

# 웹 기반 GIS 응용을 위한 변경 프로토콜

## (Update Protocols for Web-Based GIS Applications)

안성우<sup>†</sup> 서영덕<sup>\*\*</sup> 김진덕<sup>\*\*\*</sup> 홍봉희<sup>\*\*\*\*</sup>  
 (SungWoo Ahn) (Young Duk Seo) (Jin Deog Kim) (Bong Hee Hong)

**요약** 웹 기반 서비스의 급속한 성장으로 다양한 서비스를 이용할 수 있도록 웹 환경에서 공간 데이터의 동시 변경이 가능해야 한다. 웹 기반 GIS 응용은 다양한 사용자 층에 대한 요구를 만족시키기 위하여 다양한 서비스 데이터를 필요로 하고, 이 데이터들은 지속적으로 변경되어야 한다. 또한 웹 환경에서 사용하는 HTTP 프로토콜은 연결 해제(Connectionless)와 상태 비유지(Stateless)를 가정한다. 그러나 웹 환경에서 웹 클라이언트에 의한 데이터 변경 시, 긴 트랜잭션이 특징인 공간 데이터베이스 등의 응용은 클라이언트 간의 동시성과 서버 데이터의 일관성 유지에 문제점을 발생시킨다.

이 논문에서는 웹 환경에서 웹 클라이언트가 서버 측의 공간 데이터를 변경할 때 서버 데이터의 일관성 유지를 위한 해결책을 제시한다. 웹 환경에서 공간 데이터를 변경할 때 HTTP 프로토콜의 연결 해제와 상태 비유지에 의해서 발생하는 휴면 영역 잠금(Dormancy Region Lock) 문제를 영역 잠금 기법인 RX(Region-eXclusive) 잠금과 주기적인 메시지(ALIVE\_CLIENTi) 전송에 의해서 해결한다. 그리고 이 방법을 기반으로 한 프로토콜을 제시하고, 메인 메모리 공간 데이터베이스 시스템인 사이버맵을 이용한 시스템 설계 및 구현을 통하여 검증하고 있다.

**키워드** : 지리정보시스템, 인터넷 지리정보시스템, 트랜잭션, 변경 프로토콜

**Abstract** As web-based services are becoming more and more popular, concurrent updates of spatial data should be possible in the web-based environments in order to use the various services. Web-based GIS applications are characterized by large quantity of data providing and these data should be continuously updated according to various user's requirements. Faced with such an enormous data providing system, it is inefficient for a server to do all of the works of updating spatial data requested by clients. Besides, the HTTP protocol used in the web environment is established under the assumption of 'Connectionless' and 'Stateless'. Lots of problems may occur if the scheme of transaction processing based on the LAN environment is directly applied to the web environment. Especially for long transactions of updating spatial data, it is very difficult to control the concurrency among clients and to keep the consistency of the server data.

This paper proposes a solution of keeping consistency during updating directly spatial data in the client-side by resolving the Dormancy Region Lock problem caused by the 'Connectionless' and 'Stateless' feature of the HTTP protocol. The RX(Region-eXclusive) lock and the periodically sending of ALIVE\_CLIENTi messages can solve this problem. The protocol designed here is verified as effective enough through implementing in the main memory spatial database system, called CyberMap.

**Key words** : Geographic Information System(GIS), Internet GIS, Transaction, Update Protocol

<sup>†</sup> 학생회원 : 부산대학교 대학원 컴퓨터공학과

swan@pusan.ac.kr

<sup>\*\*</sup> 비 회원 : 부산대학교 대학원 컴퓨터공학과

ydseo@pusan.ac.kr

<sup>\*\*\*</sup> 비 회원 : 동의대학교 컴퓨터공학과 교수

jdk@dongeui.ac.kr

<sup>\*\*\*\*</sup> 종신회원 : 부산대학교 공과대학 컴퓨터공학과 교수

bhhong@pusan.ac.kr

논문접수 : 2001년 10월 16일

심사완료 : 2002년 5월 23일

### 1. 서론

최근 웹 환경의 급속한 성장으로 예전보다 빠른 속도로 웹 서비스를 사용할 수 있어 기존의 랜 환경에서 특정 사용자들을 위한 서비스를 제공하던 많은 분야에서 웹 환경을 통한 서비스를 제공함으로써 보다 많은 사용자가 서비스를 사용할 수 있게 되었다[1]. GIS 분야에서도 불특정 다수를 위한 웹 서비스를 통하여 다양한

다수의 사용자들이 서비스를 받을 수 있도록 하고 있다 [2,3,4].

클라이언트/서버 환경은 현재 가장 일반적인 환경으로 클라이언트가 서버의 데이터를 접근하는 것에 대해서 많은 연구가 이루어져 왔다. 클라이언트가 서버의 데이터를 접근하는 방법은 그림 1과 같이 크게 랜 환경에서의 접근 방법과 웹 환경에서의 접근 방법으로 나눌 수 있다[5]. 랜 환경에서의 접근 방법은 폐쇄 환경에서의 접근 방법과 개방 환경에서의 접근 방법으로 다시 나누어 지고[6], OGC(Open GIS Consortium)에서는 상호 운용성을 높이기 위해서 랜 환경에서의 접근 방법 중에서 개방 환경에서의 접근 방법에 대해서 표준을 제정하고 있다[7].

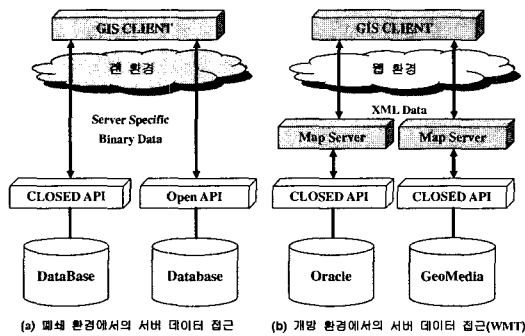


그림 1 서버의 데이터를 접근하는 방법

또한, OGC WMT [8]에서는 웹 환경이 급속한 성장에 의해 클라이언트/서버 환경에서 중요한 역할을 하게 됨에 따라, 그림 1의 (b)에서 나타내는 것과 같이 웹 환경에서 서로 다른 공간 데이터 서버의 데이터를 클라이언트에서 접근할 수 있는 방법을 제안하고 있다.

그림 2는 웹 클라이언트가 서버에 저장되어 있는 공간 데이터를 전송 받아서 화면에 출력하기 원하는 경우에 대한 주제도의 예를 보여주고 있다. 주제도의 예로는 특정 회사의 전국 대리점 망, 약도 서비스 등 다양한 데이터가 적용될 수 있다. 이러한 주제도는 수정 권한을 가진 사용자에 의해서 기본도를 바탕으로 각자 자신의 요구에 맞게 특정 서비스에 대한 용도로 정의되어야 하므로, 서버에서 주제도의 모든 데이터를 입력하고 변경, 관리하는 것은 매우 어렵고 비효율적인 문제가 발생된다. 따라서, 정확한 서버 데이터의 관리를 위해서 웹 클라이언트가 데이터의 변경이 필요할 때 데이터를 직접 변경할 수 있는 방법이 제시되어야 한다.

이 논문에서는 웹 환경에서 웹 클라이언트가 서버의

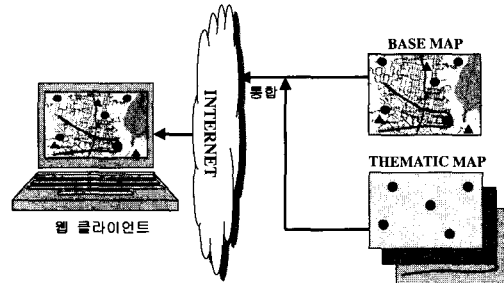


그림 2 웹 클라이언트의 공간 데이터 접근 예

공간 데이터를 변경하려고 할 때 발생할 수 있는 문제점을 제시한다. 또한, 이를 효과적으로 해결하기 위한 공간 데이터 서버와 클라이언트 간의 프로토콜을 제시하고 구현을 통하여 프로토콜을 검증한다. 문제 해결 방안으로 서버 데이터의 영역 잠금(Region Locking) 기법[9], [10]을 웹 환경에서 사용하기 적합한 형태로 확장한 RX (Region-eXclusive) 잠금을 정의하여 사용한다. 그리고 서버 측으로 클라이언트의 주기적인 메시지 전송 방법[5]을 사용하여 비정상적인 상황이 발생했을 때 서버에 남아 있을 수 있는 휴면 영역 잠금(Dormancy Region Lock)을 제거함으로써 클라이언트의 동시성 유지와 서버 데이터의 일관성 유지를 할 수 있도록 하고 있다.

이 논문의 구성은 다음과 같다. 2장에서 관련 연구에 대해서 조사하고, 3장에서는 웹 환경에서 공간 데이터를 변경하려고 할 때 고려해야 할 사항을 분석하고, 4장에서는 이러한 고려 사항에 대한 웹 환경에서의 공간 데이터 변경 방법을 제시한다. 5장에서는 이 논문에서 제시하는 공간 데이터 변경 프로토콜을 기술하고, 6장에서는 시스템 구조를 제시하며 이에 대한 설계 및 구현에 대해서 기술한 후 7장에서 결론 및 향후 연구로 끝을 맺는다.

## 2. 관련 연구

최근 웹을 중심으로 널리 알려져 있는 정보를 효율적으로 관리하고, 사용자들이 좀 더 편리하고 빠르게 정보를 접근하게 하기 위해서 정보 검색, 인공 지능 등의 데이터베이스 연동에 관해서 많은 연구가 이루어지고 있다[11]. 그러나, 랜 환경에서 동작하는 데이터베이스 응용을 웹 환경에 그대로 적용하기는 어려워, 웹 환경에서 데이터베이스 응용의 개발 방법과 고려해야 할 사항에 대한 연구 또한 진행되고 있다[12, 13, 14,15].

[12]는 HTML 기반 서비스를 기존의 응용 또는 HTML 기반 이외의 웹 응용(Java Applet, Active X

등)들과 비교할 때의 장점과 함께 기존의 응용과 비교하였을 때 발생할 수 있는 문제점으로 서버가 클라이언트에게 특정 이벤트가 발생된 사실을 알릴 수 없기 때문에 서버 중심적인 통신이 어렵고, 서버가 클라이언트에게 작업의 진척 사항을 알려주는 것이 어렵다는 점을 지적하고 있다. [13], [14], [15]는 웹 환경에서의 데이터베이스 응용 개발 방법 중에서 GIS 데이터를 처리하는 응용에 대해서 설명하고 있다. [13], [14]는 인터넷 환경에서 GIS 응용을 개발할 때 인터넷이 응용 프로그램에 미치는 영향과 응용의 개발 방법에 대해서 다루고 있다.

[16]은 웹 환경의 트랜잭션 처리를 위하여 웹 서버 또는 협동 작업 시스템과 독립적으로 동작하는 트랜잭션 서버(JPernLite)를 제시함으로써 웹 환경에서의 동시성 제어에 대한 빈틈의 해결에 대해서 주장하고 있다. 잠금 설정이나 잠금 해제에 대한 요청을 트랜잭션 서버에 요청을 하면, 이 요청에 대하여 서버는 이벤트 통지에 의해서 응답을 하는 방법으로 데이터 소스와의 통신을 하고 있다.

클라이언트/서버 환경에서 여러 클라이언트가 서버의 데이터를 변경할 때 서버 데이터의 일관성 유지를 위해서 여러 가지 방법이 사용된다. 가장 많이 사용되는 방법으로 잠금 기반 변경 방법과 체크아웃 기반 변경 방법이 있다.

[9], [6], [10]은 랜 환경에서 서버의 공간 데이터를 변경할 때 기존의 잠금 기반 변경 방법과 체크아웃 기반 변경 방법에서 고려되지 않았던 공간 관련성을 고려한 확장된 잠금 기법을 도입하여 서버 데이터의 일관성 유지와 클라이언트의 동시성 향상 문제에 접근하고 있다.

[10]은 확장된 잠금을 사용하는 영역 잠금법 및 공간 관련성 바운드 WRITE 잠금법에 대한 연구에 기반하고 있다. [10]은 공간 데이터가 정적으로 중복되어 있는 분산 환경을 대상으로 하고 있고 각 사이트가 자치성을 가지고 데이터베이스를 구축하며 잠금 정보를 각 사이트가 관리 및 타 사이트로 전파하는 방식을 사용한다. [9]는 [10]의 영역 잠금법 및 공간 관련성 바운드 WRITE 잠금법을 도입하여 확장함으로써 클라이언트 변경 트랜잭션들의 동시성을 지원하고 있다. [10]에서 제시하는 잠금법은 분산 잠금법들로 자신의 분산 사이트에 있는 데이터베이스의 객체들을 잠금 대상으로 하며 잠금 설정을 모든 분산 사이트에 전파하므로 메시지 전달 과부하의 이유로 클라이언트/서버 환경에는 그대로 적용하기가 힘들다. 따라서, [9]에서는 [10]의 잠금 방법들을 확장하여 CR(Cache-Region) 잠금과 CX(Cache-eXclusive) 잠금을 이용하는 새로운 캐쉬 영역 잠금법 프로토콜을

제시하여 클라이언트/서버 환경에 적용하고 있다.

### 3. 공간 데이터 변경 작업 시 고려 사항

#### 3.1 클라이언트의 정보 유지 방법

앞으로의 논문의 전개를 위해서 다음과 같이 랜 환경과 웹 환경에 대한 정의를 한다.

##### 정의 1 랜 환경

클라이언트/서버 환경 중 TCP/IP 소켓을 이용하여 서버와 클라이언트의 세션을 계속 유지하여 통신을 하는 환경을 랜 환경이라고 정의한다. 랜 환경의 서버는 클라이언트의 현재 상태를 감지(연결/비연결)할 수 있고, 클라이언트에게 현재 상태에 대한 이벤트를 통지할 수 있다.

##### 정의 2 웹 환경

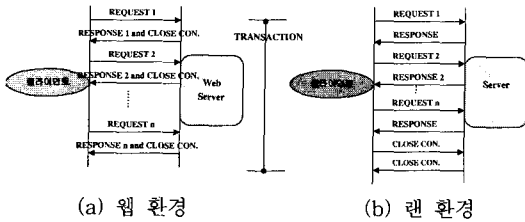
클라이언트/서버 환경 중 HTTP 프로토콜(요청/응답 프로토콜 : REQUEST/REPLY Protocol)을 이용하여 서버와 클라이언트 간의 통신을 하는 환경을 웹 환경이라고 정의한다. 웹 환경의 서버와 클라이언트는 요청/응답 기간 동안만 서로 간의 세션을 유지할 수 있고 이후에는 서로의 상태에 대해서 감지할 수 없으므로, 서버는 능동적으로 클라이언트에게 현재 상태에 대한 이벤트를 통지할 수 없다.

그리고, 이후에 제시되는 정의와 예제에서 사용하게 될 기호에 대해서 다음과 같이 정의한다.

##### 정의 3

- 공간 데이터 변경 작업을 하는 클라이언트의 집합 :  $C = \{C_1, C_2, \dots, C_i, C_n\}$
- $C_i$ 가 수행 중인 공간 데이터 변경 트랜잭션 :  $T_{C_i}$
- $C_i$ 의 서버 데이터에 대한 잠금 :  $R_{C_i}$
- $C_i$ 와  $C_j$ 가 요청하는 잠금 영역의 겹침(Overlap) :  $R_{C_i} \cap R_{C_j} \neq \phi$

그림 3에서는 웹 환경과 랜 환경에서의 통신 방법에 대한 차이점을 보여주고 있다. 기존 랜 환경(그림 3의 (b))의 서버는 클라이언트의 현재 상태(세션)를 유지하면서 현재 수행 중인 트랜잭션의 상태(처리 완료, 처리 실패, 잠금 정보 등)를 클라이언트에게 보내주고, 이 정보를 이용하여 클라이언트는 적절한 트랜잭션 작업을 할 수 있다. 그러나 그림 3의 (a)와 같이 웹 환경에서는 한 트랜잭션 단위 동안 클라이언트와 서버 간의 연결 상태가 계속 끊어지고, 서버는 클라이언트의 요청이 없이는 클라이언트에게 어떠한 이벤트도 전달할 수 없다 [12]. 따라서, 기존의 응용과 같은 클라이언트와 서버간의 통신은 웹 환경에서 기대하기가 어렵다. 이러한 이유 때문에 웹 환경을 연결 해제(Connectionless)와 상태 비유지(Stateless)[12], [15], [17]를 가정한 환경이라고 한다.



(a) 웹 환경 (b) 랜 환경  
 그림 3 웹 환경과 랜 환경의 통신 방법 비교

결국 서버와 클라이언트의 상호 작용을 할 수 있는 수단이 제공되지 않은 상태에서 트랜잭션이 수행되면 동시에 여러 클라이언트가 트랜잭션 처리 중 서로 변경 충돌이 발생하는 경우 해당 클라이언트에게 적절한 메시지를 보내 줄 수 없다. 그리고 변경 충돌 시 데이터 변경에 대한 우선 권한이 있는 클라이언트에게서 오랫동안 응답이 없을 때 해당 클라이언트가 작업을 계속 수행 중인지, 오류가 발생했는지 결정을 할 수 없다. 따라서, 서버는 어떠한 클라이언트에게 작업 권한을 넘겨 줄 지에 대해서 판단하기가 힘들어 지고 서버 데이터의 일관성 유지에도 문제가 발생할 수 있다. 특히, 긴 트랜잭션이 특징인 공간 데이터 변경 트랜잭션에서 이런 문제는 더욱 심각해서 클라이언트 작업의 동시성 유지 및 서버 데이터의 일관성 유지에 많은 문제점이 발생하게 된다. 이와 같은 문제를 피하기 위해서 서버가 클라이언트의 상태 정보를 유지할 수 있는 방법이 필요하다.

현재 웹 환경에서는 클라이언트의 정보 유지 방법으로 다음과 같은 방법들을 사용한다.

1. HTML의 FORM이나 HIDDEN 필드를 사용하여 클라이언트의 상태 정보 유지
2. 쿠키(Cookie)를 사용하여 클라이언트 측에 상태 정보를 저장
3. URL Rewriting : 클라이언트의 상태에 대한 정보를 URL에 붙여서 요청

이 논문에서는 세 가지 방법 중에서 Applet이나 ActiveX와 같은 클라이언트 응용에서 쉽게 적용할 수 있는 URL Rewriting 방법을 사용한다. 그림 4는 URL Rewriting의 예로서 클라이언트가 서버에게 작업을 요청할 때 URL 뒤에 클라이언트 ID, 트랜잭션 ID, 현재 클라이언트의 상태, 요청한 작업 등의 정보를 붙여서 보내면 서버는 이 URL을 분석하여 클라이언트의 여러 가

```
http://www.server.co.kr/cgi-bin/server.cgi?CID=cid&TID=tid&Status=status&operation=op
```

그림 4 URL Rewriting의 예

지 정보를 가지고 올 수 있다.

그러나 웹 환경에서는 서버가 상태 정보를 유지하더라도 클라이언트에게 능동적으로 메시지를 전파하는 것이 불가능하기 때문에 클라이언트 간의 메시지 전송에 의한 협동 작업이 어려운 문제는 계속 남아있다.

3.2 비정상적인 상황 발생

공간 데이터 변경 작업 중에 비정상적인 상황이 발생하여 서버와 클라이언트가 서로 통신을 할 수 없는 경우 클라이언트의 긴 대기시간과 서버 데이터의 일관성 유지에 문제가 발생하게 된다. 특히, 웹 환경에서는 3.1 절에서 제시한 것과 같이 서버에서 클라이언트의 상태를 유지하지 못한다면 비정상적인 상황이 발생하였는지 예측하는 것이 힘들어지고 클라이언트 간의 동시성과 서버 데이터의 일관성에 대한 문제가 더욱 심각해진다. 이 논문에서는 발생할 수 있는 비정상적인 상황을 그림 5와 같이 세 가지 상황으로 분류한다.

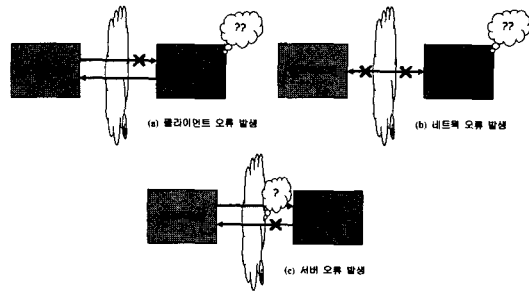


그림 5 비정상적인 상황의 종류

클라이언트 오류 발생은 공간 데이터 변경 작업을 하던  $C_i$ 가 응용의 결함에 의해  $T_{C_i}$ 를 수행하는 도중에 종료하는 경우이다. 이 경우 종료 이후에  $C_i$ 는 서버 측으로 어떤 요청도 하지 않으나 서버는  $C_i$ 의 예기치 못한 종료 상황을 예측할 수 없다. 따라서,  $R_{C_i} \cap R_{C_j} \neq \emptyset$ 인  $R_{C_j}$ 에 대해서 변경 작업을 수행하고자 하는  $C_j$ 가 접속하였을 때 서버가 판단하는 것이 불가능하므로 변경 작업의 동시성 유지에 문제가 발생된다.

네트워크 오류 발생은  $C_i$ 가  $T_{C_i}$ 를 계속 수행 중이나 서버와의 연결에 문제가 발생한 경우이다. 이 경우 서버는 네트워크 오류를 감지할 수 있지만 클라이언트의 오류 발생과 동일하게  $C_i$ 의 상황을 예측할 수 없다. 따라서, 이 경우에도 변경 작업의 동시성 유지에 문제가 발생된다.

서버 오류 발생은  $C_i$ 는 정상적으로  $T_{C_i}$ 를 수행하지만 서버에서 예기치 못한 상황이 발생하여 종료한 경우이다. 이 경우에도  $T_{C_i}$ 가 제대로 종료되지 않았으므로 서

버 데이터의 일관성 유지에 문제가 발생된다. 그러나 이 경우는 서버측의 데이터베이스에 의한 일관성처리에 의하여 처리가 가능하므로 이 논문의 논외로 한다.

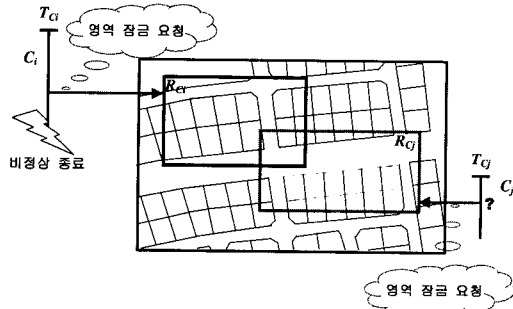


그림 6 비정상 종료에 의한 잠금 방법의 문제 발생

[9], [10]에서 제시하고 있는 영역 잠금 방법을 적용하여 이 논문에서 제시하는 웹 환경을 대상으로 공간 데이터 변경 작업을 수행할 때 그림 6에서 보는 것과 같이 잠금 정보로 인해서 트랜잭션의 동시성에 문제가 발생하게 된다. 즉,  $C_i$ 가  $T_{C_i}$ 를 시작하고 난 후 데이터 변경을 위해서  $R_{C_i}$ 를 요청하고 작업을 하던 중에  $T_{C_i}$ 를 종료하지 못한 상태에서 비정상적인 상황이 발생하여 종료를 하였을 경우 서버에서  $C_i$ 에 대한  $R_{C_i}$ 를 계속 유지한다면, 이 후  $C_j$ 가 트랜잭션  $T_{C_j}$ 를 요청하고  $R_{C_i} \cap R_{C_j} \neq \emptyset$ 인  $R_{C_j}$ 를 요청하면  $R_{C_i}$  때문에 계속 대기해야 하는 상황이 발생한다.

[18]에서는 여러 트랜잭션이 동시에 서버 데이터 변경 작업을 수행할 때 메시지 전송의 문제로 잠금 정보 유지에 문제가 발생하는 경우(False Deadlock)에 대해서 설명하고 있다. 이 논문에서도 그림 6과 같이  $C_i$ 의 오류 발생에 의해서 서버 측으로 메시지가 전송되지 않아 잘못된 잠금 정보( $R_{C_i}$ )가 남아  $C_j$ 가 대기하는 상황이 발생하는데, 이 상황에 대해서 다음과 같이 정의한다.

**정의 4 휴면 영역 잠금(Dormancy Region Lock:  $DRL(R_{C_i})$ )**  
 $T_{C_i}$ 에서 요청한 영역 잠금  $R_{C_i}$ 에 대해서,  $R_{C_i}$ 를 요청한 후에  $C_i$ 의 오류로 서버와 통신을 할 수 없게 되면  $R_{C_i}$ 가 계속 서버에 남아 있게 된다. 이 때,  $R_{C_i}$ 를 휴면 영역 잠금(Dormancy Region Lock:  $DRL(R_{C_i})$ )이라고 정의한다.

요청/응답 프로토콜에서 서버 데이터의 변경 트랜잭션을 수행할 때 비정상적인 상황이 발생하게 되면 정의 4에서 제시한 휴면 영역 잠금이 서버에 남게 되어 클라이언트 작업에 대한 동시성 및 서버 데이터의 일관성

유지에 문제점이 발생하게 된다. 4장에서 휴면 영역 잠금 문제를 해결하는 방법에 대해서 제시하고 있다.

#### 4. 웹 환경에서의 공간 데이터 변경 방법 제시

3장에서는 웹 환경에서 공간 데이터 변경 작업을 수행할 때 기존의 랜 환경에서는 고려되지 않았던 문제점에 대해서 살펴보았다. 이 장에서는 3장에서 살펴본 문제를 중심으로 공간 데이터를 변경하는 방법에 대해서 살펴보고 비정상적인 상황의 발생에 의한 휴면 영역 잠금을 해결하기 위한 방법에 대해서 기술한다.

##### 4.1 공간 데이터 변경 방법

###### 4.1.1 RX 잠금

이 논문에서는  $C = \{C_1, C_2, \dots, C_i, \dots, C_n\}$ 이 서버에 있는 공간 데이터를 변경하기 위해 [9], [10]에서 제시하고 있는 영역 잠금 방법을 사용한다. 웹 환경에서는 [9]에서 제시하는 환경과 달리 CX 잠금을 통하여 한 객체에 대한 잠금을 설정하는 것 보다 변경하고자 하는 객체를 포함하는 영역을 설정하여 그 영역과 겹치는 다른 영역에 대해서는 잠금을 설정하지 못하도록 하고 작업을 할 수 있도록 하는 것이 더 효율적이다. 따라서, 이 논문에서 사용하는 영역 잠금에 대해서 다음과 같이 정의를 한다.

**정의 5 RX(Region-eXclusive) 잠금**  
 $C_i$ 가  $T_{C_i}$ 를 수행하고  $C_j$ 가  $T_{C_j}$ 를 수행할 때,  $C_j$ 가 공간 객체 또는 공간 객체 집합을 변경하기 위해서 객체 집합을 포함하는 영역  $R_{C_j}$ 를 설정하여 변경 작업 중이라면,  $C_i$ 는  $R_{C_i} \cap R_{C_j} \neq \emptyset$ 인  $R_{C_j}$ 를 설정하여 작업을 하기 위해서는  $C_i$ 가  $R_{C_i}$ 를 해제할 때까지 대기해야 한다. 이 때  $C_i$ 의 쓰기 잠금 모드를 **RX(Region-eXclusive) 잠금**이라고 정의하고,  $C_i$ 에 의해 요청된 RX 잠금을  $RX_{C_i}$ 라고 한다.

기존의 쓰기 잠금이 객체 간의 전역 공간 관련성을 고려하지 않고 있기 때문에 공간 데이터 변경 작업은 전체 데이터 집합 또는 한 레이어 전체에 대한 잠금을 필요로 하고 결국  $C_i$ 가 공간 데이터 변경 작업을 할 동안 나머지 클라이언트는  $C_i$ 의 잠금에 대해서 공간 관련성을 가진 객체를 수정하기 위해서 계속 대기해야 한다. 그러나 이 논문에서 제시하는 RX 잠금은 겹치지 않는 영역에서 변경 작업을 하는 클라이언트 간의 동시성은 유지시켜준다. RX 잠금은 클라이언트 간의 메시지 전송에 의한 협동 작업을 고려하지 않기 때문에 [9]에서 제시하는 영역 잠금 기법에 비해서 클라이언트의 동시성이 떨어지지만 4.2절에서 제안하는 방법과 함께 HTTP 프로토콜을 이용하여 공간 데이터 변경 트랜잭션을 처

리할 수 있는 방법을 제공해 준다.

그림 7은 RX 잠금의 예를 보여 준다. 서버는 필지(Property) 레이어와 도로(Road) 레이어를 저장하고 있고  $C_i$ 와  $C_j$ 는 서버의 공간 데이터에 대해서 각각 필지 레이어와 도로 레이어의 객체들만을 가지고 있다. 만약 그림 7의 (a)와 같이  $C_i$ 가  $RXC_i$ 를 서버에 요청하여 잠금 권한을 얻으면 서버에 저장되어 있는 모든 레이어에 대해서  $RXC_i$ 가 설정되고, 결국  $C_j$ 가 저장하고 있는 도로 레이어에 대해서도  $RXC_i$ 가 설정된다. 즉, RX 잠금의 대상은 서버 데이터베이스에서 RX 잠금 영역 내에 완전히 포함되는 모든 공간 객체들이다. 따라서,  $C_j$ 가  $RXC_i \cap RXC_j \neq \emptyset$ 인  $RXC_j$ 를 설정하려면  $C_i$ 에서  $RXC_i$ 를 해제할 때까지 대기해야 한다. 그러나, 그림 7의 (b)에서와 같이 RX 잠금은  $C_i$ 와  $C_j$ 의 잠금인  $RXC_i$ 와  $RXC_j$ 에 대해서  $RXC_i \cap RXC_j \neq \emptyset$ 가 아닐 경우에는 동시에 잠금을 설정하는 것을 허용하여 두 클라이언트가 서버 데이터에 대해서 병렬 작업이 가능하도록 하고 있다.

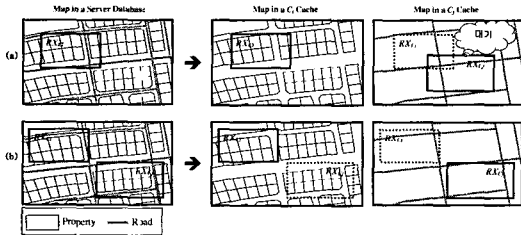


그림 7 RX 잠금의 예

4.1.2 공간 데이터 변경 트랜잭션 모델

그림 8은 RX 잠금을 이용한 공간 데이터 변경 트랜잭션 모델이다. 이 논문은 웹 환경에서의 공간 데이터 변경 프로토콜을 제시하고 있기 때문에 웹 환경의 특징인 요청/응답 프로토콜을 사용한다. 즉, 클라이언트의 요청에 대해서 서버는 응답을 한 후 클라이언트와의 연결을 끊는다.

공간 데이터 변경 트랜잭션을 시작하고 난 후 RX잠금의 영역을 설정하고 서버로 잠금 설정을 요청한다. 잠금 권한을 획득한 후 클라이언트는 영역에 포함되는 객체들에 대한 변경 작업을 수행한 후 변경한 결과를 서버로 전달해서 서버에 임시 저장하도록 한다. 클라이언트에서 모든 변경작업을 완료하면 서버는 임시 저장되어 있던 객체 변경 정보를 실제 DB에 저장한다. 그 후 RX 잠금을 해제하며 트랜잭션을 완료하게 된다. 이 때, 서버에 저장되어 있는 클라이언트의 정보와 트랜잭션 정보는 삭제된다.

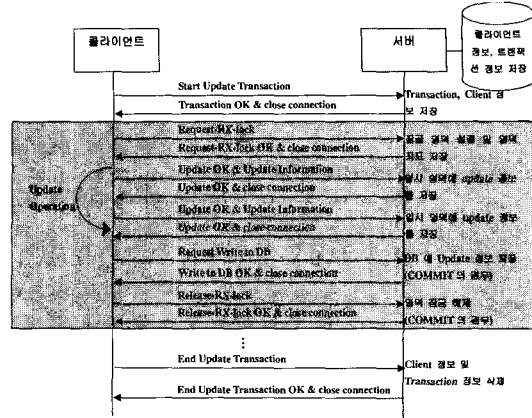


그림 8 공간 데이터 변경 트랜잭션 모델

4.2 휴면 영역 잠금 해결 방법

휴면 영역 잠금(Dormancy Region Lock)은 3장에서 언급한 바와 같이 잠금 해제를 대기할 필요가 없는 클라이언트의 대기를 초래하여 변경 작업의 동시성을 저하시키고 서버 데이터의 일관성에도 영향을 미치기 때문에 해결할 수 있는 방법이 필요하다. 이 논문에서는 비정상적인 상황이 발생했을 때 서버 중심적인 해결 방법을 사용한다. 그러므로 클라이언트의 오류발생과 네트워크 오류 발생은 동일한 오류로 취급된다.

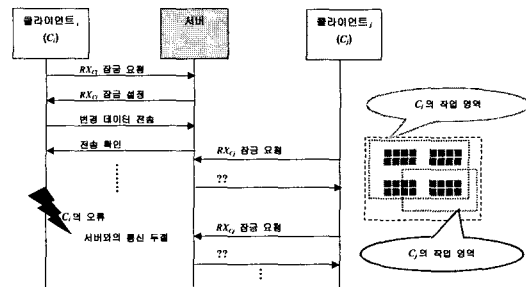


그림 9  $RXC_i \cap RXC_j \neq \emptyset$ 인 영역에 대한 변경 작업의 문제점

그림 9와 같이  $C_i$ 가 서버의 공간 데이터 변경을 위해서  $RXC_i$ 를 설정했을 때  $C_j$ 는  $RXC_i \cap RXC_j \neq \emptyset$ 인  $RXC_j$ 를 요청하기 때문에  $C_j$ 의 작업 중에는 잠금을 설정할 수 없다. 그러나 웹 환경에서는 서버는 클라이언트의 요청에만 응답을 하므로 클라이언트가 작업 요청을 해 오기 전에는 클라이언트의 현재 상태를 알 수가 없다. 만약,  $C_i$ 에게 오류가 발생하여 서버에게 작업 요청을 할 수 없는 상황에서 서버는 계속  $C_i$ 의 작업 요청을 기다리고  $C_j$ 에게 적절한 응답을 할 수 없게 된다.

그림 9에서 서버는  $C_i$ 가 처음  $RX_{C_i}$  요청을 했을 때  $C_i$ 가 작업을 하고 있기 때문에  $C_i$ 에게 대기 명령을 내리고, 두 번째  $RX_{C_i}$  요청을 했을 때  $C_i$ 가 설정한  $RX_{C_i}$ 는  $DRL(R_{C_i})$ 이기 때문에  $C_i$ 가 요청한  $RX_{C_i}$ 를 받아 들여야 한다. 그러나 단순히 클라이언트의 요청만 기다리는 응답/요청 프로토콜에서는 이 두 가지의 경우 중 어떠한 경우도 해결할 수가 없다. 서버는 위와 같은 상황에서 변경 작업을 계속 수행할 수 있도록 하기 위해서 다음의 두 가지 문제를 해결해야 한다. 첫째, 변경 작업 도중  $C_i$ 의 오류가 발생했을 때 오류를 감지하여  $C_i$ 의  $DRL(R_{C_i})$ 를 풀 수 있어야 한다. 둘째, 오류 발생 이후  $C_i$ 가 다시 작업 요청을 해 왔을 때  $RX_{C_i}$ 의 영역 안 데이터가 다른 클라이언트에 의해서 변경이 되지 않았다면 다시 작업을 계속할 수 있도록 해야 한다.

이 논문에서는  $C_i$ 에 대한 휴면 영역 잠금( $DRL(R_{C_i})$ ) 해결 방안으로 다음과 같은 방법을 제시한다.

A. 휴면 영역 잠금 감지 방안

서버는  $T_{C_i}$ 를 수행하는 동안  $C_i$ 가 어떤 상태에 있는지를 적절히 예측할 수 있어야 한다. 이와 같은 예측을 통하여 다른 클라이언트로부터  $RX$  잠금 요청이 들어왔을 때 서버 측에서 적절한 판단을 할 수 있기 때문이다. 휴면 영역 잠금 감지 방안에 대한 해결책을 제시하기 위해 다음과 같이 정의한다.

**정의 6 ALIVE\_CLIENT<sub>i</sub> 메시지 :**  $T_{C_i}$ 를 수행 중인  $C_i$ 가 서버에  $T_{C_i}$ 가 계속 수행되고 있다는 것을 알려 주기 위해서 서버 측으로 주기적으로 보내는 메시지

**정의 7 EXPIRE\_TIME :**  $C_i$ 의 오류 발생 여부를 판단하기 위한 방법으로 ALIVE\_CLIENT<sub>i</sub> 메시지가 서버에 도착해야 할 최대의 시간

정의 6에서 클라이언트가 메시지를 보내는 주기가 짧은 경우 서버측에서 과부하가 발생한다. 또 GIS의 변경 연산은 긴 시간(Long Duration) 트랜잭션이기 때문에 주기는 과도하게 짧아선 안 된다. 정의 6, 7에서의 ALIVE\_CLIENT<sub>i</sub>와 EXPIRE\_TIME을 바탕으로 서버가 휴면 영역 잠금( $DRL(R_{C_i})$ )을 판단하는 방법은 그림 10에서 제시하는 방법과 같다.

그림 10과 같이 서버가 휴면 영역 잠금을 감지할 수 있도록 하기 위해서 데이터 변경 작업을 하는  $C_i$ 는  $T_{C_i}$ 를 시작한 시점부터  $T_{C_i}$ 가 끝날 때까지 서버에게 계속 ALIVE\_CLIENT<sub>i</sub> 메시지를 보낸다. 서버에서는 ALIVE\_CLIENT<sub>i</sub> 메시지를 보고  $C_i$ 의 오류가 발생했는지에 대해서 결정을 한다. 즉, 서버는 미리 정의된 EXPIRE\_TIME 안에  $C_i$ 로부터 ALIVE\_CLIENT<sub>i</sub> 메시지가 오지

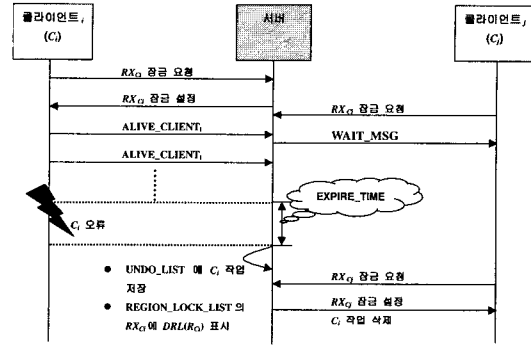


그림 10 휴면 영역 잠금 감지 방안

않으면  $C_i$ 는 오류가 발생했다고 가정을 하고 오류 회복 작업에 들어간다. 그리고 오류 회복 작업 후  $C_i$ 가  $DRL(R_{C_i}) \cap RX_{C_i} \neq \emptyset$ 인  $RX_{C_i}$  요청을 하면  $C_i$ 에게 잠금 권한을 넘겨주게 된다.

B. 오류 회복 작업

$C_i$ 의 오류가 발생하여 서버에 휴면 영역 잠금( $DRL(R_{C_i})$ )이 설정되어 있다고 서버에서 판단을 하면 서버는 오류 회복 작업에 들어간다. 서버는 오류 회복 작업을 하기 위해서 서버에 표 1과 같은 데이터를 테이블에 저장한다. 각 테이블의 역할은 다음과 같다.

- TRANSACTION\_INFO\_LIST :  $T_{C_i}$ 를 시작할 때  $T_{C_i}$ 에 대한 정보를 저장하기 위해서 유지된다.
- UNDO\_LIST :  $C_i$ 가 오류 회복 후 다시 작업을 요청해 왔을 때,  $DRL(R_{C_i}) \cap RX_{C_i} \neq \emptyset$ 인 영역에 대해서 작업을 하는  $C_i$ 에 의해서  $DRL(R_{C_i})$ 가 삭제되지 않았다면 다시 작업을 진행할 수 있도록 TRANSACTION\_INFO\_LIST의 정보를 UNDO\_LIST에 옮겨 놓는다.
- REGION\_LOCK\_INFO\_LIST :  $T_{C_i}$  수행 중  $C_i$ 가  $RX_{C_i}$ 를 요청하면 잠금에 대한 정보를 저장하기 위한 테이블이다. STATUS가 "Real\_Lock"이면  $RX_{C_i}$ 가 설정된 상태이고, "Dormancy\_Lock"이면  $C_i$ 의 오류로  $RX_{C_i}$ 가 취소된 상태( $DRL(R_{C_i})$ )이다. 즉, 휴면 영역 잠금으로 판단이 됐지만 오류가 발생했을 때  $RX_{C_i}$ 를 삭제하지 않고  $DRL(R_{C_i})$  표시만 해 둠으로써,  $DRL(R_{C_i}) \cap RX_{C_i} \neq \emptyset$ 에 대한  $C_i$ 에 의해서 잠금이 풀리지 않은 경우 다시 작업을 계속할 수 있도록 한다. 만약,  $DRL(R_{C_i}) \cap RX_{C_i} \neq \emptyset$ 에 대한  $C_i$ 에 의해서  $RX_{C_i}$  요청이 들어왔을 때 STATUS가 "Dormancy\_Lock"인 경우 테이블에서  $DRL(R_{C_i})$ 를 삭제하고  $RX_{C_i}$ 를 저장한다. 그리고, UNDO\_LIST

표 1 오류 회복을 위해 서버에 저장되는 테이블

테이블 명	필드 명	타입	설명
• TRANSACTIONS_INFO_LIST • UNDO_LIST	TID	TRANSACTION_ID_TYPE	트랜잭션 ID
	CID	STRING	클라이언트 ID
	RID	REGIONLOCK_ID_TYPE	요청한 RX 잠금에 대한 ID
	TIME_STAMP	TIME	다음 작업 초과 시간
• REGION_LOCK_INFO_LIST	RID	REGIONLOCK_ID_TYPE	RX 잠금에 대한 ID
	TID	TRANSACTION_ID_TYPE	RX 잠금의 권한을 가진 트랜잭션 ID
	CID	STRING	RX 잠금의 권한을 가진 클라이언트 ID
	Lx, Ly, Hx, Hy	MBR_TYPE	RX 잠금 영역의 MBR
	STATUS	INTEGER	• Real_Lock : 정상적인 RX 잠금 • Dormancy_Lock : DRL(RCi)
• UPDATE_OBJECT_LIST	TID	TRANSACTION_ID_TYPE	변경 트랜잭션 ID
	CID	STRING	변경 클라이언트 ID
	OPTYPE	OP_TYPE	객체 변경 작업 종류 • INSERT • DELETE • UPDATE
	OID	OID_TYPE	변경된 객체의 OBJECT ID
	GEOMETRY	GEOMETRY_TYPE	DB에 저장되는 형태의 Geometry

에서  $T_{Ci}$ 의 정보를 삭제함으로써  $C_j$ 에게 잠금의 권한을 완전히 넘긴다.

- UPDATE\_OBJECT\_LIST :  $C_i$ 에서 변경한 작업을 서버의 DB에 저장하기 전에 임시로 저장하기 위한 테이블이다. 변경 작업을 완료하고  $C_i$ 가 COMMIT 메시지를 보내면 이 테이블의 데이터를 DB에 저장한다.

5. 공간 데이터 변경 프로토콜

공간 데이터 변경 프로토콜은 그림 8에서 제시한 것과 같이 변경 트랜잭션 수행 요청을 서버에 하고 난 후 RX 잠금을 통하여 공간 데이터 변경 권한을 얻은 후에 변경 작업을 하고 변경이 완료된 후에 작업한 데이터를 서버의 DB에 저장하는 방법으로 진행된다. 각 단계의 작업에 대한 프로토콜은 다음과 같다.

5.1 트랜잭션 시작 요청

$C_i$ 가  $T_{Ci}$  요청을 하는 프로토콜은 그림 11과 같다.  $C_i$ 는 서버 측으로  $C_i$ 의 ID(CID)를 보내고, 이 메시지를

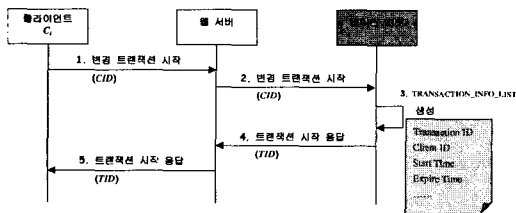


그림 11 트랜잭션 시작 요청 프로토콜

넘겨 받고서 데이터 서버는 해당되는 CID에 대해서 TRANSACTION\_INFO\_LIST를 생성한 후에  $C_i$ 에게 생성된  $T_{Ci}$ 의 ID(TID)를 보냄으로써  $T_{Ci}$ 의 시작 요청이 성공되었다는 것을 알려준다.

5.2 RX 잠금 요청

그림 12와 같이  $C_i$ 는  $RX_{Ci}$  요청을 할 때  $C_i$ 가 수행 중인 TID, CID, 그리고 영역에 대한 MBR 정보를 서버에 넘겨준다. 이 정보를 가지고 데이터 서버는 저장되어 있는 REGION\_LOCK\_INFO\_LIST에 대해서 요청한  $RX_{Ci}$ 가  $RX_{Ci} \cap RX_{Cj} \neq \emptyset$ 인 트랜잭션을 수행 중인  $C_j$ 가 있는지 조사한다. 만약  $RX_{Ci} \cap RX_{Cj} \neq \emptyset$ 인  $RX_{Cj}$ 가 존재한다면  $C_i$ 에게 대기(Wait) 메시지를 보냄으로써  $RX_{Ci}$ 가

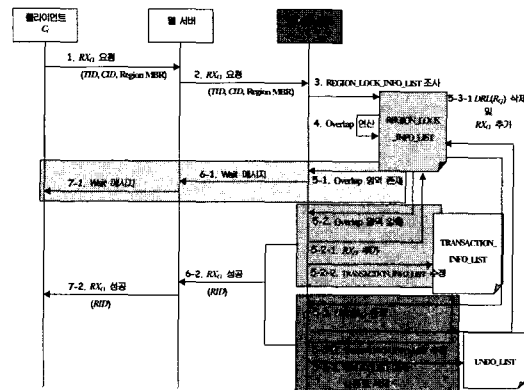


그림 12 RX 잠금 요청 프로토콜



$RX_{C_i} \neq \emptyset$ 인 영역에 대한 변경 작업으로 변경 충돌이 발생하지 않도록 한다. 만약,  $RX_{C_i} \cap RX_{C_j} \neq \emptyset$ 인  $RX_{C_j}$ 가 존재하지 않거나, 존재하지만  $DRL(R_{C_i})$ 이라면  $C_i$ 의  $RX_{C_i}$ 를 REGION\_LOCK\_INFO\_LIST에 저장한 후 영역 잠금 ID(RID)를  $C_i$ 에게 보냄으로써  $RX_{C_i}$  요청이 성공했다는 것을 알려준다. 위의 두 가지 경우 중에서  $DRL(R_{C_i})$ 이 존재하는 경우에는 잠금에 대한 권한을 UNDO\_LIST에 정보를 저장 중인  $C_j$ 로부터 가져오는 것이므로 이후에  $C_j$ 가 작업 요청을 했을 때 작업 권한을 주지 않도록 하기 위해서 UNDO\_LIST에서  $C_j$ 의 정보를 완전히 삭제한다.

**5.3 RX 잠금 해제**

$C_i$ 는 5.2절에서와 같이 RX 잠금 요청을 하고  $RX_{C_i}$ 에 대한 권한을 얻고 나면 해당 영역에 속해 있는 객체에 대해서 변경 작업을 하고 서버에 주기적으로 변경 작업을 한 객체를 저장한다. 그림 13은  $C_i$ 가  $RX_{C_i}$  요청을 하고 난 후, 서버에 저장되어 있는 변경 작업 객체를 실제 DB에 저장하려고 할 때의 프로토콜을 나타낸 것이다. 이 때,  $C_i$ 는 TID와 CID, 그리고 RID를 서버에 보낸다. 이 정보를 이용하여 서버는  $C_i$ 의 TRANSACTION\_INFO\_LIST의 잠금 상태에 대한 정보를 수정한 후 REGION\_LOCK\_INFO\_LIST에 저장되어 있는  $RX_{C_i}$  정보를 제거함으로써  $RX_{C_i} \cap RX_{C_j} \neq \emptyset$ 인  $RX_{C_j}$ 를 요청하는  $C_j$ 가 잠금 요청을 할 수 있도록 한다. 그리고 UPDATE\_OBJECT\_LIST에 저장되어 있는  $C_i$ 의 작업을 실제 DB에 저장한다. 모든 작업이 끝난 후에 서버는  $C_i$ 에게 RX

잠금 해제가 성공했다는 메시지를 보낸다.

**5.4 트랜잭션 종료**

$C_i$ 가 트랜잭션 종료 요청을 할 때 그림 14와 같이 서버 저장 정보 중 TRANSACTION\_INFO\_LIST에서  $C_i$ 의 트랜잭션 정보를 삭제한다. 서버에 저장되어 있는 데이터를 실제 DB에 저장 또는 취소하고  $RX_{C_i}$ 를 해제하기 때문에  $T_{C_i}$ 를 종료할 때는 서버에 저장되어 있는 나머지 트랜잭션 정보를 삭제하면 된다.

**5.5 클라이언트 오류 시 서버 작업**

이 논문에서는 4장에서 제시한 것과 같이  $C_i$ 가 서버에게 EXPIRE\_TIME 안에 주기적으로 ALIVE\_CLIENT<sub>i</sub> 메시지를 보냄으로써 서버가  $C_i$ 의 현재 상태를 예측할 수 있도록 하고 있다. 그림 15는 이와 같은 ALIVE\_CLIENT<sub>i</sub> 메시지가 EXPIRE\_TIME안에 전송이 되지 않았을 때 서버에서 처리하는 작업에 대해서 나타내고 있다. 이 때, 서버는  $C_i$ 가 오류가 발생했다고 판단을 하고 서버에 저장되어 있던  $C_i$ 의 정보에 대해서 다음과 같은 작업을 한다. 먼저, TRANSACTION\_INFO\_LIST에 저장되어 있던  $C_i$ 의 정보를 UNDO\_LIST에 옮겨 놓음으로써 나중에  $C_i$ 가 오류 복귀 후 다시 작업 요청을 하는 것이 가능하도록 한다. 그리고 REGION\_LOCK\_INFO\_LIST에서도  $RX_{C_i}$ 를  $DRL(R_{C_i})$ 로 전환을 함으로써 다른 클라이언트에 의해서  $DRL(R_{C_i})$ 가 삭제되지 않았을 때 나중에 다시 작업을 재개할 수 있도록 한다.

**5.6 클라이언트 오류 복귀 후 재작업 요청**

$C_i$ 의 오류에 의해서  $DRL(R_{C_i})$ 가 발생했을 때 서버에서는 그림 15와 같은 작업을 하여  $C_i$ 의 정보를 변경하였다. 그림 16은  $C_i$ 가 오류 복귀 후 재작업 요청을 했을 때 처리되는 작업에 대한 프로토콜을 나타내고 있다.

$C_i$ 가 오류 복귀 후 재작업 요청을 하기 위해서 CID와 함께 서버에 요청을 하면 서버는 일단 UNDO\_LIST와 TRANSACTION\_INFO\_LIST를 조사해서  $C_i$ 의 정보가 있는지 살펴본다. 만약 없다면  $C_i$ 의 트랜잭션 정보가 다른 클라이언트에 의해서 삭제된 경우이므로 이전 작업이 없다는 메시지를 보냄으로써  $C_i$ 의 재작업 요청에 대한 응답을 한다. 만약 UNDO\_LIST에  $C_i$ 의 트랜잭션 정보가 존재한다면 UNDO\_LIST에서  $C_i$ 의 정보를 TRANSACTION\_INFO\_LIST로 옮겨 놓는다. 그리고 REGION\_LOCK\_INFO\_LIST에 있는  $DRL(R_{C_i})$ 를  $RX_{C_i}$ 로 전환함으로써 정상적으로 트랜잭션을 수행할 수 있도록 한다. 마지막으로 서버에 저장되어 있던  $C_i$ 의 이전 작업 데이터를  $C_i$ 에게 전송을 해 줌으로써  $C_i$ 의 요청에 대한 응답을 한다.

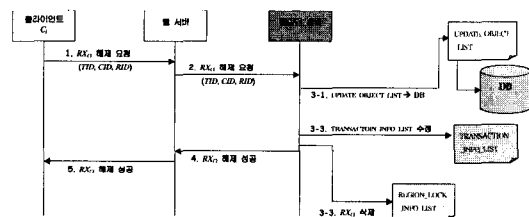


그림 13 RX 잠금 해제 프로토콜

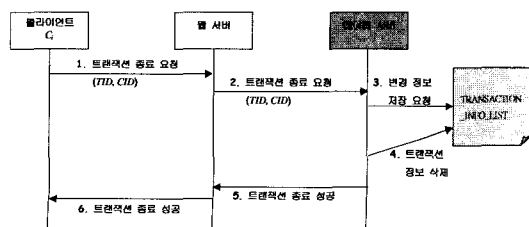


그림 14 트랜잭션 종료 프로토콜

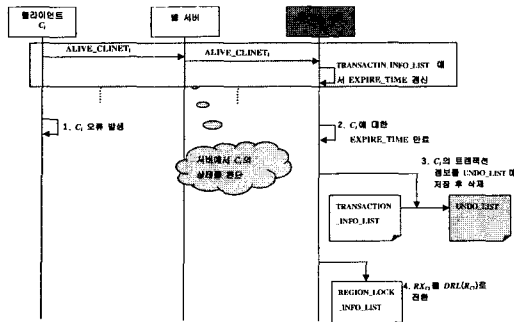


그림 15 클라이언트 오류 시 서버 작업 프로토콜

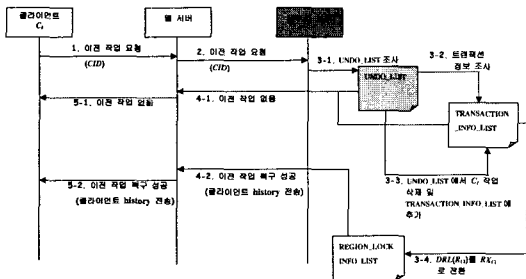
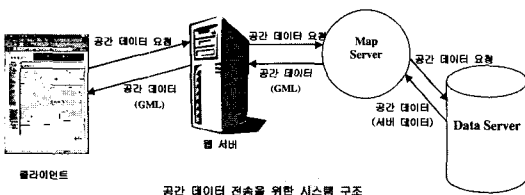


그림 16 클라이언트 오류 복구 후 재작업 요청 프로토콜

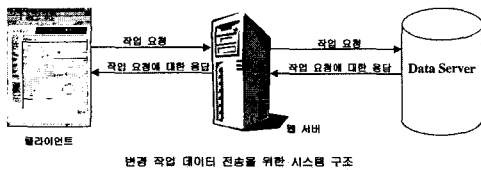
## 6. 시스템 구조 설계 및 구현

### 6.1 시스템 구조

공간 데이터 변경을 위한 시스템 구조는 그림 17과 같다. 웹 환경을 대상으로 하므로 웹 서버를 통하여 공간 데이터 서버와 통신을 하는 3-계층구조이다. 클라이언트 간의 협동작업은 하지 않는다고 가정하고 먼저 RX잠금을 요청한 클라이언트에게 잠금을 넘겨주는 방



공간 데이터 전송을 위한 시스템 구조



변경 작업 데이터 전송을 위한 시스템 구조

그림 17 공간 데이터 변경을 위한 시스템 구조

식을 사용한다. 그리고 잠금을 설정한 영역에 완전히 포함된 객체만 변경이 가능하다.

그림 17의 구조는 공간 데이터 전송 부분과 변경 작업 데이터 전송 부분으로 나누어진다. 공간 데이터 전송을 위한 시스템 구조에서의 서버는 [8]에서 제시하는 방법을 사용하여 맵 서버를 두으로써 클라이언트에게 서버의 이진 데이터를 GML 데이터로 전송을 할 수 있도록 한다.

### 6.2 시스템 설계

그림 18은 이 논문의 구현에 사용된 시스템 구조이다. 클라이언트는 Microsoft Windows 환경에서 ActiveX Control로 구현하였고 Microsoft Internet Explorer에서 ActiveX 모듈을 설치한 후 공간 데이터 변경 작업을 할 수 있도록 하였다. 서버는 현재 인터넷에서 지도 서비스를 제공하고 있는 CyberMapWorld GIS Server[2]를 사용하였다. 그림 18에서 구현한 클라이언트의 모듈은 모두 6개의 모듈로 구성되며 주요 모듈의 기능은 다음과 같다.

- GML 파싱 모듈(GML Parsing Manager): 맵 서버로부터 받은 GML 데이터를 파싱/Client data 로 변환.
- GML 디스플레이 모듈(GML Display Manager): ActiveX 컨트롤에 display.
- 트랜잭션 관리 모듈(Transaction Manager): 공간 데이터 변경 트랜잭션을 관리.
- 타임아웃 관리 모듈(EXPIRE\_TIME Manager): ALIVE\_CLIENT; 메시지 전송.

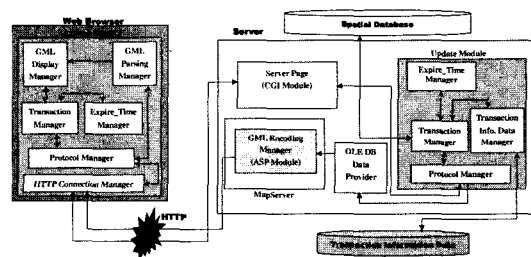


그림 18 시스템 구조

서버는 크게 맵 서버(Map Server)와 OLE DB 데이터 제공자, 서버 페이지(CGI), 그리고 변경 트랜잭션 관리 모듈(Update Module)로 구성되며 다음과 같은 역할을 한다.

- 맵 서버 : [8]에서 정의하고 있는 공간 데이터 전송 표준 중 XML 전송 표준(Data Case)에 따른

공간 데이터 전송을 담당. OLE DB 데이터 제공자로부터 서버의 데이터를 받아서 GML로 변환한 후 클라이언트에게 전송.

- **OLE DB 데이터 제공자** : 상호 운용성을 위해서 서버에 종속적인 이진 데이터를 ROWSET 형태로 변환하여 맵 서버에게 전해주는 역할.
- **변경 트랜잭션 관리 모듈**: 서버 데이터 변경 트랜잭션을 관리하는 부분으로 서버 페이지로부터 들어오는 클라이언트의 요청을 처리하고 응답을 해주는 역할.

**6.3 구현**

일반적으로 GIS에서 공간데이터의 변경 트랜잭션은 매우 긴 기간을 갖는 트랜잭션이다. 보통 수 분 이상이며, 수 십분 이상인 경우도 허다하다. 이 논문에서 고려

하고 있는 변경 연산 또한 긴 기간 트랜잭션이라 가정한다. 이와 같은 환경을 고려할 때 서버의 EXPIRE TIME은 너무 짧지 않는 것이 좋다. 또한 클라이언트가 보내는 메시지의 주기가 너무 짧으면 서버에 과부하를 초래한다. 그리고 변경 연산이 사람에 의해 수동으로 수행될 경우라도 클라이언트에서 자동적으로 주기적인 메시지를 보내도록 구현하였다. 그래서 이 논문에서 시스템 구현 시 Web의 접속시간을 고려하여 클라이언트가 보내는 메시지 주기는 5초로 책정하였으며, 서버의 EXPIRE TIME은 30초로 책정하였다.

그림 19는 구현된 공간 데이터 변경 모듈을 이용하여 두 클라이언트에서 공간 데이터를 변경하는 예를 보이고 있다. 여기서 두 클라이언트의 화면에 출력된 데이터

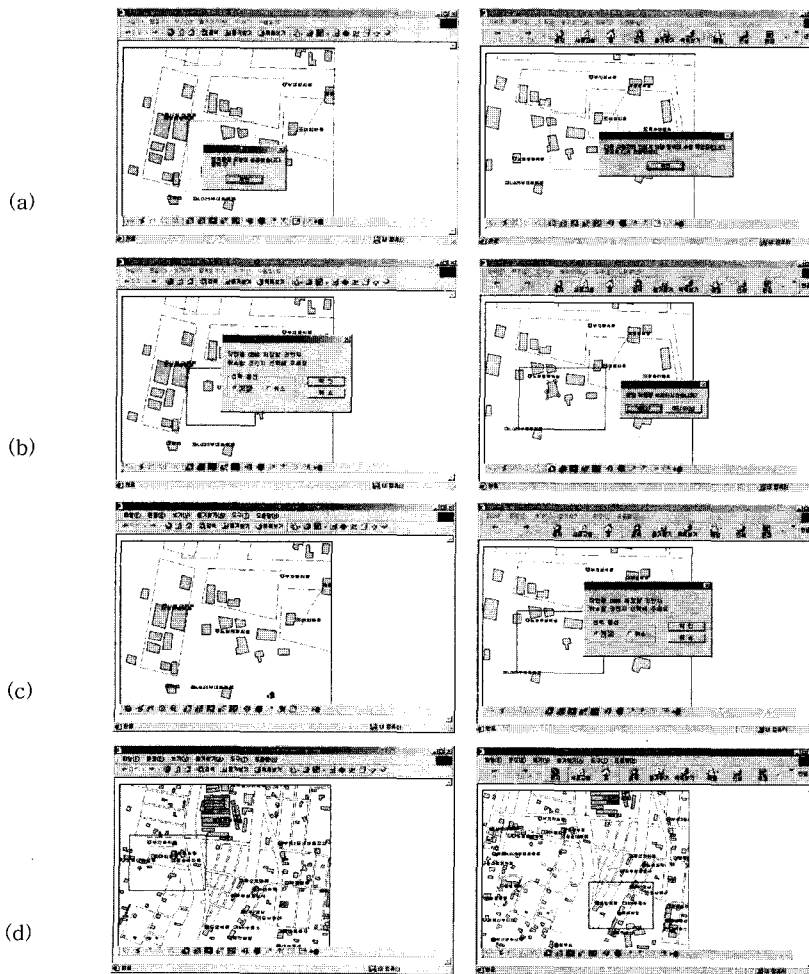


그림 19 구현된 공간 데이터 변경 모듈을 이용한 지도 수정 예

셋은 현재 CyberMapWorld GIS Server[2]에서 실제 사용하여 서비스를 하고 있는 전국 지도 데이터이다. 그리고 지도 화면 위의 사각형은 사용자에 의해 설정된 RX 영역 잠금의 영역이다. 그림 19의 (a)에서 클라이언트<sub>1</sub>은 작업할 영역을 지정하고 서버로 RX 영역 잠금을 요청하여 해당 영역에 대해서 잠금 권한을 얻고 있다. 클라이언트<sub>2</sub>는 서버로부터 잠금 권한을 얻을 수 없다는 메시지 통보를 받게 되고 클라이언트<sub>1</sub>이 RX 영역 잠금을 해제할 때까지 대기해야 한다. RX 영역 잠금을 설정한 영역에서 모든 변경 작업을 수행한 클라이언트<sub>1</sub>은 그림 19의 (b)와 같이 변경 작업을 수행한 데이터를 서버의 실제 DB에 저장할 것인지 취소할 것인지를 요구한 후 RX 영역 잠금에 대해 서버로 해제 요청을 한다. 서버에서 영역 해제 요청에 대한 응답을 받은 후 그림 19의 (c)와 같이 클라이언트의 지도 영역에는 변경 작업을 수행하고 난 후의 데이터가 출력된다. 한편, 클라이언트<sub>2</sub>는 클라이언트<sub>1</sub>이 잠금 영역을 해제하고 난 후 다시 RX 영역 잠금을 요청한 후 잠금 권한을 얻을 수 있다. 이 후 클라이언트<sub>1</sub>의 작업과 동일한 방식으로 영역에 포함되는 객체에 대해서 변경 작업을 수행하고 그림 19의 (c)와 같이 RX 영역 잠금을 해제하면서 변경 작업을 서버에 저장하게 된다. 겹치는 영역에 대해서 변경 작업을 수행한 클라이언트<sub>1</sub>과 클라이언트<sub>2</sub>는 이후에 다른 영역에 대해서 작업을 계속 수행하게 되는데 그림 19의 (d)는 두 클라이언트가 서로 겹치지 않는 영역에 대해서 작업을 할 때에는 서로 간섭을 받지 않고 변경 작업을 수행할 수 있다는 것을 보여주고 있다. 이와 같이 겹치지 않는 영역에 대해서 병렬 수정 작업을 하는 것을 허용함으로써 클라이언트 간의 동시성을 유지시켜 줄 수 있다.

## 7. 결론 및 향후 연구

웹 환경의 급속한 성장으로 현재 GIS 응용은 기존의 랜 환경에서의 서비스를 웹에서도 할 수 있게 함으로써 보다 쉽게 다양한 사용자들이 서비스를 제공 받도록 하고 있다. 웹 환경에서의 GIS 응용은 다양한 사용자 층에 대한 요구를 만족시키기 위하여 다양한 서비스 데이터를 필요로 하고, 이 데이터들은 사용자의 요구를 만족시키기 위하여 지속적으로 변경되어야 한다.

그러나 웹 환경에서 사용하는 HTTP프로토콜은 연결 해제와 상태 비유지를 가정한다. 이것은 긴 트랜잭션이 특징인 공간 데이터베이스 등의 응용에서 공간 데이터 변경 시 서버가 클라이언트의 상태를 능동적으로 인지할 수 없기 때문에 서버에 불필요한 잠금 정보가 남게

되어 클라이언트 간의 동시성 유지가 어렵고, 서버 데이터의 일관성 유지에도 문제가 발생하게 된다.

이 논문에서는 이러한 불필요한 잠금 정보를 휴면 영역 잠금이라고 정의를 하고 동시성 유지와 서버 데이터의 일관성 유지를 위한 해결책을 제시하였다. 즉, 서버 데이터의 일관성 유지를 위하여 RX 잠금을 정의하였다. 그리고 EXPIRE\_TIME안에 주기적인 메시지 전송을 통하여 휴면 영역 잠금 정보를 감지할 수 있도록 하여 휴면 영역 잠금 정보가 발생했을 때 서버에서 오류 회복 작업을 할 수 있도록 함으로써 다른 클라이언트가 휴면 영역 잠금과 겹치는 영역에 대한 RX 잠금을 요청할 때 잠금 권한을 넘겨 줄 수 있게 하였다. 이를 통하여 웹 환경에서 공간 데이터 변경 작업 시 클라이언트의 동시성 유지와 서버 데이터의 일관성 유지 문제를 해결하였다. 또한 이 논문에서는 제시된 방법을 기반으로 웹 환경에서 클라이언트가 공간 데이터를 변경할 때 발생할 수 있는 문제점을 해결하기 위한 프로토콜을 설계하고 있다. 그리고, 시스템 구현을 통하여 공간 데이터베이스의 웹 기반 변경 시 동시성 제어 및 일관성 유지를 검증하고 있다.

이 논문에서는 연결 해제와 상태 비유지를 가정한 웹 환경의 제약성 때문에 협동 작업을 잠금 정보를 이용하여 서버가 직접 일관성 제어를 하는 것으로 한정하고, 클라이언트 간의 메시지 교환에 의한 서버 데이터 일관성 유지에 대해서는 다루지 않고 있다. 그러므로, [9]에서 제시하고 있는 서버 데이터 변경 방법에 비해서 클라이언트 간의 동시성이 떨어지는 문제가 있다. 따라서, 향후 연구에서 웹 환경을 통한 서버 데이터 변경 작업을 할 때 클라이언트 간의 동시성을 높일 수 있는 방법에 대한 제시가 필요하다. 또한 제안한 방법에 대한 다양한 변경 연산의 종류, 클라이언트의 수 및 위치 변경 등에 따른 성능 평가를 수행하고자 한다.

## 참 고 문 헌

- [1] Kenneth M. Anderson, "Integrating Open Hypermedia Systems with the World Wide Web," Proceedings of ACM Hypertext '97, Southampton UK, pp.157-167, April 1997.
- [2] CyberMapWorld GIS Server Home Page, <http://www.cybermap.co.kr>
- [3] ESRI Internet Mapping Solutions Home Page, <http://www.esri.com/software/internetmaps>
- [4] Intergraph GeoMedia Web Map Demo Page, <http://www.intergraph.com/GWMDemo>
- [5] 안성우, 서영덕, 홍봉희, "웹 환경에서 공간 데이터 변

경 기업에 관한 연구”, 한국정보과학회 2000 가을 학술발표논문집, vol. 27, no. 2, pp.308-310, 2000.

[6] 박정하, 김동현, 홍봉희, “미들웨어: OLE DB를 기반으로 한 공간 데이터 변경 프로토콜의 설계 및 구현”, 한국정보과학회 2000 봄 학술발표논문집, vol. 27, no. 1, pp. 45-47, 2000.

[7] Open GIS Consortium Simple Features Specification for OLE/DB Revision 1.1, 1999, <http://www.opengis.org/techno/specs/99-050.pdf>

[8] Open GIS Consortium WWW Mapping SIG public page, <http://www.opengis.org/wwwmap>

[9] 신영상, 최진오, 조대수, 홍봉희, “클라이언트 변경 트랜잭션에서 공간 데이터의 동시성 및 일관성 제어”, 한국정보과학회 '99 가을 학술발표논문집, vol. 26, no. 2, pp.323-325, 1999.

[10] 최진오, 홍봉희, “분산된 지리정보시스템에서 새로운 잠금 기법을 이용한 중복된 공간 데이터의 변경 전파”, 한국정보과학회 논문지, vol. 26, no. 9, pp.1061-1072, 1999.

[11] Daniela Florescu, Alon Levy, Alberto Mendelzon, “Database Techniques for the World-Wide Web: A Survey,” ACM SIGMOD98, vol. 27, no. 3, pp.59-74, September 1998.

[12] Anders Kristensen, “Developing HTML Based Web Applications,” In Proc. of the First International Workshop on Web Engineering, April 1998

[13] Zhong-Ren Peng, “An Assessment of the Development of Internet GIS,” Proceedings of the ESRI User Conference, 1997.

[14] Zhong-Ren Peng, D. Nebert Douglas, “An Internet-Based GIS Data Access System,” Journal of Urban and Regional Information Systems, spring 1997.

[15] John P. Alexander, Valerie J. Warwick, “Writing GIS Applications for the WWW,” '97 ESRI International User Conference, 1997

[16] Jingshuang Yang, Gail E. Kaiser, “JPernLite: Extensible Transaction Services for the WWW,” IEEE TKDE '99, vol. 11, no. 4, pp.639-657, July/August 1999.

[17] Tim Berners-Lee, “The WWW Book: An attempt to describe most aspects of W3,” <http://www.w3c.org/History/1995/WWW/Paper/the-www/book.ps>

[18] Stefano Ceri, Giuseppe Pelagatti, “Distributed Databases-Principles & Systems,” McGraw-Hill, 1984.



안 성 우

1999년 2월 부산대학교 컴퓨터공학과 졸업(공학사). 2001년 2월 부산대학교 대학원 컴퓨터공학과 졸업(공학석사). 2001년 3월 ~ 현재 부산대학교 대학원 컴퓨터공학과 박사과정 재학중. 관심분야는 지리정보 시스템, 모바일 GIS, 웹 GIS, 공

간 색인



서 영 덕

1997년 2월 부산대학교 컴퓨터공학과 졸업(공학사). 1999년 2월 부산대학교 대학원 컴퓨터공학과 졸업(공학석사). 1999년 3월 ~ 현재 부산대학교 대학원 컴퓨터공학과 박사과정 재학중. 관심분야는 지리정보시스템, 병렬 지리정보시스템, 객체 지향 데이터베이스

체 지향 데이터베이스



김 진 덕

1993년 2월 부산대학교 컴퓨터공학과 졸업(공학사). 1995년 2월 부산대학교 대학원 컴퓨터공학과 졸업(공학석사). 2000년 8월 부산대학교 대학원 컴퓨터공학과 졸업(공학박사). 1998년 3월 ~ 2001.2 : 부산정보대학 정보통신계열 전임강사.

2001년 3월 ~ 현재 : 동의대학교 컴퓨터공학과 전임강사. 관심분야는 객체지향 DB, 지리정보시스템, 공간 질의어, 공간 색인, 모바일 데이터베이스



홍 봉 희

1982년 2월 서울대학교 컴퓨터공학과 졸업(공학사). 1984년 2월 서울대학교 대학원 컴퓨터공학과 졸업(공학석사). 1988년 8월 서울대학교 대학원 컴퓨터공학과 졸업(공학박사). 1987년 4월 ~ 현재 부산대학교 공과대학 컴퓨터공학과 교수. 현재 부산대학교 컴퓨터 및 정보통신 연구소 연구원, 부산대학교 대학원 GIS 학과 주임 교수

현재 부산대학교 컴퓨터 및 정보통신 연구소 연구원, 부산대학교 대학원 GIS 학과 주임 교수