

非同期通信による高インタラクティブ WebGIS フレームワーク — TMAP2 の設計と実装 —

田中 龍一[†] 泉 陽一郎 岩根 一樹 井上 潮

[†] 東京電機大学大学院工学研究科 〒101-8457 東京都千代田区神田錦町 2-2

E-mail: [†] 06gmc13@ed.cck.dendai.ac.jp

あらまし WebGIS の多くは導入に当たって多大なコストがかかるため、エンドユーザとして利用することも、自らシステム構築し運営することは難しいという問題があった。そこで、我々はオープンソースソフトウェアを利用して、低コストなシステムを簡便に構築する方法を検討している。既存のオープンソース WebGIS フレームワークの多くは、サーバ側とクライアント側が同期を取って通信(同期通信)をするため、サーバからのレスポンスが返ってくるまでユーザは新たに次の操作をすることができない。WebGIS では、地図上で拡大・縮小を連続で行ったり、地図上を移動し続けたりというように連続的な操作が頻繁に行われるため、同期通信の使い勝手が悪いという問題があった。また、地図の提供機能と付加機能が混在しているため、新たな機能の追加や、既存 Web システムへの組み込みが困難であった。そこで本論文では、同期をとらない通信(非同期通信)を適用したインタラクティブ性が高く、既存の Web システムへの組み込みも容易な汎用的フレームワーク TMAP2 を提案する。

キーワード WebGIS フレームワーク, 非同期通信, Ajax, オープンソース

A Highly Interactive WebGIS Framework by Asynchronous Communication — A design and implementation of TMAP2 —

Ryuichi TANAKA[†], Yoichiro IZUMI, Kazuki IWANE and Ushio INOUE

[†] Graduate School of Information and Communication Engineering, Tokyo Denki University

2-2 Kanda-nishikicho, Chiyoda-ku, Tokyo, 101-8457 Japan

E-mail: [†] 06gmc13@ed.cck.dendai.ac.jp

Abstract Most of the current WebGIS are so expensive for installation that they are difficult for a user to build his/her own system rather than to use a ready-made system as an end-user. Using open source software is a solution to the problem. Most of the existing open source WebGIS frameworks, however, use synchronous communication between client and server, where the user cannot input a new command from the client before the response of the current command is returned from the server. Since continuous operations, such as zooming in and out or moving around, are very frequent, the synchronous communication declines the interactivity of the system. Their frameworks are also monolith, and mapping and additional functions are mixed-up. This makes difficult to add a new function or incorporate them to existing web systems. This paper proposes a new WebGIS framework, TMAP2, which is highly interactive due to the asynchronous communication between client and server, and is easy to incorporate to existing web systems due to the multipurpose organization.

Key words WebGIS Framework, Asynchronous Communication, Ajax, Open Source

1. はじめに

近年、GIS(Geographic Information System)が大手企業を中心として導入され、インターネット上では

Google Maps[1]を初めとする WebGIS が注目を浴びている。しかし、WebGIS の多くは、導入に当たって多大なコストがかかるため、エンドユーザとして利用することはできても、自らシステムを構築し、運営するこ

とが難しいという問題があった。

最近では、LinuxOS, Apache などのオープンソースソフトウェアを利用してシステムを構築することが盛んに行われている。これは、コストの削減が重要な問題であるためである。

そこで、我々はオープンソースソフトウェアを利用して、低コストな WebGIS を構築する方法を検討している[9]。オープンソースの GIS エンジンの代表的なものに MapServer[2],[3]がある。MapServer は、NASA がスポンサーとなり、ミネソタ大学で開発中の Web マッピングエンジンである。マルチプラットフォームの WebGIS サーバであるので、Unix や Linux, Windows, MacOS で動作が可能である。また、主要なデータベースである PostgreSQL, MySQL, Oracle などと連動させて動作させることも可能である。ただし、MapServer の機能は、地図データを元に地図画像を生成するだけであるため、MapServer とは別に Web 上で動作する WebGIS フレームワークが必要となる。

従来の WebGIS フレームワークの多くは、サーバ側とクライアント側が同期を取って通信(同期通信)する。この同期通信でも通常の Web ページであれば特に問題は無いが、WebGIS においては問題がある。同期通信は、ユーザが操作を確定させた状態(例えば、リンクを介してページの遷移を開始するような状態)において、サーバからのレスポンスが返ってくるまではユーザは新たに次の操作をすることができない。もし、レスポンスが返ってくる前に新たな操作を行うと、その前の操作に対するレスポンスは反映されず、新たな操作に対するレスポンスのみ反映がされる。WebGIS では、地図上で拡大・縮小を連続して行ったり、地図上を移動し続けたりというように連続的な操作が頻繁に行われる。そのような利用形態において同期通信を行うと、拡大処理や移動処理中に連続して操作ができないことになり、WebGIS としての操作性は非常に悪いものとなる。

また、従来のフレームワークは、地図の提供機能と付加機能が混在しているため、新たな機能の追加が困難であった。地図と各種機能が一体となってしまっているため、すでに構築済みの Web システムに WebGIS を組み込むなどの既存システムへの応用も困難であり、実際はフレームワークとは呼び難いのが実状である。

本論文では、同期をとらない通信(非同期通信)を適用したインタラクティブ性の高い WebGIS フレームワークを提案する。このフレームワークは、従来のフレームワークのような地図と各種機能が一体となって、一つのシステムとして完成したシステムを提供するのではなく、既存の Web システムへの組み込みも容易な汎用的フレームワークを目指したものである。以下、2

章で関連研究について、3章では、高インタラクティブ性の実現、4章では、WebGIS フレームワーク TMAP について、5章では、TMAP2 の性能評価、そして6章でまとめと今後の課題について述べる。

2. 関連研究

MapServer をエンジンとして利用する WebGIS フレームワークとして、欧米や日本で多く利用されているものに、p.mapper[4], Ka-Map[5]がある。しかし、p.mapper は、同期通信を用いるため、操作性が良いとは言えない。Ka-Map は非同期通信を用いているものの、一つの完成したシステムとなっているため再利用性が低く、新たに機能を追加したり変更することが困難である。また、どちらのフレームワークも既存システムやサービスとのマッシュアップが困難であり、Web システムとして十分なものとは言えない。GoogleMaps は、公開されている API[6]により既存システムへの組み込みが比較的容易で、操作性も良いため、現在多く利用されている。しかし、GoogleMaps 自体が Google のシステムに依存しているため、自ら地図データの変更や更新をすることはできないという問題点がある。

3. 高インタラクティブ性の実現

高インタラクティブとは、通常のインタラクティブよりもさらに高い双方向性と定義し、人間がコンピュータへ指示を与えていることを意識させない性質とした。具体的には、通常の Web ページよりも操作性向上に主眼を置き、ユーザに処理の待ち等のストレスを与えないような仕組みを実現することである。操作内容を確定させてから通信を開始する同期通信では高インタラクティブを実現することは難しい。同期通信では、通信中にユーザ側は待ち状態になり、処理が終るまでは次の操作をすることができないためである。また、同一のページ内では地図部分のみを更新することが難しく、ページ全体を更新しなければならず、データの送受信量が增大する。このことから、高インタラクティブな WebGIS にするための効率的な通信方法および処理方法を考える必要がある。

高インタラクティブを実現するために、次の3つのアプローチを採用した。

(1) 通信方法の改良

従来の通信方法であった同期通信の様子を図 1 に、我々の通信方法である非同期通信の様子を図 2 に示す。非同期通信を実現させる方法に Ajax(Asynchronous Javascript + XML)[7],[8]がある。Web ブラウザ上の操作を Ajax engine が受け取り、サーバへ送信する。通信は全て Ajax engine を介しバックグラウンドで行うため、

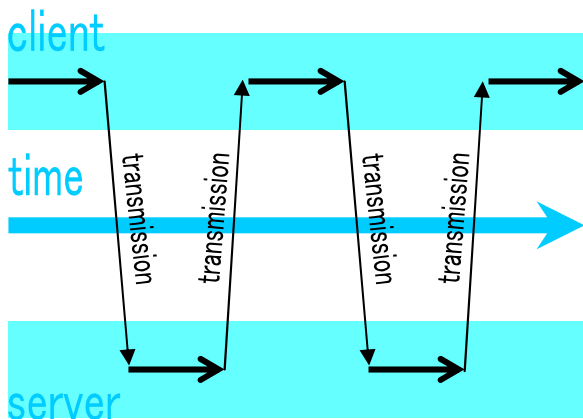


図 1 同期通信

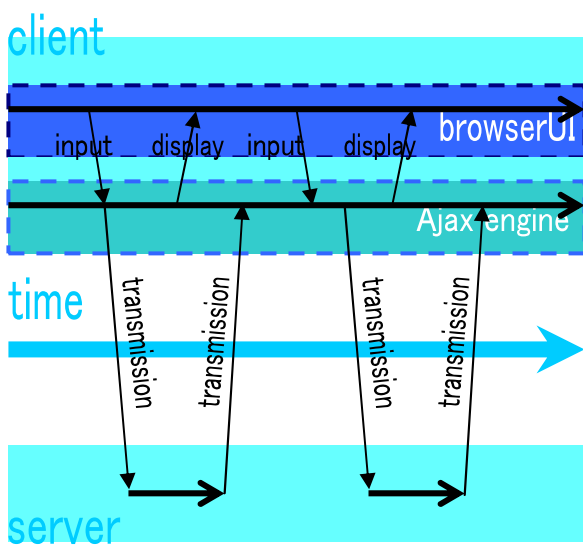


図 2 非同期通信

その間ユーザは待ち時間無く操作を続行させることができる。Ajax 自体は、既存技術の組み合わせであるが、近年、同期通信を用いた Web アプリケーションに対する使い勝手の悪さに対する不満や、XML や DOM(Document Object Model)などの Web 関連技術の標準化が進んでいること、高性能な Web ブラウザの普及などの要因により注目を浴びている。

(2) 通信データ量の削減

従来、WebGIS における地図画像を表示領域に対して 1 枚の画像として表示する方法を用いていた。これは、同期通信による WebGIS に適したものであった。我々は、非同期通信を用いるためこのような方法ではなく、地図画像を分割する方法を用いる。表示の際に分割画像をタイリングし、ユーザの操作(地図上の移動、拡大・縮小等)に応じて必要な画像のみの読み込みを行う。このとき、ユーザの操作による処理は出来る限りクライアント側にさせ、サーバとの通信データ量

は表示に必要な地図画像のみに抑える。このようにすることで、操作性向上およびサーバの負荷軽減になる。

(3) 描画処理量の削減

MapServer では、図 3 のように生成した地図画像は一時ディレクトリに保存されるが、一度使われるだけで再利用されない仕組みになっている。しかし、画像分割の効果を高めるためには、サーバ側においても再利用する仕組みが有効となる。そこで、我々は図 4 のように保存ディレクトリに地図画像を保存(キャッシング)する仕組みを設けた。一時ディレクトリに保存する方法とは異なり、地図画像に位置情報を表す ID をそれぞれ付与し、この ID を基に再利用する。この仕組みにより、一度地図画像を描画すれば次にその地点の画像が呼び出されるときは保存ディレクトリから読み出すだけで良いので、地図画像生成にかかるオーバーヘッドを減らすことができる。

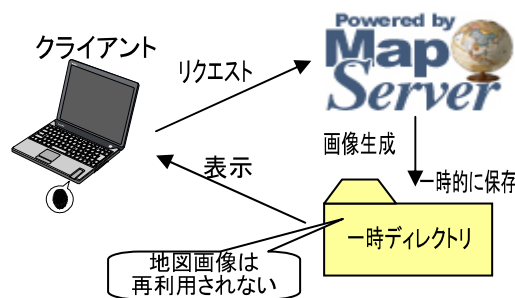


図 3 従来方式の画像生成と保存方法

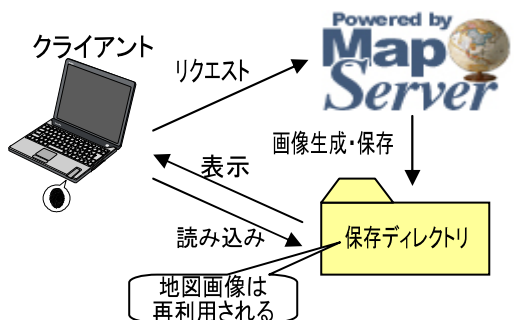


図 4 提案方式の画像生成と保存方法

4. WebGIS フレームワーク「TMAP2」

4.1. 高インタラクティブな地図画像生成・表示

前章で掲げた 3 つのアプローチを元に高インタラクティブな WebGIS フレームワーク「TMAP2(TDU Map Access Processor Ver.2)」を開発した。

TMAP2 は、Ajax の非同期通信を用いた従来の TMAP[9]に、タイリング機能を追加することにより、性能向上を図るものである。以下、高インタラクティ

ぶな地図画像生成・表示と地図操作の実現方法について説明する。

図 5 に TMAP2 を利用した WebGIS の処理の流れを、図 6 に TMAP2 の主な処理系を示す。

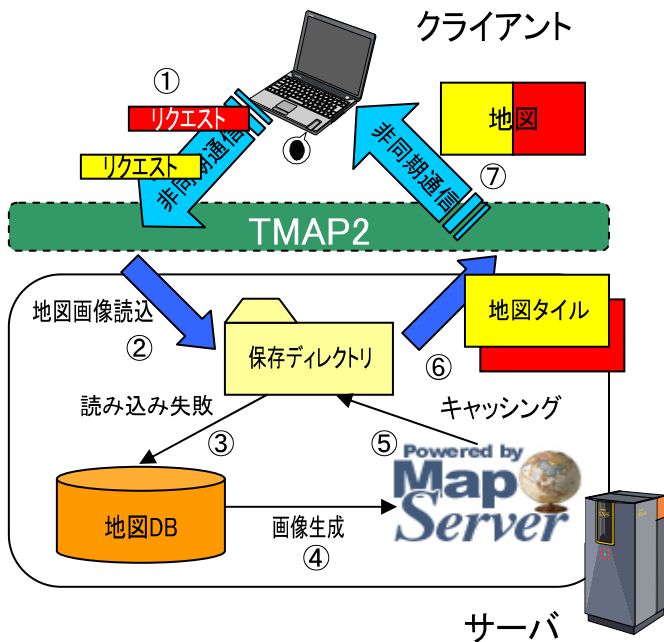


図 5 TMAP2 を利用した WebGIS の処理の流れ

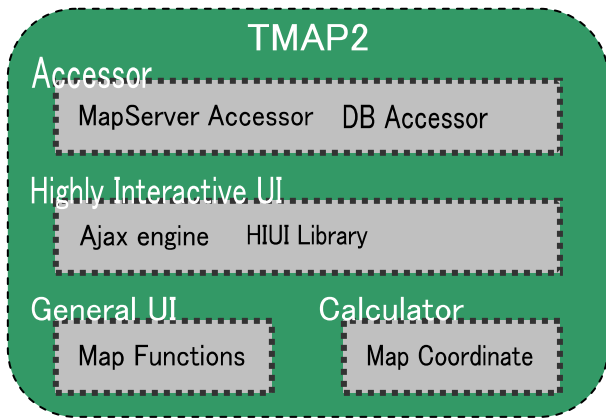


図 6 TMAP2 の処理系

TMAP2 では、アプローチ(1)の非同期通信を利用している。TMAP2 内に埋め込まれた Ajax engine がサーバへ送るリクエストには、

- 座標情報
- スケール
- 画像幅・高さ

が含まれている。これらの情報は、各地図タイルの領域が有している情報である。ただし、座標情報の扱いがクライアント側とサーバ側で異なる点に注意が必要である。これは、WebGIS では通常、座標空間は経緯度で管理されるが、Web ブラウザでの座標空間はピクセル(pixel)で表される。そのため、クライアント側からサーバ側へ座標情報を送るとき、サーバ側で処理する前に TMAP2 内で座標変換処理を行う必要がある。その後、変換された座標に対応した地図画像を保存ディレクトリから読み出す。しかし、座標に対応した地図画像が保存ディレクトリ内に存在しない、つまり、地図画像が未描画の場合がある。そのため、地図画像が保存ディレクトリに無い場合は、図 5 の

①→②→③→④→⑤→⑥→⑦

の順に処理を行い、地図画像が保存ディレクトリにある場合は、

①→②→⑥→⑦

の順に処理を行う。

4.2. 高インタラクティブな地図操作

(1) 地図表示領域の構造

TMAP2 では、地図画像を一枚の画像として扱うのではなく、タイリングして表示する。図 7 に TMAP2 の地図表示領域の構造の様子を示す。地図画像の表示領域は 2 つのフレーム「inframe」と「outframe」から構成される。これらの 2 つのフレームは、outframe が上層、inframe が下層にあるという関係にある。inframe は地図をタイル状に並べる領域、outframe は inframe を可視化して Web ブラウザに表示させるための領域である。inframe は、Web ブラウザから直接見えるわけではなく、outframe を通して表示する仕組みになっている。Web ブラウザ上においてマウス操作を行う場合、一つのオブジェクトしか操作することができない。つまり、outframe と inframe の両方を一度に操作することはできない制約がある。そこで、TMAP2 では inframe だけを操作する仕組みにしている。実際は inframe を移動させ、outframe は常に同じ位置にある。地図の移動操作をする際には、inframe 上を outframe が移動し Web ブラウザに表示されるように見えるので、ユーザにとっては、地図上を移動しているように見える。

(2) inframe 操作の伝播制御

図 8 は、図 7 の状態から東へ移動したときの様子を示している。地図上から見ると東へ移動したように見えるが、実際には inframe を左へ動かしている。この処理は、上位の階層への inframe 操作の伝播制御により実現することができる。この inframe の移動に従って新たに未表示領域が outframe を通して Web ブラウザに表示されるので、非同期でサーバへリクエストを送り

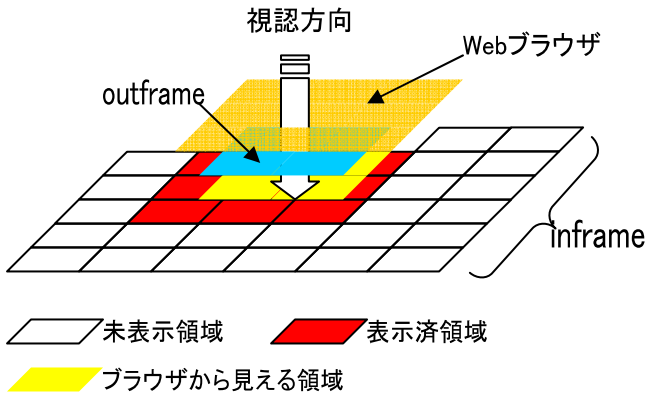


図7 TMAPの地図表示領域の構造

座標に対応した地図画像を読み込む。このとき、outframeに隣接するinframe未表示領域のみを読み出すため、画像を読み出したまたは描画のための通信データ量は最小限に抑えられる。

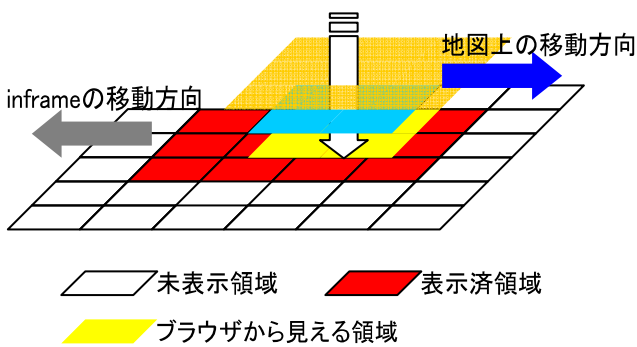


図8 地図上の移動と移動処理の方向の違い

このようにTMAPではinframeを操作する体系になっているが、このときのイベント伝播方向について注意を払う必要がある。イベントとは、Webブラウザ上で起こすことができるアクションのことで、マウスを動かしたり、クリックしたり、ボタンを押すなどを差す。onmouseover, onclickなど、それぞれのイベントには名前がついている。Javascriptの性質上、イベント発生領域が多層構造になっている場合は上層から下層または下層から上層へイベントが伝播[6]してしまう。つまり、inframeで発生したイベントを得る場合、outframeに伝播させずに得なければ、正しく動作しない。具体的には上記の地図の移動処理の場合、inframeをマウスでドラッグし操作をするが、もし、inframeで発生したイベント(ドラッグ)が上層のoutframeに伝播してしまうと、outframeまでドラッグしてしまい、表示が崩れてしまう。また、outframeの大きさ(表示上の枠の大きさ)は解像度にあわせて自動的にリサイズ、もしくは、管理者が任意の大きさに固定させることが

できるが、この仕組みが下層のinframeに伝播してしまったり、地図画像のタイリングそのものが崩れてしまう。このようなイベント伝播は、あらかじめ制御させておかなければならない。制御の仕組みを図9に示す。

イベントの伝播[10]方向は2つあり、DOM[11]ツリーの頂点から深い階層へ向かう方向をキャプチャ・フェーズ、DOMツリーの最も深い階層から順に上に戻る方向をバブリング・フェーズと呼ぶ。TMAPでの地図の移動処理は、イベントをキャプチャし、そのイベントを上位の階層へバブリングしないように制御している。

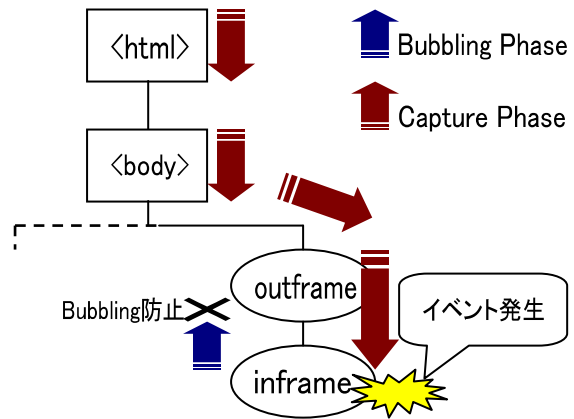


図9 イベント伝播の制御

4.3. プロトタイプシステムの構築

本章で述べた技術によって、地図のタイリングと地図画像の読み出しを効率よく行うことで、これまでのWebGISよりも操作性が向上する高インタラクティブ性を得ることができるということを確認するためのプロトタイプシステムを構築した。表1にシステムのサーバ環境を示す。

表1 プロトタイプシステムのサーバ環境

OS	FedoraCore4 Linux
Webサーバ	Apache 2.0.54
DBMS	PostgreSQL 8.0.3/ PostGIS 1.0.0
Mapエンジン	MapServer 4.6.0
使用言語	PHP 5.0.4 JavaScript 1.3(※1)
地図データ	国土地理院 数値地図 2500, 25000

※1 JavaScript1.3以上で動作確認
InternetExplorer6, FireFox1.5で動作確認

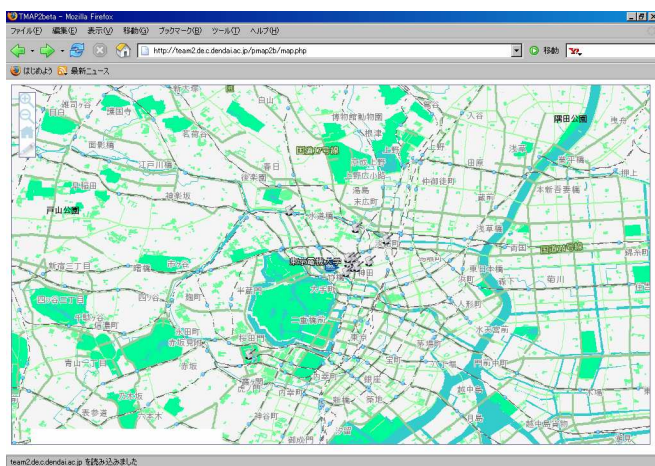


図 10 TMAP を適用した WebGIS

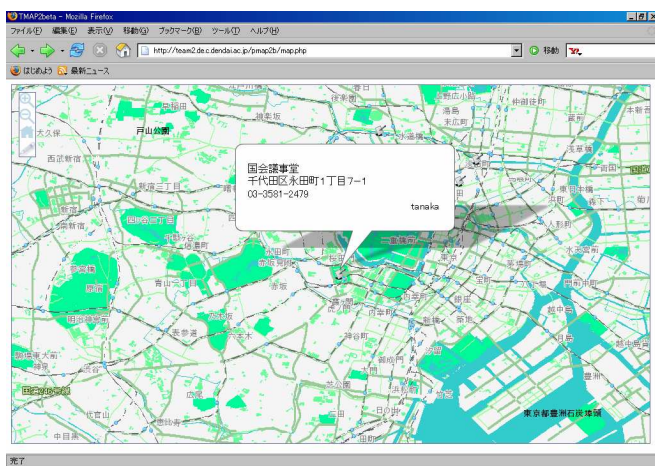


図 11 登録した地点のポップアップ

図 10 は TMAP2 を適用した WebGIS を Web ブラウザ (Firefox1.5) で閲覧している様子である。地図の部分以外に地図操作に必要な機能アイコン(図 10 左上)を表示している。TMAP2 では、以下の機能を実現している。

- (1) 地図の拡大
- (2) 地図の縮小
- (3) 初期位置へ戻る
- (4) 地点登録

(1), (2)は、地図操作で必要となる基本的な機能である。(3)は、地図の初期位置(ブラウザで初めに表示される位置)に戻る機能である。初期位置は、システムの管理者がパラメータを変更することで自由に変更することが可能である。(4)は、利用者が自由に地図上に地点情報を登録できる機能である。これらの 4 つの機能は、システムの管理者が必要に応じて、表示・非表示にすることができる。例えば、地点登録の機能が不要と考

える TMAP2 の管理者であるならば、パラメータを変更するだけで、機能から除外することができる。

これらの機能は、図 6 の General UI の Map Functions に属している。

このほかに TMAP2 では、機能を高インタラクティブ化するためのライブラリを新たに追加することができる。その一例を図 11 に示す。図 11 は、地点登録の機能によって登録された情報をポップアップさせるという、ライブラリである。これは、図 6 の Highly Interactive UI の HIUI Library に属している。登録した地点をクリックすると、地図中央までスムーズに移動し、登録情報がポップアップで表示される。

5. TMAP2 の検証

本章では、まずタイリングの効果がもっとも高いタイル長の検証を行い、そのタイル長を適用した TMAP2 と TMAP の操作性の比較を行い、タイリングと画像キャッシングの有用性を検証する。検証するシステム環境は表 1 の環境であり、画面解像度と、地図画像表示領域(outframe)の大きさは表 2 の通りである。

表 2 検証環境の画面解像度と地図画像表示領域

解像度	1024×768[dpi]
表示領域(outframe)	995×515[dpi]

5.1. 検証方法

(1) 最適なタイル長の検証方法

まず、タイルの幅・高さを変化させて outframe に表示されるタイルの総容量を測定する。このとき表示領域に表示されるタイルの総容量は、タイル 1 枚の実測容量に表示されるタイルの枚数を乗算して求める。また、タイルの枚数をもっとも効率よく表示されるときの枚数(最小枚数)で検証する。その結果から、タイリング効率が最も良いタイル長の検証を行う。

(2) TMAP2 と TMAP の比較方法

その後、そのタイル長を適用した TMAP2 と TMAP の移動処理の比較を行い、タイリング処理と画像キャッシングの有用性を検証する。それぞれ 1~4 回の移動処理からその処理時間を測定する。移動回数が複数の場合は、連続で処理を行う。移動時間は、地図移動時間、画像表示時間の合計時間とする。このときの測定における地図移動距離は全て同じとする。

5.2. 検証結果

(1) 最適なタイル長

図 12 にタイルの幅を固定して高さを変化、図 13 にタイルの高さを固定して幅を変化させたときの様子を

示す. 図 12 では, タイル高さ 100Pixel の時に特にタイル容量が大きくなっていることが分かる. これは, タイル単体では容量は小さいが, タイル数が多いためである. タイル数 T_{cnt} は(1)式で求めることができる.

$$T_{cnt} = RoundUp(O_w / T_w) \times RoundUp(O_H / T_H) \quad (1)$$

O_w , O_H , T_w , T_H はそれぞれ, outframe 幅, outframe 高さ, タイル幅, タイル高さを示す. 表 3 はタイルの大きさと表示されるタイル数 T_{cnt} の関係である.

表 3 タイル長と表示領域に表示されるタイル数

幅[Pixel] \ 高さ[Pixel]	100	200	300	400	500	600	700	800	900	1000
100	60	30	24	18	12	12	12	12	12	6
200	30	15	12	9	6	6	6	6	6	3
300	20	10	8	6	4	4	4	4	4	2
400	20	10	8	6	4	4	4	4	4	2
500	20	10	8	6	4	4	4	4	4	2
600	10	5	4	3	2	2	2	2	2	1
700	10	5	4	3	2	2	2	2	2	1
800	10	5	4	3	2	2	2	2	2	1
900	10	5	4	3	2	2	2	2	2	1
1000	10	5	4	3	2	2	2	2	2	1

タイル Width

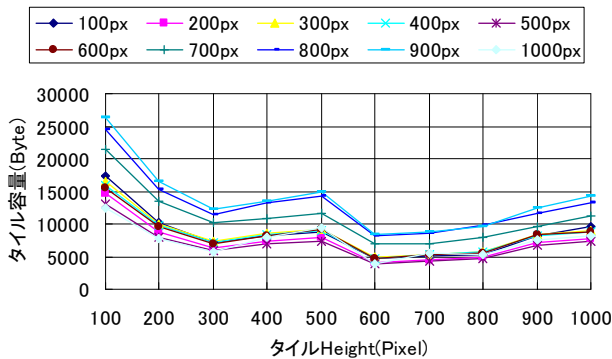


図 12 タイルの高さを固定させたときの総タイル容量の比較

タイル Height

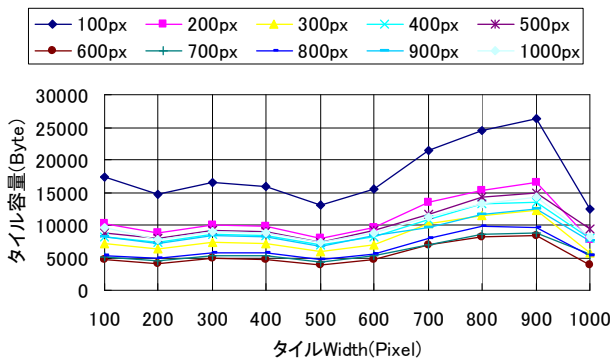


図 13 タイルの幅を固定させたときの総タイル容量の比較

図 13 では, タイル幅 700~900Pixel の時にタイル容量が大きくなっていることが分かる. これは, outframe の外に表示しきれないタイルがはみ出すためである. タイル幅 700~900Pixel は余剰部分が多いため, タイル容量多くなっている. また, タイルの幅が小さいほどタイル容量が増えているが, これは, タイル幅が小さいときはタイルが縦に長い形状になるためである.

また, 表 3 より, タイル幅がタイル高さよりも長いほうが表示タイル数が少なくなっている. 従って, タイルは長方形の形状になるタイルになるように設定するほうが良いと言える.

outframe はブラウザのウィンドウの大きさに合わせて可変する. outframe が小さければ, タイルは小さい方が余剰部分小さくなる. outframe が大きければ, タイルは大きいほうが表示タイル数は抑えられる. 本環境では outframe は 995×515[dpi] ほどなので, タイルは大きすぎないほうが良いと思われる.

これらのことより,

- (1) タイルは横長の形状にする
- (2) タイル幅は 600Pixel 以下にする
- (3) タイル高さは 100Pixel 以上にする
- (4) タイルは大きすぎないほうが良い

と言える. 従ってプロトタイプシステムではタイル幅 300Pixel, タイル高さ 200Pixel とした. このタイル長であれば, 余剰部分が比較的少なくタイル自体も大きすぎない. また, タイルの形状も長方形であるため(1)~(4)の条件を全て満たすことになり, 妥当なタイル長であると言える.

(2) TMAP2 と TMAP の比較

図 14 に TMAP2 と TMAP の移動処理時間の測定結果を示す. タイリング処理と画像キャッシングを行っていない TMAP に比べて, 行っている TMAP2 は処理時間が短いことが分かる. さらに, 処理が連続で行われるほど処理時間に差が生じていることも分かる. 従って, タイリング処理と画像キャッシングを実装することで, 処理時間が改善されたと言える.

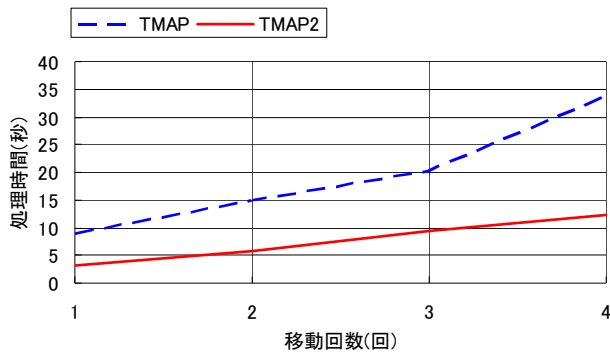


図 14 TMAP2 と TMAP の移動処理時間の比較

6. まとめ

本論文では、非同期通信とタイリング技術を用いた高インタラクティブ WebGIS フレームワーク TMAP2 を提案した。地図表示領域の 2 つの層である `outframe` と `inframe` によるタイリングを行うことによって TMAP よりも操作性が向上していることを示した。本論文では、新たに開発した TMAP2 の内部構成、設計方法と実装方法について特に議論した。そして、タイル長の最適値とタイリングと画像キャッシングの有用性を検証した。

7. 今後の課題

今後の課題としては、TMAP2 のライブラリの追加、TMAP2 の各種パラメータの最適化、そして既存システムへの組み込み方法の検討が挙げられる。現状では最低限の基本機能のみの実装となっているので、新たな機能の実装および高インタラクティブなライブラリの追加が必要である。本論文では、TMAP2 のタイリングの検証を行い、最適なタイル長を決定した。タイル長以外にも TMAP2 ではタイルの幅、高さ以外にも非同期通信のタイミングや地図移動時のタイルの再描画の判定などのパラメータを指定している。今後それらのパラメータについての検証をしていく必要がある。また、TMAP2 を ブログ、SNS、掲示板などとマッシュアップしていく検討していく予定である。

文 献

- [1] Google Maps
<http://maps.google.co.jp/>.
- [2] MapServer
<http://mapserver.gis.umn.edu/>.
- [3] A MapServer PHP/MapScript Framework.
<http://pmapper.sourceforge.net/index.shtml>.
- [4] p.mapper
<http://www.pmapper.net/>.
- [5] Ka-Map
<http://ka-map.maptools.org/>.
- [6] Google Maps API

<http://www.google.com/apis/maps/>.

- [7] Ajax
<http://www.scriptet.net/ajax-garrett.html>.
- [8] 羽田野太巳：“AJAX Web アプリケーションアイデアブック”，秀和システム，pp. 191-202 (2005).
- [9] 田中龍一，井上潮：“非同期通信による高インタラクティブ WebGIS フレームワークの研究”，第 15 回 GISA 学術研究発表大会 (2006).
- [10] XML Events
<http://www2u.biglobe.ne.jp/~oz-07ams/prog/xml-events/Overview-ja.html>.
- [11] Document Object Model (DOM) Technical Reports
<http://www.w3.org/DOM/DOMTR>.