

GIS データベースの分散管理と利用— GeomapZ 及び DEMloader における URL アクセス

雷 興林・長谷川 功・村上 裕

Distributed GIS database – URL access in GeomapZ and DEMloader

Xinglin LEI, Isao HASEGAWA, and Yutaka MURAKAMI

産業技術総合研究所地球科学情報研究部門

Institute of Geoscience, National Institute of Advanced Industrial Science and Technology (AIST),

Higashi 1-1-1, Tsukuba, Ibaraki-ken, 305-8567, Japan.

E-mail: xinglin-lei@aist.go.jp; i-hasegawa@aist.go.jp; y.murakami@aist.go.jp

Key words: GIS, URL, WEB database, GIS application

1. はじめに

GIS (Geographical Information Systems) データがあらゆる分野から利用されるのでその整備と標準化は情報化時代においては急務である。これまで、おもに CD-ROM などの形で GIS データが提供されてきた。しかし、GIS データは時間の推移及び調査の詳細程度に伴い変化するものも多く、最新の情報を即時に社会に提供するためには WEB などのインターネット上での提供が望ましい。近年の IT (Information Technology) 革命はインターネットの普及及び高速化を進展させ、地球上のいかなる場所からもネットワークを通じて必要なデータの取得を可能にさせた。そこでユーザはいつでも最新の情報を得ることができる。また国際標準に準じる API, GIS アプリケーションの開発と実用化が着実に進んでいる。そこで、CD-ROM 等のローカル環境に保存されている地学データベースを表示・検索する専用ソフトウェアとして開発した GeomapZ (雷他, 1990; 雷・長谷川, 2000) を、インターネット上に分散しているデータベースを利用できるように拡張した。例えばインターネット上にある DEM (Digital Elevation Model) にアクセスし自分の独自のデータと統合的に利用できるようになった。さらに、任意の範囲の DEM データを抽出でき、HTML ページに埋め込んで利用できる DEMloader を開発した。ここで、GeomapZ と DEMloader を例にし簡単な WEB データベースの構築及び URL (Uniform Resource Locator) アクセスについて紹介する。

2. WEB データベースの構築と利用

インターネット上に GIS データベースを構築する手法は様々ある。専門業者に依頼しなければいけない部分 (例えばハードウェアシステムの構築) もあれば、研究者自らの手で構築した方がよい部分 (データベース) もある。データベースが研究者自らの手で構築すればデータの更新と保守等の作業が容易にできるだけでなく、データの特性と様々な利用法に応じて柔軟なシステムとして開発することができる。データベース活用を促進させるためには、相互利用が容易に行えるように異種システム間での互換性を高めるための標準化が重要である。国際標準化機構

(International Organization for Standardization : ISO) においても、1994 年に地理情報の標準化を図るための専門委員会 (TC211) を設置し、地理情報に関する標準の検討を進め、最終国際規格案 (FDIS) に至っている。この案は A4 判で 1000 頁以上となっている。データを公開する際、標準の規格になるべく従うべきであるが、それに完全に縛られる必要はない。処理速度とデータ容量を考慮すれば、独自の仕様でデータベースを構築する利点もあるからである。今回は簡単のためデータベースを WEB (WWW: World Wide Web) ページと同じサイトに構築する方式を採用する。この方式では、データベースと WEB ページを混在させることができ、WEB ページにデータベースのファイル (画像、HTML など) のリンクを貼ることもできる。

GeomapZ では URL (具体的には HTTP : Hyper Text Transfer Protocol) にある DEM, 地質図 DLG 等のデータを直接アクセスすることが可能で、処理速度を向上させるため独自仕様の内部フォーマットファイルを採用している。これによって、CD-ROM の場合と変わりなくスムーズに表示・検索できる。GeomapZ はブラウザのように働き、基本的に WEB 上にあるデータファイルを必要に応じて直接アクセスして情報を取得する (Fig.1)。WEB サイトには特別なサービスは必要なく、余分な負担をかけることもない。データベースの構築も非常に簡単になる。CD-ROM などのローカル環境に保存されているデータファイルをそのまま HTTP フォルダにコピーすれば済む。

3. GeomapZ の拡張と DEMloader の開発

WEB データベースにアクセスする方法は通常 WEB ページとして提供する。静的な WEB ページと違うのは、ユーザのパラメータに応じて動的に WEB ページを作成しなければならないことである。この処理をサーバーにさせることも出来ればクライアントにさせることも出来る。サーバーに処理を任せれば、ユーザのコンピュータに WEB ブラウザ以外のソフトウェアのインストールは必要ないので非常に便利である。しかし、GeomapZ のようにパラメータに応じて画像を作成し、更に複雑な情報の検索・処理を行う場合、WEB ページとして提供するだけでは不十分である。今回は GeomapZ を従来の機能をそのまま維持

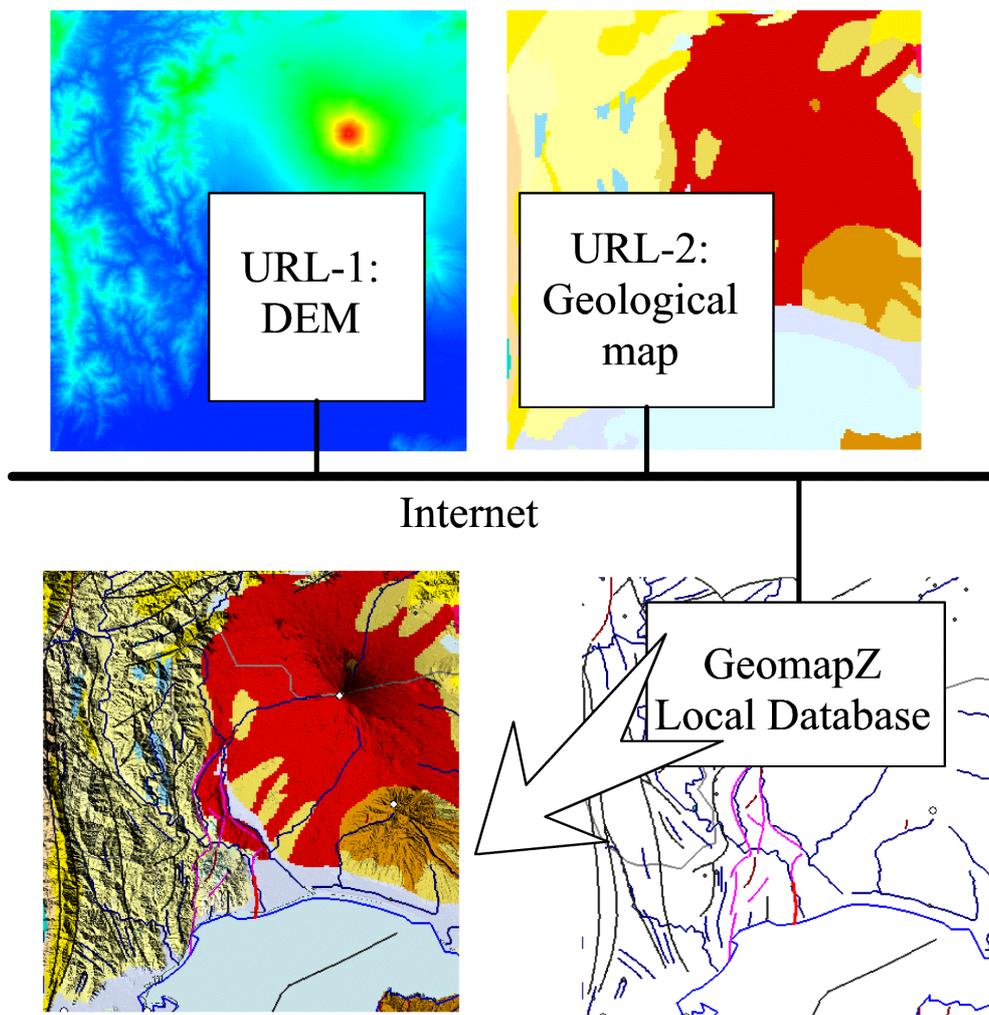


Figure.1 Database access through URL and local storage in GeomapZ

しつつ専用ブラウザのようなアプリケーションに拡張した。GeomapZはもともと地学データベースのCD-ROM出版のために開発した簡易GIS閲覧ソフトなので、この機能を拡張しHTTPファイルをCD-ROM等のローカルファイルのように扱うことが出来れば良いことになる。GeomapZはVC++を利用して開発したWindowsアプリケーションであるから、HTTPファイルにアクセスさせるのは簡単である。MFC (Microsoft Foundation Class) には Fig.2 の階層図に示すような HTTP ファイルサービスが提供されている。CHttpFile は CStdioFile の派生クラスだから、CStdioFile と同じようにテキストモードとバイナリモード両方でファイルにアクセスすることができる。GeomapZ では CLeioFile クラスを定義し、アクセスするファイルの保存先を自動判断して CStdioFile 或いは CHttpFile のインスタンスを作成する。このインスタンスを指すポインタを使えば同じコードで HTTP ファイルとローカルをアクセスすることができる(付録A)。

また、DEMloader は任意の範囲の DEM データを抽出し画像(段彩図、陰影図等)或いは標高データとして保存できる小さいアプリケーションである。独立に使える DEMloader.exe と HTML ページに埋め込む型の DEMloader.class との 2 つのバージョンがある。DEMloader.exe は VC++ を利用し GeomapZ と同様に

HTTP ファイルへアクセスする。国土地理院の 250m・50 m DEM (CD-ROM) を直接に利用することもできる。WEB 上に構築した GTOPO30 及び ETOPO5 の DEM データを利用することもできる (Fig. 3)。DEMloader.class は Java で書かれたアプレットであり、WEB ページに埋め込んで利用する。

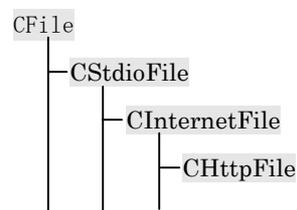


Figure 2. File service in MFC 6.0

4. 終わりに

分散管理のためには個々のデータをそれぞれ所有機関のサイトに置くべきである。こうすれば、個々のデータベースがそれぞれの機関により更新できる。そのデータが WEB 上で公開され、GeomapZ がそのデータにアクセスできるよ

うに機能を強化すれば、いつでも最新の情報が得られるようになるだろう。

GeomapZ及びDEMloaderに関する最新の情報及びダウンロードが試用サイト (<http://staff.aist.go.jp/xinglin-lei/>) から入手できる。ETOPO5, GTOPO30 などを利用した試用WEBデータベースも同サイトに構築した。

参 考

雷 興林・長谷川 功・野呂 春文・脇田 浩二, 地質情報表示・解析用簡易 GIS ソフトーGeomapZの開発. 情報地質, 第10巻, 第4号, 247-255頁, 1999年.

雷 興林・長谷川功, 地質調査所における地質情報データベースの作成と公開, 情報地質, 第11巻, 第3号, 167-177頁, 2000年.

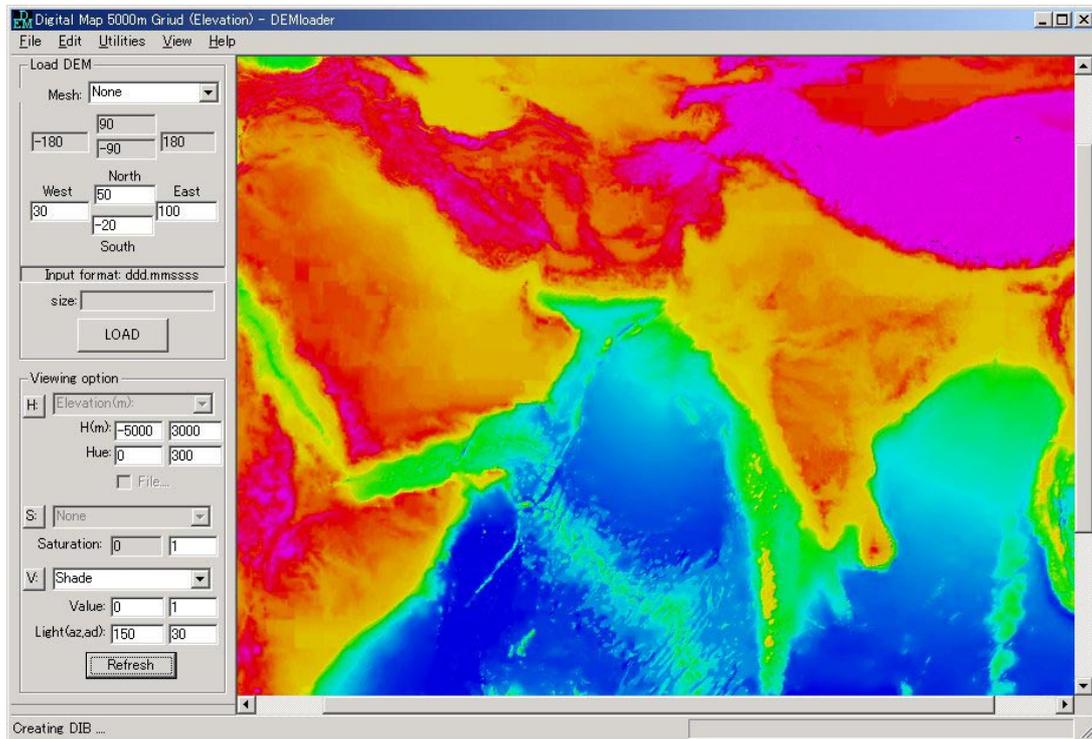


Figure 3. DEMloader connected to the Etopo5 database in a test HTTP site.

付録 A CleiioFile クラス

LeioFile.h //Head file

```
#include "afxinet.h"
class CLeioFile
{
public:
    CStdioFile *m_file;
    CLeioFile();
    ~CLEiioFile();
    CInternetSession m_session;
    CStdioFile *Open(CString csFile, UINT wFla);
    UINT Read(void* lpBuf, UINT nCount);
    UINT ReadTextFile(CString &sText);
};
```

LeioFile.cpp //CPP source file

```
#include "LeioFile.h"
CLEiioFile::CLEiioFile()
{
    m_file = NULL;
```

```

}

CLeioFile::~CLeioFile()
{
    if (m_file) {
        m_file->Close();
        delete m_file;
    }
}

CStdioFile *CLeioFile::Open(CString sFile, UINT wFlag)
{
    if (m_file) { m_file->Close(); delete m_file;}

    if (sFile.Find("http://") > -1 || sFile.Find("ftp://") > -1) { // HTTP または FTP ファイルである
        DWORD wFlag = INTERNET_FLAG_TRANSFER_BINARY |
            INTERNET_FLAG_KEEP_CONNECTION;
        m_file = m_session.OpenURL(sFile, 0, wFlag);
    }else{ // ローカルファイルである
        m_file = new CStdioFile();
        if (! m_file->Open(LPCTSTR(sFile), wFlag)) {
            delete m_file;
            m_file = NULL;
        }
    }
    return m_file;
}

UINT CLeioFile::Read(void* lpBuf, UINT nCount) // 指定されるバイト数のデータをパーファーに読み込む
{
    BYTE *cF = (BYTE *)lpBuf;
    UINT nRead, nNeed = nCount;
    do {
        nRead = m_file->Read(cF, nNeed);
        if(nRead<2) break;
        nNeed -= nRead;
        cF += nRead;
    }while(nNeed>0);
    nRead = cF - (BYTE *)lpBuf;
    return nRead;
}

UINT CLeioFile::ReadTextFile(CString &sText) // ファイル全体をテキストとして読み込む
{
    CString line;
    sText.Empty();
    while (m_file->ReadString(line)) {
        sText += line;
    }
    return sText.GetLength();
}

```