

WIPI platform기반 LBS 맵 브라우저 클라이언트

Map Browser Client for LBS using WIPI platform

고상보, 좌정우*

제주대학교 통신공학과*

Ko Sang-Bo, Jwa Jeong-Woo*

Department of Telecommunication Engineering,
Che-Ju University*

요약

이동통신망 진화와 함께 다기능 단말기를 기반으로 한 무선인터넷 서비스가 활성화되고 있다. 무선인터넷 서비스는 브라우저 기반의 정보서비스와 다운로드 플랫폼을 사용하는 이동멀티미디어 서비스가 제공되고 있다. 다기능 단말기 보급이 확산됨에 따라 이를 기반으로 한 신규서비스 개발이 활발하게 이루어지고 있다. 본 논문은 무선인터넷 서비스에서 핵심서비스로 발전할 위치기반 서비스를 위한 모바일 맵 브라우저로, 개발된 맵 브라우저는 국내 무선인터넷 플랫폼인 WIPI를 사용하여 GIS 서버와 연동하여 지도정보를 제공한다. 또한 개발된 시스템은 위치정보 서비스 이외에 텔레메틱스를 위한 GIS 플랫폼으로 사용되고, 향후 WIPI 플랫폼 확산과 더불어 PDA, 텔레메틱스 단말기 등에서도 서비스가 가능할 것이다.

Abstract

Mobile internet service based on CDMA network evolution and multifunction mobile terminal is popular in Korea. Mobile internet service is classified as the information service using browser and mobile multimedia service using download platform. Killer applications based on multifunction mobile terminals have been developed for new business in mobile internet services. One of killer applications is location based service (LBS) using mobile location information. In this paper, we develop mobile map browser using WIPI platform that is the main function in LBS. Mobile map browser interworking with GIS server provides the map information. The developed mobile map browser can be working on PDA and telematics terminals when WIPI is porting on these devices.

I. 서론

최근 들어 무선 인터넷 시장이 형성되면서 이동성을 기반으로 하는 서비스에 대한 관심이 급증하고 있다. 그리고, 휴대폰, PDA와 같은 휴대용 단말기의 사용이 일반화된 가운데 GPS를 기반으로 하는 서비스들도 점차적으로 보편화되어 가고 있다[1,2]. 이러한 변화에 따라 이동성을 지원하는 무선 단말기를 기반으로 한 Mobile GIS 환경에서의 위치 정보 활용은

커다란 관심을 불러들이고 있다. Mobile GIS란 다양한 Mobile device에 지도 등의 공간 데이터를 포함한 GIS 응용 어플리케이션을 장착하여 시간과 장소의 제약 없이 공간정보 획득이 가능한 시스템을 말하며, Mobile GIS 서비스 중 가장 활발한 분야는 CNS(Car Navigation System)와 PNS(Personal Navigation System)로 나타나고 있다.

최근 텔레메틱스 단말기를 통해 교통지리정보를 제

공하는 CNS서비스도 나타나고 있으며, 이러한 서비스를 효율적으로 제공하기 위해 위치기반 서비스(LBS: Localization Based Service)가 널리 이용되고 있다[3-6]. LBS는 길 안내 서비스, 교통 정보 서비스, 관광 정보 서비스, 부동산 정보 서비스, 물류 운송 정보 서비스, 버스/지하철 노선 안내 서비스, 지역 정보 서비스, 사람/차량 위치 추적 서비스 등 실생활에 아주 밀접하게 관련되어 있어 다양한 부가 서비스의 창출이 가능하기에 앞으로 기대되는 비즈니스 모델로 꼽히고 있다.

현재 국내에서만 이미 3000만 명 이상이 휴대폰을 보유하고 있어 이젠 사실상 노인과 어린이등을 제외한 인구의 대부분이 이동 중에도 통신을 할 수 있게 되었다. 이러한 점에서 PDA를 이용한 모바일 콘텐츠 보다는 휴대폰을 이용한 LBS가 많이 연구되어 지고 있다[8,9]. 또한, 다양한 콘텐츠 공급과 폭넓은 사용자층의 확대를 위한 기반 조성을 위한 표준 플랫폼 요구에 의해 이동통신 단말기용 응용 프로그램 실행 환경을 표준화한 WIPI(Wireless Internet Platform for Interoperability)표준화 규격이 한국무선인터넷 표준화 포럼(KWISF : Korea Wireless Internet Standardization Forum)과 한국전자통신연구원(ETRI : Electronics and Telecommunications Research Institute)의 활동에 의해 개발되면서 WIPI 플랫폼 기반에서의 공통 컴포넌트 개발이 활발히 이루어지고 있다. 표준 플랫폼 WIPI의 탄생은 기존 이동통신사업자의 다중 플랫폼에 의한 다양한 콘텐츠 공급의 문제점과 콘텐츠 개발의 중복투자에서 비롯되는 비용과 엔지니어링 시간과 비용문제 등을 해결할 뿐 아니라 1회의 콘텐츠 개발로 모든 이동통신사업자들이 공급할 수 있어 사용자로 하여금 폭넓은 콘텐츠의 선택을 제공하고 있다.

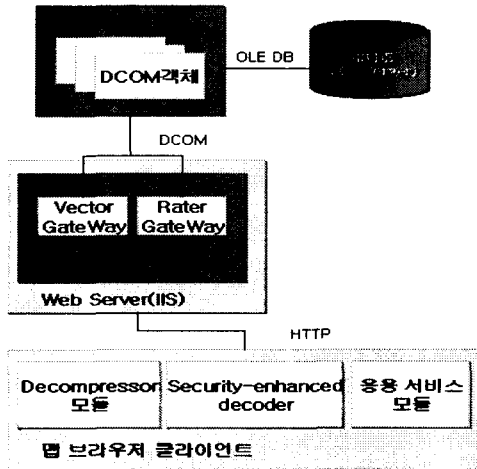
본 논문에서는 이러한 이동성과 LBS를 위한 위치 정보 시스템으로 표준 플랫폼 WIPI를 기반으로 모바일 맵 브라우저를 구현한다.

II. 시스템 구성

1. GIS 시스템

전체 시스템의 구성은 크게 GIS DB(SQL Server), GIS Server, Web Server 그리고 클라이언트로 구성된다. 시스템은 클라이언트가 구동되면서 DLL이 호출되고 GIS Server가 공간 데이터를 WKB타입의 이진 데이터로 저장되어있는 SQL 서버로부터 SQL문으로 조회를 통해 데이터를 읽어 오게 된다. 그리고 클라이언트에서 요청이 있을 시 DLL은 미리 읽어둔 GIS Server의 데이터를 받아 데이터를 경량화 시켜 클라이언트로 요청한 데이터를 넘겨주고 데이터를 넘겨받은 클라이언트는 Geometry_type에 따라 데이터를 다시 실수형 데이터로 파싱을 하게 되고 파싱된 데이터는 이미지로 변환되어 이미지 형태로 화면에 보이게 된다. 그림 1에서 전체 시스템 구성도를 보여주고 있다. GIS DB는 공간 데이터를 WKB형태의 이진 파일로 저장하고 있는 데이터 베이스로 본 논문에서는 SQL Server를 사용하였다. 데이터는 OpenGIS 컨소시엄의 WKB(Well-Known Binary)로 Well-Known Binary 표현에 대한 기본 빌딩 블록은 점에 대한 바이트 스트림이며, 이는 두 개의 DOUBLE 값으로 구성되어 있으며 다른 기하학에 대한 바이트 스트림은 이미 정의된 기하학에 대한 바이트 스트림을 이용하여 구성한다[10]. GIS Server는 서비스 구동과 동시에 데이터베이스에 저장되어 있는 이진 파일에서 SQL 질의를 통해 전체 데이터를 읽어들이게 된다. 이는 클라이언트에서 요청시 화면에 도시하는데 시간을 줄이기 위함으로 미리 읽어 들인 데이터를 DLL을 통해서 전달하게 된다. Web Server는 ISAPI (Internet Server Application Programing)를 사용한다. 이는 웹 서버의 가장 중요한 기능 중의 한가지인 서버에 대한 웹 사용자의 액세스를 관리하기 위하여 ISAPI필터는 개발자들이 이러한 사적인 영역을 보호하는 임의 제작된 보안 모델을 만들 수 있도록 한다. 시스템의 암호화 모듈로는 DES(Data Encryption Standard) 암호

화 모듈을 사용한다[11]. 클라이언트는 Java로 프로그래밍 되었으며 메인 클래스는 geom패키지 클래스, theme 패키지 클래스, ui 패키지 클래스, util 패키지 클래스, cfg 패키지 클래스, security패키지 클래스, symbol 패키지 클래스로 구성된다.

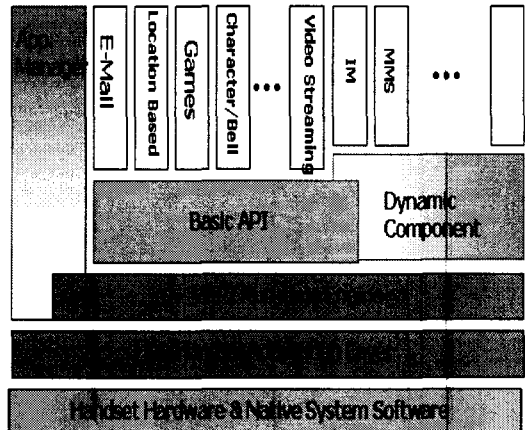


▶▶ 그림 1. 서비스 시스템 구성도

2. WIPI platform

현재 WIPI는 이동통신3사가 WIPI를 탑재한 단말기 출시계획을 발표 했고, 점차 WIPI폰 생산을 확대한다는 계획을 가지고 있다. 정보통신부와 이동통신업계는 위피를 국제 표준으로 만들기 위해 2002년 6월 국제무선인터넷표준화기구(OMA)에 국제 표준으로 제안하였고 또한 위피 해외진출 계획도 본격 추진되고 있다. 또한 한국무선인터넷표준화포럼은 작년 11월 중국 신식산업부 관계자들에게 위피 상용폰을 시연하고 중국시장 진출 가능성을 본격 타진하고 돌아와 앞으로 위피에 대한 많은 가능성을 보여주고 있다. WIPI 플랫폼의 개념적 구조를 살펴보면 그림 2에서 구조를 보여주고 있다. 모바일 플랫폼이란 모바일 표준 플랫폼 규격에 따라 작성된 응용프로그램을 실행 시킬 수 있는 단말기의 실행 환경(Runtime Execution Environment)을 모바일 플랫폼이라 하며 응용프로그램 관리와 API 관리 기능을 포함해야

한다. HAL(Handset Adaptation Layer)이란 플랫폼의 하드웨어 독립성을 유지하기 위한 추상화 계층으로 상위 레이어들은 HAL위에서 Native System과 무관하게 즉, 하드웨어 독립적으로 플랫폼이 구성되며 하단의 단말기 기본 소프트웨어와 플랫폼을 연결해 주는 역할을 한다. 단말기 기본 소프트웨어(Native System Software)는 플랫폼이 탑재되는 기본 소프트웨어를 말한다. 기본 API(Basic API)란 응용프로그램 개발자가 사용하는 플랫폼에서 지원하는 기본 API 모음으로, Java와 C API로 구성되어 있다.



▶▶ 그림 2. 위피 플랫폼 구성도

III. 시스템 설계

1. 데이터 전송 포맷

OGC의 표준 전송 포맷인 WKB를 경량화시켜 무선 네트워크 환경에 적합한 구조로 변환한다[10]. 무선 네트워크 환경과 무선 기기의 하드웨어적 제한성을 고려하여 실수형 데이터의 정수형 변환 작업을 통한 경량화 작업을 수행한다. 즉, 실수형 8바이트 데이터를 4바이트 정수형 데이터로 변환하여 보내주는데 변환시 발생할 수 있는 데이터 overflow현상을 방지하기 위해 실수형 데이터에 일정크기의 origin 실수값을 뺀 후 multiple 값으로 곱해주면 8바이트 실수

형 데이터를 정수형으로 변환하는 작업이 완료되고 클라이언트에서는 다시 정수형 데이터를 실수형 데이터로 재변환하게 된다. 그림 3,4에서 변화과정을 보여주고 있다.

```

m_dOrgX = 106698.0
m_dOrgY = 3659233.0
m_dMultiple = 10.0

int xmin = (int)((dMinX-m_pLayerInfo->m_dOrgX)
              *m_pLayerInfo->m_dMultiple);
int ymin = (int)((dMinY-m_pLayerInfo->m_dOrgY)
              *m_pLayerInfo->m_dMultiple);
int xmax = (int)((dMaxX-m_pLayerInfo->m_dOrgX)
              *m_pLayerInfo->m_dMultiple);
int ymax = (int)((dMaxY-m_pLayerInfo->m_dOrgY)
              *m_pLayerInfo->m_dMultiple);

```

▶▶ 그림 3. 서버에서의 변환작업

```

double x = lx / m_pLayerInfo->m_dMultiple +
          m_pLayerInfo->m_dOrgX;
double y = ly / m_pLayerInfo->m_dMultiple +
          m_pLayerInfo->m_dOrgY;

```

▶▶ 그림 4. 클라이언트에서 재변환작업

2. 압축 필터

압축률이 좋은 Arithmetic Coding 기법으로 경량화 된 WKB Data블록을 더욱 경량화시켜 데이터를 전송한다. Arithmetic encoding 과정은 다음의 그림 5과 같다. 서버에서 전송받는 스트림(WKB 이진 데이터)을 버퍼에 저장한 후 버퍼를 인코딩하게 된다.

```

UINT nLength = pStream->GetStreamSize();
BYTE* pBuffer = pStream->Detach();

CArithCompressor com;

com.StartModel();
/* Set up other modules */
com.StartOutputtingBits();
com.StartEncoding();

for(int i = 0; i < nLength; i++) {
    unsigned char ch = pBuffer[i];
    int symbol = com.m_nCharToIndex[ch];
    com.EncodeSymbol(symbol);
    com.UpdateModel(symbol);
}

```

▶▶ 그림 5. Arithmetic encoding 과정

```

{
    m_oOffImage = null;
    int nLength = CompUtils.readStreamInt(fis);
    int nCmpLength = CompUtils.readStreamInt(fis);

    byte buffer[] = new byte[nLength];
    byte CmpBuffer[] = new byte[nCmpLength];
    CompUtils.readStream(fis, CmpBuffer);
    fis.close();

    ArithCompressor com = new ArithCompressor();
    com.StartModel();
    com.StartInputtingBits();
    com.StartDecoding(CmpBuffer);

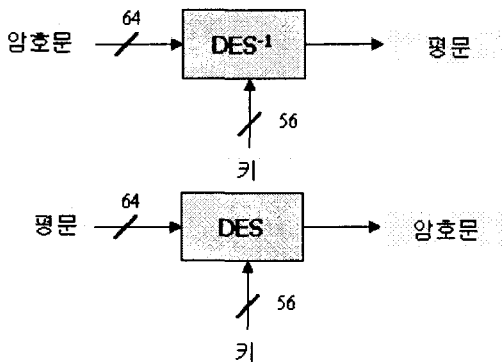
    for (int i = 0; i < nLength; i++)
    {
        int symbol = com.DecodeSsymbol();
        if (symbol == com.EOF_SYMBOL)
        {
            break;
        }
        buffer[i] = (byte) com.m_cIndexToChar[symbol];
        com.UpdateModel(symbol);
    }
    int nSize = 0;
    ByteArrayInputStream in =
        new ByteArrayInputStream(buffer);
    streamToMap(in);
    in.close();
}

```

▶▶ 그림 6. 클라이언트에서 Arithmetic Decoding 과정

3. 보안화 필터

DES는 1977년 미 상무성의 국립 표준국(NBS, National Bureau of Standards)에서 채택한 64비트 대칭형 블록암호 알고리즘으로 56비트 키를 사용하고(64비트 중 8비트는 parity check로 사용) 16라운드의 Feistel 구조를 갖는다. DES는 평문을 64비트로 나뉘 56비트의 키를 이용해 다시 64비트의 암호문을 만들어 낸다[11].



▶▶ 그림 7. DES를 이용한 암호화 & 복호화

4. 데이터 전송 규칙

데이터 전송을 위한 규칙은 다음과 같다.

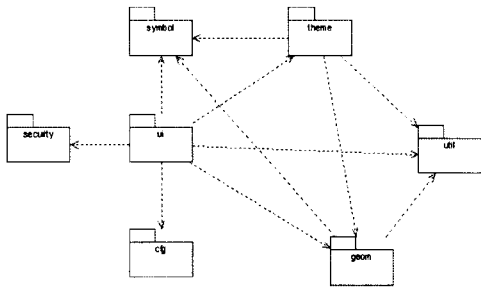
1. 전달되는 데이터 패킷은 요청 URL에 명시된 레이어의 순서대로 구성한다.
2. 지도 데이터의 경량화를 통해 변환된 정수 좌표 값을 클라이언트에서 실수화를 통하여 복원한다.
3. 지도 데이터 경량화의 정수 변환을 위한 스케일 값을 지도 데이터 전송에 앞서 복원을 위해 제공한다.
4. 각 레이어의 데이터 패킷에 레이어의 도형 타입을 지정하는 1Byte의 값이 전달된다.
5. 하나의 레이어는 단일한 유형의 Geometry type으로 구성한다.
6. 데이터의 엔디언 타입은 little endian으로 한다.
7. Geometry에는 레이블이 뒤따르며 2바이트의 문

자열 길이(바이트 수), 길이에 해당하는 문자열의 실제 내용이 포함된다.

8. 문자열은 unicode로 구성한다.

5. 단말 클라이언트

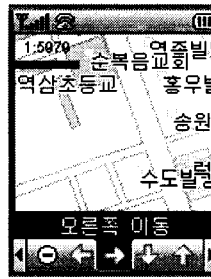
geom 패키지 클래스는 각종 도형(Point, Line, LinearRing, Polygon 등)의 상위 클래스인 Geometry를 통한 도형 정보를 토대로 축척별로 layer 정보를 가지고 있는 Theme에 속성 정보인 label을 취합하여 최종적인 맵을 도출하게 되고, symbol 패키지 클래스는 각종 도형의 속성 정보 및 도형 종류를 분류하는 클래스들의 모임이다. 특정 도형의 종류는 GroupSymbol에 의해 구분되고 각 도형의 속성 정보 중 label의 유무에 따라 labelSymbol, GeometrySymbol을 사용하게 되며 각 도형의 GroupSymbol별로 일대일로 매칭되는 이미지의 표현은 ImageSymbol에 의해 표현한다. theme 패키지 클래스는 GeoLayer의 각 Geometry를 드로잉 해주는 GeoVectorTheme (Geo Theme을 상속)으로 구성되고 GeometryContainer에 추가된 geometry를 GeoLayer를 통해 Theme에서 드로잉하게 된다. ui 패키지 클래스는 사용자가 직접적으로 입력하거나 조회할 때 화면에 보여주는 Boundary 클래스로 화면의 GUI와 지도의 경계를 결정한다. GeoMap클래스는 GIS 엔진의 데이터를 보여준다. util 패키지 클래스는 맵브라우저 클라이언트 프로그램을 작성 시 필요한 유용한 유틸리티용 클래스를 모은 패키지이다. 특히 ArithCompressor에서 데이터 압축을 실시한다. cfg 패키지 클래스는 맵 브라우저 클라이언트 프로그램 환경을 설정, 관리하는 클래스들의 패키지이다. Configuration클래스는 프로그램이 시작하면 서버로부터 Configuration 파일을 다운 받아 추가 환경을 세팅하여 준다. security 패키지 클래스는 위치 및 속성 데이터를 암호화 하여 주는 클래스들을 모아 만든 패키지이다. 그림 6에서 메인 클래스들의 다이어그램을 보여주고 있다.



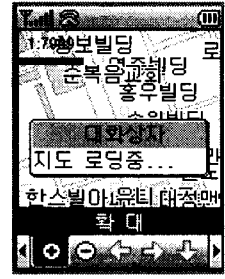
▶▶ 그림 6. 메인 클래스 다이어그램

IV. 구현 결과

구현한 결과는 WIPI Emulator를 통하여 확인하였으며 맵 브라우저의 기능은 기본적인 확대, 축소, 이동, 검색의 4가지 기능만을 구현하였다. 구현에 사용된 지도 데이터는 서울특별시 데이터이며 TM좌표계에 맞춘 데이터이다. 초기 화면에 보여지는 지도의 좌표는 X=326374, Y=4150627, 축척은 0000이다. 중심 좌표 지역을 기준으로 경량화 과정에서 필요한 Orgin값은 OrgX=6000, OrgY=106698, Multiple=10.0이며, 지도 표출이 가능한 최대 스케일 값은 30000이며 최소 스케일 값은 2000이고 지도 도시는 2차원으로 하였으며 이동시 한번 클릭 시 이동되는 지도의 범위는 30m이다. 이 거리는 상, 하, 좌, 우 모두 동일한 값이고 확대, 축소의 기능 시 보이는 지도의 범위는 한번 클릭 시 현재 스케일에 1:2000의 축척으로 확대 혹은 축소된다. 검색기능은 데이터의 속성정보를 SQL문을 통하여 검색하여 검색한 결과를 도시하여주며, 검색 결과를 선택 시 해당 건물의 중심 좌표값을 얻어와 그 좌표값을 중심으로 화면의 범위만큼을 도시한다. 검색 시 검색 결과가 될 때까지 소요되는 시간은 2-3초가 소요되고 지도 로딩 시 소요되는 시간은 평균 3-5초가 걸림을 확인 할 수 있었다. 그림 7은 Emulator에서 맵 브라우저 실행 결과를 캡처한 화면이다.



<초기 화면>



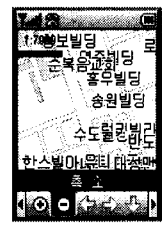
<지도 로딩 화면>



<축소>



<확대>



<오른쪽 이동>

▶▶ 그림 7. WIPI Emulator 실행결과

IV. 결론

본 논문에서 GIS를 이용하여 WIPI platform에서 위치기반 서비스를 위한 맵 브라우저 클라이언트를 구현하였다. 전송하는 데이터를 무선 데이터 통신에 적합하도록 경량화하고 Arithmetic Coding을 통한 압축으로 3~5초 사이의 신속한 지도 표출이 가능하였다. 본 논문에서 구현한 맵 브라우저 클라이언트의 목적은 GIS를 이용한 모바일 맵 브라우저 서비스이다. 이는 향후의 더 많고 다양한 LBS 응용 모바일 서비스를 제공하기 위한 가장 기본적인 서비스로 차후 LBS플랫폼과 연동하여 긴급한 상황 발생시 자신의 위치를 알려거나 친구의 위치를 검색, 교통 혼잡 및 네비게이션 정보등 매우 다양한 분야의 콘텐츠에 시스템을 확장하여 활용이 가능할 것이라 사료된다. 또한 국내 표준 플랫폼 WIPI의 확산과 더불어 PDA, 텔레메틱스 단말기 등에서도 서비스가 가능할 것이다.

■ 참고문헌 ■

- [1] 문형돈, "LBS 기술 및 시장 동향", ETRI 주간기술동향 1080호, 2003.
- [2] 윤재관, 한기준, "LBS(Location Based Service)의 기술동향 및 활성화 전망.", KISDI IT FOCUS, 2002
- [3] 한기준, "이동 객체 DBMS 기술 저장 컴포넌트 개발.", 2003년도 LBS 기술 워크샵 발표집, 2003.
- [4] Hayes. I, "Wireless Location Based Services : Finding your Way Out of the Box", Executice Update, Cutter Consortium, 2001.
- [5] Jagoe. A, "Mobile Location Services : The Definitive Guide", Prentice-Hall, 2003.
- [6] Pundt. H. and Brinkkotter-Rundle,. K, "Visualization of spatial data for field based GIS". Computers & Geosciences 26, 2000
- [7] Derekenaris. G, Garofalakis. J, Makris. C, Prentzas. J, Sioutes. S,& Tsakalidis. A, Integrating GIS, GPS and GSM technologies for the effective management of ambulances, Computers, Environment and Urban System, 25, 2001.
- [8] 안중호, 황명호, 박철우, "LBS 응용 모바일 서비스의 사용요인에 관한 연구", 2003
- [9] Yun. J. K, Kim. D. O & Han. K. J, "Development of a Real-Time Mobile GIS supporting the Open Location Based Service.", Proc. of Geotec Event Conference, 2003.
- [10] OGIS, "OLE/COM을 위한 개방형 GIS 인터페이스에 관한 표준", pp.72-74.
- [11] 김광조, "DES의 선형 보완법에 관한 해설", 통신정보보안학회지 제 3권, 1993.