Localhostで運用するボーリングデータ表示アプリケーションの開発

中田　文雄*,**

Development of Borehole Data Display Application at Localhost

Fumio NAKADA*,**

*特定非営利活動法人地質情報整備活用機構 Geological Information Utilization and Promotion Initiative,
URL: http://www.gupi.jp/　E-mail: nakadafumio@gupi.jp
**川崎地質株式会社 Kawasaki Geological Engineering Co., Ltd.,　URL: http://www.kge.co.jp/　E-mail: nakadaf@kge.co.jp

キーワード：ボーリングデータ，アプリケーションソフト，地質情報
Key words ：Borehole Data, Application Software, Geological Information

1. はじめに

ボーリングデータを検索する目的で構築されているウェブサイトで，圧倒的に多いのは地図検索サイトである。電子地図上にボーリング地点が複数表示されるので，場所を限定した検索には極めて便利である。しかし，マーカーをクリックしないとメタデータが表示されないことが多く，目的のボーリングデータを探し出すためには，極めて多くのクリック作業が必要になるなど，“探す”ということに関しては必ずしも効率的ではない点もある。

このため，筆者が自身で利用するためのシステムとして，メタデータを一覧表として表示するアプリケーションソフトウエア(以後，アプリ)を開発した。 開発条件としては，通信環境に左右されることのないように，Windows7 パソコンに元々装備されているウェブ管理ツール「IIS, http://localhost/でアクセス」を利用した。

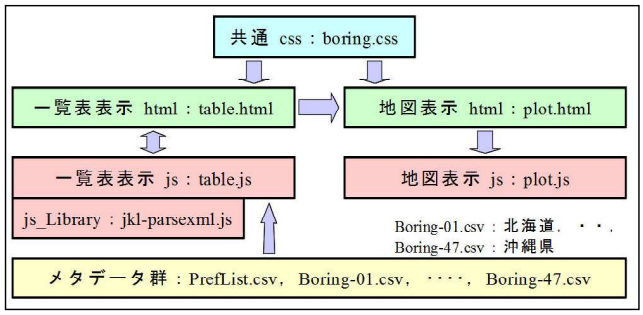
本文は，開発したアプリの概要を示して，その基本的な機能などについて報告するものである。

2. 検索・表示アプリとメタデータ

2.1 アプリの概要

第 1 図は，開発したアプリとメタデータの構成である。なお，本書末に「table.html」および「table.js」の主要部分を抜粋したので参照されたい。

・table.html：一覧表表示アプリの I/O 処理を行う。検索文字列の入力と結果の表示が可能である
・table.js：文字検索，表示範囲の設定処理と出力用 html の合成処理などを行う
・jkl-parsexml.js[1)]：csv で保管されているメタデータを変数に格納するために使用するフリーの js ライブラリーである
・plot.html，plot.js：選択した 1 箇所のボーリング地点を電子地図上にプロットする



第 1 図 アプリとメタデータの構成

2.2 メタデータ

本アプリのメタデータは，本書末の第 1 表に示した「固有コード」や「業務名称」などの 24 項目である。

このうち，一覧表として表示するメタデータは，「引用先」，「固有コード(ID)」，「業務件名」，「孔番号」，「事業者」，「調査業者」および「推定住所」の 7 項目である。

3. アプリケーション

3.1 特徴

第 2 図は，開発したアプリの表示画面(例)である。

・ボーリング 20 本分のメタデータ(2.2 参照)の表示

第 2 図 一覧表表示画面(部分・例)

```
//*****各都道府県のボーリングメタデータ
function SelectPref() {
   var id =Number(document.PrefBox.PrefNumber.value);  //***「id」は html のリストボックスで選択した数値
   var file_name = String(Prefdata[id][5]); //***配列変数「Prefdata」で 0 番から 5 番目が都道府県別のメタデータファイル名
   var httpObj = new JKL.ParseXML.CSV( "metadata/" + file_name ); //***使用する処理ライブラリーの書式
   var rows = httpObj.parse();  //***使用する処理ライブラリーの書式。 csv の値は 2 次元配列変数「rows[i][j]」に分解格納
   metadata = rows.concat(metadata); //***都道府県を選択する度に「rows[i][j]」の各値は「metadata[k][l]」に追加コピーされる
   AllDisplay();
}
//*****キーワード検索
function keywordkensaku() {
    var j= 0;
    for (var i = 0; i < metadata.length; i++) {
        var taisyouword = String(metadata[i][1]) + ・・・ + String(metadata[i][14]); //***被検索文字列の合成
        if (taisyouword.indexOf(keywordx) != -1) {          //***被検索文字列に対し「1 文字一致」でヒット
                dispdata[j] = metadata[i]; //***メタデータ用の変数から表示用配列変数へのコピー
                j = j + 1;
        }
    }
}
```

第3図　table.js(抜粋)

・任意の 1 本についての詳細なメタデータの表示
・スイッチにより，詳細メタデータ表示中のボーリング位置をダイヤログボックスの地図上へのマーキング

3.2 「table.js」

第 3 図は，開発した「table.js」の抜粋である。本アプリには FOSS である「jkl-parsexml.js」を組み込んであり，公開サーバのボーリングデータの閲覧や電子地図の使用を除いて，インターネットへの接続を必要としない。

一覧表を表示させるために，以下のように 2 次元配列変数を 2 種類用意した。

metadata[i][j]：読み込んだメタデータの格納用
dispdata[k][l]：検索結果の格納用。全数表示の場合は単純に全データをコピーしている。

表示ページを変更する場合でも再検索する必要がないため，クライアントがよく持つ「イライラ感」の低減に寄与している。

なお，国土地盤情報検索サイト(KuniJiban)のボーリングデータを対象とした場合，5 万本程度のメタデータ表示でもパソコンがクラッシュすることはなかった。

4. 通信環境による表示時間の差

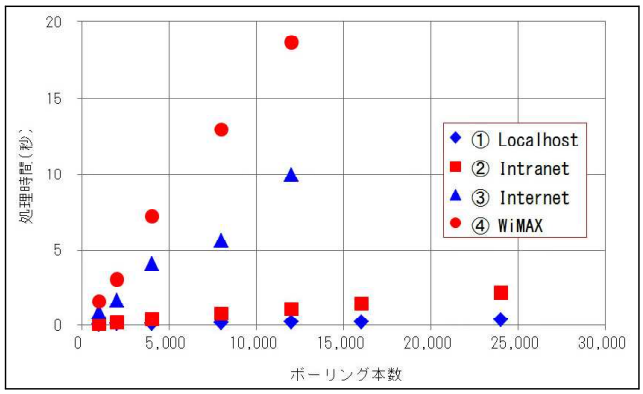
KuniJiban のボーリングデータから任意の 1,000 本を収集し，24 項目のメタデータを抽出して csv ファイルを作成した。各データを繰返すことにより 2,000 本，4,000 本，・・・，24,000 本の試験用メタデータファイルを作成した。なお，1,000 本の試験用メタデータの容量は 538.4kbyte(24,000 本では約 12.9Mbyte)であった。

メタデータの選択時刻と表示終了時刻の差をプログラムで自動計測した。なお，ブラウザの履歴クリアと再読込を行って，前回のキャッシュが残らないようにした。

結果を第 4 図に示す。

24,000 データをロードして一覧表を表示するために必要な時間は，Localhost の場合 0.5 秒弱，イントラネットでは 2 秒強であった。一方，一般用のインターネットでは 12,000 本で 10 秒程度の時間が必要であり，WiMAX では 20 秒近くとなったため，これ以上の試験は中止した。

この結果から，メタデータを一覧表表示するまでの時間はパソコン自身の性能ではなく，メタデータをサーバから転送する時間に支配されていることがわかる。



注 メタデータを格納した場所による表示までの時間

第 4 図 通信環境による表示時間の差

5. おわりに

紙面の関係により，これらの各アプリやメタデータ群を Localhost にセットする方法については省略した。

本アプリは商用ではないため，開発に当たっては「table」タグを随所に使用して，最もマンパワーが必要である css(スタイルシート)の開発は殆ど省略した。

筆者のノートパソコンに 24 万本強のボーリングメタデータをセットして利用しているが，通信環境に左右されないため，極めて快適であることを付記しておきたい。

本アプリについては，講演後にしかるべきウェブサーバから公開する予定であるが，メタデータは筆者らの資源であるため，将来的にも非公開としたい。

なお，KuniJiban は，公開サーバ上のボーリングデータ(XML)と電子柱状図(PDF)への直接アクセスを認めていないので，予め一定数の各データをダウンロードして自身のパソコンに格納する必要がある。KuniJiban には，この目的のために 100 本までの一括ダウンロード機能があるので，利用されるとよい。

【参考文献】

1) Yusuke Kawasaki（2010），JKL.ParseXML/ajax 通信処理ライブラリ，
http://www.kawa.net/works/js/jkl/parsexml.html

第１表 ボーリングメタデータ

項　目　名	変数	一覧表	項　目　名	変数	一覧表
引用先	metadata[i][0]	○	高さ	metadata[i][12]	
固有コード(ID)	metadata[i][1]	○	掘進長	metadata[i][13]	
業務件名	metadata[i][2]	○	掘削終了期日	metadata[i][14]	
孔番号(Boring No)	metadata[i][3]	○	柱状様式	metadata[i][15]	
事業者	metadata[i][4]	○	基盤地質	metadata[i][16]	
事業主体	metadata[i][5]		N≧50の出現深度	metadata[i][17]	
調査業者	metadata[i][6]	○	同合計N値	metadata[i][18]	
担当者	metadata[i][7]		同合計貫入量	metadata[i][19]	
経度	metadata[i][8]		孔内水位	metadata[i][20]	
緯度	metadata[i][9]		電子柱状図(pdf等)のURL(リンク)	metadata[i][21]	
推定住所	metadata[i][10]	○	ボーリングデータ(XML)のURL(リンク)	metadata[i][22]	
高さ基準	metadata[i][11]		都道府県コード	metadata[i][23]	

注１推定住所：国土交通省から発表されているジオコード表を使用して，経緯度から推定した住所
注２高さ基準：電子納品制度が確立する前のボーリング柱状図では，孔口の高さ基準が統一されていなかった。
　　一般的に，建築関係のボーリングでは仮ベンチマーク(KBM)からの相対表示が多い。
注３基盤地質：名称は基盤地質とあるが，表層から最深層までの全ての地層名を抽出している(重複は排除)
注４Ｎ≧50の出現深度など：標準貫入試験結果で，最も浅くＮ値が50を超えた深度，そのＮ値と貫入量

```
//*****使用する Java Script
<script type="text/javascript" src="jkl-parsexml.js"></script>
<script type="text/javascript" src="kunijibantable.js"></script>

//*****都道府県の選択
<form name="PrefBox">
  <select size="8" name="PrefNumber">
    <option value="0" selected>北海道</option>
    <option value="1">青森県</option>
    <option value="2">岩手県</option>
          ・・・・・・・
    <option value="44">宮崎県</option>
    <option value="45">鹿児島県</option>
    <option value="46">沖縄県</option>
  </select>      <input type="button" id="move1" value="表示" onclick="SelectPref()">
</form>

//*****キーワード検索(部分)
<form name="KeyBox">キーワード：
  <input type="text" font-size: 1.0em; name="keyw" value="" size="16">→[次行に続く，以下同じ]
  →<input type="button" id="move2" value="検索" onclick="keywordkensaku()">
</form>

//*****選択データ数の表示枠
<div id="alldata"></div>

//*****メタデータのタイトル
<td width="5%">No</td><td width="6%">引　用　先</td>
<td width="16%">固有コード</td><td width="20%">業務件名</td>
<td width="12%">孔　番　号</td><td width="15%">事　業　者</td>
<td width="14%">調査業者</td><td width="12%">推定住所</td>

//*****メタデータの表示枠
<div id="datalatlon0"></div>
<div id="datalatlon1"></div>
          ・・・・・・・
<div id="datalatlon18"></div>
<div id="datalatlon19"></div>

//*****個別メタデータの表示枠
<div id="nowdata"></div>
<div id="detel_metadata1"></div>
<div id="detel_metadata2"></div>

//*****地図表示ボタン
<input type="button" value="地図表示" onclick="MarkingDispData()">
```

【参考 table.html(抜粋)】

```
//*****都道府県データの読込み。のち待機
function initialize() {
   var httpObj = new JKL.ParseXML.CSV( "metadata/PrefList.csv" );
   Prefdata = httpObj.parse(); //***変数 Prefdata[i][5]に都道府県別メタデータのファイル名を格納した
}
//*****************全数表示
function AllDisplay(){
   dispdata = [];
   dispdata = metadata.concat(dispdata);
   maxnumber = dispdata.length;
   startnum = 0;
   if (maxnumber < 20){endnum = maxnumber; } else {endnum = startnum + 20; }
   DataDisplay(dispdata);
   document.getElementById("alldata").innerHTML = (maxnumber-1);
}

//******文字出力
function  DataDisplay(dispdata) {
   for (j = 0; j < 20; j++) {
      var textpos = "datalatlon" + String(j);
      document.getElementById(textpos).innerHTML = (" ");
   }
   for (j = startnum; j < endnum; j++) {
      htmlout = "<table border=¥"0¥" width=¥"100%¥" class=¥"underline¥" align=¥"center¥">";
      htmlout +="<tbody><tr><td width=¥"5%¥" class =¥"left3¥">" + String(j+1) + "</td>";
      htmlout += "<td width=¥"6%¥" class =¥"left3¥">" + dispdata[j][0]
                + "</td><td width=¥"16%¥" class =¥"left3¥">" + dispdata[j][1]
                + "</td><td width=¥"20%¥" class =¥"left3¥">" + dispdata[j][2].substring(0,32) + "</td>";
      htmlout += "<td width=¥"12%¥" class =¥"center3¥">" + dispdata[j][3]
                + "</td><td width=¥"15%¥" class =¥"left3¥">" + dispdata[j][4].substring(0,26)
                + "</td><td width=¥"14%¥" class =¥"left3¥">" + dispdata[j][6].substring(0,20) + "</td>";
      htmlout += "<td width=¥"12%¥" class =¥"left3¥">" + dispdata[j][10].substring(0,20)
                + "</td></tr></tbody></table>";
      var jj = j % 20;
      var textpos = "datalatlon" + String(jj);
      document.getElementById(textpos).innerHTML = (htmlout);
   }
   dispnumber = startnum;
   Infomation(dispdata, dispnumber)
}
//******詳細メタデータ関連
//******メタデータ表示　注 外部データを閲覧する場合は，インターネットに接続のこと
function Infomation(dispdata, k){
      htmlout1 = "<div><table border='0'>";
      htmlout1 +="<tr><td width='100'>引　用　先</td><td width='335'> "
                 + dispdata[k][0] + "  </td></tr>";
      htmlout1 +="<tr><td>固有コード</td><td> " + dispdata[k][1] + "  </td></tr>";
         ・・・・・・・・・・・・・・・
      htmlout1 +="<tr><td>掘削住所</td><td> " + dispdata[k][10] + "  </td></tr></table></div>";
      document.getElementById("detel_metadata1").innerHTML = (htmlout1);
      htmlout2 = "<div><table border='0'>";
      htmlout2 +="<tr><td width='100'>掘削標高</td><td width='335'> "
                 + dispdata[k][11] + " "+ dispdata[k][12] +"m</td></tr>";
      htmlout2 +="<tr><td>掘　進　長</td><td> " + dispdata[k][13] + "m</td></tr>";
         ・・・・・・・・・・・・・・・
      htmlout2 +="<tr><td>柱　状　図</td><td> ";
      htmlout2 +="<a href=¥"" + dispdata[k][21] + "¥" onclick=¥"window.open(this.href, 'mywindow',
                'width=1000, height=800, menubar=no, toolbar=no, scrollbars=yes'); return false;¥">";
      htmlout2 +="pdf の閲覧</a></td></tr>";
      htmlout2 +="<tr><td>ボーリングデータ</td><td> ";
      htmlout2 += "<a href=¥"" + dispdata[k][22] + "¥" onclick=¥"window.open(this.href, 'mywindow',
                'width=1000, height=800, menubar=no, toolbar=no, scrollbars=yes'); return false;¥">";
      htmlout2 += "XML のダウンロード</a></td></tr></table></div>";
      document.getElementById("detel_metadata2").innerHTML = (htmlout2);
      document.getElementById("nowdata").innerHTML = (k+1);
}
//******地図用ダイアログボックス表示とデータ転送。注 インターネットに接続のこと
function MarkingDispData(){
      senddatas = new Array();
      senddatas[0] = dispdata[dispnumber][9];
      senddatas[1] = dispdata[dispnumber][8];
      senddatas[2] = dispdata[dispnumber][5];
      senddatas[3] = dispdata[dispnumber][1];
      senddatas[4] = dispdata[dispnumber][10];
      reterndatas = window.showModalDialog("boringplot.html" , senddatas,
                "dialogHeight: 260px; dialogWidth: 440px; center: 1;");
}
```

【参考 table.js(抜粋)】