

# OGIS OLE DB 기반의 공간관련성을 가진 분산 공간객체의 변경 프로토콜

김 동 현<sup>†</sup> · 문 상 호<sup>††</sup> · 홍 봉 희<sup>†††</sup>

## 요 약

지리정보시스템에서 이질적인 다중 데이터 소스간의 상호운용성을 지원하기 위하여 OGIS OLE DB 데이터 제공자를 이용하여 분산된 공간 데이터에 접근한다. 그러나 OGIS OLE DB 데이터 제공자인 OLE/COM 구현 명세는 공간데이터의 발견 및 접근을 위한 표준은 제시하고 있지 않으나, 공간데이터 변경을 위한 인터페이스는 정의하고 있지 않다. 이 논문에서는 이질적인 공간데이터 서버에 분산되어 있는 공간객체들의 변경을 지원하기 위하여 OGIS OLE DB를 기반으로 정의된 서비스 제공자에 공간객체 변경 인터페이스와 변경된 공간객체의 분산 공간관련성 검사를 위한 공간관련성 검사 인터페이스를 추가한다. 또한, 동시 변경된 분산 공간객체들 간의 일관성을 보장하기 위하여 공간관련성 기반의 2단계 완료 프로토콜을 제시한다. 그리고 이 논문에서 제시한 추가 인터페이스와 2단계 완료 프로토콜의 실현을 위한 프로토타입 시스템을 설계 및 구현한다.

## Update Protocol for Distributed Spatial Objects with Spatial Relationships based on OGIS OLE DB

DongHyun Kim<sup>†</sup> · SangHo Moon<sup>††</sup> · BongHee Hong<sup>†††</sup>

## ABSTRACT

For supporting interoperability among heterogeneous data sources in different geographic information systems, it is required to use open API like OGIS OLE DB proposed in OGC. The OLE/COM Implementation Specification of OGIS only provides interfaces to discover and access spatial data, but does not support interfaces for updating distributed spatial data. To solve this problem, we propose and add interfaces for updating spatial objects distributed over heterogeneous data sources and examining the consistency of spatial relationships between updated objects. We also devise a 2-phase commit protocol based on spatial relationships in order to guarantee the consistency of updated objects. To justify the update scheme using the proposed interfaces and 2-phase commit protocol, we design and implement a prototype system for updating distributed spatial objects.

**키워드** : 공간 데이터베이스(Spatial Database), 변경 프로토콜(Update Protocol), 갱신 트랜잭션(Update Transaction), OGIS OLE DB, 일관성 제어(Consistency Control)

## 1. 서 론

지리정보시스템에서 공간데이터 변경 작업은 크게 두 가지 방식에 의해 수행된다. 첫 번째는 일괄 변경(batch update) 방식으로 데이터베이스의 전체 데이터를 일괄적으로 변경하는 방법이다. 두 번째는 사용자 대화식 변경(user-interactive update) 방식으로 각 공간객체를 사용자와의 대화식 작업을 통하여 변경하는 방법이다. 이 방식은 일괄 변경 방식에 비하여 사용자의 직접적인 변경 작업을 필요로 하는 단점이 있지만, 소규모의 데이터를 쉽게 변경할 수 있으며 변경 기간 동안에 다른 사용자가 공간데이터를 이용하는 것

을 허용하는 장점이 있다. 기존의 지리정보시스템은 특정 사용자 그룹에 의해서만 활용되기 때문에 사용자 대화식 변경 방식에 의한 변경 작업은 폐쇄적 클라이언트-서버(closed client-server) 환경에서 특정 관리자에 의해 주로 이루어졌다. 이러한 폐쇄적 환경하에서 공간데이터 변경 작업은 협동 작업을 통하여 수행되었으며 공간관련성에 따른 데이터 종속성을 이용하여 변경된 공간데이터의 일관성을 보장하였다[1, 2]. 그러나 네트워크 기술과 다양한 지리정보시스템 응용의 발달로 인하여 컴퓨팅 환경이 폐쇄적 클라이언트/서버 환경에서 개방형 환경으로 바뀌고 있다. 이러한 개방형 환경에서는 클라이언트가 이질적인 다중 데이터 소스에 위치하는 공간데이터에 접근 및 변경하는 기법이 필요하다.

일반적으로 개방형 환경에서 다중 데이터 소스에 위치하는 공간데이터의 발견 및 접근을 위해 OGC(OpenGIS Con-

<sup>†</sup> 정 회 원 : 동서대학교 소프트웨어 전문대학원 교수

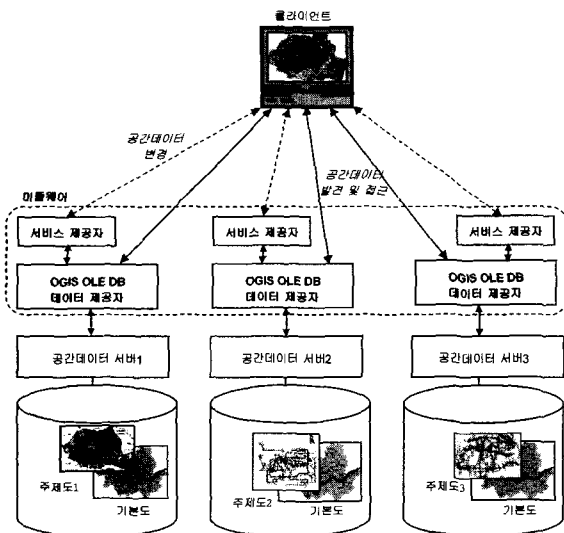
<sup>††</sup> 정 회 원 : 부산외국어대학교 컴퓨터공학과 교수

<sup>†††</sup> 정 회 원 : 부산대학교 컴퓨터공학과 교수

논문접수 : 2003년 7월 14일, 심사완료 : 2003년 11월 27일

sortium)에서 개방형 환경의 표준으로 제안하는 OGIS OLE DB 데이터 제공자를 이용한다[3]. 이질적인 데이터 소스들의 공간데이터를 사용하기 위한 일관된 인터페이스를 제공하기 때문에 OGIS OLE DB 데이터 제공자는 다양한 데이터 소스간의 상호운용성(interoperability)을 지원할 수 있는 장점이 있다. 그러나 OGIS OLE DB 명세인 OLE/COM 구현 명세는 공간데이터의 발견 및 접근을 위한 표준 인터페이스를 제시하지만, 공간데이터 변경을 위한 표준은 제공하지 않는 문제가 있다. 또한 분산 공간데이터의 변경을 지원하기 위하여 OGIS OLE DB 데이터 제공자 자체를 변경하는 방법은 OGIS OLE DB 데이터 제공자가 제공하는 상호운용성을 저해하는 문제가 있다. 따라서 다중 데이터 소스에 저장되어 있는 분산 공간데이터의 변경을 위해서는 OGIS OLE DB 데이터 제공자를 기반으로 하는 서비스 제공자에 변경 인터페이스를 추가할 필요가 있다.

분산 공간데이터 변경을 위한 서비스 제공자는 다음 사항들을 지원해야 한다. 첫째, 이질적 구조의 다중 데이터 서버에 접근할 수 있는 상호운용성을 제공해야 한다. 클라이언트는 이질적인 데이터 소스에 저장되어 있는 공간데이터에 대해서도 일관된 인터페이스를 통하여 변경할 수 있어야 한다. 둘째, 다양한 데이터 소스들에 분산되어 저장되어 있는 객체들 간의 공간관련성에 따른 일관성을 유지할 수 있어야 한다. 만약 서로 다른 데이터 소스에 존재하는 두 객체들 간에 공간관련성이 존재한다면, 한 객체의 변경에 의해 다른 데이터 소스에 저장된 객체와 비일관적인 공간관련성이 생성될 수 있기 때문이다.



(그림 1) 미들웨어를 이용한 분산 공간데이터 변경

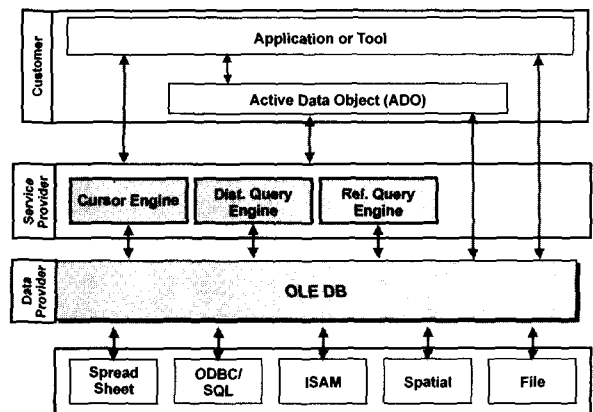
(그림 1)은 이 논문에서 다루고자 하는 대상 환경을 보여준다. 다수의 공간데이터 서버별로 주제도와 공통된 기본도를 저장하고 있으며, 미들웨어는 데이터 제공자와 서비스 제공자로 구성된다. 클라이언트는 데이터 제공자의 인터페이스

를 이용하여 분산되어 있는 공간데이터에 대한 발견 및 접근 작업을 수행하고 서비스 제공자의 인터페이스를 이용하여 공간데이터의 변경 작업을 수행한다. 데이터 제공자는 OpenGIS 구현명세에 따른 OGIS OLE DB 데이터 제공자를 이용하며, 서비스 제공자는 OLE DB 데이터 제공자를 기반으로 공간데이터 변경을 위한 인터페이스를 제공한다.

이 논문에서는 (그림 1)과 같이 이질적인 다수의 공간데이터 서버에 분산되어 있는 공간객체를 OGIS OLE DB를 이용하여 변경하기 위한 서비스 제공자의 공간데이터 변경 인터페이스와 동시 변경된 공간객체의 일관성을 유지하기 위한 공간관련성 유지 인터페이스를 제안한다. 제안한 서비스 제공자의 인터페이스는 이질적인 공간데이터 소스에 대한 상호운용성을 보장하기 위하여 OGIS OLE DB 데이터 제공자의 인터페이스만을 이용하며, 공간관련성에 의한 데이터 종속성을 알아내기 위하여 분산 객체들 간의 공간관련성을 검사한다. 그리고 동시 변경된 공간객체들 간의 데이터 일관성을 유지하기 위하여 트랜잭션의 수행 중에 변경된 공간객체를 관련 서비스 제공자에게 전송하는 서비스 제공자 간섭을 이용한 2단계 완료 프로토콜을 제시하고, 제시한 프로토콜을 사용하는 서비스 제공자 기반의 공간데이터 변경 프로토타입 시스템을 설계하고 구현한다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서 OGIS OLE DB 데이터 제공자에 대하여 기술한다. 4장에서 기존 변경 인터페이스의 사용시 발생하는 문제점을 정의하고 이를 해결하기 위한 분산 공간관련성의 정의와 서비스 제공자 기반의 트랜잭션 수행 방법을 제시한다. 5장에서 분산 공간관련성 기반의 2단계 완료 프로토콜에 대하여 기술하고 추가된 변경 인터페이스와 메타정보 테이블을 제시한다. 6장에서 서비스 제공자 기반의 공간데이터 변경 프로토타입 시스템의 설계 및 실험 내용을 제시하고 7장에서 결론을 기술한다.

## 2. 관련 연구



(그림 2) OGIS OLE DB 구조

(그림 2)는 OGC 표준안에서 기본 골격으로 사용하는 OG

IS OLE DB 구조로, 크게 데이터 제공자와 서비스 제공자인 두 가지 계층 구조로 구성되어 있다[3-5]. 데이터 제공자는 데이터 소스의 종류에 관계없이 데이터에 접근할 수 있는 인터페이스를 제공하며, 세부적으로 데이터베이스시스템을 포함하여 다양한 형태로 구축되어 있는 문서, e-mail 시스템, COM 컴포넌트, 스프레드 시트 등 모든 종류의 데이터 소스들에 대해 동일한 인터페이스를 통하여 접근이 가능하도록 지원해준다. 서비스 제공자는 중간자로 위치하며 데이터 제공자의 인터페이스를 사용하여 데이터 제공자가 제공해 주지 못하는 기능을 보충해 주는 역할을 하며, 일반적으로 질의 처리 기능, 커서 기능, 분산 처리 기능 등을 수행한다.

기존의 OLE DB는 일반데이터를 위한 접근 및 검색 인터페이스와 함께 행집합(rowset)을 이용한 변경 인터페이스를 정의하고 있기 때문에 이를 기반으로 확장된 OGIS OLE DB도 역시 일반데이터를 위한 변경 인터페이스를 제공한다. 그러나 동시 변경된 분산 공간데이터를 대상으로 공간관련성에 의한 데이터 비일관성을 검사하기 위한 인터페이스는 정의하고 있지 않다. 즉, 공간관련성을 가진 분산 공간객체들의 변경에 대한 인터페이스는 정의하지 않았다. 따라서 만약 일반데이터를 위한 변경 인터페이스를 이용하여 공간데이터를 동시 변경하면 공간 객체간에 발생하는 공간관련성에 의한 데이터 일관성을 유지할 수 없는 문제가 발생한다.

분산 환경에서 데이터 변경을 지원하기 위한 연구로는 분산 트랜잭션 모델[6-8]이 있다. 분산 트랜잭션은 크게 지역 트랜잭션과 전역 트랜잭션으로 구성된다. 지역 트랜잭션은 트랜잭션이 초기화된 데이터 서버에 위치한 데이터를 접근하며 각각의 데이터 서버에 의해 독립적으로 제어된다. 전역 트랜잭션은 트랜잭션이 초기화된 사이트와 다른 한 사이트 또는 다수의 다른 사이트들에 저장되어 있는 데이터에 접근하며 각각의 데이터 서버가 아닌 전역 트랜잭션 매니저에 의해 제어된다. 분산 트랜잭션 모델에서 전역 트랜잭션의 동시 수행시 전역 순서화(global serializability)를 이용하여 데이터 일관성을 보장하고 있으나 공간관련성에 따른 데이터 일관성을 유지하지 못하는 문제가 있다.

완전 중복된 데이터 서버간에 공간데이터 변경 트랜잭션을 위한 확장 쓰기 잠금 기법과 협동 작업을 기반으로 한 변경 프로토콜이 [1,2]에서 제시되었다. 확장 쓰기 잠금 기법은 공간객체들 간의 공간관련성을 이용하여 변경하고자 하는 객체와 연관된 다른 객체들에게도 쓰기 잠금을 설정한다. 그리고 변경 사용자간의 협동 작업을 통하여 변경한 객체의 정확성을 보장한다. 그러나 데이터 서버간에 객체가 완전히 중복되어 있지 않고 일부만 중복되어 있을 때 데이터 서버는 모든 계층의 공간 객체를 가지고 있지 않다. 만약 변경하고자 하는 객체에 확장 쓰기 잠금을 설정하면 데이터 서버에서 모든 객체에 대하여 쓰기 잠금 간의 변경 충돌을 검사할 수 없는 문제가 있다. 따라서 동일한 공간데이터가 완전 중복되어 저장된 데이터 서버들을 대상으로

한 [1,2]의 변경 프로토콜은 서로 다른 데이터 서버에 분산된 서로 다른 공간 객체를 변경하기 위한 트랜잭션 수행시 적합하지 않다.

[9]에서 Egenhofer는 두 개의 공간객체간에 기하로 인해 발생하는 관련성이 있다고 정의하고 크게 disjoint, meets, equals, inside1, inside2, cover1, cover2, overlaps의 8가지로 분류하였다. 객체 A의 기하는 내부(interior) 프리디킷 IA, 외부(exterior) 프리디킷 A 그리고 경계(boundary) 프리디킷 A으로 표현되며 8가지의 공간 관련성은 이들 프리디킷간의 9-교집합에 의하여 정의된다. 예를 들어 두 공간 객체 A와 B간의 meet 공간 관련성은 <표 1>과 같이 정의된다.

<표 1> Meet 공간관련성

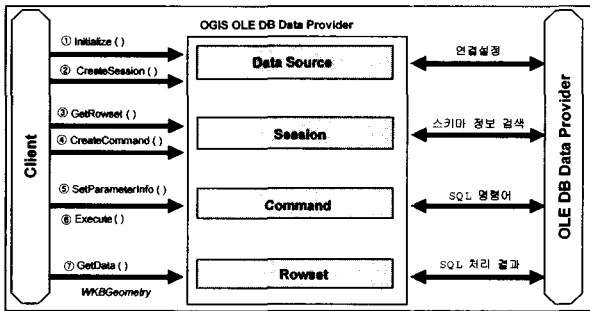
	iB	∂B	-B
iA	∅	∅	∅
∂A	∅	≠ ∅	∅
-A	∅	∅	∅

### 3. OGIS OLE DB 데이터 제공자

OGIS OLE DB 데이터 제공자는 공간데이터를 지원하기 위해 스키마 정보와 기하 정보에 대한 인터페이스를 제공한다[3]. OGIS OLE DB 데이터 제공자가 기존의 OLE DB 데이터 제공자와 다른 점은 기존의 인터페이스 외에 추가적으로 공간데이터 접근을 위한 인터페이스를 지원한다는 것이다. 데이터 소비자가 OGIS OLE DB의 공간데이터를 위한 인터페이스를 이용하지 않고 기존의 OLE DB 인터페이스를 이용하여 공간데이터에 접근하는 것은 가능하다. 그러나 이 방법에서는 데이터 소비자가 공간데이터를 활용하기 위해 필요한 메타 데이터, 공간참조 정보, 공간연산자 등을 사용할 수 없으며, 필요한 공간데이터에 올바르게 접근할 수 없는 문제점이 있다. 따라서 공간데이터를 사용하기 위한 데이터 소비자는 OpenGIS 표준 인터페이스를 지원하는 OGIS OLE DB 인터페이스를 사용하는 것이 필요하다.

OGIS OLE DB 데이터 제공자는 크게 데이터 소스(data source) 객체, 세션(session) 객체, 명령어(command) 객체, 행집합(rowset) 객체로 구성되어 있다. 각각의 객체에서 제공하는 기능은 다음과 같이 3가지로 분류할 수 있다. 첫째 기능은 데이터 소스와의 연결(Connection)을 설정하는 것이다. OGIS OLE DB의 연결 기능은 클라이언트가 데이터 소스에 대해 연결 설정을 수행하게 한다. 데이터 소스 객체와 세션 객체는 각각 데이터 소스에 대하여 연결하는 역할과 데이터 소스에 대한 세션을 여는 역할을 한다. 두 번째는 연결된 데이터 소스에서 수행할 SQL 명령을 설정하는 것이다. 이를 위해 명령어 객체는 연결된 데이터 소스에서 사용할 수 있는 질의 처리 서비스를 캡슐화하고 질의 형식, 질의 파라미터 설정, 질의 실행 등을 수행하는 인터

페이스들을 제공한다. 마지막으로 세 번째는 질의 수행으로 얻어진 데이터들을 테이블 형태로 저장하여 제공하는 것이다. 행집합 객체는 세션 및 명령어 객체를 통하여 생성되고 행집합 객체 내에 저장된 공간데이터들에 대한 접근 인터페이스들을 제공한다.



(그림 3) OGIS OLE DB 데이터 제공자를 통한 데이터 접근

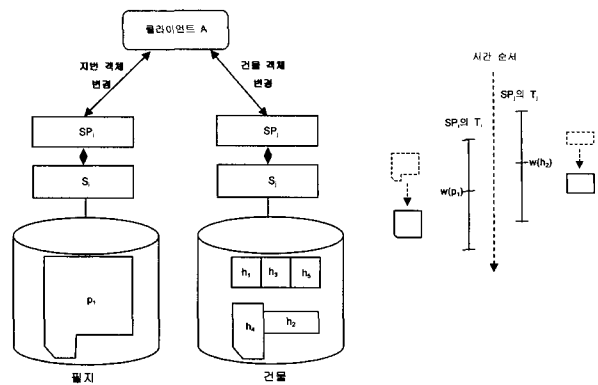
(그림 3)은 클라이언트가 OGIS OLE DB 데이터 제공자 인터페이스를 이용하여 공간데이터에 접근하는 과정을 보여준다. OGIS OLE DB 데이터 제공자 인터페이스를 이용한 공간데이터에 대한 접근 과정은 다음과 같다. 먼저, 클라이언트가 데이터 제공자의 GUID를 이용하여 데이터 소스 객체를 생성하고 초기화하는 단계로, 여기서는 데이터 소스 객체 인터페이스를 이용하여 세션 객체를 생성한다. 두 번째 단계에서는 명령어 객체를 생성하고 세션 객체의 GetRowset 인터페이스를 이용하여 스키마 정보를 읽어서 공간데이터에 접근하기 위한 질의 파라미터를 설정한다. 세 번째 단계에서는 SQL 질의를 명령어 객체의 Execute 인터페이스를 이용하여 수행한 후에 행집합 객체를 생성한다. 네 번째 단계에서는 클라이언트가 질의 수행 결과로 생성된 행집합 객체로부터 접근자(accessor)를 이용하여 필요한 공간데이터를 읽는다. 따라서 클라이언트는 데이터 소스의 다양한 데이터 타입과는 상관없이 행집합의 일관된 인터페이스를 통하여 공간데이터를 사용할 수 있다.

4. 서비스 제공자 기반의 트랜잭션 수행

이 장에서는 서비스 제공자를 기반으로 공간데이터를 변경하기 위한 트랜잭션 수행에 대하여 기술한다.

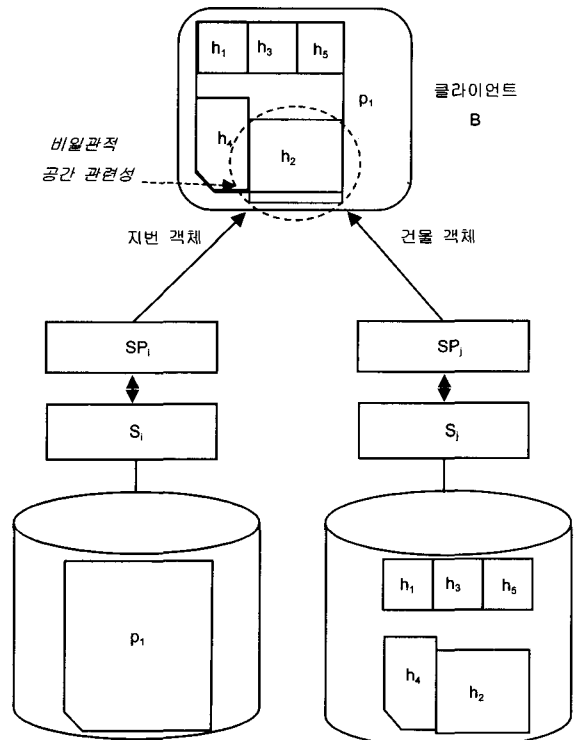
4.1 서비스 제공자에 의한 동시 변경

클라이언트는 기존의 OGIS OLE DB 서비스 제공자에서 제공하는 인터페이스를 이용하여 이질적인 데이터 소스들에 존재하는 공간객체에 접근할 수 있으며, 각 객체들을 동시에 변경할 수 있다. (그림 4)는 서로 다른 서비스 제공자들에 의한 공간객체의 동시 변경 예를 보여준다. 클라이언트 A는 서비스 제공자 SP<sub>1</sub>의 인터페이스를 이용하여 데이터 서버 S<sub>1</sub>의 지번 객체 p<sub>1</sub>을 변경하고, 서비스 제공자 SP<sub>2</sub>



(그림 4) 서로 다른 서비스 제공자들에 의한 공간객체 동시 변경의 예

의 인터페이스를 이용하여 동일 위치에 있는 데이터 서버 S<sub>2</sub>의 건물 객체 h<sub>2</sub>를 변경한다. 여기서 p<sub>1</sub>과 h<sub>2</sub>는 서로 다른 데이터 서버에 존재하고 다른 공간객체이기 때문에 p<sub>1</sub>과 h<sub>2</sub>에 설정된 잠금은 충돌하지 않는다. 따라서 기존의 잠금 기법을 사용할 때 동시 변경이 가능하다. 여기서 동시 변경된 p<sub>1</sub>과 h<sub>2</sub>간에 Inside 공간관련성이 존재한다면, p<sub>1</sub>과 h<sub>2</sub>는 비록 다른 데이터 서버에 저장되어 있지만 동시 변경되면 (그림 5)와 같이 비 정상적인 공간관련성이 생성될 수 있다. 그러나 기존의 잠금 기법은 분산된 공간객체에 대한 각 서비스 제공자들의 동시 변경을 인해 발생하는 데이터 비일관성을 방지할 수 없다.



(그림 5) 서비스 제공자들의 동시 변경으로 인해 생성된 비일관적인 공간관련성의 예

4.2 분산 공간관련성

Engenhofer는 공간객체들 간의 공간관련성을 크게 disjoint, meets, equals, inside1, inside2, cover1, cover2, overlaps의 8가지로 정의하였다[9]. 이 정의에 기반하여 OpenGIS에서 공간관련성이 다음의 특성을 가진다고 정의하였다[3].

- 서로 배타적(mutual exclusive)이다.
- 같거나 다른 차원의 두 기하에 대해 공간관련성을 적용할 수 있다.
- 각 공간관련성은 대응하는 차원 확장 9-교집합 행렬 패턴으로 표현할 수 있다.
- 차원 확장 9-교집합 행렬은 커브에 대해 시작점, 종료점에 대한 메소드와 기하의 경계에 대한 메소드가 주어진다면 공간관련성을 통해 불린(boolean) 표기법으로 표현될 수 있다.

서로 다른 데이터 서버에 속하는 공간객체들 간에도 공간관련성이 존재하며, 이것을 이 논문에서 분산 공간관련성이라 한다. 이 관련성에 대한 정의는 다음과 같으며, 여기서  $\alpha_i.G$ 를 공간객체  $\alpha_i$ 의 기하 영역이라 하고  $DS(S_i)$ 를 데이터 서버  $S_i$ 의 데이터 집합이라 한다.

정의 1 :  $\alpha_i \in DS(S_i), \alpha_j \in DS(S_j)$ 인 두 객체  $\alpha_i, \alpha_j$ 가 있을 때  $\alpha_i$ 와  $\alpha_j$ 간의 공간관련성을 분산 공간관련성이라 한다.

정의 2 :  $\alpha_i \in DS(S_i), \alpha_j \in DS(S_j)$ 인 두 객체  $\alpha_i, \alpha_j$ 가 있을 때 만약  $\alpha_i.G \cap \alpha_j.G \neq \emptyset$ 이면  $\alpha_i$ 와  $\alpha_j$ 간에는 Disjoint를 제외한 하나의 분산 공간관련성이 존재한다.

만약  $\alpha_i$ 와  $\alpha_j$ 가 각각 서브트랜잭션  $T_i$ 와  $T_j$ 에 의하여 동시 변경된다면  $\alpha_i$ 와  $\alpha_j$ 간의 공간관련성이 달라질 수 있다. 따라서 만약  $\alpha_i$ 와  $\alpha_j$ 가 분산 공간관련성을 가지면  $\alpha_i$ 와  $\alpha_j$ 가 서로 다른 데이터 서버에 속한 객체이더라도  $\alpha_i$ 와  $\alpha_j$ 간의 공간관련성을 검사할 필요가 있다. 그리고 만약  $\alpha_i$ 와  $\alpha_j$ 가 disjoint 이외에 다른 공간관련성을 가지고 있다면  $\alpha_i$ 와  $\alpha_j$ 는 동시 변경될 수 없다.

정의 3 :  $\alpha_i \in DS(S_i), \alpha_j \in DS(S_j)$ 인 두 객체  $\alpha_i, \alpha_j$ 가 있을 때 만약  $T_i$ 가  $\alpha_i$ 를 그리고  $T_j$ 가  $\alpha_j$ 를 동시 변경하고  $\alpha_i.G \cap \alpha_j.G \neq \emptyset$ 이면  $\alpha_i$ 와  $\alpha_j$ 의 변경 내용은 충돌한다.

이 논문에서 공간객체들 간의 분산 공간관련성을 검사하기 위하여 서비스 제공자에서 분산 공간관련성 검사 인터페이스를 <표 2>과 같이 추가한다. 공간객체의 분산 공간관련성을 검사하기 위해서 먼저 서브트랜잭션의 서비스 제공자와 관련된 다른 서비스 제공자를 파악할 필요가 있다. ISpatialObjectUpdateService :: GetParticipantInfo 인터페이스는 분산 공간관련성을 검사하기 위하여 관련된 서비스 제공자의 부가 정보를 검색한다. 그리고 ISpatialObjectUpdateService :: CheckSpatialRelationShip 인터페이스는 하나의 서비스 제공자 내에서 두 공간객체 간의 공간관련성을 검사하고 동시 변경 여부를 판단한다.

<표 2> 분산 공간관련성 검사를 위한 인터페이스 정의

메소드 이름	설 명
ISpatialObjectUpdateService :: CheckSpatialRelationShip	공간관련성 정보에 따라 변경 가능성 여부를 검사
ISpatialObjectUpdateService :: GetParticipantInfo	참여서비스 제공자의 메타 정보를 읽음

4.3 서비스 제공자 간섭

분산 공간관련성은 서로 다른 데이터 서버들에 있는 공간객체들 간에 존재하는 관련성이므로 데이터 서버  $S_i$ 에서 수행되는 서브트랜잭션  $T_i$ 는 변경한 공간객체  $\alpha_i$ 의 정확성 여부를 확인하기 위해서는, 다른 데이터 서버  $S_j$ 의 서브트랜잭션  $T_j$ 가 동시 변경중인 객체  $\alpha_j$ 와의 공간관련성을 검사해야 한다. 이를 위해서  $T_i$ 는  $\alpha_i$ 의 변경을 마쳤을 때  $\alpha_j$ 가 위치한  $S_j$ 의 서비스 제공자에게 변경된 객체  $\alpha_i$ 에 대한 분산 공간관련성 검사를 의뢰한다.  $S_j$ 의 서비스 제공자는 자신의 서브트랜잭션인  $T_j$ 가 변경중인  $\alpha_j$ 에 대하여  $\alpha_i$ 와의 분산 공간관련성을 검사하고 만약  $\alpha_i.G \cap \alpha_j.G = \emptyset$ 이면  $\alpha_i$ 의 변경에 대하여 동의한다. 그러나 만약 정의 3에 따라  $\alpha_i$ 와  $\alpha_j$ 가 충돌하면 일관성 유지를 위하여  $T_j$ 의 서비스 제공자는  $\alpha_j$ 의 변경을 거부하고  $T_i$ 는  $\alpha_i$ 의 변경 내용을 취소한다.

$S_i$ 의 서브트랜잭션  $T_i$ 가 변경한  $\alpha_i$ 의 완료를 위하여  $S_i$ 의 서비스 제공자가  $S_i$ 와 관련된 데이터 서버인  $S_j$ 의 서비스 제공자에게 변경 동의를 요청할 때  $S_i$ 의 서비스 제공자를 조정서비스 제공자 CS라 하고  $S_j$ 의 서비스 제공자를 참여서비스 제공자 PS라 한다.

정의 4 : 조정서비스 제공자(Coordinating Service Provider : CS)

데이터 서버  $S_i$ 의 서브트랜잭션  $T_i$ 가 공간객체  $\alpha_i$ 를 변경한 후에  $\alpha_i$ 의 변경 내용을 완료하고자 할 때  $S_i$ 의 서비스 제공자는 조정서비스 제공자가 되며  $S_i$ 에서  $\alpha_i$ 의 완료를 책임진다. 그리고 참여서비스 제공자의 동의에 따라  $\alpha_i$ 의 완료 여부를 결정한다.

정의 5 : 참여서비스 제공자(Participating Service Provider : PS)

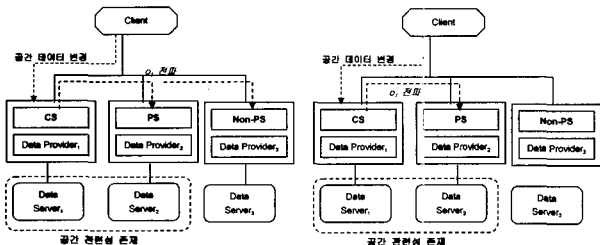
데이터 서버  $S_i$ 의 공간객체  $\alpha_i$ 를 변경하는 서브트랜잭션  $T_i$ 가 있다고 가정하자.  $\alpha_i$ 의 완료를 위하여  $S_i$ 의 서비스 제공자가  $S_j$ 의 서비스 제공자에게  $\alpha_i$ 의 변경에 대한 동의를 요청할 때  $S_j$ 의 서비스 제공자는  $\alpha_i$ 를 위한 참여서비스 제공자이다. 참여서비스 제공자는  $S_i$ 와 다른 데이터 서버  $S_j$ 에 위치하며  $\alpha_i$ 와 동시에 변경된 객체  $\alpha_j$ 와의 공간관련성 검사를 책임진다. 그리고 검사 결과에 따라  $\alpha_i$ 의 변경에 대한 동의 여부를 결정한다.

변경 객체의 공간관련성에 따른 데이터 일관성을 유지하기 위하여 조정서비스 제공자가 참여서비스 제공자들에게 변경에 대한 동의를 요청하는 작업을 서비스 제공자간 간섭이라 정의한다.

정의 6 : 서비스 제공자간 간섭

조정서비스 제공자 CS가 분산 공간관련성에 따른 공간데이터 일관성을 유지하기 위하여 CS의 서브트랜잭션  $T_i$ 에 의해 변경된 객체  $o_i$ 를  $k$ 개의 참여서비스 제공자들인  $PS_1 \sim PS_k$ 로 전파하고 각 참여서비스 제공자의 동의를 획득하는 과정이다.

변경 객체의 공간관련성 검사를 위해 참여서비스 제공자들에게 변경 객체를 전파하는 방법은 크게 두 가지가 있다. 첫 번째는 (그림 6)(a)와 같이 시스템의 모든 서비스 제공자에게 변경 객체를 전파하는 방송 전파(broadcast propagation) 방법이다. 이 방법에서는 참여서비스 제공자는 전파된 객체인  $o_i$ 에 대하여 공간관련성 검사를 수행한 후에 동의 여부를 조정서비스 제공자에게 알리지만, 비 참여서비스 제공자는 공간관련성이 없음을 알린다. 그러나 이 방법에서는 모든 서비스 제공자에게 변경 객체를 보내주기 때문에 네트워크 부하를 증가시키는 문제점이 발생할 수 있다. 두 번째는 (그림 6)(b)와 같이 실제 참여서비스 제공자가 되는 서비스 제공자들만 선별하여 변경 객체를 전파하는 선별 전파(selective propagation) 방법이다. 선별 전파 방식은 전파하기 전에 참여서비스 제공자를 선별해야 하는 부하가 있지만 네트워크 부하를 줄이면서 효과적으로 참여서비스 제공자의 동의를 획득할 수 있는 장점이 있다.



(a) 방송 전파 방법 (b) 선별 전파 방법  
(그림 6) 변경 전파 방식

<표 3> 영역 잠금 지원 인터페이스

메소드 이름	설 명
IlockService :: SetRegionLock	사용자가 변경할 영역에 대해 영역 잠금을 설정
IlockService :: ReleaseRegionLock	영역 잠금을 해제

조정서비스 제공자가 참여서비스 제공자를 선별하기 위하여 [1, 2]에서 제시된 영역 잠금 기법을 활용한다. 영역 잠금은 weak SIX 속성의 잠금으로 트랜잭션이 영역 안에 포함되는 공간객체들을 변경하고자 하는 의도를 나타낸다. 영역 잠금은 트랜잭션 시작 시 해당 트랜잭션에 의해 설정된다. 만약  $o_i$ 를 변경하고 있는 트랜잭션  $T_i$ 의 영역 잠금과  $T_j$ 의 영역 잠금이 겹친다면  $T_j$ 가 변경 중인 객체  $o_j$ 가 위치하는 서비스 제공자는 참여서비스 제공자가 된다. <표 3>은 서브 트랜잭션이 영역 잠금을 설정하기 위하여 서비스 제

공자에 추가한 인터페이스를 보여준다.

4.4 서비스 제공자 기반의 변경 트랜잭션

서비스 제공자를 기반으로 한 공간데이터 변경 트랜잭션은 서비스 제공자에 정의된 인터페이스를 이용하여 (그림 7)의 순서대로 수행된다. 클라이언트는 먼저 OGIS 데이터 제공자에 정의되어 있는 인터페이스를 이용하여 변경하고자 하는 영역에 위치하는 공간객체들의 행집합들을 얻는다. 그리고 서비스 제공자에 정의된 ITransaction Service의 인터페이스를 이용하여 트랜잭션  $T_i$ 를 시작한다.  $T_i$ 를 시작할 때 ILock Service에 정의되어 있는 SetRegionLock 인터페이스를 이용하여 변경하고자 하는 영역에 대한 영역 잠금을 설정한다.

```

Transaction Ti
BEGIN
    ItransactionService :: startTransaction()
    IlockService :: setRegionLock()
    . . . . .
    While(oi for updating)
    {
        IlockService :: setExclusiveLock(oi)
        Update oi
        ItransactionService :: midCommit(oi)
    }
    . . . . .
    ItransactionService :: commit()
    IlockService :: releaseExclusiveLock()
    IlockService :: ReleaseRegionLock()
END
    
```

(그림 7) 서비스 제공자를 이용한 트랜잭션 수행

영역 잠금이 설정된 후에 변경하고자 하는 공간객체  $o_i$ 가 결정되면 ILock Service를 이용하여 쓰기 잠금을 설정하고  $o_i$ 를 변경한다.  $o_i$ 의 변경 작업이 끝나면  $T_i$ 는 서비스 제공자간 간섭 작업을 수행하기 위하여 ITransaction Service의 mid-commit 인터페이스를 이용하여 각 참여서비스 제공자에게  $o_i$ 의 변경 내용에 대한 동의를 요청한다. 변경하고자 하는 모든 객체  $o_i$ 에 대한 변경 작업이 완료되었을 때  $T_i$ 는 ITransaction Service의 commit 인터페이스를 호출하여 완료되고  $T_i$ 가 설정한 영역 잠금을 해제한다.

5. 완료 프로토콜

5.1 분산 공간관련성 기반의 2단계 완료 프로토콜

분산 환경에서 조정서비스 제공자는 동시 변경한 공간객체  $o_i$ 의 데이터 일관성을 유지하기 위하여  $o_i$ 의 변경을 마칠 때 서비스 제공자간 간섭 작업을 수행한다. 이를 위하여 이 논문에서는 기존의 2단계 완료 프로토콜을 기반으로 한 분산 공간관련성 기반의 완료 프로토콜을 제시한다. 조정서비스 제공자는 객체  $o_i$ 를 변경하는 트랜잭션  $T_i$ 와 관련된 모든 참여서비스 제공자들과 잠금 정보를 교환하고 분산 공간관련성 기반의 2단계 완료 프로토콜에 따라  $o_i$ 의 변경 내용을 완료한다.

데이터 서버  $S_i$ 의 서비스 제공자의 서브 트랜잭션  $T_i$ 가 공간객체  $o_i$ 를 변경하면  $S_i$ 의 서비스 제공자는  $o_i$ 를 위한 조정서비스 제공자 CS가 된다. 2단계 완료 프로토콜의 첫 번째 단계는 변경 동의를 필요한 참여서비스 제공자를 결정하고 객체  $o_i$ 의 변경 내용을 전파하는 단계이다. 참여서비스 제공자를 결정하기 위하여  $T_i$ 의 시작 시 설정된 영역 잠금을 이용한다. 만약  $T_i$ 의 영역 잠금과 데이터 서버  $S_j$ 에 설정되어 있는  $T_j$ 의 영역 잠금이 disjoint이외에 다른 공간관련성이 있다면  $S_j$ 의 서비스 제공자는  $o_i$ 를 위한 참여서비스 제공자  $PS_j$ 가 된다. 모든  $PS_j$ 가 결정되면  $o_i$ 의 변경 내용을  $PS_j$ 에게 전파하고 모든  $PS_j$ 로부터 응답이 올 때까지 대기한다. 반면에 각  $PS_j$ 는 CS로부터  $o_i$ 의 변경 정보를 전파받으면 현재  $PS_j$ 를 통하여  $T_j$ 가 변경하고 있는  $o_j$ 와의 공간관련성을 검사한다. 만약  $o_i$ 와  $o_j$ 간의 공간관련성이 disjoint이면  $PS_j$ 는  $o_i$ 의 변경을 승인하고 CS에게 변경 동의 메시지를 전송한다. 그러나 만약  $o_i$ 와  $o_j$ 가 disjoint이외의 공간관련성을 가지면  $PS_j$ 는 변경 거부 메시지를 전송한다.

두 번째 단계는 각  $PS_j$ 의 응답에 따라  $o_i$ 의 변경 완료 여부를 결정하는 단계이다. 만약 모든  $PS_j$ 가 변경 동의 메시지를 전송하면  $o_i$ 의 변경을 승인한 것이기 때문에 CS는 각  $PS_j$ 에게  $o_i$ 의 변경이 완료되었음을 알리고  $o_i$ 의 변경 작업을 완료한다.  $o_i$ 가 중복되어 저장되어 있는  $PS_j$ 는 전파된  $o_i$ 의 변경 완료 메시지에 따라  $o_i$ 의 변경 내용을 저장한다. 그러나 하나 이상의  $PS_j$ 가 변경 거부 메시지를 전송하면  $o_i$ 의 변경 내용은 취소된다.

```

RegionLock( $T_i$ ).G : the spatial extent of  $T_i$ s Region Lock
Operation midCommit(  $T_i$ ,  $o_i$  )
BEGIN
RegionLock( $T_i$ ) ← read( $T_i$ )
FOR all  $T_j$  where  $i \neq j$ 
  BEGIN
    PS( $T_j$ ), RegionLock( $T_j$ ) ← read( $T_j$ )
    IF RegionLock( $T_i$ ).G ∩ RegionLock( $T_j$ ).G ≠ ∅
      THEN add PS( $T_j$ ) to PS_list( $T_i$ )
    END FOR

IF( PS_list = ∅ )
THEN BEGIN
  Execute globalCommit( $o_i$ )
  Exit midCommit
END IF

FOR all PS( $T_j$ ) in PS_list( $T_i$ ) /* Phase 1 */
  BEGIN
    Propagate  $o_i$  to PS( $T_j$ )
    Wait for reply( $T_j$ ) from PS( $T_j$ )
    Add reply( $T_j$ ) to reply_list( $T_i$ )
  END FOR

FOR all reply( $T_j$ ) in reply_list( $T_i$ ) /* Phase 2 */
  IF ( reply = midReject )
  THEN BEGIN
    Execute globalAbort( $o_i$ )
    Exit midCommit
  END IF
Execute globalCommit( $o_i$ )
End
    
```

(그림 8) midCommit 인터페이스 알고리즘

분산 공간관련성 기반의 2단계 완료 프로토콜은 ITransaction Service에 정의된 midCommit 인터페이스에 의해 시작된다. midCommit 인터페이스의 알고리즘은 (그림 8)과 같다. 먼저 CS는 영역 잠금을 이용하여  $PS_j$ 를 결정한다. 모든  $PS_j$ 가 결정되면  $T_i$ 가 변경한 객체  $o_i$ 의 정보를 각  $PS_j$ 에게 전파하고  $PS_j$ 의 응답을 기다린다. 만약 하나 이상의  $PS_j$ 로부터 midReject를 전송받으면  $o_i$ 의 변경 내용을 취소하기 위하여 globalAbort를 수행한다. globalAbort는 모든  $PS_j$ 에게  $o_i$ 의 변경 내용이 취소되었음을 알린다. 그러나 만약 모든  $PS_j$ 로부터 midAccept를 전송받으면  $o_i$ 의 변경 내용을 완료하고 globalCommit을 수행한다. globalCommit은 모든  $PS_j$ 에게  $o_i$ 의 변경 내용이 완료되었음을 알린다.

<표 4>는 OGIS OLE DB를 기반으로 분산된 공간데이터를 변경하기 위하여 서비스 제공자에 추가적으로 정의된 인터페이스를 보여준다. 변경 트랜잭션은 <표 4>에 정의된 인터페이스를 이용하여 트랜잭션 연산을 수행한다.

<표 4> 분산 공간데이터 변경 트랜잭션을 위한 서비스 제공자 인터페이스

ITransactionService Interface	IProtocolService Interface
HRESULT startTransaction()	HRESULT informTransaction()
HRESULT endTransaction()	HRESULT requestRegionLock()
HRESULT Commit()	HRESULT requestExclusiveLock()
HRESULT Abort()	HRESULT requestMidCommit()
HRESULT Rollback()	HRESULT requestRegionLockRelease()
HRESULT acceptMidCommit()	HRESULT getBlockedList()
HRESULT rejectMidCommit()	HRESULT getNDJ()
HRESULT rollbackMidCommit()	HRESULT getDeltaInfo
HRESULT MidCommit()	HRESULT getObjectInfo
HRESULT MidRollback()	HRESULT acceptMidCommit()
	HRESULT rejectMidCommit()
ILockService Interface	ISpatialDataUpdateService Interface
HRESULT SetRegionLock()	
HRESULT setExclusiveLock()	
HRESULT setSharedLock()	HRESULT setSpatialData()
HRESULT releaseRegionLock()	HRESULT acceptSpatialObject()
H R E S U L T	HRESULT rejectSpatialObject()
releaseExclusiveLock()	HRESULT checkSpatialRelationship()
HRESULT releaseSharedLock()	
HRESULT waitRegionLock()	

5.2 메타 정보 테이블

분산 공간관련성 기반의 2단계 완료 프로토콜을 수행하기 위해서는 OGIS OLE DB 데이터 제공자가 제공하는 메타 데이터 외에 추가적으로 메타 데이터가 필요하다. 프로토콜 수행을 위한 메타 데이터는 서비스 제공자 정보 테이블(SP\_Info)과 영역 잠금 정보 테이블(RegionLock\_Info)인 두 가지 테이블로 구성된다.

조정서비스 제공자는 완료 프로토콜을 시작하기 위하여 먼저 참여서비스 제공자를 결정할 필요가 있다. 만약  $T_i$ 가 수행중인 조정서비스 제공자 CS에 정의된 영역 잠금과  $T_j$ 의 영역 잠금이 중첩되면  $T_j$ 가 수행중인  $PS_j$ 는 CS의 참여서비스 제공자가 된다. 서브트랜잭션들의 영역 잠금 간 중첩 검사를 위하여 CS는 <표 5>에 정의된 영역 잠금 정보

테이블을 이용한다. 영역 잠금 정보 테이블은 서브 트랜잭션 식별자, 영역 좌표, 잠금의 상태 정보로 구성되고 서브 트랜잭션의 시작 시 관련 레코드가 생성되며 서브 트랜잭션 완료시 제거된다. 영역 잠금 간 중첩 검사는 T<sub>i</sub>가 첫 번째 공간객체 O<sub>n1</sub>을 변경하고 O<sub>n1</sub>을 위한 변경 완료 프로토콜을 시작할 때 수행된다.

〈표 5〉 영역 잠금 정보 테이블(RegionLock\_Info)

필드명	데이터 타입	설명	
TID	VARCHAR2(40)	트랜잭션 식별자	
Ltx	NUMBER(10)	영역 좌표	
Lty	NUMBER(10)		
Rtx	NUMBER(10)		
Rty	NUMBER(10)		
STATUS	NUMBER(1)	영역 잠금 상태	
		0	잠금
		1	대기
		2	해제

〈표 6〉 서비스 제공자 정보 테이블(SP\_Info)

필드명	데이터 타입	설명	
TID	VARCHAR2(40)	트랜잭션 식별자	
SName	VARCHAR2(15)	서버명	
STATUS	NUMBER(1)	서비스 제공자 상태	
		0	미 참여서비스 제공자
		1	조정서비스 제공자
		2	참여서비스 제공자

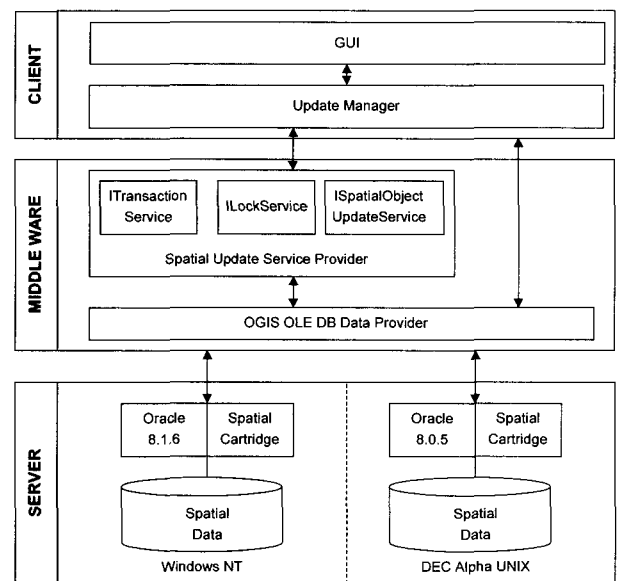
서브트랜잭션 T<sub>i</sub>가 첫 번째 객체 O<sub>n1</sub>에 대한 완료 프로토콜을 수행한 이후에 T<sub>i</sub>가 수행중인 CS를 위한 참여서비스 제공자들은 동일하기 때문에 그 이후의 변경 객체에 대한 완료 프로토콜을 수행하기 위하여 영역 잠금 간 중첩 검사를 반복할 필요가 없다. T<sub>i</sub>가 변경한 두 번째 이후의 객체에 대한 완료 프로토콜 수행시 <표 6>의 서비스 제공자 테이블을 이용한다. 서비스 제공자 테이블은 트랜잭션 식별자, 서버 이름, 서비스 제공자의 상태 정보로 구성된다. 서비스 제공자의 상태 정보는 미 참여서비스 제공자 상태, 조정서비스 제공자 상태 그리고 참여서비스 제공자 상태의 값을 가진다.

6. 구현 및 실험

6.1 시스템 구조

(그림 9)는 서비스 제공자 기반의 공간데이터 변경 프로토타입 시스템의 구조를 보여준다. 프로토타입 시스템은 Windows NT 플랫폼 환경에서 Visual C++ 6.0과 ATL을 사용하여 구현하였다. 그리고 기존의 OLE DB 데이터 제공자의 인터페이스를 이용하여 OGIS OLE DB 데이터 제공자와 공간데이터 변경을 위한 서비스 제공자를 DLL의 형태로 구현하였다. 실험을 위한 공간데이터 서버는 Windows NT 플랫폼을 이용하는 Oracle 8.1.6 데이터 서버와 DEC ALPHA UNIX 플랫폼을 이용하는 Oracle 8.0.5 데이터 서버를 사용하였다. 각 데이터 서버는 공간데이터 지원을 위하여 공간

카트리지(Spatial Cartridge) 패키지를 사용한다. 클라이언트는 Windows NT 환경에서 운용되며 공간객체를 변경하기 위하여 사용자 대화식 방법을 이용하도록 구현하였다. 또한 (그림 9)에서 보듯이 구현된 서비스 제공자의 인터페이스만을 이용하여 데이터 서버의 객체를 변경하도록 설계하였다. 실험을 위한 공간데이터는 국립지리원에서 제작된 1:25,000 축척의 국내 시도 지역의 데이터를 사용하였다. 공간데이터는 20여 개의 주제별 계층으로 구성되어 있으며 각 계층은 각각의 데이터 클래스로써 데이터 서버에 분산되어 저장되었다. 즉 두 개의 데이터 서버는 서로 다른 클래스의 주제도 데이터를 보유하고 있다.



(그림 9) 서비스 제공자 기반의 공간데이터 변경 프로토타입 시스템

클라이언트는 기존 OGIS OLE DB 데이터 제공자가 제공하는 데이터 소스, 세션, 명령, 행집합으로 구성된 4가지 객체의 인터페이스를 이용하여 분산 공간데이터에 대한 발견, 접근, 질의 명령을 수행한다. 데이터 제공자의 인터페이스를 이용하여 서비스 제공자는 분산 공간데이터에 대한 변경 트랜잭션을 위한 인터페이스를 제공한다. 클라이언트는 ITransaction Service에 정의된 인터페이스를 이용하여 분산 공간데이터를 위한 변경 트랜잭션을 시작하고 변경된 객체의 완료를 위한 midCommit 연산을 수행한다. ITransaction Service의 midCommit 연산은 분산 공간관련성에 따른 2단계 완료 프로토콜에 따라 각 데이터 서버에 분산되어 저장되어 있는 공간객체에 대한 완료 연산을 수행한다. 또한 ILock Service는 영역 잠금과 쓰기 잠금을 위한 인터페이스를 제공한다.

6.2 실험

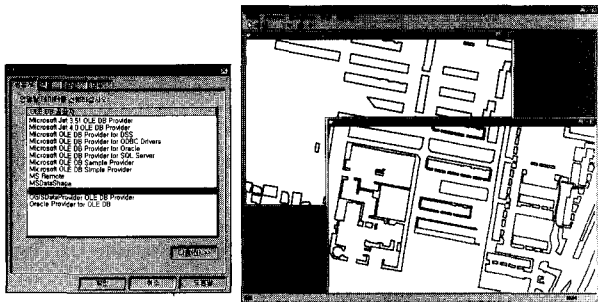
변경 실험을 위하여 이 논문에서는 Window NT 플랫폼에 저장된 일반 건물 주제도와 DEC Alpha UNIX 플랫폼



에 저장된 경지 경계 주제도를 사용하였으며, (그림 10)은 각 서버에 저장되어 있는 공간데이터들을 보여준다.

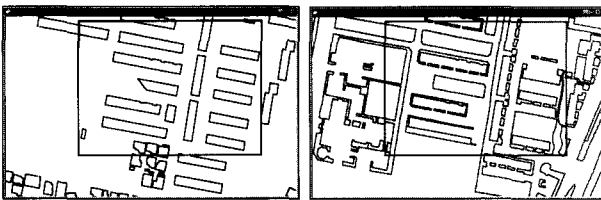


(그림 10) 경지 경계 주제도와 건물 경계 주제도

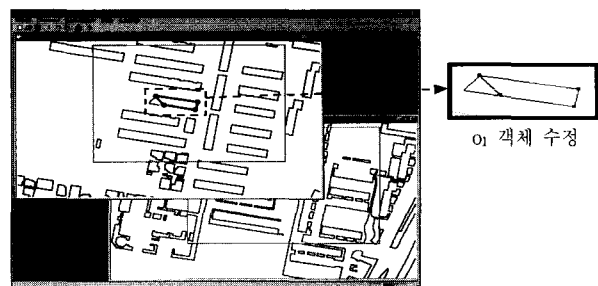


(a) OGIS 데이터 제공자 (b) OGIS 데이터 제공자를 이용한 공간 데이터 접근

(그림 11) OGIS OLE DB 데이터 제공자를 이용한 공간데이터 접근



(a) T<sub>1</sub>의 영역 잠금 설정 (b) T<sub>2</sub>의 영역 잠금 설정



(c) T<sub>1</sub>의 o<sub>1</sub> 객체 수정

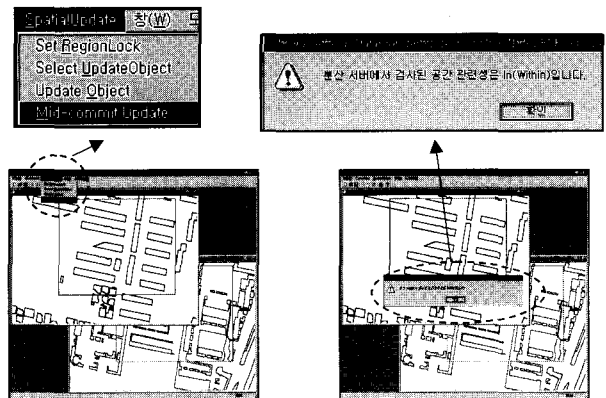
(그림 12) 영역 잠금 설정 및 T<sub>1</sub>의 객체 변경

(그림 11)(a)는 분산 공간데이터에 접근하기 위하여 OGIS OLE DB 데이터 제공자에 연결하는 것을 보여준다. 클릭이

언트인 공간데이터 소비자들은 OGIS 데이터 제공자의 인터페이스를 이용하여 각 데이터 서버에 연결하고 (그림 11)(b)와 같이 공간데이터를 분산 서버로부터 읽는다.

서브 트랜잭션 T<sub>1</sub>은 건물 주제도를 변경하고 T<sub>2</sub>는 경지 경계 주제도를 변경하도록 설정한다. T<sub>1</sub>은 트랜잭션 시작 시 (그림 12)(a)와 같이 변경할 영역에 대하여 영역 잠금을 설정하고 (그림 12)(c)와 같이 건물 객체 o<sub>1</sub>을 선택한 후에 변경한다. T<sub>2</sub>는 T<sub>1</sub>의 변경 영역과 중첩되는 영역을 변경하도록 설정하고 (그림 12)(b)와 같이 영역 잠금을 설정하였다.

T<sub>1</sub>은 (그림 12)(c)와 같이 o<sub>1</sub>에 대한 변경 작업을 수행한 후에 (그림 13)(a)와 같이 o<sub>1</sub>에 대한 midCommit을 요청한다. 이 때 o<sub>1</sub>이 저장되어 있는 데이터 서버의 서비스 제공자는 조정서비스 제공자가 되고 분산 공간관련성 기반의 2단계 완료 프로토콜을 수행한다. T<sub>1</sub> 영역 잠금의 영역과 T<sub>2</sub> 영역 잠금의 영역이 중첩하므로 T<sub>2</sub>가 작업중인 서비스 제공자는 참여서비스 제공자가 된다. 따라서 조정서비스 제공자는 참여서비스 제공자인 T<sub>2</sub>의 서비스 제공자에게 o<sub>1</sub>의 변경 내용을 전파한다. 이 때 참여서비스 제공자는 전파된 o<sub>1</sub>의 객체와 T<sub>2</sub>가 현재 변경중인 공간객체와 비교하고 (그림 13)(b)와 같이 공간관련성이 유지되기 때문에 o<sub>1</sub>의 완료에 동의한다. 조정서비스 제공자는 참여서비스 제공자의 동의에 따라 (그림 14)와 같이 o<sub>1</sub>의 변경을 완료한다.



(그림 13) 분산 공간관련성 기반의 2단계 완료 프로토콜에 따른 midCommit 수행



(그림 14) 참여서비스 제공자의 동의에 따른 변경 객체 완료

## 7. 결 론

이질적인 다중 데이터 소스에 저장되어 있는 공간데이터를 사용하기 위하여 OGC의 표준으로 제안된 OGIS OLE DB 데이터 제공자를 이용할 수 있다. 그러나 OGIS OLE DB 데이터 제공자는 데이터 발견 및 접근을 위한 인터페이스만을 정의하고 있으며 공간데이터 변경을 위한 인터페이스는 제공하지 않는다. 따라서 개방형 환경에서 분산 공간데이터의 변경을 위하여 OGIS 데이터 제공자에 기반으로 변경 인터페이스를 정의하여 제공할 필요가 있다.

이 논문에서는 이질적인 다수의 공간데이터 서버에 분산되어 저장된 공간객체의 동시 변경을 위하여 OGIS OLE DB를 기반으로 시스템을 구현하였다. 세부적으로 서비스 제공자에 공간객체 변경 인터페이스와 공간관련성 검사 인터페이스를 추가하였다. 그리고 이를 기반으로 하여 동시 변경된 공간데이터의 정확성을 검증하기 위하여 서비스 제공자간의 간섭을 통한 분산 공간관련성 기반의 2단계 완료 프로토콜을 제시하였다. 변경된 공간객체는 2단계 완료 프로토콜에 기반하여 조정서비스 제공자가 참여서비스 제공자들의 동의에 따라 완료되며 참여서비스 제공자를 변경 객체에 대한 동의를 위하여 전달된 공간객체에 대한 공간관련성 검사를 수행한다.

이 논문에서 제시한 2단계 완료 프로토콜은 기본적으로 영역 잠금을 이용하여 참여서비스 제공자를 결정하기 때문에 정확한 참여서비스 제공자의 결정이 어렵다는 문제가 있다. 이로 인하여 불필요한 변경 객체 전달이 발생할 수 있다. 향후 연구로는 클라이언트 트랜잭션과 서버 트랜잭션의 동시 수행 시 공간관련성에 따른 일관성 유지 기법이 필요하다.

## 참 고 문 헌

- [1] 최진오, 홍봉희, "분산된 지리정보시스템에서 새로운 잠금 기법을 이용한 중복된 공간데이터의 변경 전파," 한국정보과학회논문지, Vol.26, No.9, pp.1061-1072, 1999.
- [2] Jin-oh Choi, Young-sang Shin, Bong-hee Hong, "Update Propagation of Replicated Data in Distributed Spatial Databases," Proc. of Int'l Conf. on DEXA, pp.952-963, 1999.
- [3] OpenGIS Consortium, OpenGIS Simple Features Specification for OLE/COM Revision 1.1, 1999.
- [4] Microsoft Press, Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK, 1998.
- [5] Jose A. Blakeley, "Data Access for Masses through OLE DB," Proc. of ACM SIGMOD, 1996.
- [6] Silberschatz, Korth and Sudarshan, Database System Concepts, 3<sup>rd</sup> Edition.
- [7] Yuri Breitbart, Avi Silberschatz, "Multidatabase Update Issue," Proc. of ACM SIGMOD, pp.135-141, 1988.
- [8] Yuri Breitbart, Raghavan Komondoor, Rajeev Rastogi, S. Seshadri, Avi Silberschatz, "Update Propagation Protocols For Replicated Databases," Proc. of ACM SIGMOD, 1999.
- [9] M. J. Egenhofer, "Reasoning about binary topological relations," Int'l Symposium on SSD, pp.143-160, 1991.
- [10] K. Stathatos, S. Kelly, N. Roussopoulos, J. S. Baras, "Consistency and Performance of Concurrent Interactive Database Applications," Proc. of Int'l Conf. on Data Engineering, pp.602-608, 1996.
- [11] 신영상, 최진오, 조대수, 홍봉희, "클라이언트 변경 트랜잭션에서 동시성 및 일관성 제어," 한국정보과학회 가을 학술발표논문집, Vol.26, No.2, pp.323-325, 1999.
- [12] 윤우진, 한성룡, 조대수, 홍봉희, "OLE/COM을 기반으로 한 OpenGIS 미들웨어 설계," 한국개방형GIS연구회, pp.95-106, 1999.



### 김 동 현

e-mail : pusrover@pusan.ac.kr

1995년 부산대학교 컴퓨터공학과(학사)

1997년 부산대학교 대학원 컴퓨터공학과 (석사)

2003년 부산대학교 대학원 컴퓨터공학과 (박사)

2004년~현재 동서대학교 소프트웨어 전문대학원 전임강사  
관심분야 : 지리정보시스템, 모바일 GIS, 모바일 트랜잭션, 이동체 색인



### 문 상 호

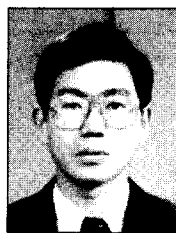
e-mail : shmoon87@pufs.ac.kr

1991년 부산대학교 컴퓨터공학과(학사)

1994년 부산대학교 대학원 컴퓨터공학과 (석사)

1998년 부산대학교 대학원 컴퓨터공학과 (박사)

1998년~2002 위덕대학교 컴퓨터멀티미디어공학부 조교수  
2002년~현재 부산외국어대학교 컴퓨터공학부 조교수  
관심분야 : 공간DB, 공간뷰, 데이터마이닝, GIS 표준, 정보 시스템 감리, 전산교육 등



### 홍 봉 희

e-mail : bhong@pusan.ac.kr

1982년 서울대학교 컴퓨터공학과(학사)

1984년 서울대학교 대학원 컴퓨터공학과 (석사)

1988년 서울대학교 대학원 컴퓨터공학과 (박사)

1987년~현재 부산대학교 공과대학 컴퓨터공학과 교수  
관심분야 : 데이터베이스, 공간 데이터베이스, 이동체 데이터 베이스, 이동체 색인, 트랜잭션