

버전 관리 기능을 지원하는 모바일 시설물 관리 시스템의 구현

The Implementation of Mobile Facility Management System Supporting Version Management Facility

이영걸¹⁾ · 배상호²⁾

Lee, Young Geol · Bae, Sang Ho

Abstract

Mobile facility management system is a on-site management tool to collect and update various facility data. But there is a chance to make a data inconsistency between lot of concurrent workers. And facility management system requires to keep records about work log within a certain period of time. This paper describes the design and implementation of mobile facility management system, which provide efficient data synchronization mechanism using version management technique on large facility data in concurrent work environment. And proposed system is able to keep workers update log persistently.

Keywords : Mobile GIS, Spatial database, Version management, Facility management

초 록

모바일 시설물 관리 시스템은 시설물 데이터에 대한 이동 관리를 지원하여 현장에서 발생하는 다양한 정보를 실시간으로 수집, 갱신할 수 있게 하지만, 해당 지역의 다수 작업자간의 동시 작업으로 인해 데이터의 불일치성이 발생할 수 있으며, 또한 다른 작업자의 갱신 내용을 실시간으로 반영하기 어렵다는 단점을 가지고 있다. 또한 시설물 데이터에 대한 작업 이력은 시설물 관리 시스템에서 매우 중요한 의사 결정 정보로 시스템에서 영속적으로 관리할 필요가 있다. 따라서 본 논문에서는 기존의 버전 관리 기법을 이용하여 다른 동시 작업자와 서버에게 효율적으로 데이터를 동기화할 수 있으며, 작업자의 작업 내역을 영속적으로 관리할 수 있는 모바일 시설물 관리 시스템을 구현한다.

핵심어 : 모바일 지리정보시스템, 공간데이터베이스, 버전 관리, 시설물 관리

1. 서 론

휴대용 컴퓨터의 대중화와 무선 통신 기술의 급속한 발전은 사용자에게 어디서나 필요한 컴퓨팅 환경을 제공할 수 있게 하였고, 현재 다양한 사용자의 요구를 수용하기 위하여 모바일 컴퓨팅에 관한 연구가 활발히 진행되고 있으며 특히 공간적인 특성을 정보화하여 처리하는 지리정보시스템(GIS:Geographic Information System) 분야에서는 모바일 컴퓨팅을 접목하여 데이터의 영속성을 실시간으로 보장해주는 연구가 활발하게 진행되고 있다(Evaggelia Pitoura; 1998, G. H. Forman; 1992, Rafael Alonso; 1993).

이와 같이 모바일 컴퓨팅 기술과 지리정보시스템의 한 분야인 시설물 관리 시스템이 결합되어 이동성과 실시간성을 향상한 정보 시스템을 모바일 시설물 관리 시스템이라고 한다. 모바일 시설물 관리 시스템은 어디서나 원하는 데이터를 조작할 수 있는 장점을 제공하지만, 무선 환경의 데이터 전송 제약으로 인해 방대한 규모의 데이터에 대한 서버와 다른 사용자간의 일치성을 보장하기 어렵다는 단점을 가지고 있다. 이로 인해 대부분의 모바일 시설물 관리 시스템에서는 일괄 동기화 기법이나 단방향 갱신 데이터 전송 방법을 사용하고 있는 실정이다. 이로 인해 다수 작업자와 서버 간에 데이터의 불일치성이 발생하며,

1) 정희원 · 대림대학 컴퓨터정보계열 부교수(E-mail:yglee@daelim.ac.kr)

2) 연결저자 · 대림대학 토목환경과 조교수(E-mail:shbae@daelim.ac.kr)

데이터의 후처리 작업에 많은 비용과 시간이 소비된다. 이러한 문제점을 해결하기 위해 본 논문에서는 기존의 버전 관리 기법을 이용하여 실시간으로 데이터의 연속성을 유지할 수 있는 모바일 시설물 관리 시스템을 구현한다. 구현 시스템은 작업자나 서버 간의 실시간 갱신 내역을 별도의 버전 데이터로 관리함으로써 실제 대용량의 데이터베이스와는 별도로 실시간으로 동기화할 수 있게 해준다. 이러한 버전 데이터들은 데이터의 연속성을 보장해주는 역할 이외에도 작업 내역을 저장하여 이력 정보로도 활용할 수 있다는 장점을 가지고 있다. 본 논문의 구성은 다음과 같다. 2장에서는 기존의 버전 관리 기법과 모바일 시설물 관리 시스템에 대해 살펴보고, 3장에서는 개발 시스템의 시스템 구조와 데이터 구조에 대해 소개하고, 4장에서는 개발 시스템의 구현 방안과 산출물을 소개하며, 마지막으로 5장에서 결론을 내린다.

2. 관련 연구

2.1 버전 관리 기법

버전 관리 기법은 분산화된 지역 데이터베이스와 서버 데이터베이스의 일치성을 보장하기 위한 가장 일반적인 기법으로 알려져 있다(김희택; 2005, Nnwell, R.G; 1994). 버전은 서버 데이터베이스와 논리적으로 동일한 시간 종속적인 데이터베이스를 말한다. 분산된 작업자들은 각자 특정 버전의 데이터에 대해 갱신 업무를 수행하게 되며 다른 작업자들은 특정 작업자의 갱신 내용을 알 수 없다. 그러므로 작업자들의 모든 데이터들은 모두 물리적으로 일치하지 않으며 해당 작업자들은 자신의 버전 데이터에 대한 갱신 데이터만을 저장하고 있다. 작업자들에 의해 갱신된 데이터는 서버로 전송되어 기록되는 게시(post) 연산과 다시 게시된 변경 데이터를 다른 작업자에게 전송하는 병합(merge) 연산을 통해 데이터 간의 일치성을 보장받을 수 있게 된다. 일반적으로 버전 관리 기법에서는 낙관적인 동시성 제어 기법을 사용하며, 작업자간의 중복되는 갱신에 대해서는 병합 연산 시 타임스탬프를 이용하여 해결한다. 이와 같은 버전 관리 기법은 개별 작업자의 갱신 연산에 대해 별도의 추가적인 시간이 요구되지 않아 분산화된 지리정보시스템, 데이터베이스 관리 시스템 분야에서 데이터의 일치성을 보장하는 표준 기법으로 자리 잡고 있다.

버전 관리 기법의 구현 방안으로는 데이터베이스 관리 시스템이 가지는 버전 관리 기법을 그대로 사용하는 심층

버전 관리(deep version management) 기법과 별도의 응용 프로그램 내부에 자체적인 버전 관리 기능 모듈을 포함하는 지층 버전 관리(shallow version management) 기법이 있다. 전자의 경우 시스템의 구현이 용이하며, 개발자에게 데이터의 독립성을 보장한다는 장점을 가지고 있지만 순수한 버전 차원의 데이터 만을 지원하여 이력과 관련된 별도의 제어 기능을 추가하기 어렵다는 단점을 가지고 있다. 후자의 경우 응용 시스템 구현이 복잡하지만 이력과 관련된 별도의 제어 기능을 추가하기가 용이하다는 장점을 가지고 있다(김희택; 2005). 본 논문에서는 지층 버전 관리 기법을 이용하여 모바일 시설물 관리 시스템을 구현한다.

2.2 모바일 시설물 관리 시스템

모바일 시설물 관리 시스템은 휴대폰, PDA, 태블릿 컴퓨터, 노트북 컴퓨터 등 휴대용 단말기를 이용하여 시공간적 제약 없이 시설물과 관련된 정보를 수집, 갱신, 분석, 출력할 수 있는 컴퓨터 응용 시스템을 말한다(한승희; 2002, Hughes; 2000). 이것은 지형공간에 관한 모든 공간 데이터와 관련된 속성 정보인 비공간 데이터를 통합하여 관리할 수 있는 효율적인 공간 연산 및 위상 연산의 제공과 공간 제약 조건인 무결성 유지 등의 기능이 필수적으로 요구된다.

모바일 시설물 관리 시스템은 GIS 기술, 모바일 컴퓨팅 기술, GPS 기술이 통합되어 연동하는 정보 기술의 집합체로 기존의 클라이언트-서버 기반의 시설물 관리 시스템에서 클라이언트가 제공하던 역할을 이동형 단말기를 통해 수행함으로써, 이동성, 위치 독립성, 실시간성을 제공하고 있다(Hughes; 2000). 그림 1은 모바일 시설물 관리 시스템의 시스템 개념도를 나타내고 있다.

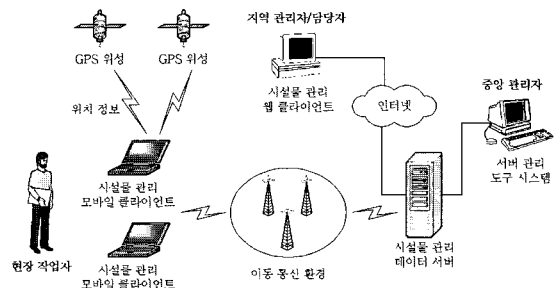


그림 1. 모바일 시설물 관리 시스템의 개념도

3. 모바일 시설물 관리 시스템의 설계

3.1 시스템 구조

개발 시스템은 크게 태블릿 컴퓨터를 이용하는 시설물 관리 모바일 클라이언트, 웹을 이용하는 시설물 관리 웹 클라이언트 그리고 서버 관리 도구 시스템으로 구성된다. 모바일 클라이언트는 GPS RTK 장치와 연동하여 위치 정보를 수치 지도에 사상하거나 속성 정보를 갱신할 수 있는 기능을 갖는 현장 작업자용 시스템이다. 모바일 클라이언트는 수치 지도와 속성 정보를 출력하고 검색할 수 있는 데이터 브라우저(data browser), 로컬 데이터에 대한 분석 및 질의 처리 기능을 담당하는 질의 처리기(query processor), 갱신 데이터를 서버와 동기시키는 기능을 담당하는 버전 관리자(version manager), 서버와 이동 통신을 기반으로 통신 기능을 담당하는 통신 관리자(communication manager), 특정 버전의 로컬 데이터에 대한 입출력 기능을 담당하는 데이터 관리자(data manager)로 구성된다. 웹 클라이언트는 웹 브라우저를 이용하여 서버의 데이터를 출력, 검색하거나 지역의 시설물 관리 업무를 처리하는 기능을 갖는 지역관리자나 담당자용 시스템이다. 웹 클라이언트는 수치 지도와 속성 정보를 출력하고 검색할 수 있는 데이터 브라우저, 서버의 데이터에 대한 분석 및 질의 처리 기능을 담당하는 질의 처리기, 지역의 시설물

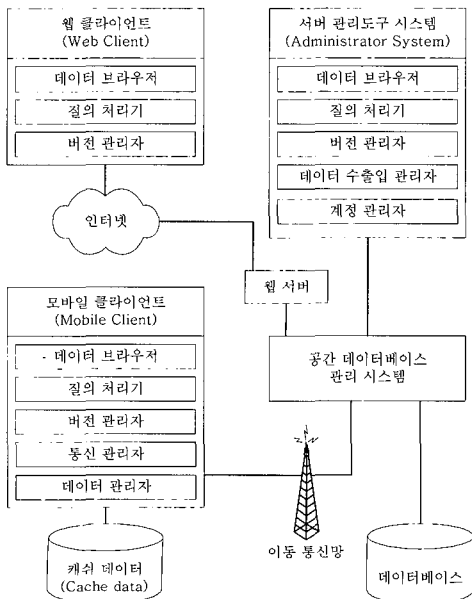


그림 2. 모바일 시설물 관리 시스템의 구조

관리 업무를 처리하는 기능으로 구성된다. 웹 클라이언트의 모든 프로세스는 서버를 기반으로 이루어진다. 서버 관리 도구 시스템은 서버의 데이터에 출력, 검색 기능과 서버의 데이터, 사용자 자원에 대한 관리 기능을 갖는 중앙 관리자용 시스템이다. 서버 관리 도구 시스템은 수치 지도와 속성 정보를 출력하고 검색할 수 있는 데이터 브라우저, 서버 데이터에 대한 통계, 분석 및 질의 처리 기능을 담당하는 질의 처리기, 지역이나 현장 작업자로부터 전송된 데이터를 반영 또는 취소 기능을 담당하는 버전 관리자, 작업 데이터를 표준 공통 호환 형식으로 수입하고 버전별로 수출하는 기능을 갖는 수출입 관리(import/export manager), 시설물 관리 시스템의 사용자 계정에 대한 생성, 수정하는 기능을 갖는 계정 관리자 기능으로 구성된다. 모바일 시설물 관리 시스템의 시스템의 구조는 그림 2와 같다.

3.2 버전 데이터를 포함하는 시설물 객체의 구조

본 논문에서 제안하는 시설물 객체는 실제계에 존재하는 시설물 정보가 가지는 공간, 비공간, 이력 정보를 포함하는 추상화된 정보 구조이다. 시설물 객체는 지형, 위치, 위상 정보를 포함하는 공간 데이터, 시설물이 가지는 속성 정보를 포함하는 비공간 데이터, 시설물 객체의 이력 정보의 버전을 포함하는 이력 데이터로 구성된다. 또한 시설물 객체는 분류된 시설물 객체들의 공통적인 특성(feature)을 별도의 식별자로 참조할 수 있도록 특성 정보를 별도로 관리한다(배상호 ; 2005). 이러한 특성 정보는 공통적으로 분류된 시설물에 대해 기하 형식, 레이어 명칭, 적용 연산 등을 사상하여 객체의 무결성을 유지할 수 있는 자료구조로 사용된다. 뿐만 아니라 시설물 객체가 표현하는 데이터의 규모를 최소화할 수 있게 해준다. 다음의 그림 3은 시설물 객체의 개념적 구조를 보여준다.

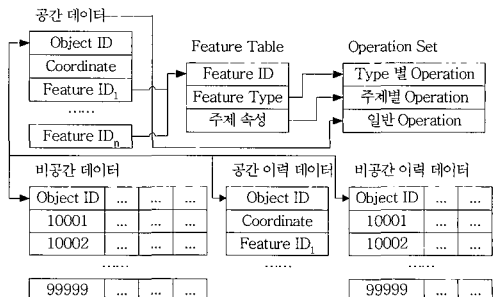


그림 3. 시설물 객체의 개념적 구조

갱신 이력이 저장되는 시설물 객체는 다중 버전 기법을 통하여 서버와 분산된 클라이언트간의 데이터를 동기화할 수 있다. 버전은 클라이언트의 갱신 연산에 의해서 생성된 데이터의 복사본을 의미한다. 서버로 전송된 데이터는 여러 개의 버전이 존재하며 검색 연산과 갱신 연산은 타임스탬프를 이용해 각각 다른 버전을 사용할 수 있다. 결국 다수의 버전에 대한 병합을 통해 데이터의 일치성을 보장하며, 연산의 충돌을 제거하여 모바일 클라이언트가 데이터를 점유하고 기다리는 블록킹 상황을 제거할 수 있다(김희택 ; 2005). 버전은 공간 데이터와 비공간 데이터를 별도의 테이블로 구분하여 생성 및 관리한다. 이러한 방법은 기존의 공간 데이터베이스 관리 시스템이 속성 데이터나 공간 데이터만 변경되어도 공간 객체 전체에 대한 버전을 저장하는 낭비와 시간 소비를 방지할 수 있다. 버전 테이블은 기존의 공간 데이터나 속성 테이블이 가지고 있는 속성 이외에 버전 식별자, 타임스탬프, 연산 구분, 계정 식별자 속성을 포함한다. 버전 식별자는 클라이언트의 복사본 데이터가 가지는 데이터베이스의 버전 식별자이다 서버로 데이터를 전송할 때 함께 보내지며 병합 연산 및 게시 연산의 기준이 된다. 타임스탬프는 갱신 연산이 반영된 절대적 시간으로 시스템 내에서 유일한 값을 가진다(이영걸 ; 2007). 분산된 시스템의 시간 차이를 극복하기 위해 서버의 버전 테이블에 반영된 절대적 시간으로 통일한다. 연산 구분은 적용된 갱신 연산의 구분을 나타내며 데이터의 복구 연산이나 이력을 확인하는데 사용된다. 계정 식별자는 갱신 연산을 수행한 계정의 식별자이다. 다음의 그림 4는 버전 테이블의 개념적 구조이다.

버전 테이블의 이력 정보는 시설물의 갱신 이력을 영속적으로 관리할 수 있게 해준다. 버전 테이블과 실제 공간 또는 비공간 테이블의 조인으로 발생한 테이블은 관리자에게 작업자를 기준으로 생성, 변경, 삭제 이력을 검색할 수 있게 해준다. 시설물 객체는 현재 상태의 객체 이외에 버전 테이블을 참조하여 삭제되거나 변경전의 공간 데이터에 대해 별도의 특성값(feature value)을 부여해 다양한 의사 결정에 사용된다.

객체 식별자	공간데이터/ 비공간데이터	버전 식별자	타임 스탬프	연산 구분	계정 식별자
--------	------------------	--------	--------	-------	--------

그림 4. 버전 테이블의 개념적 구조

3.3 시설물 객체의 조작 연산

시설물 객체의 연산은 로컬 데이터와 서버 데이터 간의 일치성을 보장할 수 있도록 구현되며 갱신된 데이터가 다른 클라이언트에게도 반영되어 전체 데이터가 논리적으로 일관되도록 유지한다.

검색 연산은 로컬 데이터의 버전이 최신인지를 서버의 버전 테이블을 참조하여 확인하고 최신이 아닌 경우 서버의 버전 테이블로부터 로컬 버전 이후의 데이터들을 전송받아 데이터를 동기화 시킨 후 검색 연산을 수행한다. 만약 로컬 버전 이후의 버전 데이터가 존재하여 동기화가 수행되면 전송된 버전 데이터를 로컬의 버전 테이블에 기록(logging)된다.

갱신 연산이 클라이언트에서 발생하면 갱신 데이터 정보가 서버로 전송되어 버전 테이블에 저장 된다. 이때 버전 테이블에 동기화하지 않은 동일한 객체에 대한 버전 데이터가 존재한다면 클라이언트에게 동기화 요청 메시지를 전송한다. 이때 클라이언트 사용자는 해당 객체에 대한 동기화 정보를 확인하고 갱신 연산을 재수행할지 여부를 다시 서버에 전송한다. 재수행이 요청되면 다시 버전 테이블을 확인하고 갱신 정보를 서버의 버전 테이블에 기록한다. 서버는 버전 데이터를 해석하여 실제 데이터베이스에 반영한다. 갱신 연산의 알고리즘은 다음과 같다.

표 1. 시설물 객체에 대한 갱신 연산 알고리즘

```

Algorithm for updata operation at client-side
Input
R : Updated Record
RTL : Recent Timestamp of Local version
output
commit or abort

Begin
extendedR = Add_additional_information(R);
ServerConnction();
retCode = SendToServer(extendedR,RTL);
if( retCode==NeedSync ) {
    While ( MergeFromServer() ) {
        extendedR = FetchVersion();
        retCode = CheckConflict(extendedR);
        if ( retCode==Conflict )
            if ( UserConfirm()==0 ) return abort;
        InsertIntoLocalVersionTable(extendedR);
    } // end while
} else {
    InsertIntoLocalVersionTable(extendedR);
}
return commit;
End

```

3.4 시설물 객체의 동기화

동기화 연산은 분산화된 시설물 데이터베이스의 복사본들과 서버의 시설물 데이터베이스의 일치성을 보장하기 위한 일련의 연산이다. 모바일 클라이언트와 서버간의 동기화를 시도하는 시점으로는 주기적 또는 일정 시간에 서버 데이터베이스와 동기화 시켜주는 방법과 모바일 클라이언트 또는 서버의 데이터베이스 갱신 연산이 발생했을 경우 동기화를 시켜주는 방법이 있다(서동환 ; 2006). 전자의 방법은 특정 시간동안 데이터의 불일치성이 발생하는 단점이 있으며, 후자와 같은 실시간 동기화의 경우 모바일 클라이언트의 갱신에 대해 다른 클라이언트와 서버가 매번 함께 갱신하는데서 발생하는 시간적 비용이 크다는 단점을 가지고 있다. 제안 시스템은 후자의 실시간 동기화 방법을 수동적으로 방법으로 변형하여 구현하였다. 수동적 동기화 방법은 모바일 클라이언트에서 질의가 발생하는 경우 동기화 여부를 판단하게 한다. 다시 말해 동기화가 요구되는 시점을 실제 갱신 시점이 아닌 질의 요구 시점을 사용하게 된다. 이와 같은 방법은 데이터의 연속성을 보장하면서 동시에 데이터 통신 비용을 최소화할 수 있다는 장점을 가지고 있다.

동기화를 하는 방법으로는 전체 데이터베이스를 복제하는 방법과 변경된 로그(log)만을 동기화시켜주는 방법이 있다. 전자의 방법은 동기화에 발생하는 데이터 전송량이 방대하여 많은 시간 비용이 요구된다. 반면 후자의 방법은 변경된 내역인 로그만을 동기화하므로 시간 비용을 감소시킬 수 있다. 개발 시스템에서는 버전 테이블을 서버와 클라이언트에 로그로 저장한다. 동기화가 요구되는 시점에서 해당 버전 테이블 간의 데이터를 동기화 통해 데이터베이스 간의 일치성을 보장한다.

3.5 시설물 객체의 이력 관련 연산

시설물 객체에 이력 관련 연산은 시설물을 생성, 갱신, 삭제한 절대적인 시간을 기반으로 처리되는 일련의 질의 연산이다. 속성값을 비교하는 검색 질의, 공간 위상 관계를 이용하는 공간 검색 질의와 함께 시간 위상을 이용한 시간 검색 질의를 가능하게 한다. 이력 관련 연산은 절대 시간간의 위상 관계를 조건으로 사용하여 결과를 반환한다. 개발 시스템에서는 Allen이 정의한 13가지 시간 위상 관계를 구현하여 사용한다(Allen, F.J ; 1983). 표 2는 Allen이 제시한 위상 관계 중 7개의 시간 연산자의 의미를 나타낸다. 나머지 6개의 시간 관계는 시간 간격 A, B

표 2. 시간 위상 관계

위상 관계	의 미
A Before B	A.To < B.From 이면 참을 반환
A Equals B	(A.From=B.From) and (A.To=B.To) 이면 참을 반환
A Meets B	A.To=B.From 이면 참을 반환
A Overlaps B	(A.To<=B.To) and (A.From<=B.From) 이면 참을 반환
A During B	(A.To>B.To) and (A.From>B.From) 이면 참을 반환
A Starts B	(A.To=B.To) and (A.From<B.From) 이면 참을 반환
A Finishes B	(A.To>B.To) and (A.From=B.From) 이면 참을 반환

사이의 역관계로 성립하므로 별도로 기술하지 않는다.

4. 모바일 시설물 관리 시스템의 구현 및 적용

4.1 구현 환경

개발 시스템의 구현 환경은 윈도우즈 서버에 탑재된 공간 데이터베이스 관리 시스템과 연동하기 위해 모바일 클라이언트의 운영체제는 윈도우즈 태블릿 에디션을 사용하였으며 개발도구로는 윈도우즈 비주얼 C++를 사용하였다. 표 3은 시스템 구축을 위해 개발환경에 사용된 개발도구, 통신 방식, 그리고 GPS 장비 등을 나타낸 것이다.

4.2 모바일 클라이언트의 구현

데이터 브라우저는 검색된 공간 데이터와 비공간 데이터를 사용자의 요구에 의해 화면에 출력하고 사용자의 입력 요구를 받아들이는 기능을 담당한다. 데이터 브라우저는 윈도우의 다양한 인터페이스 컨트롤들을 이용하여 사

표 3. 개발 환경

종 류	내 용
하드웨어(모바일)	P-1510 후지쯔 태블릿 컴퓨터
운영체제(모바일)	Microsoft Windows XP-Tablet
개발도구(모바일)	Microsoft Visual Studio
공간 DBMS	SANTrek-KGI
DB 인터페이스	OLE DB
이동통신방식	CDMA-TCP/IP
GPS 통신방식	시리얼 통신(RS 232C)
GPS	GARMIN GPS16 HVS

용자와 대화하는 기능을 제공하며, 시설물 및 지형도를 레이어 별로 가시성, 편집성, 선택성을 제어할 수 있게 하는 레이어 관리자, 시설물 및 공간 데이터를 화면 좌표계로 변환하여 확대, 축소 출력할 수 있는 지도 브라우저, 비공간 데이터인 테이블 데이터를 출력하는 기능을 제공하는 테이블 브라우저로 구성된다.

질의 처리기는 사용자의 질의 요구를 분석하여 공간 데이터와 비공간 데이터에 대한 물리적 연산 절차로 변환하여 수행하는 역할을 한다. 이 과정에서 응답 시간을 최소화하기 위한 최적화 과정을 수행하며 서버와 통신하여 동기화하는 버전 관리자와 연동한다.

데이터 관리자는 디스크 장치의 데이터와 버전 데이터를 메모리 버퍼에 저장하여 질의 처리기가 효과적으로 사용할 수 있도록 하는 기능을 제공한다. 데이터 관리자는 데이터와 분리된 버전 데이터를 단일한 테이블인 것처럼 사용하게 하는 조인 뷰(join view)의 형태로 메모리 버퍼에 적재한다. 다음의 그림 5는 모바일 클라이언트의 시스템 구조이다.

4.3 시스템 적용

개발한 시스템의 적용 실험을 위해 기존의 수치지도도를 활용할 수 있는 경기도 안양시 지역을 선정하였다. 특히 본 적용 실험을 위해서는 유지 보수 및 교체가 빈번한 시

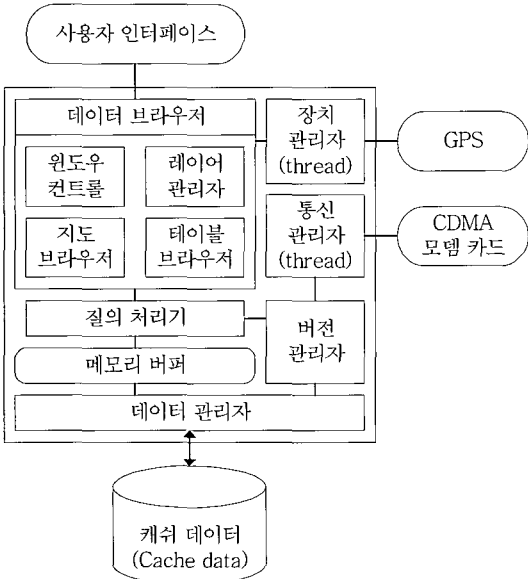


그림 5. 모바일 클라이언트의 구성 요소

설물 중 하나로 통신 시설물을 선정하였다. 관리 대상이 되는 통신 시설물은 분기국사, 공동구, 지하관로, 맨홀, 전주, 단자함, 전진배치국사, 광기지국, 광단국, 지선, 중간 절체반, 접지, 시내케이블, 광케이블, 접속점, 분배함 등을 레이어로 분리하여 관리하였다.

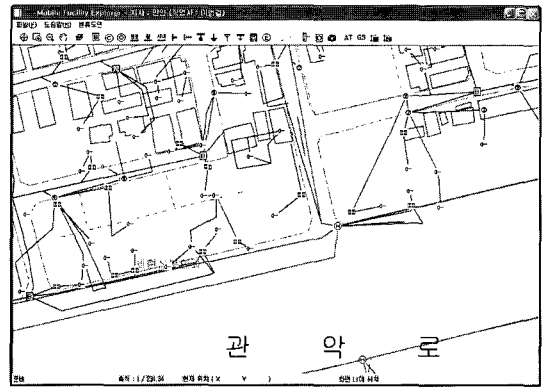


그림 6. 작업 영역의 시설물 정보의 도시

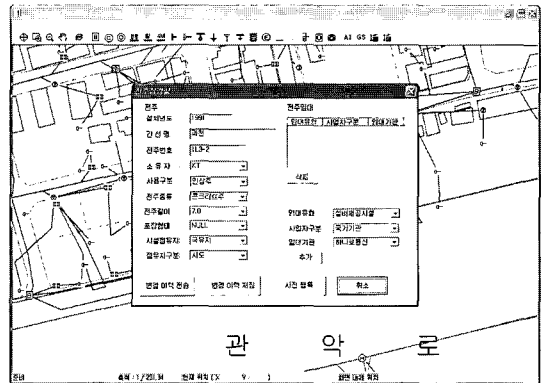


그림 7. 특정 시설물의 속성 정보 보기

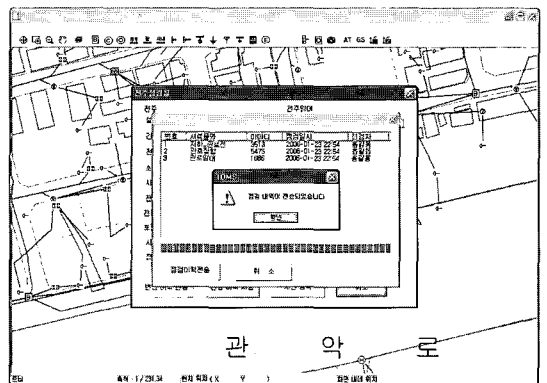


그림 8. 갱신 정보 전송 후 완료 화면

참고문헌

그림 6은 작업 지역의 선택으로서, 현재 작업해야할 시설물 객체 있으며, 시설물의 특성에 따라 선분 집합과 심볼로 표시하고 있다. 그림 7은 특정 시설물 객체의 속성 정보를 보여주는 화면이고, 그림 8은 갱신 정보를 서버로 전송하고 동기화를 완료한 화면이다. 이와 같이, 본 연구에서 개발한 모바일 시설물 관리 시스템을 이용하여 현장에서 작업을 보다 원활하게 수행해 갈 수 있었으며, 현장 작업자가 별도로 갱신이나 작업 내역을 보고하는 과정을 거치지 않아 작업 시간의 단축 뿐만아니라 관리자의 의사 결정 정보에 정확성을 실시간으로 제공할 수 있게 해준다.

5. 결 론

본 논문에서는 버전 관리 기법을 이용하여 분산된 시설물 데이터베이스에 대한 일치성을 보장하며, 동시에 이력 데이터를 연속적으로 관리할 수 있는 모바일 시설물 관리 시스템을 구현하였다. 특히 대용량의 공간 데이터베이스에 대한 물리적 동기 시간 비용을 최소화할 수 있도록 공간 데이터와 비공간 데이터를 분리하여 버전을 관리하며, 데이터베이스 전체가 아닌 갱신 이력인 로그 정보에 대해서만 동기화를 수행한다. 또한 시설물 관리에서 필수적인 작업 이력에 대한 시공간적인 분석을 지원하여 시설물 관리 업무를 실시간으로 원활하게 수행할 수 있다. 향후에는 시설물 설계 시스템과 통합할 수 있는 미래 시간의 시설물 객체를 표현할 수 있는 자료 구조를 확장하고, 시설 계획에 대한 분석 질의를 처리할 수 있는 질의 처리기에 대한 설계가 이루어져야 할 것이다.

- 김희택 (2005), 이동 객체의 논 블로킹 검색을 위한 공간 레코드 다중 버전 관리 기법, 석사학위논문, 인하대학교, pp. 8-27.
- 배상호, 이영걸 (2005), 토타스태이션을 이용한 PDA기반의 맵핑시스템 개발, 한국지형공간정보학회지, 한국지형공간정보학회, 제 13권, 제 4호, pp. 11-17.
- 서동환 (2006), SyncML을 이용한 공간 데이터 동기화 시스템 구현, 석사학위논문, 서울산업대학교, pp. 15-25.
- 이영걸 (2007), 데이터베이스 이론 및 SQL 서버 2005 실습, pp. 56-77.
- 한승희, 이용욱, 이형석 (200), 시설물관리를 위한 Mobile GIS 솔루션 활용, 한국측량학회지, 한국측량학회, 제 20권, 제 1호, pp. 59-68.
- Allen, F. J. (1983), Maintaining Knowledge About Temporal intervals, *Communications of ACM* 26, pp. 832-843.
- Evaggelia Pitoura and George Samaras (1998), *Data Management for Mobile Computing*, Klueer Academic Publishers, pp. 5-15.
- G. H. Forman and J. Zahorjan (1992), The challenge of mobile computing, *IEEE Computer*, pp. 8-20.
- Hughes, J. R. (2000), GIS goes Mobile, *GeoWorld*, Vol. 13, No. 6, pp. 8-15.
- Nnwell, R. G. and Batty, P. M. (1994), GIS databases are different, *Proceedings of AM/FM Conference XVII*, pp. 279-288.
- Rafael Alonso and Henry F. Korth (1993), Database System Issues in Nomadic Computing, In *Proceedings of the ACM International Conference on Management of Data*, pp. 388-392.

(접수일 2007. 2. 23, 심사일 2007. 4. 23, 심사완료일 2007. 4. 23)