

# 실시간 맵 업데이트를 위한 모바일 공간 DBMS 개발

## The Development of Mobile Spatial DBMS for Real-time Map Air Update

민경욱\*, 안경환, 김주완, 진성일

Kyoung-Wook Min\*, Kyoung-Hwan An, Ju-Wan Kim, Sung-Il Jin

\*한국전자통신연구원 텔레매틱스연구부, 충남대학교

{kwmin92\*, mobileguru, juwan}@etri.re.kr, {sijin}@cnu.ac.kr

### 요약

현재 상용화되어 서비스되고 있는 내비게이션의 맵 데이터는 유선 인터넷을 이용하여 업데이트한다. 내비게이션을 위한 모바일 단말에서의 전국 데이터는 보통 1GB ~ 2GB의 크기를 가지며 이 데이터를 유선 인터넷을 이용하여 업데이트하는데 수십 분이 소요된다. 하지만 실제 변경된 데이터는 일부분에 해당하지만 전체 맵 데이터를 업데이트함으로써 많은 비용을 낭비하게 된다. 이에 본 연구에서는 모바일 단말에서의 부분 맵 데이터를 업데이트하기 위하여 모바일 공간 데이터베이스관리시스템 개발과 이를 이용한 내비게이션 시스템 개발에 대해서 살펴보고자 한다.

### 1. 서론

기존 내비게이션 시스템의 맵 데이터는 상세정도에 따라 다르지만 시간이 지날수록 데이터의 크기가 방대해져 보통 수 기가바이트에 달한다. 이러한 맵 데이터가 변경된 경우 지금까지는 유선 인터넷을 이용하여 전체 맵 데이터를 업데이트 하여왔다. 실제 변경된 데이터는 일부분임에도 불구하고 전체 데이터를 업데이트하는데에는 많은 비용을 초래하게 된다. 지금까지 모바일 단말에서 저장/관리되는 맵 데이터를 부분 업데이트 하지 못하는 이유는 모바일 단말의 리소스 제한으로 인하여 맵 데이터가 단말에 최적화 된 형태 (PSF: Physical Storage Format)로 저장/관리되어 부분 업데이트가 불가능하였다. 더군다나, 내비게이션에서 가장 중요한 도로 네트워크 데이터의 경우, 노드와 링크가 복잡한 관계를 가지고 있어 하나의 링크

크 또는 노드가 변경될 경우 다수 데이터가 변경되어야 하며 이럴 경우 최적화 되어 있는 파일의 구조를 유지할 수가 없다. 이에, 부분 업데이트가 가능한 데이터 구조 및 시스템에 대한 연구가 진행되기 시작하였으며 아직까지는 상용화되어 서비스되고 있는 시스템이 국내에는 존재하지 않는다. 내비게이션 응용에서 맵 데이터의 부분 업데이트가 가능하도록 하기 위한 요구사항은 크게 두 가지이다. 첫 번째는 변경된 부분 배경, 검색용 POI 데이터, 경로탐색을 위한 도로 네트워크 데이터가 업데이트 가능하여야 하며, 두 번째로 업데이트가 가능한 시스템에서 맵 데이터 검색 및 경로 검색 성능을 보장하여야 한다. 이에 본 연구에서는 위의 두 가지 조건을 만족하는 모바일 공간 DBMS(Database Management System)의 개발과 그 성능에 대해서 평가하고자 한다.

## 2. 관련연구

[1]의 ActMap 프로젝트에서는 2004년부터 부분 맵 업데이트를 위한 교환 포맷 및 시스템에 대한 연구 프로젝트를 진행하여 프로토타입 수준의 시스템을 만들었으나 상용화에 실패하였다. [2]의 혼다 인턴비 내비게이션에서는 주요 도로에 대해서만 실시간 업데이트를 지원하며 변경 및 삭제는 지원하지 않는다. [3]에서는 기존 내비게이션용 맵 데이터가 PSF 구조를 따르고 있음으로 해서 발생하는 단점인 복잡한 데이터 구조와 관리 및 상호 호환성 문제점을 극복하고자 SQL-Lite에 공간질의 처리 기능을 확장하여 시스템을 개발하였다. 하지만, 이 연구에서는 메쉬(Mesh)단위의 맵 데이터를 업데이트하기 때문에 본 연구의 객체단위의 업데이트하는 방식에 비해 업데이트해야 하는 데이터의 크기가 커지는 단점이 있으며, SQL-Lite DBMS내부에 공간 데이터 처리 기능을 추가한 것이 아니라 상위에서 이러한 기능을 추가 구현하여 본 연구에 비해 공간 데이터 검색 성능이 느리며, 특히 경로탐색의 경우 노드/링크의 반복적인 접근을 위하여 query-by-query 검색을 수행하기 때문에 그 성능이 많이 떨어진다. [4]에서는 모바일 디바이스에서의 저장 매체로 가장 많이 활용되는 Flash Memory의 특성을 반영한 모바일 관계형 DBMS를 개발하였다. 본 연구에서는 관계형 DBMS 뿐만 아니라 추가적으로 공간 데이터 모델[5]을 지원하는 모바일 DBMS를 개발하였다.

## 3. 시스템 구성

본 연구에서 개발한 시스템은 크게 서버(MAUS: Map Air Update Server)와 단말(FUNs: Flash-aware Ubiquitous Navigation System)으로 구성된다. 본 논문에서는 단말(이하 FUNs) 시스템 기술에 대해서 논의하도록 한다. FUNs는 그림1과 같이 FUNs Core Engine과 Data Loader, Administration Tool, FUNs Navi App로 구성

된다. Data Loader는 원본 데이터(SHP 포맷)를 FUNs 데이터베이스 포맷으로 변환하는 기능을 제공한다. Administration Tool은 데이터 테이블 브라우징, 데이터베이스, 테이블스페이스, 테이블등을 생성/삭제할 수 있는 관리 툴이다. FUNs Navi App는 FUNs Core Engine을 이용하여 구현한 내비게이션 응용프로그램이다.

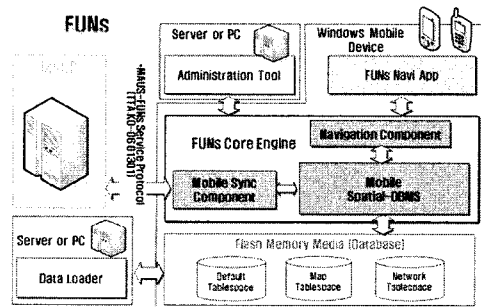


그림 2 FUNs 시스템 구성

FUNs Core Engine의 모바일 Sync 컴포넌트는 MAUS와 무선통신을 이용하여 부분 맵 데이터를 전송받은 후 데이터베이스에 반영하는 기능을 수행한다. 이 때 MAUS와 FUNs의 교환 포맷은 표준[6]을 이용한다. 내비게이션 Component는 Mobile Spatial-DBMS의 API를 이용하여 경로 검색/맵매칭 등의 API를 제공한다. 이 때, 빠른 경로 검색을 위하여 별도 API를 이용하며, 경로탐색 방법은 [7,8]에서의 Dijkstra 알고리즘과 멀티레벨 경로탐색 방법을 참조하여 구현하였다. 내비게이션을 위하여 저장 관리하는 데이터는 메타 정보를 관리하는 default 테이블스페이스 파일과 맵 표출용 배경 데이터 및 검색용 POI 데이터 등의 테이블이 저장되어 있는 map 테이블스페이스 파일, 내비게이션을 위한 도로네트워크 등의 데이터 테이블들을 포함 있는 network 테이블스페이스 파일이 있다. 모바일 Spatial-DBMS에 대해서는 4장에서 자세히 살펴보도록 한다.

#### 4. 모바일 공간 DBMS

FUNs 모바일 공간 DBMS는 테이블스페이스 단위의 파일을 관리한다. 테이블스페이스에는 다수 테이블이 포함된다. 특징은 표1과 같다. 응용과의 인터페이스는 SQL-like한 API를 제공하고 있으며, 속성 데이터 검색의 경우는 일반 DBMS에서 제공하는 검색기능이외에 한글 초성검색기능을 지원한다. 공간 데이터 검색은 window 검색, k-nearest neighbor 검색, [5]의 공간 위상관계 연산 기능을 제공한다. 빠른 경로 검색을 위하여서는 물리적 레코드 ID 검색을 제공한다.

표1. FUNs 모바일 DBMS 특징

Function		FUNs
SQL/API		C/C++ API
DDL		Create / Drop Table, Index
DML		Insert, Update, Delete
Search		Attribute Search (Initial Sound, Match) Spatial Search (Relational Search, Windows, K-NN) Network Search (RID Search)
Transaction, Recovery		Begin, Commit, Redo, Undo
Object		Table (General Table, Network Table) Access Path (B+-Tree, GRID, R*-Tree)
Data Type	General	char, varchar, int, uint, short, double, float, binary, date, time, blob
	Spatial	point, lineString, polygon
Storage Media		Disk, Flash-Memory
Supported OS		Windows, WinCE (Windows Mobile, Pocket PC)

일반적으로 모바일 DBMS에서는 다중 사용자 지원을 위한 concurrency control 기능을 제공하지 않으며, FUNs 모바일 공간 DBMS에서는 트랜잭션 처리를 위하여 logging 기법을 이용한 회복 기능을 제공한다. 빠른 데이터 접근을 위한 색인으로는 속성 검색을 위한 B+-Tree와 공간 검색을 위한 GRID, R\*-Tree를 지원한다. 공간데이터 타입으로는 [5]의 Point, Line String, Polygon을 지원한다.

#### 5. 실험

본 연구에서 개발한 FUNs를 이용하여 실제 내비게이션을 구현하였다. 사용된 데이터는 전국 데이터로써, 6개 레벨로 구성된 배경 표출용 데이터(약 400M - 68개 테

이블), 검색 POI 데이터(약 300M - 50만 건 레코드), 네트워크 데이터(약 400M, 9개 테이블, Node - 60만건 레코드, Link - 70만건 레코드)로 구성되어 총 데이터 크기는 약 1.1Gbyte이다. 여기서 네트워크 데이터는 전체 네트워크 데이터를 포함하는 1레벨 데이터와 일반적으로 제외한 네트워크 데이터로 구성된 2레벨 데이터를 이용하여 경로탐색을 수행한다. 테스트 단말기는 LG KC1 단말기이며, CPU 800 Mhz, 128M DRAM, 2G 플래시메모리를 이용하였다.

그림 2는 FUNs Core Engine을 이용하여 개발한 내비게이션 응용 화면이다. 실제 동탄 신도시 배경 데이터 업데이트 후 화면 표출과, 도로네트워크 업데이트 후 경로검색을 수행한 화면 표출을 보여주고 있다.

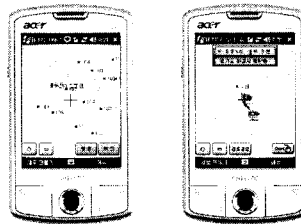


그림 2 FUNs를 이용한 내비게이션 응용

본 연구에서는 무선통신(CDMA, Wibro)을 이용하여 실제 맵 데이터 업데이트를 테스트하였다. 업데이트 데이터는 광주지역 도로 네트워크 데이터와 동탄 신도시 도로네트워크 및 배경데이터를 이용하여 업데이트를 테스트하였다. 광주데이터의 경우 Node/Link 삭제 49건, 추가 264건, 변경 89건으로 총 업데이트 데이터 크기는 40Kbyte이다. 이때 무선통신을 이용한 전송시간을 제외한 DB 업데이트 시간은 12초이다. 그리고 동탄 신도시 업데이트 데이터의 경우 Node/Link 삭제 107건, 추가 1,103건, 변경 124건으로 총 100Kbyte이며 업데이트 시간은 20초, 배경 공간데이터의 경우 추가 객체 2,532건이며 데이터

크기는 130Kbyte, 업데이트 시간은 8초가 소요되었다. 네트워크 업데이트의 경우 서버에서 전송하는 데이터는 하위레벨만 전송하고 FUNs에서 상위 2레벨을 생성하는 부분과 빠른 경로검색을 위한, 별도 정보 생성을 위한 로직이 추가되기 때문에 배경 객체 업데이트에 비해 시간이 많이 소요된다.

그림 3에서는 FUNs에서의 경로탐색 성능을 보여주고 있다. 일반 모바일 DBMS를 이용할 경우 Node/Link의 연속 검색을 위하여 query-by-query를 수행하기 때문에 성능이 느리다. 본 연구에서는 경로탐색을 위하여 별도 물리적 레코드 ID정보를 구성하고 있기 때문에 현재 서비스되고 있는 일반 내비게이션과 성능이 유사하다. 그림 3의 성능 테스트는 직선거리 10Km 미만, 10~50Km, 50~100Km, 100Km 이상에 해당하는 각각 1,000개의 전국 POI 데이터를 무작위로 추출하여 경로탐색 평균 시간을 나타낸 것이다. 장거리(100Km 이상)의 경우 경로탐색 시간이 약 7초 정도의 성능을 보인다.

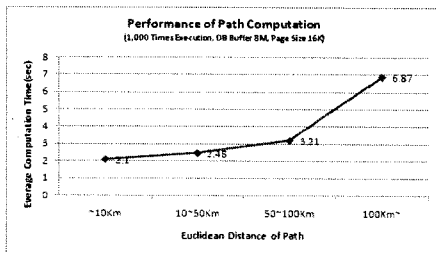


그림 3 FUNs의 경로 검색 성능

## 6. 결론

본 연구에서 개발한 FUNs는 내비게이션 뿐만 아니라, LBS (Location-based service), 텔레매틱스, 모바일 GIS 등 다양한 모바일 응용에서 활용이 가능하며, 모바일 환경에서의 효율적인 데이터 관리와 빠른 응용 개발을 가능하게 하는 기능과 성능을 제공한다.

## 감사의 글

본 연구는 건설교통부 첨단도시기술개발 사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C05)에 의해 수행되었습니다.

## 참고문헌

- [1] <http://www.etrico.com/en/activities/activities/actimap.htm>
- [2] <http://www.internavi.ne.jp/>
- [3] Y.J. Joo, J.Y. Kim, S. H. Park, "Design and Implementation of Map Databases for Telematics and Car Navigation Systems using an Embedded DBMS," The Journal of GIS Association of Korea, Vol.14, No. 4, pp.379-389, December 2006
- [4] G.J. Kim, S.C. Baek, "LGeDBMS: a small DBMS for embedded system with flash memory," Proceedings of the 32nd international conference on VLDB, September 12-15, 2006, Seoul, Korea
- [5] OpenGIS Consortium, Inc., "The OpenGIS Simple Feature Specification for OLE/COM Revision 1.1", 1999
- [6] MCP-MAUS간서비스 프로토콜(TTA.KO-06.0129), MAUS-단말간 서비스 프로토콜(TTA.KO-06.0130), [www.tta.or.kr](http://www.tta.or.kr)
- [7] 이재우, 김종훈, 전홍석, "차량 항법 시스템의 경로 탐색을 위한 탐색 알고리즘들의 성능 비교," 한국정보교육학회논문집, 제2권 제2호, pp.252-259, 1998
- [8] G. R. Jagadeesh, T. Srikanthan, and K. H. Quek, "Heuristic Techniques for Accelerating Hierarchical Routing on Road Networks," IEEE Trns. Intelligent Transportation Systems, Vol. 3, No 4, pp.301-309, 2002