

GIS 기반 텔레매틱스 POI 단말기 설계

변환식* 김동균 이상정
 순천향대학교 컴퓨터공학부
 e-mail : keep2smile@naver.com, {kdk70, sjlee}@sch.ac.kr

Desing of GIS-based Telematics POI Terminal

Hwan-Sik Byun⁰, Dong-Kyun Kim, Sang-Jeong Lee
 Dept. of Computer Science and Engineering, Soonchunhyang University

요 약

사용자에게 제공하는 일반적인 텔레매틱스 서비스들은 광역 서비스로 한정되어 있고 특정 관심 지점(POI, Point-of-Interest)에 대한 경로와 간략한 안내 서비스만을 제공하고 있다. 그러나 사용자들은 광역 서비스뿐만 아니라 POI 지점의 상세 서비스들도 제공받기를 원한다. 따라서, 각 POI 지점에서의 상세 서비스를 지원하고 이를 관리 제어할 수 있는 텔레매틱스 POI 응용 서비스가 요구되고 있다. 본 논문에서는 POI 지점의 상세 서비스를 제공하기 위하여 GIS 기반 텔레매틱스 POI 응용 서비스를 설계하고 텔레매틱스 POI 단말을 설계/구현한다. 구현된 텔레매틱스 POI 단말은 GPS 모듈로부터 좌표를 수신하고 Shape 뷰어에서 파싱하여 현재위치를 구한다. ESRI Shape 파일 및 DBF 파일을 파싱하여 현재위치와 매칭시킨 후 POI 상세 서비스를 디스플레이 한다.

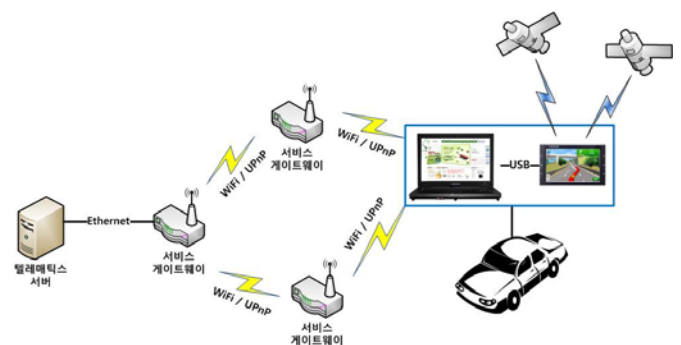
1. 서론

최근의 텔레매틱스 서비스는 기존의 서비스들을 수용하는 종합서비스로 발전하고 있다. 이에 관련 사업자 및 학계에서 활발한 연구 개발이 진행 중이다 [1, 2]. 텔레매틱스 서비스는 사용자에게 교통안내, 위치 기반 서비스, 긴급구난 정보 및 방송 서비스 등을 제공한다. 또한 GPS(Global Positioning System), GIS(Geographic Information System) 기술과 텔레매틱스 서비스 제공자(TSP, Telematics Service Provider)에 의해 실시간 교통 정보 및 생활 정보 서비스를 제공 받는 네비게이션 서비스는 대중화가 이루어져 있다. 그러나 이와 같은 정보 서비스들은 광역 서비스로 한정되어 있다. 즉, 중앙의 TSP 들은 특정 관심지점인 학교, 주유소, 주차장, 음식점 등의 POI(Point-of-Interest, 특정 관심지점) 서비스에 대해서 간략한 안내와 경로 서비스만을 제공하고 있으며, POI 지점의 상세 서비스들을 제공하지 않는다. 사용자들은 광역 서비스뿐만 아니라 POI 지점의 상세 서비스들도 제공받기를 원한다.

따라서 본 논문에서는 GIS 를 기반으로 한 텔레매틱스 POI 상세 서비스를 디스플레이 하기 위한 텔레매틱스 POI 단말을 설계 및 구현하였다. 구현된 텔레매틱스 POI 단말은 GPS 모듈로부터 좌표를 수신하고 Shape 뷰어에서 파싱하여 현재위치를 구한다. 파싱된 데이터로 Shape 파일 및 DBF 파일을 파싱하고 POI 상세 서비스를 디스플레이 한다.

2. 서비스 시나리오

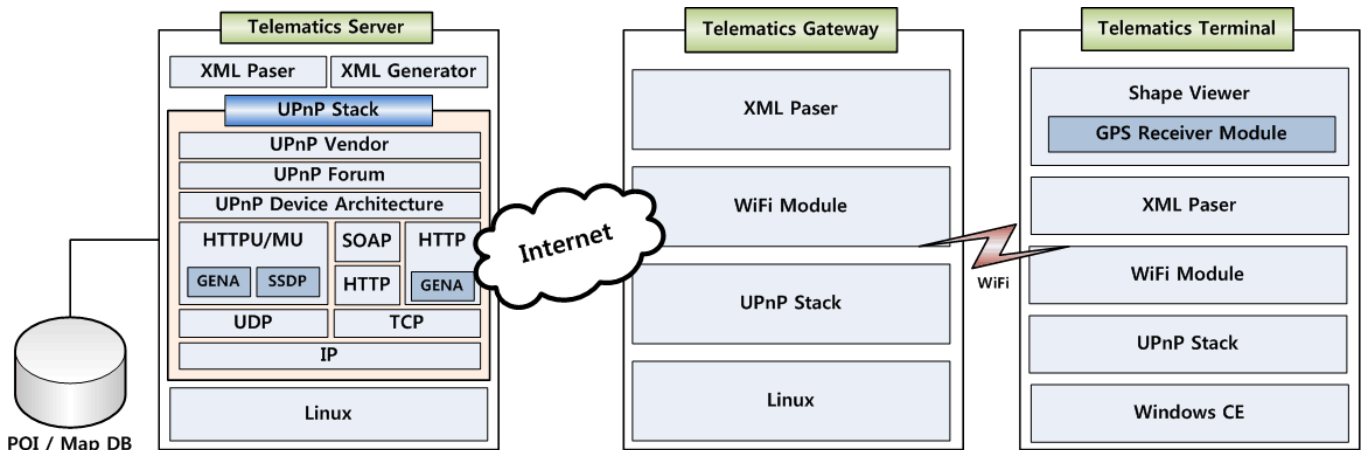
사용자가 대학내 학술행사 참가를 위하여 텔레매틱스 단말(PDA/네비게이터)에 POI 서비스 지점을 설정하고 서비스를 요청하는 시나리오를 가정한다. 교내 안내시스템을 적용하여 사용자가 POI 서비스 지점을 설정하는 부분부터 POI 서비스 지점을 자동 발견하고 상세 서비스를 받는다.



(그림 1) POI 응용 서비스 전체 구성도

POI 응용 서비스 구성은 크게 텔레매틱스 서버와 서비스 게이트웨이, 텔레매틱스 POI 단말로 이루어진다. 그림 2 는 POI 응용 서비스의 전체구성도이다. 텔레매틱스 서버와 서비스 게이트웨이는 이더넷으로 통신하고 서비스 게이트웨이들과 텔레매틱스 POI 단말은 WiFi 로 통신 한다. 또한 텔레매틱스 POI 단말에 탑재된 GPS 모듈은 위성으로부터 좌표정보를 받아

* 이 논문은 2006 년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-511-D00319)



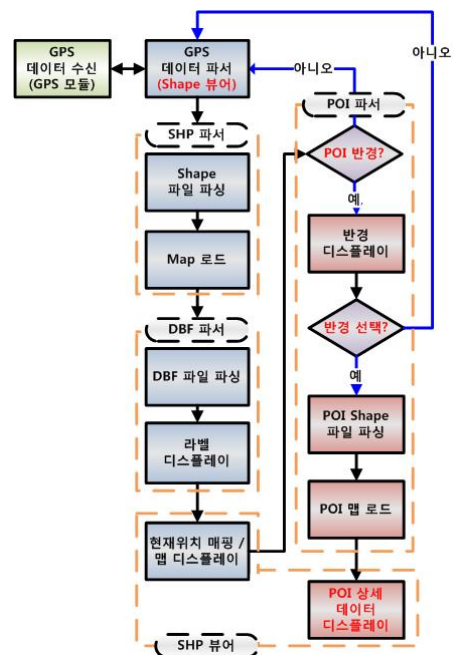
(그림 2) GIS 기반 텔레매틱스 POI 응용 서비스 플랫폼 아키텍처

파싱하여 현재위치를 구한다. 그림 2 는 GIS 기반 텔레매틱스 POI 응용 서비스의 전체적인 구성 아키텍처를 나타낸다. 텔레매틱스 서버는 리눅스를 기반으로 UPNP 스택과 XML 제너레이터/파서를 탑재하여 사용자가 요청한 서비스를 제공하는 역할을 한다. 서비스 게이트웨이는 리눅스 기반으로 UPNP 스택, WiFi 모듈, XML 파서를 탑재하여 텔레매틱스 서버에서 받은 XML 메시지를 텔레매틱스 POI 단말에게 전송한다. 또한 텔레매틱스 POI 단말은 Windows CE 기반으로 UPNP 스택과 WiFi 모듈, XML 파서 및 Shape 뷰어/GPS 수신 모듈을 탑재하여 데이터를 파싱하여 디스플레이 하는 역할을 한다. 그리고 이들은 인터넷과 WiFi 로 통신하며 GIS 정보와 POI 상세 서비스 데이터를 주고 받는다.

3. 텔레매틱스 POI 단말 설계

텔레매틱스 단말은 교내 세부 맵 데이터와 POI 상세 서비스 정보를 서비스 게이트웨이에서 전송 받아 파싱한 후 디스플레이 하는 역할을 한다. GPS 로부터 좌표를 제공 받고 이를 Shape 뷰어에서 파싱하여 현재 위치를 구한다. 사용자가 교내 POI 상세 서비스 지점 부근에 도달하게 되면 근처 서비스 게이트웨이가 방송하는 POI 서비스 안내 메시지를 수신 받게 된다. POI 서비스 안내 메시지를 받은 텔레매틱스 단말은 POI 응용 서비스 리스트를 출력하고 리스트들 중에 앞서 사용자가 설정한 POI 지점에 해당하는 POI 응용 서비스를 찾아 ACK 신호를 보낸다. ACK 신호는 서비스 게이트웨이를 경유하여 텔레매틱스 서버에게 전달 된다. ACK 신호를 받은 텔레매틱스 서버는 텔레매틱스 단말을 UPNP 네트워크에 추가하고 텔레매틱스 단말에게 GIS 정보와 POI 데이터를 XML 메시지 형식으로 전송한다. XML 메시지를 받은 텔레매틱스 단말은 XML 파서를 이용하여 메시지를 파싱한다. Shape 뷰어는 이 파싱한 데이터를 이용하여 사용자에게 POI 응용 서비스를 제공하며 ESRI Shape file 형식의 데이터를 시각적으로 보여주는 기능을 제공한다. ESRI Shape file 은 Geospatial data 에 대한 기하학적 위

치와 속성 정보를 저장/제공 해주는 비위상 구조의 데이터 포맷이다. 공간 형상(feature)의 기하학 정보는 일련의 벡터 좌표들로 구성되는 shape 에 저장된다. 위상구조를 가지지 않기 때문에 빠르게 공간 데이터를 디스플레이, 에디팅이 가능하다. 또한 디스크 용량을 감소시키며, 데이터를 읽고, 쓰는데 용이하다. ESRI Shape file 은 Main file(.shp), Index file(.shx), dBASE table(.dbf)로 구성된다[3,4].



(그림 3) 텔레매틱스 POI 단말에서의 Shape 뷰어 동작 플로우차트

그림 3 은 텔레매틱스 POI 단말에서의 Shape 뷰어 동작 플로우차트이다. Shape 뷰어는 GPS 로부터 받은 좌표를 파싱을 하고 파싱된 GPS 데이터로 현재 위치를 구하여 매칭되는 Shape 파일과 DBF 파일을 로드하여 파싱하게 된다. 그림 4 와 5 는 수신된 GPS 데이터의 예와 GPS 데이터의 일부를 파싱하는 부분이다. 그림 6 과 7 은 Shape 파일과 DBF 파일을 파싱하는 코드 부분이다. 파싱된 데이터를 이용하

여 POI 반영여부를 묻는다. 사용자가 POI 반영에 도달하면 그 반영이 Shape 뷰어에 디스플레이 되며 이를 선택하게 되면 POI 상세 서비스에 대한 Shape 파일을 파싱하고, 맵을 로드하여 텔레매틱스 POI 단말에 POI 상세 데이터를 디스플레이 하게 된다.

```

$GPGGA,141113.999,3730.0308,N,12655.2369,E,1.06,1.7,98.9,M,,,,0000*3E
$GPGSA,A,3,02,07,01,20,04,13,,,,,3.7,1.7,3.2*31
$GPRMC,141113.999,A,3730.0308,N,12655.2369,E,19.77,195.23,101200,*3C
$GPGGA,141114.999,3730.0264,N,12655.2351,E,1.07,1.2,98.8,M,,,,0000*3C
$GPGSA,A,3,02,07,01,20,24,04,13,,,,,2.3,1.2,1.9*3E
    
```

(그림 4) 수신된 GPS 데이터 예

```

void CNMEAParser::ProcessGPGGA(BYTE *pData)
{
    BYTE pField[MAXFIELD];
    CHAR pBuffer[10];

    //
    // Time
    //
    if(GetField(pData, pField, 0, MAXFIELD))
    {
        // Hour
        pBuffer[0] = pField[0];
        pBuffer[1] = pField[1];
        pBuffer[2] = '\0';
        m_btGGAHour = atoi(pBuffer);
        ...

        //
        // Latitude
        //
        if(GetField(pData, pField, 1, MAXFIELD))
        {
            m_dGGALatitude = atof((CHAR *)pField+2) / 60.0;
            pBuffer[2] = '\0';
            m_dGGALatitude += atof((CHAR *)pField);
        }
        if(GetField(pData, pField, 2, MAXFIELD))
        {
            if(pField[0] == 'S')
            {
                m_dGGALatitude = -m_dGGALatitude;
            }
        }
    }
}
    
```

(그림 5) GPS 데이터 파서 부분 코드

```

switch(m_nShapeType)
{
case SHPT_POINT:
    m_pSHPObject = new shpPoint;
    if(!((shpPoint*)m_pSHPObject)->Open(_filename))
        return false;
    break;

case SHPT_ARC:
    m_pSHPObject = new shpPolyLine;
    if(!((shpPolyLine*)m_pSHPObject)->Open(_filename))
        return false;
    break;

case SHPT_POLYGON:
    m_pSHPObject = new shpPolygon;
    if(!((shpPolygon*)m_pSHPObject)->Open(_filename))
        return false;
    break;

case SHPT_MULTIPPOINT:
    m_pSHPObject = new shpMultiPoint;
    if(!((shpMultiPoint*)m_pSHPObject)->Open(_filename))
        return false;
    break;

default:
    return false;
}
    
```

(그림 6) Shape file 형태 및 DBF 처리 부분 코드

```

void shpDBF::WriteHeader()
{
    uchar abyHeader[32];
    int i;

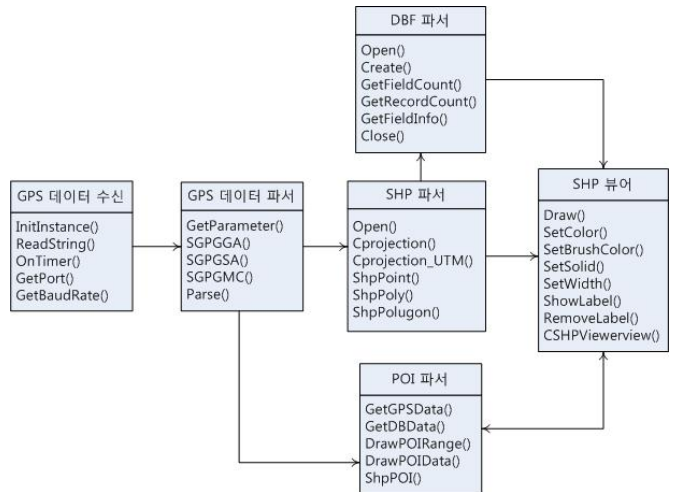
    if(!m_bNoHeader)
        return;

    m_bNoHeader = FALSE;

    for(i = 0; i < 32; i++)
        abyHeader[i] = 0;
    ...
}

int shpDBF::Open(const char *pszFilename, const char *pszAccess)
{
    uchar *pabyBuf;
    int nFields, nRecords, nHeadLen, nReclen, iField;
    char *pszDBFFilename;
}
    
```

(그림 7) DBF 파일 파싱하는 코드 부분

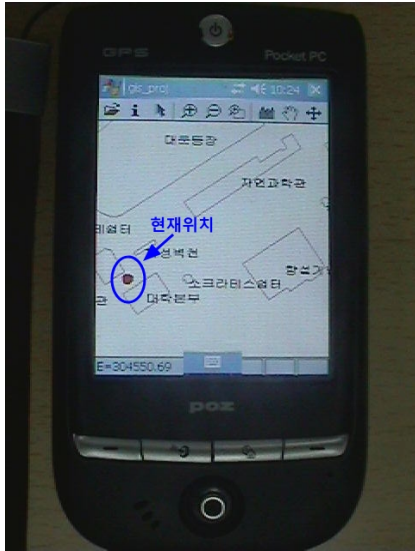


(그림 8) Shape 뷰어 클래스 다이어그램

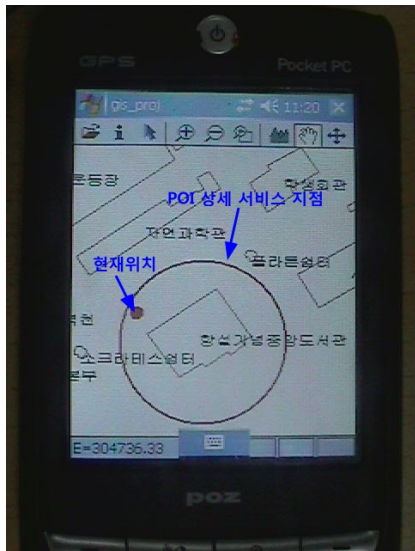
그림 8 은 Shape 뷰어에서 데이터를 처리하는 클래스 다이어그램을 나타낸다.

4. 구현

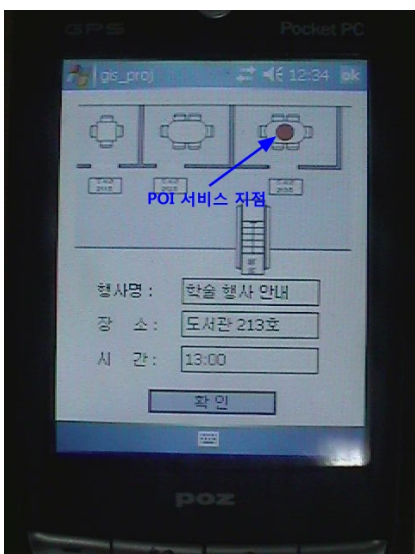
테스트 구현 환경으로는 크게 구분하여 텔레매틱스 서버와 서비스 게이트웨이, 텔레매틱스 POI 단말 등을 설계하고 구현하였다. 텔레매틱스 서버는 리눅스를 포팅하고 UPnP 미들웨어를 탑재하여 테스트하였다. 서비스 게이트웨이는 저가형 상용 유/무선 공유기인 Linksys 사의 WRT54G Ver4.0 에 RS232C 하드웨어 인터페이스를 추가로 부착시켜 리눅스 포팅 및 응용프로그램을 작성하고 UPnP 미들웨어를 탑재하여 테스트하였다. 텔레매틱스 POI 단말은 사이버뱅크사의 POZ CP-G300 PDA 에 UPnP 미들웨어 및 Shape 뷰어/GPS 모듈을 탑재하여 테스트 하였으며 Shape 뷰어는 EVC++ 4.0 으로 작성 되었다. 그림 9 는 텔레매틱스 POI 단말에서의 POI 응용 서비스 동작을 보여 준다. 그림 9 의 가)는 현재위치를 보여준다. 그림 9 의 나)는 사용자가 POI 상세 서비스 지점에 근접했을 경우 POI 상세 서비스 지점 영역을 보여준다. 이 때 이 원을 클릭 하게 되면 그림 9 의 다)와 같이 POI 상세 서비스의 상세 내역을 디스플레이 한다.



(가)



(나)



(다)

그림 9 텔레매틱스 POI 단말에서의 POI 응용서비스 동작

5. 결론

본 논문에서는 텔레매틱스 POI 응용 서비스를 받아 디스플레이하는 텔레매틱스 POI 단말을 설계하고 제안하였다. POI 지점의 상세 서비스를 제공하기 위하여 제안된 시스템은 교내에 적용하여 테스트 하였다. 텔레매틱스 POI 단말에 Shape 뷰어와 GPS 모듈을 탑재하여 위성에서 좌표정보를 수신하여 파싱한 후 현재 위치를 구한다. ESRI Shape file 을 파싱하여 현재 좌표와 매칭시켜 디스플레이 하고 사용자가 POI 영역에 접근하면 POI 반경을 표시하여 선택시 POI 상세 서비스를 제공하였다.

참고문헌

- [1] 안선일, 장병준, 이윤덕 “텔레매틱스 동향 및 기술개발 방향”, 한국 정보과학회 학회지, 제 23 권 제 2 호 p.77-82, 2005년 2월
- [2] 김기영, 김동균, 이상정, “텔레매틱스 단말기용 서비스 플랫폼 설계 및 구현”, 한국인터넷정보학회 논문지 제 7 권 제 3 호, p.13-30, 2006년 6월
- [3] ESRI, <http://www.esri.com/library/whitepapers/pdfs/shapfile.pdf>
- [4] 고려대학교 사회환경시스템 공학과, <http://civil.korea.ac.kr/getFile.asp?filename=shapfile.doc>
- [5] Intel UPnP, <http://www.intel.com/education/highered/Wireless/labs/L05.pdf>
- [6] Tae-Kyung Moon, Joon-Hyung Cho, Seong-Hee Jeong, Jun-Nam Oh, “Hot-Drive - Preliminary retrieved POI cognitive driver navigation system”, Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on 28-01 June 2006 Page(s):1 - 6
- [7] Y.Liong, Y.Ye, “Effect of UPnP advertisements on User Experience and Power Consumption”, Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE 3-6 Jan. 2005 Page(s):91 - 97
- [8] 변환식, 김동균, 전병찬, 이상정, “GIS 기반 텔레매틱스 POI 응용 서비스 플랫폼” 2007 한국정보과학회 한국컴퓨터종합학술대회 논문집(A), 제 34 권 1 호, p123-124, 2007년 06월
- [9] 김동균, 이상정, 이주원, 최영길, “로컬 핫스팟 서비스를 위한 UPnP 기반 텔레매틱스 미들웨어”, 2006 한국정보과학회 한국컴퓨터종합학술대회 논문집(D), 제 33 권 1 호, p.265-267, 2006. 06