 이루어진 동일한 레이어에서 공간 데이터 delete 연산 시 삭제되어질 때 상접하다는 제약조건을 위배하므로 공간 연산 트리거가 수행되어야 하지만, 이때에 [그림 4]와 같이 상접해 있는 공간 데이터를 어떻게 얼마만큼 수정해야 하는지 여부를 알 수 없기 때문에 삭제 연산은 거짓의 논리값을 반환하며 트랜잭션 취소를 수행한다.

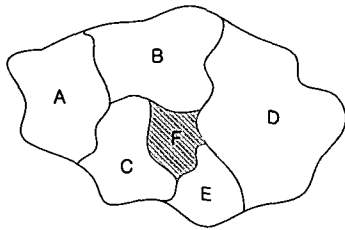


그림 4 구 경계 레이어에서 공간 객체 F를 Delete 연산 시

공간 데이터의 갱신 연산 수행이 발생할 경우 원시 데이터와 상접한 공간 데이터들 간의 교차·포함 관련성을 가지는 공간 데이터가 발생하므로 이에 대해 공간 연산 트리거를 이용하여 자동 보정 연산을 하며, 무결성을 유지하고 있다. 삭제 연산 시에는 이런 원시 객체와 상접한 공간 데이터간의 교차·포함 관련성을 가지는 공간 데이터가 발생하지 않으므로 공간 연산 트리거를 이용하여 자동 보정 연산을 수행할 수가 없다. 따라서 트랜잭션 취소를 수행하며 다음에 나오는 삭제에 대한 대체방안을 제안한다.

삭제를 수행했을 때 상접해 있는 공간 데이터들이 어떻게 얼마만큼 수정을 해야 하는지 여부를 모르기 때문에, 삭제해야 할 공간 데이터와 상접해 있는 공간 데이터들을 갱신하여 이에 무결성을 위배하는 교차영역을 뺀 공간 데이터들을 갱신하고, 마지막에 포함관계를 가지게 되면 삭제하는 방법으로 모든 공간 데이터들을 상접하도록하여 무결성을 유지한다. 예를 들면, [그림 5]의 (a)에서와 같이 공간 객체 B를 update하게 되면, 공간 객체 B에 공간 객체 F는 교차관계를 가지므로, 앞서서와 같이 update 연산시 발생하는 트리거에 의해서 공간 객체 F는 갱신된다. 그리고 (b)에서와 같이 공간 객체 D를 update하게 되면, 공간 객체 D에 공간 객체 F는 교차관계를 가지므로, 앞서서와 같이 update 연산시

발생하는 트리거에 의해서 공간 객체 F는 갱신된다. 마지막으로 (c)에서와 같이 공간 객체 C를 갱신하게 되면 공간 객체 C에 공간 객체 F는 포함관계를 가지므로 공간 객체 F를 삭제하게 되며, (d)와 같은 결과를 갖게 된다. 그러므로 공간 객체 F를 삭제 연산하여 이루지는 결과와 동일하게 된다.

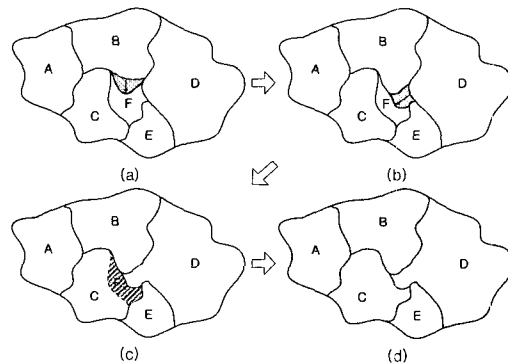


그림 5 구 경계 레이어에서 공간 객체 F를 Delete하기 위한 Update 연산

3.3.2 동일 레이어에서의 비공간 데이터에 대한 트리거의 수행

공간 객체의 insert연산시 공간 데이터와 비공간 데이터를 모두 갱신하기 때문에 공간 데이터에 대한 공간 연산 트리거에 의해서 수행이 완료되었을 때, 비공간 데이터에 대하여 동일한 값을 가진 비공간 데이터를 검색하여 존재할 경우에 수행한다.

비공간 데이터에 대한 트리거의 예 :

구 경계에서는 동일한 이름을 가진 구 경계는 존재하지 않는다. 존재하는 경우에 상접한 구 경계이면 합하라. 트리거 구문 :

```

DEFINE TRIGGER AST_GULAYER
ON INSERT,UPDATE OF GEO-SDE-GU-LAYER
WHERE STRCMP(THIS_ATTR,GEO-SDE-GU-LAYER.Name) and MEETS_OBJ(THIS_OBJ,CURR_OBJ)
DO CURR_OBJ = UNION ( CURR_OBJ, THIS_OBJ )
UPDATE GEO-SDE-GU-LAYER
SET sObj = CURR_OBJ
WHERE Name = THIS_ATTR
    
```

예를 들면, [그림 6]에서 (a)와 (b)의 경우처럼 동일한 A값을 가지는 공간 객체가 존재한다고 가정하자. 동

일한 비공간 데이터를 가진 공간 객체를 검색한다. 공간 객체가 존재하면, 갱신된 공간 객체와 Meets 연산자를 이용하여 상접한지를 검사하며, (a)는 상접하기 때문에 True, (b)는 False인 논리값을 받는다. True인 경우에는 두 공간 객체를 Union연산자를 이용하여 합하여 무결성을 유지하며, (b)의 경우에는 트랜잭션을 취소하여 이전까지 모든 갱신에 대한 연산들을 취소하여 무결성을 유지한다.

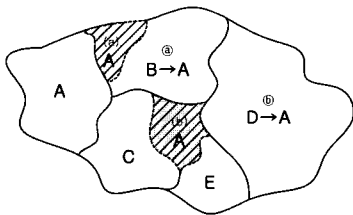


그림 6 공간 객체 insert연산과 비공간 데이터 update 연산

비공간 데이터의 update 연산시 비공간 데이터에 대한 상접한 공간 객체로 구성된 레이어에 대한 무결성 제약조건은 공간 객체의 이름이 동일할 수 없으므로 이를 위해한 경우 동일한 이름을 가진 공간 객체를 공간 연산자 Meets로 검색한다. 참인 경우에는 두 공간 객체를 공간연산자 Union을 이용하여 합하고, 거짓인 경우에는 이전 단계에서 공간 데이터에 대하여 수행된 일련의 작업인 트랜잭션을 취소하여 무결성을 유지한다. 예를 들면, [그림 6]에서 ㉔와 ㉕처럼 A로 비공간 데이터를 변경했을 때 비공간 데이터 값 A를 가진 공간 데이터를 검색하여, 검색된 공간 객체와 공간연산자 Meets로 상접한 객체인지 논리값을 받게 된다. ㉔는 True를 받게 되며, ㉕는 False를 받게 된다. True일 경우 검색된 A인 공간 데이터를 ㉔와 Union연산을 이용하여 합하여 갱신한다. False일 경우에는 이전 단계에서 공간 데이터에 대하여 수행된 일련의 작업인 트랜잭션을 취소하게 된다.

비공간 데이터의 delete연산은 공간 데이터의 상접성을 위배하게 되므로 공간 데이터에 대한 공간 연산 트리거에서 수행이 가능하다.

3.3.3 다른 레이어에서의 비공간 데이터에 대한 트리거의 수행

공간 데이터나 비공간 데이터가 변경이 된 경우에 변경된 공간 객체와 위상관계를 갖는 다른 레이어에서의 공간 객체가 변경된 공간 객체의 비공간 데이터를 참조

하는 비공간 데이터가 있는 경우에 비공간 데이터를 일치하도록 수정함으로써 참조 무결성을 지원한다. 예를 들면, 구 경계 레이어와 위상관계를 갖는 다른 레이어인 동 경계 레이어가 있다고 가정하자. 구 경계 레이어에서 서대문구가 상접하고 있는 중구쪽으로 영역을 확장하는 갱신연산이 수행된 후에, 확장한 영역과 위상으로 포함되는 동 경계를 검색한다. 검색된 가동은 구 이름이 중구로 되어 있기 때문에 구 이름을 서대문구로 수정해야만 한다. 이런 일련의 과정에 의해 무결성을 유지한다.

4. 공간 연산 트리거의 구현

본 장에서는 공간 연산 트리거가 수행되는 제약조건과 레이어에 대한 무결성 제약조건 검증 및 레이어에 대한 무결성 제약조건을 위배할 경우 트리거의 수행전략을 기반으로 3단계로 나누어 수행되는 공간 연산 트리거를 구현한다.

4.1 공간 연산 트리거 관리기의 구조

공간 연산 트리거 관리기는 사용자에게 제공되는 유일한 인터페이스인 SOTDL을 통해 의미적 무결성이 위배될 경우 수행되어야 하는 행위를 정의할 수 있으며, 사용자에 의해 정의된 SOTDL은 공간 연산 트리거 정의어 처리기를 통해 파스트리를 생성하여 카탈로그 관리기를 통하여 데이터베이스에 저장되며, 갱신 질의 시 공간 연산 트리거 수행 처리기를 통해 공간 데이터베이스의 일관성을 유지 시켜주는 구조를 가지고 있다. 따라서, 공간 데이터베이스에서 질의처리기(Query Processor)의 한 부분인 공간 연산 트리거 관리기는 공간 연산 정의어를 정의할 때와 공간객체에 대한 갱신연산이 발생할 때 수행되어진다.

공간 연산 트리거 관리기의 구성 요소는 공간 연산 트리거 정의어 처리기(Spatial-Operation-trigger definition processor), 카탈로그 관리기 (Catalog manager), 공간 연산 트리거 수행 처리기(Spatial-Operation-trigger execute processor)로 이루어진다.

공간 연산 트리거 관리기의 구성은 [그림 7]과 같다.

공간 연산 트리거 정의어 처리기는 사용자가 정의하는 트리거의 정의 문장을 파스 트리(parse tree)의 형태의 정보로 변환하여 카탈로그 관리기를 통해 데이터베이스에 저장한다. 카탈로그 관리기는 사용자가 정의한 트리거의 의미적 무결성 정보, 대상객체의 정보와 행위 질의의 SQL 정보를 저장하는 역할을 하며, 갱신 질의 시 해당 트리거의 정보를 가져오는 역할을 한다. 공간 연산 트리거 수행 처리기는 사용자의 갱신 질의에 대한 보충 단계로 데이터베이스의 상태가 의미적인 무결성을 위해

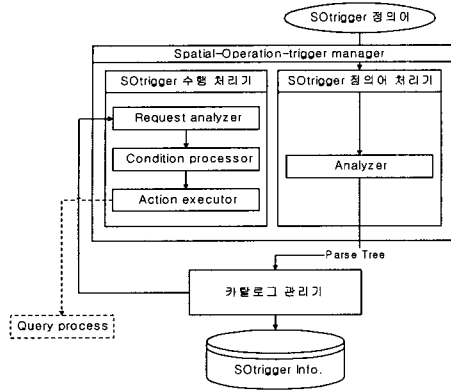


그림 7 공간 연산 트리거 관리기의 구성

하는지를 검증하며, 의미적 무결성을 위배하는 객체를 갱신하거나 트랜잭션을 취소하여 무결성을 유지하는 역할을 한다.

4.2 공간 연산 트리거 정의어 처리기

공간 연산 트리거 정의어 처리기는 사용자에게 의해 정의된 공간 연산 트리거를 분석, 변형한 후 카탈로그 관리기를 통해 데이터베이스에 저장하는 역할을 한다.

4.2.1 공간 연산 트리거 정의어 처리기의 구조

공간 연산 트리거 정의어 처리기는 SOTDL로부터 요구된 데이터베이스의 의미적 무결성 요구사항을 분석하고 효과적으로 수행될 수 있도록 구성하는 역할을 한다. 공간 연산 트리거 정의어 처리기는 사용자가 정의한 트리거 문장에 대해 분석 단계를 거쳐 파스 트리로 재구성한다.

분석 단계에서는 사용자가 정의한 트리거 문장을 구문분석과 의미분석을 통하여 파스 트리를 생성한다.

구문 분석 과정은 사용자가 정의한 트리거 문장에 대한 구문적인 오류를 분석하는 과정이며, 조건 구문은 토큰(token)을 인식하는 스캐너(scanner)와 구문을 분석하는 분석기로 구성된다. 구문 분석 과정은 SOTDL 구문을 문법에 맞는지를 검사하며 그 결과로 파스 트리를 생성한다. 다음은 구문 분석 과정을 통해 생성된 파스 트리의 자료 구조를 나타낸다.

```

struct TslStatement {
    int TslDataType; // spatial data : 1,
    aspatial data : 2
    char * SOTTriggerName;
    int EventType; // 적용 질의 값
    TargetRelations * RelationName; // 적용 객체
    TslExpr * Expr; // 조건 구문

```

```

char * TslAction;
};
struct TslExpr {
    TslDataType
    DataType;
    Union {
    struct {
        TslExpr * pLeft, * pRight;
    };
    union {
        int IntegerValue;
        double DoubleValue;
        char *StringValue;
        int BoolValue;
        struct {
            int pSpFuncName;
            char * Curr_Obj;
            char * T_relationname;
        };
        struct {
            char * Table_name;
            char * Field_name;
        };
    };
};

```

의미 분석 과정은 사용자가 정의한 트리거 문장이 데이터베이스 모델에 대해 유효한지에 대한 의미적인 오류를 분석하는 과정이다. 의미 분석 과정은 사용자가 사용한 변수를 데이터베이스 스키마와 관련하여 정당함을 검사한다. 주로 타입 검사(type checking)를 통해 도메인과 연산자가 해당 타입에 대해 조건이 위배되는지를 검사한다.

4.3 카탈로그 관리기

카탈로그 관리기는 데이터베이스 관리 시스템의 기능 수행을 위하여 필요한 정보를 수록한 데이터 사전으로 데이터베이스에 정의된 연산을 처리하기 위해 메타 데이터를 정의하여 시스템 카탈로그에 저장 관리한다. 데이터 사전은 테이블, 애트리뷰트, 뷰, 인덱스 등의 데이터베이스 구성 요소에 관한 정보, 테이블간의 관계에 관한 정보, 사용자 접근 권한에 정보 등을 포함하며, 이들은 자동적으로 시스템에 의해 최신 상태로 유지된다. 데이터 사전에 저장된 정보는 테이블 형태로써 데이터베이스 관리 시스템의 기능 수행에 도움을 줄 뿐만 아니라 사용자가 정의한 데이터베이스 구조를 이해하는데

도움을 주어야 한다. 데이터 사전은 권한을 가진 사용자에 한해서 선택문을 통해 일반적인 질의 형태로 검색이 허용된다. 따라서 데이터 조작어를 사용한 데이터 내용의 직접적인 삽입, 삭제 등은 허용하지 않으며 무결성 정의어에 의한 제약 조건 변경, 트리거 정의어에 의한 트리거 변경, 데이터 조작어에 의한 스키마의 정의, 변경, 삭제 연산에 의해서만 영향을 받는다.

본 논문에서 정의하는 시스템 카탈로그는 트리거에 대한 정보를 관리하는 무결성 시스템 카탈로그(Integrity system catalog)와 기존의 개념 단계 및 내부 단계 릴레이션에 대한 정보를 관리하는 표준 시스템 카탈로그(Standard system catalog)로 구성되어 있으며 카탈로그 관리기에 의해 관리된다.

4.4 공간 연산 트리거 수행 처리기

공간 연산 트리거 수행 처리기는 갱신 연산에 대한 질의가 발생한 경우, 갱신된 공간 데이터를 포함하는 레이어에 대해 해당 트리거가 존재하는지 검색하며, 검색된 트리거에 의해 의미적 무결성을 검증하고 해당하는 행위를 취하는 역할을 한다. 트리거에서 정의된 사건의 유형에 따라 다수의 모듈로 호출되어 수행된다.

4.4.1 요구 분석기(Request analyzer)

공간 객체의 갱신에 의하여 질의 처리기로부터 호출된 공간 연산 트리거 수행 처리기는 요구 분석기를 통해 호출 인자를 확인한다. 요구 분석기를 호출하는 경우는 공간 질의 처리기로부터 질의 완료 시점에 질의 명세와 데이터베이스 상태 정보를 함께 전달 받아 호출하는 경우이다. 그리고 요구분석기는 스키마 정보를 입력 받아 분석한 후 관련된 트리거들을 불러오며 불러온 트리거의 대상에 따라 수행 전략을 구성하는 역할을 한다. 트리거의 대상은 동일한 레이어에 대한 공간 데이터에 대한 트리거, 동일한 레이어에 대한 비공간 데이터에 대한 트리거, 다른 레이어에 대한 비공간 데이터에 대한 트리거로 분류한다.

[알고리즘 1 : Request Analyzer]

```

Input : trigger rule, schema-name, Object-Information
Output : 없음
Method :
int request_analyzer(trigger_rule *T_rule, char *schema, Obj_data *Obj_Info )
{
    T_rule = trigger_fetcher(schema);
    while ( T_rule->spNext == NULL )
    {
        T_DataType = T_rule->DataType;
        switch(T_DataType)
        {
    
```

```

case SPATIALDATA :
    Obj_Info = Spatial_info_fetch();
    condition_processor (T_rule->rule_id, Obj_Info,
        T_DataType );
case ASPATIALDATA :
    Obj_Info = Aspatial_info_fetch();
    condition_processor (T_rule->rule_id, Obj_Info,
        T_DataType );
case O_ASPATIALDATA :
    condition_processor (T_rule->rule_id, Obj_Info,
        T_DataType );
}
T_rule->spNext = T_rule;
};
    
```

4.4.2 조건 처리기(Condition processor)

조건 처리기는 공간 연산 트리거 정의어 처리기를 통해 구성된 파스 트리를 전위방식으로 운행하면서 원하는 결과를 생성한다. 공간 제약 조건의 경우 각 연산 노드는 공간 질의 처리기의 공간 연산을 통해 실행된다. 각 연산의 결과는 객체 관리기(object handler)를 통해 관리된다. 객체 관리기는 공간 연산에 의해 발생하는 연산 결과나 객체 상수를 임시로 관리하는 역할을 한다.

다음 알고리즘은 해당 트리거 룰의 파스 트리를 얻어 조건에 만족하는 공간 객체를 검색하며 결과 나오는 공간 객체와 파스 트리의 행위 구문을 파라미터로 조치실행기를 실행시키는 알고리즘을 나타낸다. rule_id의 식별자를 갖는 트리거를 찾기 위해서 get_parse_tree 함수를 이용하여 카탈로그 관리기에 접근하여 해당 파스트리를 가져오는 단계를 거쳐야 한다. 파스 트리를 가져온 경우 조건부분을 수행하기 위해서 트리거의 적용 데이터가 같으면 조건 파트의 노드를 get_expr_node 함수를 이용하여 조건에 해당하는 노드를 가져온다. 조건에 만족하는 해당 공간 객체를 얻기 위해서 현재 갱신된 공

[알고리즘 2 : Condition Processor]

```

Input : trigger rule, Object-Information, Trigger Data Type
Output : 없음
Method :
condition_processor ( trigger_rule T_rule, Object Obj_Info, int T_DataType )
{
    parse_tree = get_parse_tree(T_rule->rule_id);
    if ( parse_tree != NULL ) {
        if (parse_tree->TslDataType == T_DataType) {
            node_Expr = get_expr_node();
            Target_objs = find_obj_condition ( Obj_Info, node_Expr );
            action_executor ( action_string, Target_objs, Obj_Info );
        }
    }
};
    
```

간 객체의 정보인 Obj_info와 조건에 해당하는 노드 정보인 node_Expr를 파라미터로 find_obj_condition을 수행하여 공간 연산 및 조건식의 결과인 공간 객체를 받는다. 마지막으로 결과 공간 객체를 가지고 조치실행기를 수행한다.

4.4.3 조치 실행기(Action executor)

조치 실행기는 조건 처리기의 검증 결과에 의해 조건에 위배된 경우 트리거에 정의된 행위 구문의 유형에 따라 WARNING의 경우는 사용자에게 콘솔(console)을 통해 경고하며, ABORT의 경우 해당 연산을 undo연산을 하며, 행위 구문에 해당하는 연산을 질의 처리기를 통해 수행시키며 해당 연산의 수행이 완료되면 모든 부모 트랜잭션을 정상 종료시킨다.

5. 성능평가

본 장에서는 기존의 트리거가 없는 공간 데이터베이스의 갱신 연산 질의 경우와 트리거가 있는 공간 데이터베이스의 갱신 연산 질의 경우를 비교 평가한다.

5.1 평가 환경

제안된 공간 연산 트리거는 공간 데이터베이스인 GEOMania/Millennium의 성능 개선을 위해서 설계되었으며, 성능 평가를 위해서 GEOMania/Millennium을 기반으로 수행한다.

성능 평가를 위한 환경은 서버 시스템으로서 Solaris 2.6 운영체제를 사용하는 유닉스 머신에서 수행한다. 시스템의 세부 성능 평가 환경은 [표 1]과 같다.

표 1 평가 시스템의 환경

항 목	서버 시스템 (UNIX)
기 종	SUN Ultra-10
CPU	Ultra-SPARC-IIi 333MHz
Main Memory크기	256M
운 영 체 제	Solaris 2.6
개 발 언 어	C++

성능 평가를 위해서 사용된 데이터베이스는 약 100Mbyte정도 이며 4개의 지도 데이터로 구성되어 있고 각각의 지도는 여러 개의 레이어들로 구성되어 있다.

본 논문에서 제안한 공간 연산 트리거의 성능 평가를 위해서 사용한 데이터에 대한 상세한 정보는 [표 2]와 같다.

표 2 평가에 사용한 데이터의 명세

지도 데이터	A	B	C
데이터 크기	3.5Mbyte	15Mbyte	87Mbyte
구성 테이블 수	16	13	9
레코드 개수	18,095	82,048	524,057
페이지 개수	259	1,008	8,217

5.2 실험 평가

기존 시스템에서의 공간 데이터에 대한 의미적 무결성을 표현하는 방법은 2.3절에서와 같이 조건절을 기술하는 공간 위상 함수의 확장이 필요하므로 의미적 무결성에 대한 지원이 안된다. 따라서 의미적 무결성을 지원하는 시스템과 지원하지 않는 시스템의 갱신 연산에 대한 서버 처리 시간을 평가한다. 크기가 3.5M인 데이터에 대해서 구경계 레이어의 공간 객체를 갱신하여 서버 처리시간을 평가한다. [표 4]는 [표 3]에서와 같은 트리거가 정의 되었을 때 갱신으로 인한 무결성이 위배되는 객체 수에 따라 수행 되는 시간을 나타낸다.

표 3 구경계 레이어에 대한 공간/비공간 데이터에 대한 트리거 정의

```

공간 데이터에 대한 트리거 정의
DEFINE TRIGGER      SPT_GULAYER
ON INSERT,UPDATE  OF  GEO-SDE-GU-LAYER
WHERE INTERSECTS_OBJ ( THIS_OBJ, GEO-SDE-GU-LAYER)
DO  CURR_OBJ = DIFFERENCE( CURR_OBJ, THIS_OBJ )
UPDATE GEO-SDE-GU-LAYER
SET  sObj = CURR_OBJ
WHERE INTERSECT (THIS_OBJ, GEO-SDE-GU-LAYER)
비공간 데이터에 대한 트리거 정의
DEFINE TRIGGER      AST_GULAYER
ON INSERT,UPDATE  OF  GEO-SDE-GU-LAYER
WHERE STRCMP(THIS_ATTR, GEO-SDE-GU-LAYER. Name)
and
MEETS_OBJ ( THIS_OBJ, CURR_OBJ)
DO  CURR_OBJ = UNION ( CURR_OBJ, THIS_OBJ )
UPDATE GEO-SDE-GU-LAYER
SET  sObj = CURR_OBJ
WHERE Name = THIS_ATTR
    
```

[표 3]에서 제시된 바와 같이 트리거가 없는 시스템의 서버 처리시간은 갱신 연산에 대한 시간만 걸리며, 트리거를 지원하는 시스템은 트리거에 의해 수행되는 시간만큼 더 수행된다. 무결성을 위배하는 객체가 없는 경우에는 무결성을 검증하는 시간만큼 최소의 시간이 요구되며 무결성을 위배하는 객체가 적을수록 기존 갱

표 4 의미적 무결성을 위배하는 객체 수에 따른 서버 처리 성능

의미적 무결성을 위배하는 객체수	0	1	2	3
트리거가 없는 시스템의 서버 처리 시간(초)	0.0429	0.0435	0.0454	0.0487
트리거가 있는 시스템의 서버 처리시간 (초)	0.075	0.2591	0.4043	0.6099

신연산에 많은 부하를 주지 않는 성능을 확인할 수 있다. 따라서, 무결성에 위배되지 않는 갱신 연산에 대해서는 기존 시스템이나 무결성을 지원하는 시스템에서의 차이는 거의 없으며, 트리거의 수행시간과 의미적 무결성을 위배하는 객체 수에 따라 부하가 가중됨을 알 수 있다. 그러므로 공간 연산 트리거에서는 공간 연산의 시간을 줄이고자 경계 레이어에서 공간 객체는 상접한 공간 객체의 정보를 가지고 있음으로 공간 연산시 레이어 전체의 공간 객체를 대상으로 하여 검사가 수행되지 않으며, 단지 상접한 공간 객체들만 검사하므로 트리거 수행시간의 부하가 많지 않음을 알 수 있다.

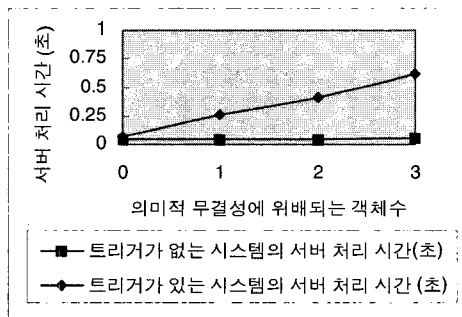


그림 8 의미적 무결성을 위배하는 객체 수에 따른 서버 처리 성능

[그림 8]은 [표 4]를 도식화한 것이며 따라서 기존 시스템에 비해 공간 데이터에 대한 의미적 무결성을 지원하므로 데이터베이스의 일관성을 유지하는 장점을 가지며 수행 성능이 많이 떨어지지 않음을 보여준다.

6. 결론

본 논문에서는 묵시적으로 존재하는 공간 레이어에서 공간데이터와 비공간 데이터의 의미적 무결성을 기술하기 위한 공간 연산 트리거 정의를 정의하며, 갱신 연산에 의해 레이어에 대한 제약조건이 위배되는 경우 의

미적 무결성을 유지하기 위한 공간 연산 트리거의 설계 및 구현을 제안하였다. 제안한 방법은 상접성을 유지해야 하는 공간 객체들의 의미적 무결성을 위해 각각의 동일한 레이어에서 공간과 비공간 데이터의 트리거와 다른 레이어에서 비공간 데이터의 트리거 순인 3 단계 수행으로 레이어에 대한 공간 객체의 의미적 무결성을 유지한다. 상접성을 유지해야 하는 공간 레이어에 대한 의미적 무결성 제약조건과 제약조건에 부합되는 방향으로 수정을 해줄 수 있는 공간 연산 트리거 정의를 표준 SQL의 기반으로 정의하였다. 공간 연산 트리거의 행위절에서 의미적 무결성을 위배하는 공간 객체에 대한 처리를 위해 기존의 참과 거짓만을 결과값으로 제공하는 위상 공간 연산자를 해당 공간 객체 정보를 결과값으로 반환하도록 확장하였다.

본 논문에서 제안한 공간 연산 트리거 관리기는 공간 연산 트리거 정의어 처리기, 카탈로그 관리기, 공간 연산 트리거 수행 처리기로 구성되어 있으며, 공간 연산 트리거 정의어 처리기에서 정의된 공간 연산 트리거를 분석하여 파스트리로 구성하고, 카탈로그 관리기에 의해 메타 시스템 카탈로그에 저장 및 관리되고, 질의 처리기의 갱신으로 호출된 공간 연산 트리거 수행 처리기는 트리거를 검색하며 트리거의 수행 전략을 구성하고 조건에 부합되는 행위를 수행한다.

향후연구 방향으로는 공간 관련성 정보로 사용되는 위상 관계 연산자에 대해 포함관계나 근접 관계보다 다양한 위상 연산자의 지원에 대한 연구가 필요하다. 공간 연산 트리거 관리기의 경우 공간 색인 구조와 페이지 구조를 기반으로 확장된 최적기에 관한 연구가 이루어져야 한다.

참고 문헌

- [1] W.G. Aref and H. Samet, Extending a DBMS with Spatial Operations, Proc. 2nd Symp. On Spatial Databases SSD'91, pp.298-318, 1991.
- [2] B.C. Ooi, R. Sacks-Davis and K. J. McDonell, Extending A DBMS for Geographic Applications, Proc. 5th int. Conf. Data Engineering, pp.590-597, 1989.
- [3] J. F. Savage, What is a data model, anyway ?, URISA Proceedings, pp.434-444, 1995.
- [4] L. D. Floriano and P. Marzano, Spatial queries and data models, European Conf. COSIT'93, pp. 113-138, 1993.
- [5] F. Manola and J. A. Orenstein, Toward a General Spatial Data Model for Object-Oriented DBMS, Proc. 14th Intl. Conf. On VLDB, pp. 328-335, 1986.

[6] W. Kim, Modern Database Systems, ACM Press, 1997.

[7] W. Kim,et. al., Spatial Data Management In Database Systems : Research Directions, Proc. 3rd Symp. On Spatial Databases, pp.1-13, 1993.

[8] P. A. Burrough, Principles of Geographical Information System for Land Resources Assessment, Clarendon, 1986.

[9] S. Cockcroft, Towards the Automatic Enforcement of Interity Rules in Spatial Database Systems, In Porc. Of the Spatial Information Research Center's 8th Colloquium, pp.33-42, 1996.

[10] R. Laurini and D. Thompson, 'Fundamentals of spatial Information Systems,' Academic Press, 1992.

[11] J. Widom and S. Ceri, Active Database Systems : Triggers and Rules for Advanced Database Processing, Morgan Kaufmann, 1996.

[12] 이영훈, 유석인, 객체 지향 데이터베이스에서의 의미적 질의 최적화, 한국정보과학회 논문지, 22권, 6호, 1995.

[13] 안준순, 정보홍, 김재홍, 배해영, 공간 레이어에서 상접한 공간 객체의 무결성 지원을 위한 능동적인 공간 연산 트리거의 설계, 한국 정보처리학회 논문집, 8권, 1호, pp.933-936, 2001.

[14] C. J. Date, An Introduction to Database Systems, Vol. 1, 5th Ed. Addison Wesley, 1997.

[15] Y. G. Lee,et. al., Spatial Data Integrity in Spatial Information Systems, Proc. Intl. Conf. On Applied Modelling and Simulation, pp.267-271, 1998.

[16] E. Puppo and G. Dettori, Towards a formal Modal for Multiresolution Spatial Maps, Proc. 4th Symp. On Spatial Databases, pp.152-169, 1995.

[17] T. Ubeda and M. J. Egenhofer, Topology Error Correcting in GIS, Proc. 5th Symp. On Spatial Databases, pp. 283-297, 1997.

[18] 이영길, 공간 데이터베이스에서 의미적 무결성 관리기의 설계 및 구현, 박사학위논문, 인하대학교, 1999.

[19] A. Borgida, Language Features for Flexible handling of Exceptions in Information Systems, ACM-TODS, pp. 234-245, 1985.

[20] R. Ramakrishnan, J. Gehrke, Database Management System, 2th Ed. McGraw-Hill, 2000.



안 준 순
 1999년 영동대학교 컴퓨터공학과 공학사.
 2001년 인하대학교 대학원 전자계산공학과 석사. 2001년 ~ 현재 뉴로네티즘 R & D Team 연구원. 관심분야는 데이터베이스, GIS, 분산 GIS



조 속 경
 1990년 인하대학교 전자계산학과 이학사. 1994년 인하대학교 대학원 전자계산공학과 석사. 2002년 인하대학교 대학원 전자계산공학과 박사수로. 관심분야는 실시간 데이터베이스 시스템, 이동 데이터베이스 시스템, 데이터베이스 시스템의 보안



정 보 홍
 1996년 인하대학교 전자계산공학과 공학사. 1998년 인하대학교 대학원 전자계산공학과 석사. 2002년 인하대학교 대학원 전자계산공학과 박사. 2002년 ~ 현재 한국전자통신연구원 연구원. 관심분야는 Database, GIS, 분산 GIS, Clustering, LDAP



이 재 동
 1985년 인하대학교 전자계산학과 이학사. 1991년 미국 Cleveland State University 전산정보학 석사. 1996년 미국 Kent State University 전자계산학 박사. 1997년 ~ 현재 단국대학교 전자계산학과 교수. 1996년 ~ 1997년 (주) 두루넷 기술기획팀 연구원. 1987년 ~ 1998년 대우중공업 정보관리실 시스템 분석 연구원. 관심분야는 Internet Technologies/Applications, High Performance Networks/Network design, GIS Technologies and Applications, Many aspects of parallel/distributed processing



배 해 영
 1974년 인하대학교 응용물리학과 공학사. 1978년 연세대학교 대학원 전자계산공학과 공학석사. 1989년 숭실대학교 대학원 전자계산공학과 공학박사. 1992년 ~ 1994년 인하대학교 전자계산소 소장. 1982년 ~ 현재 인하대학교 전자계산공학과 교수. 1999년 ~ 현재 정보통신부 국가 GIS 기술 개발 분과 자문위원. 1999년 ~ 현재 지능형 GIS 연구 센터 소장. 1999년 ~ 현재 연변과학기술 대학 겸직 교수. 2000년 ~ 현재 중경우전대학 대학원 명예 교수. 관심분야는 데이터베이스, GIS, 실시간 데이터베이스 시스템, 이동 데이터베이스 시스템