

공간/비공간 데이터베이스 분산관리를 위한 Replicator 설계 및 구현

김봉석*, 최현영**, 민성기**

*고려대학교 컴퓨터정보통신대학원 정보통신공학과

**고려대학교 정보통신대학원 컴퓨터학과

{bskim*, neongas**, sgmin**}@korea.ac.kr

The Design and Implementation of Replicator for Distributed Management of Spatial/non_spatial Database

Bong-Seok Kim*, Young-Hyun Choi**, Sung-Gi Min**

*Graduate School of Computer and Information Technology, Korea University

**Dept of Computer Science and Engineering, Korea University

요 약

부동산문제등과 관련하여 지적(地籍) 및 용도지역지구 자료의 중요성이 부각되면서, 복잡한 법적 절차를 갖춘 시스템이 구축되고 있다. 그 중 한국토지정보시스템(KLIS)은 각 지자체별 지적 및 도시계획 관련 데이터 구축, 업무시스템 및 건축 인허가등의 민원발급, 부동산 관련 다양한 통계 등을 제공하고 있다. 그러나, 이 시스템이 설치되어 있는 시군구 중 일반구(區)를 포함한 시(市)의 경우는 법적인 편집 권한이 다른 기관에서 공통 데이터베이스의 편집에 따른 데이터 무결성 문제와 네트워크 장애에 따른 업무 지연 등의 업무 연속성 문제를 야기할 수 있다. 본 논문에서는 일반구가 포함된 시(市)에서의 지적도편집, 용도지역지구도 편집 등 이중의 기관에서 이루어지는 분산 환경 데이터베이스 편집에 따른 동기화의 문제점을 제기하고, 이러한 문제점을 보완할 수 있도록 트랜잭션 단위로 공간 및 속성 데이터의 복제를 담당하는 Replicator를 이용한 분산 데이터베이스 동기화 아키텍처를 제안하고, 동기화부분과 표준화부분, 트랜잭션관리부분으로 구성되어 있는 Replicator의 구성요소를 설계 및 구현하고 Replicator를 이용한 분산 데이터베이스의 변경부분을 미반영 데이터베이스에 반영하는 처리 절차와 처리 후 결과를 제시하여, 지적 및 용도지역관련 데이터를 두 개 이상의 기관에서 동시에 편집한 내용이 실시간으로 두 기관의 분산 데이터베이스에 변경 분이 반영되어 모두 동기화 되어 있는 것을 공간데이터 편집기로 확인해 볼 수 있었다.

1. 서론

토지정보화사업은 토지행정업무가 내포하고 있는 문제를 합리적으로 해결하고 한정된 토지자원의 이용을 극대화함으로써, 토지에서 발생하는 소득과 부가 공평하게 분배될 수 있도록 지원하는 사업이다. 이 사업의 기반은 지적도와 용도지역지구도의 전산화에 있으며, 현재 230여개 시군구에 토지종합정보망 시스템이 설치되어 있다.

지적도와 용도지역지구도는 GIS(Geographic Information System)의 성격을 가지고 있는 공간(Spatial)데이터와 속성(Attribute)데이터로 이루어져 있으며, GIS 수치지도는 위치좌표(Georeference)를 가지고 있고,

위상(Topology) 인식이 가능하도록 되어 있기 때문에 아주 다양하고 복잡한 공간분석을 행할 수 있다. 일반 데이터와 저장체계가 다른 구조인 Spatial 데이터는 포인트(Point), 라인(Line), 폴리곤(Polygon)으로 이루어져 있으며, 피쳐(Feature) 테이블과 그리드(Grid)테이블에 저장된다. 이런 공간정보는 속성테이블과 밀접하게 연결되어 있으므로, 공간데이터를 복제하기 위해서는 Spatial 데이터와 Non_spatial 데이터를 하나의 트랜잭션으로 관리하여 복제가 될 수 있도록 구성하여야 한다[1].

특히 지적도와 용도지역지구도의 편집은 공간데이터에 대한 특정지역에 대한 수정이 일어나기도 하지만 이동정리 등으로 한개 시군구가 한 번에 수정이 일

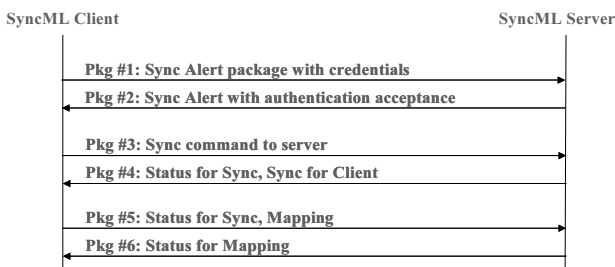
어날 수 있으므로 무엇보다도 트랜잭션의 관리가 중요하다 할 수 있다. 그리고 용량이 큰 트랜잭션을 저장하여, 정확하게 변경전의 서버로 이동시키기 위한 기술이 반영되어 설계 되어야한다.

본 논문의 구성은 다음과 같다. 2장은 분산 환경에서의 데이터 동기화를 위한 표준인 SyncML 기술과 OGIS(Open GIS)공간데이터의 표준화인 GML(Geographic Markup Language)과 Replicator의 분산객체 전달을 위한 기술인 CORBA에 대하여 알아보고, 3장에서는 현재의 일반구에 사용하고 있는 아키텍처의 문제점과 분산 환경에서 보완할 수 있는 분산 아키텍처를 제안한다. 4장에서는 Replicator 구성요소를 설계하고 구현방안 및 제약사항등을 기술한다, 5장에서는 결론 및 향후의 연구에 대해서 논의한다[2].

2. 공간/비공간데이터 동기화를 위한 표준

2.1 Replicator 의 동기화를 위한 SyncML

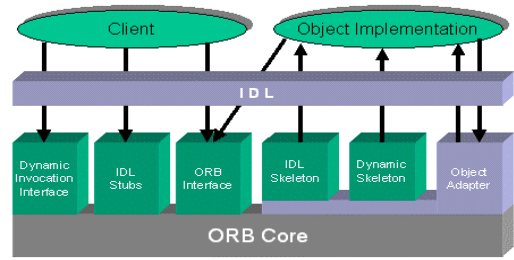
SyncML은 여러 플랫폼 환경에서의 원격 데이터 동기화를 위한 프로토콜로서 기존의 Mobile환경 등의 사용으로 주로 연구되어 지고 있다. 일반 구에서 공간 데이터 동기화를 위한 Replicator를 설계하기 위하여 변경된 데이터를 SyncML로 전달할 수 있는데 이런 Replicator 사이의 프로토콜을 단순화시키면 (그림 1)과 같다[3].



(그림 1) Synchronization protocol

2.2 분산 환경 통신을 위한 CORBA 통신사용

CORBA(Common Object Request Broker Architecture)는 서로 다른 분산 응용 프로그램 간을 연결하는 소프트웨어 버스의 역할을 하는 모듈로서 OMA 환경에서 가장 핵심이 되는 부분이라 할 수 있으며 현재 CORBA 2.0이 발표되어 있다. CORBA는 분산 컴퓨팅과 객체지향 기술을 하나로 합친 표준 아키텍처로서 CORBA의 구조는 (그림 2)와 같으며 그 구성 요소로는 IDL Stub, 동적 호출 인터페이스(Dynamic Invocation Interface), 코어와 인터페이

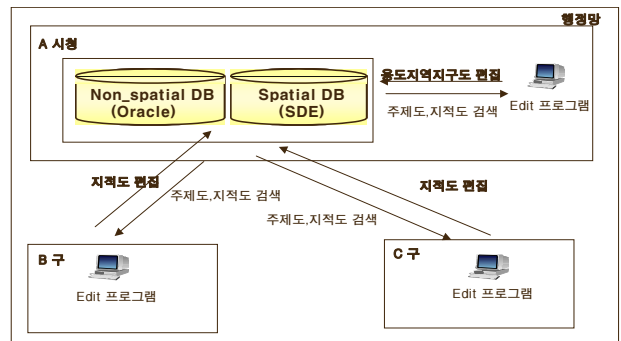


(그림 2) CORBA의 구조

스로 이루어진 ORB(Object Request Broker), 객체 어댑터(Object Adapter), Skeleton 등이 있다[3].

3. 업무개선을 위한 분산 아키텍처 제안

현재 일반구인 경우의 업무처리 아키텍처는 (그림 3)과 같으며, 공통의 데이터베이스를 분산 환경에서 다른 권한을 부여받아 데이터를 변경함으로써 데이터의 무결성에 문제가 야기될 수 있고, B 구청과 C 구청의 경우는 최신의 데이터를 가지고 있지 않음으로 시청과의 통신이 두절되었을 경우에 업무의 연속성을 해칠 수 있는 위험성이 존재한다. 그리고 도시계획 및 지적관련 중요한 데이터가 시청 한곳에서만 관리되어 지고 있으므로, 분산 환경으로의 백업 등에 대한 대안도 필요하다.



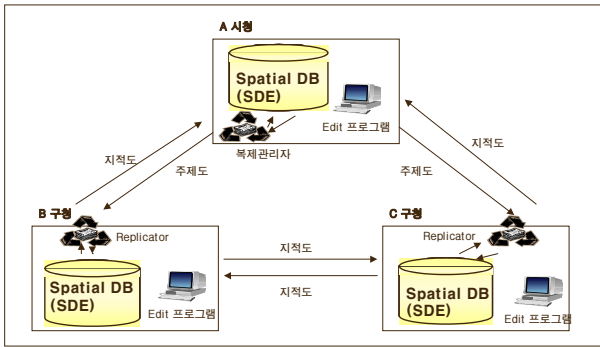
(그림 3) 토지종합정보망 일반구인 경우 아키텍처

각 구청은 별도의 서버와 소프트웨어 장비가 있음에도 불구하고 관리의 어려움 때문에 시청의 데이터를 공유하여 사용하고 있으며, 별도의 백업은 정기적인 수작업 백업으로 이루어지고 있다.

전주시와 같이 일반 구를 포함하는 시청에서는 용도지역지구 관련 레이어 입력 및 편집을 담당하고, 덕진구와 완산구에서는 지적관련 연속지적도와 편집지적도의 입력 및 편집 작업을 하고 있다. 그리고 이런 200개 이상의 용도지역지구 및 지적관련 레이어의 공간 및 속성 데이터는 시청의 데이터베이스 서버에 저장되어 있다. 이를 이용하여, 토지이용계획

확인원등의 민원업무를 처리하고 있다. 현재 공통 데이터베이스의 작업은 먼저 접속한 사용자가 작업하려는 지역에 Lock을 걸어 동시 편집이 될 수 없도록 작업제한을 두고 있는 어플리케이션을 사용함으로써 해결하고 있다[4].

이런 문제를 해결하기 위하여 (그림 4)와 같이 분산 DB와 에이전트를 이용한 분산아키텍처를 제안한다. 공통의 DB를 분산 환경에서 편집함으로써 생기는 문제를 해결할 수 있다. A시청, B 구청, C 구청이 각각 동기화된 데이터베이스를 가지고, 자신의 데이터만을 수정하고 에이전트가 실시간으로 공간 및 속성데이터의 변경된 부분만 트랜잭션단위로 패키징하여 복제를 수행함으로써, 데이터 무결성, 실시간 백업, 업무연속성을 증가시킬 수 있다. (그림 4)는 이런 분산 환경에서의 개선된 아키텍처의 경우이다.



(그림 4) 분산환경에서의 개선된 아키텍처

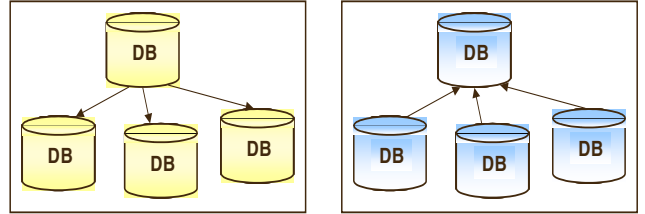
물론 이런 분산 환경에서 100%의 데이터를 처리하기는 힘든 부분이 있다. 특히 지적 및 용도지역지구 관련 데이터는 개인의 재산권이 변동될 수 있는 중요한 데이터임을 감안할 때 분산 환경에서의 처리가 오히려 부작용이 될 수도 있다. 따라서 정확한 예외 처리 및 트랜잭션 처리가 무엇보다도 중요하며, 업무의 정의에 따라 대규모 이동정리와 같은 1개의 트랜잭션에 2만 건 이상의 공간데이터 편집이 포함되어 있는 용량이 큰 트랜잭션인 경우에는 별도의 옵션에 따른 기존의 수작업 처리가 필요하다.

4. Replicator의 설계 및 구현

4.1 Replicator의 부하를 줄이기 위한 2-Way Sync

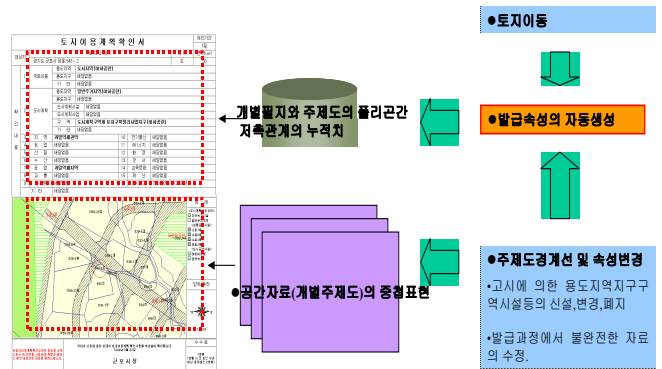
위의 개선된 아키텍처에서 Replicator는 분산 환경에서의 실시간 데이터 동기화를 위하여 (그림 5)와 같이 1:N 동기화 및 N:1 동기화가 필요하다. 예를 들어 시청에서 데이터가 변경된 경우 B 구청과 C 구청에 변경된 데이터의 복제가 이루어져야 하며, B구청의 데이터가 변경된 경우에도 동기화 작업이 이루어

어져야 한다. 따라서 변경이 발생된 소스 측 Replicator와 변경된 내용을 수용해야 할 타겟 측 Replicator의 역할을 먼저 결정해야 한다[6].



(그림 5) 1:N 방식 과 N:1 방식

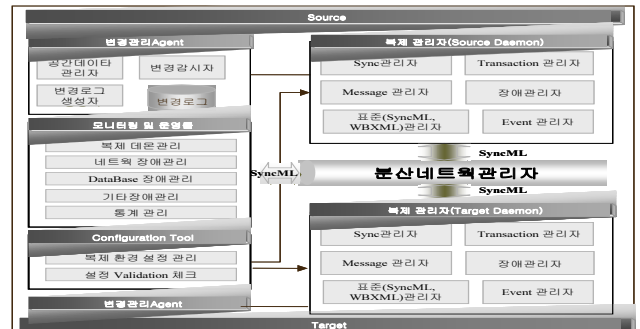
현재 일반구와 같은 상태에서는 1:N과 N:1 방식을 혼용하면 2-Way Sync와 같은 효과로 3곳의 데이터베이스를 동기화 시킬 수 있다.



(그림 6) 시청에서의 용도지역지구도 업무처리절차

(그림 6)은 시청에서 실제 업무처리 절차로서, 용도지역지구도의 편집 작업을 나타낸 것으로 시청에서 공간데이터의 편집 작업이 이루어지면 시청의 해당 레이어에 대한 변경사항이 트리거에 의하여 변경로그로 저장되어 지고, 그 변경로그는 B, C 구청 Replicator의 변경감시자에 의하여 감지되며 B, C 구청은 시청의 변경내용을 확인하여 속성데이터는 XML형태로 공간데이터는 GML형태로 저장되어 있는 데이터를 패킷 전송하여 시청과 데이터를 Sync 한다.

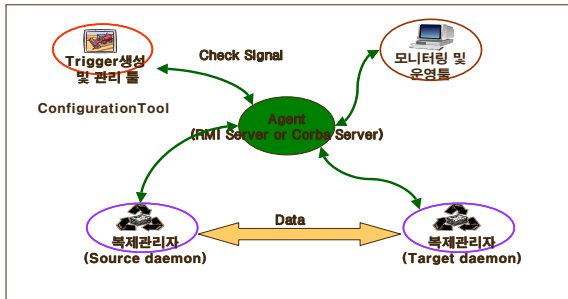
4.2 Replicator 구성요소 및 처리절차



(그림 7) Replicator 구성도

Replicator의 구성요소는 (그림 7)과 같이 먼저 데이터 변경이 발생했을 때 데이터에 대한 변경을 감지하여, 변경이 발생했음을 감지하는 Sync 관리자, 장애가 발생했을 때 장애사항을 감지하는 장애 관리자, 변경된 데이터를 XML, GML로 Packaging 하는 표준관리자, 공간 및 속성데이터를 하나로 전송하는 트랜잭션관리자로 크게 나눌 수 있다[6].

다음의 (그림 8)은 Replicator의 처리 흐름도로서 아래의 순서로 복제 작업이 진행된다.

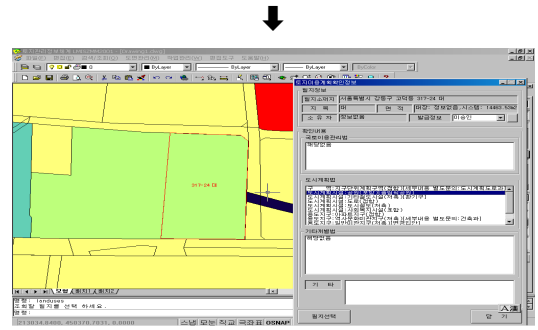
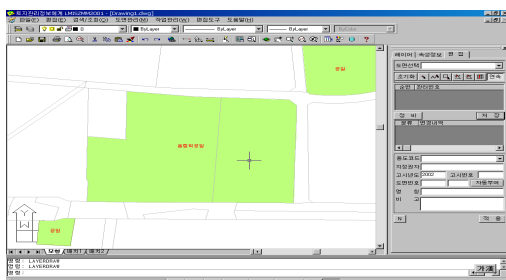


(그림 8) Replicator 처리 흐름도

1. 공간 및 속성데이터의 편집부분을 트랜잭션 단위로 트리거를 이용하여 변경부분에 대한 로그정보를 표준포맷 형태로 생성한다.
2. 데이터 변경이 발생한 소스측 Replicator 데몬 (Daemon)은 타겟측에게 변경사실을 통보한다.
3. Target측 데몬은 변경부분의 내용을 자신의 DB 버전 정보와 비교한다.
4. 버전 변경 시 변경된 속성 데이터와 공간데이터 부분을 트랜잭션 단위로 각각의 DB에 반영한다.
5. 버전테이블을 업데이트하여 소스측과 타겟측의 정보를 동기화 시킨다.

4.3 Replicator를 이용한 분산 환경에서의 복제수행

(그림 9)는 Replicator 데몬을 시청과 B 구청에 설치한 후 시청에서 특정지역의 용도지역지구를 공원지역으로 변경한 후(그림 9 상단) 그 결과를 B 구청에서 확인 것으로 환경설정에서 Timer를 1분으로 맞추어 놓으면 거의 실시간으로 반영되는 것을 확인할 수 있다.



(그림 9) Replicator를 이용한 분산 환경에서의 데이터편집 및 결과

5. 결론 및 향후 연구

본 연구는 지적도와 용도지역지구도와 같은 공간 및 속성데이터의 처리에 있어 공통 데이터베이스를 사용함으로써 생기는 아키텍처의 문제점을 제시하고 분산 데이터베이스를 사용함으로써 백업 및 데이터 무결성, 업무 연속성 등을 고려한 아키텍처를 제안하였고, 분산 데이터 복제를 위한 Replicator를 설계하여 제안된 아키텍처를 구현하였다. 위에서 언급한 바와 같이 지적도와 용도지역지구도는 재산권과 결부된 중요한 데이터로서 데이터의 무결성이 편리성보다 중요하다고 할 수 있다. 분산 환경은 장애처리의 우수성이 담보되어야 하며, 정확한 트랜잭션처리 특히 공간데이터의 특성상 자주 발생하는 용량이 큰 트랜잭션에 대한 처리 등이 수반되어야 하며, 이런 처리들은 지자체의 업무특성에 맞추어 공통DB사용 부분과 분산DB 사용부분을 분리하여 업무에 적용할 수 있다.

참고 문헌

- [1] ESRI , "Spatial Data Standards and GIS Interoperability",<http://www.esri.com/library/whitepapers/pdfs/spatial-data-standards.pdf>
- [2] Ron Lake, "Geography Mark-Up Language (GML) (Foundation for the Geo-Web", John Wiley & Sons Inc 2004
- [3] SyncML Specification, Version 1.0 , <http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html#V101>
- [4] David Maguire, Michael Batty, "GIS, Spatial Analysis, and Modeling" First Ed. ESRI Press
- [5] Steve Muench, " Building Oracle XML Applications", O'Reilly 2000
- [6] Philippe Rigaux, "Spatial Databases: With Application to GIS", 2nd Ed, Morgan Kaufmann