再利用可能な地図データの考察と WWW用変換ツールの作成について

平成 17 年 2 月 10 日

情報電子工学科
齊藤 真理子

目 次

1 はじめに **1**

2 Unix の描画ソフト **1**

3 クリッカブルマップ **4**

4 SVG 画像形式 **6**

4.1 SVG とは 6

4.2 SVG の画像のサイズと表示 6

4.3 SVG の文書の構造 7

4.4 基本的な形状 8

4.5 テキストを描画する要素 9

4.6 グループ 9

4.7 座標系の変換 10

4.8 パス 10

4.9 リンクを作る要素 10

4.10 地図エディタに向いている機能 11

5 地図エディタを作成する GUI のツール **12**

5.1 Tcl と Tk とは 12

5.2 構造と機能 12

6 データを SVG で読み込む **14**

6.1 地図データ 14

6.2 プログラムと実行結果 15

7 HTML 形式への変換 **16**

7.1 プログラム 17

8 地図の一部分切り取り **17**

8.1 プログラムと実行結果 17

9 まとめ **18**

参考文献 **20**

概要

WWW 用の地図を作る場合、Unix 上には定番といえる簡単なソフトは見当たらない。Unix には一般的に、Tgif や Xfig、Xpaint 等の描画ソフトがあるが、それらの描画ソフトは地図を書くのに向いるとはいえず、またそのデータは再利用するのが難しい。そこで、本研究では、Unix 上で WWW 用の地図データを作成するツール、そしてそのデータを加工するツールとして、地図データを画像形式に変換するもの、地図データの一部分を切り取るもの、地図データからクリッカブルマップの HTML 部分を作成するもの、などの作成について考察する。

1　はじめに

研究室で、大学付近の飲食店マップを WWW 上で公開している。この地図は Tgif で作られており、道路や一部の建物だけを使って別な地図を作ったり、地図の一部分だけを切り出すのが難しい。もし、実際に道路だけを取り出そうとすると、建物を一つ一つ消していかなければならない。それと、Tgif ではクリッカブルマップが作りにくい。なぜなら、建物がある位置 (座標) を計算するのが大変だからである。以上のことから、Tgif は Unix の一般的な描画ソフトだが、地図を書くのには向いていないことがわかる。そこで、もっと簡単に地図を作成できるような専用のツールの作成について考察していこうと思う。

(現在公開されている地図を以下に示す)

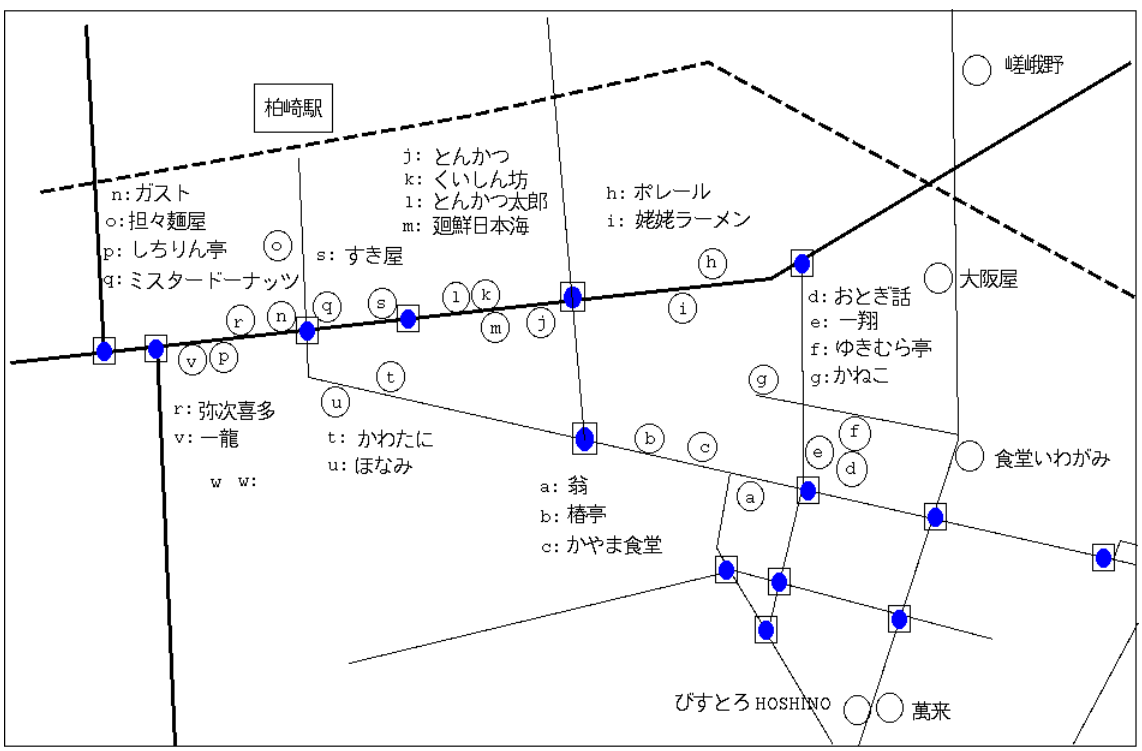

Fig. 1.1　研究室で公開しているマップ

2　Unix の描画ソフト

一般に図を作成する場合、Unix には Tgif、Xfig、Xpaint などの描画ソフトがある。Tgif と Xfig はドロー系の描画ソフトで、Xpaint はペイント系の描画ソフトである。次にそれ

ぞれの特徴を挙げる。

ドロー系描画ソフト　ドロー系描画ソフトとは、2 次元ドローイングツールで、図を線分や円などの図形要素の組み合わせで表現している。Tgif の画面の例を以下に示す。

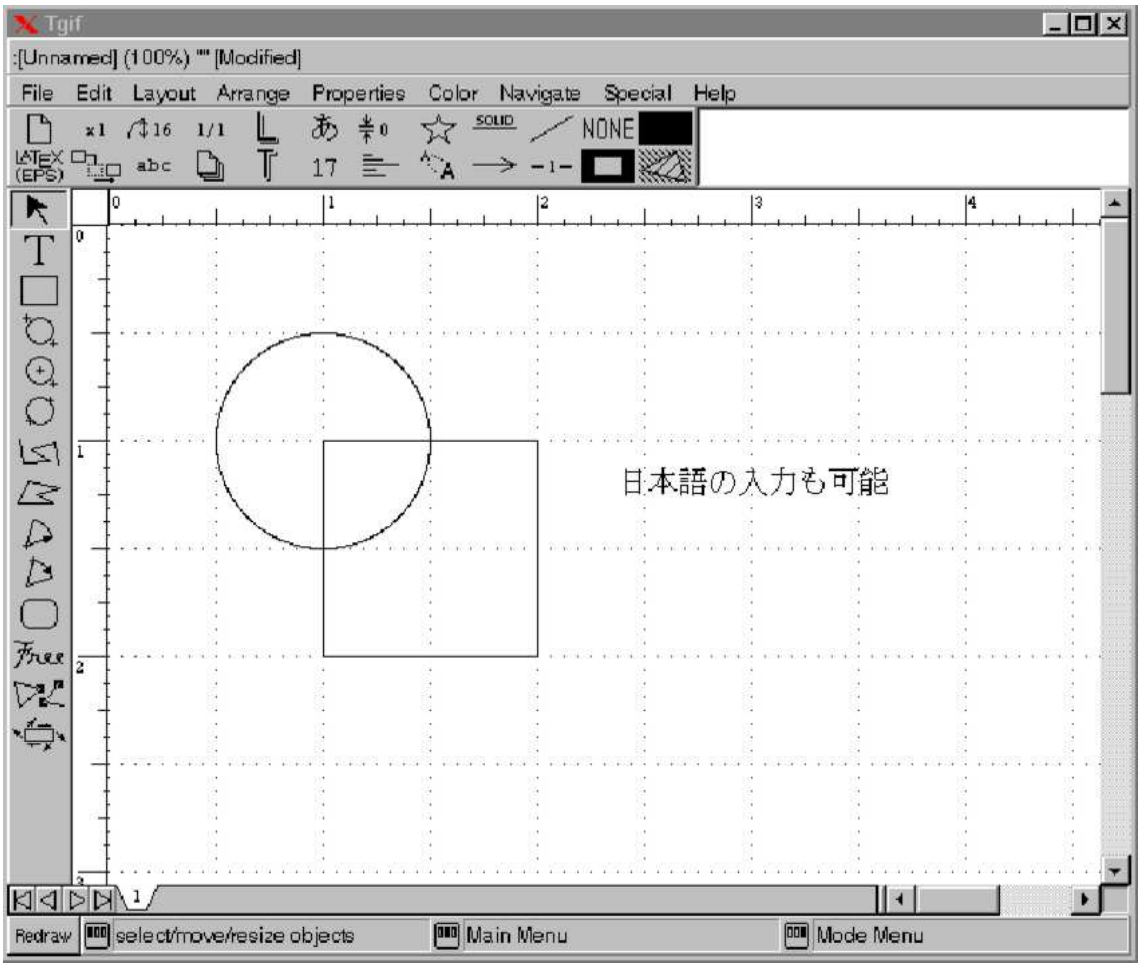

Fig. 2.1 Tgif

以下に Tgif の特徴を記す。

- 円、楕円、直線、折れ線、曲線、長方形などの部品が使えるが、部分的な削除等の操作が困難である
- 点線、矢印、塗りつぶし、拡大、縮小などの機能がある
- 色々な大きさ、色々なフォントの文字が書ける

- EPS、PS 等の形式の出力も行なえるし、逆に EPS ファイルを取り込むこともできる

Tgif で例えば直線や円の図を書いた場合には、「座標 (1,1) から座標 (2,3) までの直線と、中心点 (2,2) 半径 1 の円」というような形で情報が保存されてる。そのため、図形を書くのに適している。ドロー系の描画ソフトの Tgif と Xfig は、ほぼ同じ機能を持っている。

ペイント系描画ソフト　ペイント系描画ソフトが作る、ビットマップ形式の画像ファイルは、画面上の一つ一つの点に色がついていて、それが集まって絵のように見える。そのため、絵を書くのに適している。以下に Xpaint の特徴を記す。

- 点を一つ一つまで操作が可能。
- ブラシやエアーブラシ、ぼかしなどいろいろなペイントモードがある
- 様々な画像形式を利用できる (GIF、JPEG、PNG、PPM 等)

WWW 上で利用できる地図を、これらの汎用の描画ソフトで作成する場合地図を作成する問題点を以下に記す。

- 地図用の部品を Tgif で作っていくのは面倒
- Tgif の図は、一部分の再利用などがしにくい

  ドロー系の描画ソフトは図は 1 つ 1 つ部品で構成されているため、1 つの部品の途中を切ることができない。そのため地図を作成してから一部分の範囲を切り取ることなどができない。
  ビットマップ画像は、建物だけを取り出すようなことは難しいが、情報も入らないため建物の種類別に表示するような機能をつけることは不可能である。

- 地図の情報を入れられない。

  ビットマップ画像では見た目だけの地図となってしまい、建物の名前、場所、種類などの情報を入れられない。
  Tgif のデータも建物の情報等を入れることはできない。円ならどこに円があるかのような情報しか持っていない。

- クリッカブルマップを作りにくい

汎用のツールで地図を作成しようと考えた場合、これらの問題点がある。専用の地図描画ソフトがあるとよいのだが、MS–Windows にはあるが、Unix にはないのでそれを作成することを行なっていく。

地図を簡単に作成でき、なおかつ再利用可能にしなければいけない。そこで、研究室にある GUI の Tcl/Tk のツールを使用する。このツールは地図の情報を入力できる地図エディタで、GUI を使って、数値 (座標) を入力することなく、視覚的に部品を配置し、部分の配置データ (地図データ) を作成、編集できる。そして、このツールを使って得たデータをもとに、次のようなことができるツールを考えていこうと思う。

- 地図上の建物に、その建物の情報が見れるようにクリッカブルマップにする
- 地図データの一部分を切り取る、連結させる、レイヤー、のような機能をつける

3 クリッカブルマップ

クリッカブルマップとは、WWW ブラウザ内に表示された画像上をクリックすることで、指定した URL にジャンプできるような仕組みをいう。クリッカブルマップは、画像のいろいろな部分にリンク可能領域を設定して、一つの画像から複数の指定された URL にジャンプするようにできる。クリッカブルマップには、サーバサイドクリッカブルマップとクライアントサイドクリッカブルマップの二種類がある。

1. サーバサイドクリッカブルマップ

   サーバサイドクリッカブルマップとは、画像上のリンクの貼られた領域のデータやリンク先のアドレスのデータをサーバ側に記憶しておき、リンク領域がクリックされると、その場所からリンクの行き先を判断する処理をサーバ側で行なうクリッカブルマップのこと。

   使用方法は、ファイル形式が gif か jpg の画像を用意し、用意した画像のどこをクリックしたらどのページにジャンプする、などの指定を行なう。(Map Edit などの座標と目的 URL を指定する専用ソフトを利用すると簡単。) そして指定したファイルをマップファイルとして保存する。

   HTML の記述は、画像を表示させる HTML ファイルに次のような記述をする。

   ```
   <a href = "マップファイル名"><img src = "画像ファイル名" ismap></a>
   ```

   そして、マップファイル、画像、HTML ファイルをサーバにアップロードすれば、クリッカブルマップが動作する。

   長所は以下の通り。

   - サーバの好きなように作れること。
   - サーバ側で行われる処理は一般には CGI などのプログラムを使って行われるが、最近はクリッカブルマップ処理用のサーバプログラムがあるので、これが

用意された一部のプロバイダでは、特別なプログラムなしでサーバサイドクリッカブルマップが利用できる。

短所は以下の通り。

- 自分の登録しているプロバイダのサーバが、サーバサイドクリッカブルマップをサポートしていない場合には、CGI を使って作らなければならない。また CGI を使うにはプロバイダーの許可が必要となる。
- CGI を使うクリッカブルマップでは、サーバに対してリンク先を問い合わせる処理があるので、どうしても待ち時間が長くなる。
- CGI クリッカブルマップでは、サーバに問い合わせるまでリンク先はわからないので、リンク先を示すステータスバーには意味不明な座標の値が表示される。

2. クライアントサイドクリッカブルマップ

クライアントサイドクリッカブルマップとは、Web ブラウザ側で用意されている、クリッカブルマップ機能を利用して、イメージからのリンクを行なうこと。そのため、サーバサイドクリッカブルマップとは違い、どこををクリックしたらどのページにジャンプする、という指定を HTML に含めなければならない。HTML の記述は次のようになる。

```
<img src="画像ファイル" usemap="#MAP" border="0">
<map name="MAP">
<area shape="circle" coords="340,192,4" href="リンク先"/>
</map>
```

このように書けば画像ファイルの中心座標 (340,192)、半径 4 の円は「リンク先」にリンクされていることになる。

長所は以下の通り。

- HTML に指定する位置と URL を記述するため、後から変更する場合でも、HTML の修正だけで対応できる。またローカル環境でもきちんとリンクするので、確認が簡単に行なえる。
- HTML ソースにリンク先が書いてあるので、ダイレクトにジャンプできるため CGI を使ったものより早くできる。そして HTML ソースを参照されるので、ユーザーにもリンク先がわかるようになる。

短所は以下の通り。

- 座標でリンクする場所と範囲を決めるため、計算が大変である。
- 画像が全部ダウンロードされるまで使えない。そのため、ユーザーは少なからず待たされることになる。もし、画像の自動読み込みを外しているユーザーだとどこにも行けなくなる。

- リンクされている部分がはっきりわかる画像でないかぎり、マウスをいちいち動かさないと、どこにリンクポイントがあるのか、何ヶ所リンクしているのかわからない。

WWW で利用する地図を作っていく上で、簡単に修正や、確認が行なえるクライアントクリッカブルマップで作成するのがよいのではないかと考えた。それに、地図エディタから作成していくため、画像のデータを持っているので座標計算ができているから、短所も補える。そのため、CGI を利用するサーバサイドクリッカブルマップより、簡単に作れると思う。

4 SVG 画像形式

4.1 SVG とは

SVG 画像形式は、テキストファイルで線画データを表現する画期的な汎用の画像形式で、W3C が提唱している WWW 用に開発された画像形式である。二次元空間のグラフィックスを記述するための言語の 1 つで、XML という言語を土台にして作られた言語になっている。XML というのは、インターネットで交換される文書を作るための汎用的な言語である。特徴を以下に記す。

- XML 形式のテキストファイルでグラフィックを表現できるため、エディタとブラウザさえあれば開発できる
- 画像に直接リンクを貼れるため、クリッカブルマップが作りやすい
- ベクター画像なので拡大・縮小が容易である
- XML 形式なので SVG 文書を生成するプログラムが書きやすい

ブラウザが直接 SVG をサポートすれば、画像ファイルに直接リンクを記述してクリッカブルマップとできるようであるが、現在はまだ SVG をサポートするブラウザがあまりない。

4.2 SVG の画像のサイズと表示

SVG で書かれたグラフィックスをコンピュータの画面に表示するソフトウェアは、SVG のビューワーと呼ばれる。現在、SVG のビューワーとしては、Adobe SVG Viewer というものが最もよく使われている。このビューワーは、WWW のブラウザーのプラグイン

として動作するので、これをインストールすると、WWW のブラウザーのウィンドウの中に、SVG で書かれたグラフィックスが表示される。

SVG では、グラフィックスがその上に描画される平面のことをキャンバスと呼ぶ。SVG のキャンバスは無限の広さを持っているため、その全体をビューワーに表示させることは出来ない。ビューワーは有限の広さの長方形の領域をキャンバスから切り取って、その部分だけを表示する。ビューワーがキャンバスの一部分を切り取る長方形の領域はビューポートと呼ぶ。

4.3 SVG の文書の構造

SVG というのは XML を土台にして作られた言語なので、SVG 文書も XML の文書と同じように、XML 宣言、文書型宣言、ルート要素という 3 つの部分から構成されている。

- XML 宣言

  XML 宣言とは、文書を書くために使われている言語が XML であるということや、その XML のバージョンが何であるかと言うような記述。以下に XML 宣言の例を挙げる。

  ```
  <?xml version="1.0" encoding="EUC-JP" standalone="no"?>
  ```

  XML 宣言は必ず先頭に記述する必要がある。version の内容は常に 1.0 である。encoding で指定されるものは、XML 文書がどのような文字コードで記述されているかを明示するもの。standalone は、その XML 文書が、スタンドアロン文書かどうかを、yes か no で記述する。

  > スタンドアロン文書とは、外部のファイルを参照を処理しなくても、すべて正しい文書が得られる XML 文書を示す。
  > XML 文書を処理するアプリケーションプログラムにおいては、外部のファイルは処理されないかも知れない。その場合でも、正しく処理できる XML 文書であればスタンドアロン文書であるという。

- 文書型宣言

  文書型宣言とは、基本的には「文書型定義」Document Type Definition(DTD) と呼ばれるものから構成されている。文書型定義というのは、XML という言語を土台にして個別の言語を構築するための記述のこと。以下に例を挙げる。

  ```
  <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
   "http://www.w3.org/TR/SVG/DTD/svg10.dtd">
  ```

DTD というのはスキーマ言語の一種である。スキーマとは XML 文書を取り得る構造を記述したものである。つまり、要素や属性の配列に関して、正しい並び方と間違った並べ方をコンピュータ言語として、明確に記述したものという意味である。スキーマを記述するための言語を、スキーマ言語という。

DTD は、スキーマ言語としての機能のほかに実体参照などの定義も含んでおり、スキーマ言語以外の機能を利用される場合もある。

- ルート要素

  ルート要素とは、XML の文書の本体に相当する部分だと考えることができる。XML の文書の本体は、「要素」と呼ばれる文法上の単位が何段階にも入れ子になっている、という構造になっている。ルート要素というのは、入れ子構造になっている要素の階層のうちで最も上にある要素のことだ。

  要素は、「タグ」と呼ばれるテキストを使って作られる。タグというのは小なり (<) で始まって、大なり (>) で終るテキストのこと。要素は、いくつかの属性と呼ばれるものを持っていて、属性で要素に付加的な情報を与えることができる。

  ひとつの要素は、1 組の開始タグと終了タグのペアによって作られるか、またはひとつの空要素タグによって作られる。開始タグと終了タグの間に書かれたテキストは、要素の内容と呼ばれる。空要素タグは、それ自体が 1 つの要素となり、内容を持たない。以下に例を挙げる。

```
<svg width="340" height="280" viewBox="60 100 340 280"
 xmlns:xlink="http://www.w3.org/2000/svg">
```

  このような大きさの指定をして、この後に図形の要素を書いていくことで画像ができる。

4.4 基本的な形状

SVG では以下の基本的な形状を描画する要素が定義されている。

形状	要素
長方形	rect
円	circle
楕円	ellipse
直線	line
折れ線	polyline
多角形	polygon

基本的な形状を描画する要素は、すべて空要素タグを使って作る。そのタグの中にどんな属性指定を書けば良いかというのはそれぞれの要素型ごとに違うが、共通する属性指定の例を挙げる。

線の幅	stroke-width
線の色	stroke
内部を塗りつぶす色	fill

以下に要素を使った例を挙げる。

```
<circle cx="30" cy="40" r="4" fill="blue"/>
```

これを実行すると、座標 (30,50) に半径 5 の青い円が表示される。

4.5 テキストを描画する要素

テキストを描画したいときは、text という要素型の要素を書く。text 要素は、空要素タグではなくて、開始タグと終了タグのペアを使って作る。そうすると、text 要素の内容、つまり開始タグと終了タグとのあいだに書かれたテキストが描画されることになる。日本語の入力も text 要素を使うことで入力することができる。

text 要素を書くときには、次のような属性に対して属性値を設定する必要がある。

x	テキストの左端の x 座標
y	テキストのベースラインの y 座標
font-size	フォントの大きさ
fill	テキストの色

以下に例を挙げる。

```
<text x="30" y="50" font-size="12" font-family="serif" fill="black">
店の名前 </text>
```

このようなソースを作れば、座標 (30,50) の位置に「店の名前」と表示される。

4.6 グループ

SVG には、何個かのグラフィックスをひとつにまとめる機能がある。この機能を使えば、色とか線の幅とかの属性が共通する何個かのグラフィックスを描画する場合に、その

記述を完結にすることができる。

また、グラフィックスのグループには、名前を付けるということも可能である。グループに名前を付けておくと、その名前を指定することによってグループを描画することができるので、ひとつのグループを何回も使い回しするということが簡単にできるようになる。

4.7 座標系の変換

SVG でグラフィックスを描画するときに使う座標系というのは、ずっと同じものを使い続けないといけないというわけではなく、必要に応じて移動や拡大や回転などの変更を加えることができるようになっている。

座標系を変換したいときは、グラフィックスを描画する要素が持っている transform という属性に対して、どのような変換をするかということを指定する属性値を設定する。

4.8 パス

SVG では、直線または曲線の集合のことをパスと呼ぶ。パスは path という要素を使うことによって、キャンバスの上に描画することができる。

path 要素は、空要素タグを書くことによって作る。基本的な図形を描画する要素と同じように、path 要素も同じ属性を持っていて、それらに値を設定することによって、線の幅と線の色と内部の色を指定することができる。以下に例を挙げる。

```
<path d="m 10,10 l 70,10 c 100,40 50,40 30,40"
stroke-width="2" stroke="black" fill="silver"/>
```

path のあとに書かれている最初の d は、属性でパスを表しているデータのこと。次の m は、引数として 1 個の座標を受け取って、カレントポイントをそこに移動させる、という動作を行なう。そして、l は直線を書くためのコマンド。c は曲線を書くためのコマンドである。これを実際に実行すると以下のような図になる。

4.9 リンクを作る要素

SVG でも、キャンバス上のグラフィックスをアンカーとする別のデータへリンクを作ることが出来る。リンクを作りたいときは、a という名前の要素を使う。

Fig. 4.1 path 要素で作成した図

a 要素は、開始タグと終了タグのペアで作られる要素だ。a 要素の内容として何らかのグラフィックスを描画する要素を書くと、そのグラフィックスをアンカーとするリンクが作られることになる。以下に例を示す。

```
<a xlink:href="URL">
<circle cx="340" cy="192" r="4" fill="blue"/>
</a>
```

このようにすると、中心点 (340,192) の半径 4 の青色の円にリンクが貼られ、クリックすると URL へいけるようになる。

4.10 地図エディタに向いている機能

SVG の機能を調べると、地図作成に向いている機能が 2 つあった。

1. キャンバスの一部分を切り取る長方形の領域を指すビューポートの機能を使えば、一部分だけの表示が簡単にできる。
2. SVG では画像に直接リンクを貼れるためクリッカブルマップが簡単にできる。

1 の例を以下に挙げる。

300 × 300 の地図を作ったとする。その中の一部だけ表示したいと考えた場合、ビューポートの指定を座標 (100,100) から 100 × 100 の範囲と設定するだけで、真ん中の部分だけ表示することとなり、見た目では切り取ったようになる。

このようにすれば、一部分の切り取りができ、再利用が可能になる。しかし、地図に向かないところもあった。それは、画像を描くには形と座標をソースに書かなくてはならな

い。描画ソフトなら目で確認しながら、その位置に書き込んでいけばいいのだが、それができないので地図を作るのに手間がかかってしまう。

そこで、研究室にある GUI のツールを使って作成した地図エディタのデータを読み込んで、SVG 画像にするということを考えた。

5 地図エディタを作成する GUI のツール

研究室にある GUI のツールは Tcl/Tk で作られている。まず、Tcl/Tk について説明たあとに、GUI のツールの構造について説明していこうと思う。

5.1 Tcl と Tk とは

Tcl とはツール・コマンド・ランゲージの略で、C 言語のライブラリ・パッケージとして組み込め、かつ汎用な目的に使えるようにと開発されたスクリプト言語である。

そして、Tk とはツール・キットの略で UNIX の X Window の環境下で GUI に必要な部品を、Tcl で簡単に扱えるようにする目的で作られた。このように Tcl と Tk は通常ペアで用いられる。

Tcl/Tk は、C 言語などのコンパイラと違ってスクリプト言語である。Tcl/Tk では平易なコマンド郡によって GUI アプリケーションが実現でき、しかも言語の拡張性に優れているため、多くの UNIX ユーザーに支持されている。

Tcl はスクリプト言語であるため、コンパイル・リンクの手間が省けて即実行でき、プログラム開発期間を大幅に短縮することができる。また、Tk によって簡単に GUI を実現できるのが最大の特徴である。GUI のための widget(部品) がいくつも用意されており、簡単なスクリプトを書くだけで、基本的な機能が実現できる。

しかし、スクリプト言語はコンパイルされたプログラムより実行速度が遅いことが弱点である。

5.2 構造と機能

このツールは、建物と道をマウスで位置を決め、書いていくことができる。建物は公的施設と食料品店、道路は一般道路と幹線道路の二種類ずつあり、それぞれ名前をつけることも可能である。できあがった地図はテキスト形式のデータファイルとして保存される。

簡単に使い方と機能について説明する。まず、実行時に入力ファイル名を指定する。入力ファイルはテキスト形式のデータファイルで、この入力ファイルのデータを読み込み起動する。この入力ファイルは次のような形になっています。

size 行	データの左下の座標と右上の座標を表す
build 行	建物一つを表す
road 行	道路一つを表す

上の表を実際に入力ファイルを作成する時の書き方の例を以下に示す。

```
size 0 0 100 100
build "病院" 20 30 h
road "" 0 25 100 25 n
road "" 80 0 80 100 w
build "ローソン" 75 20 c
```

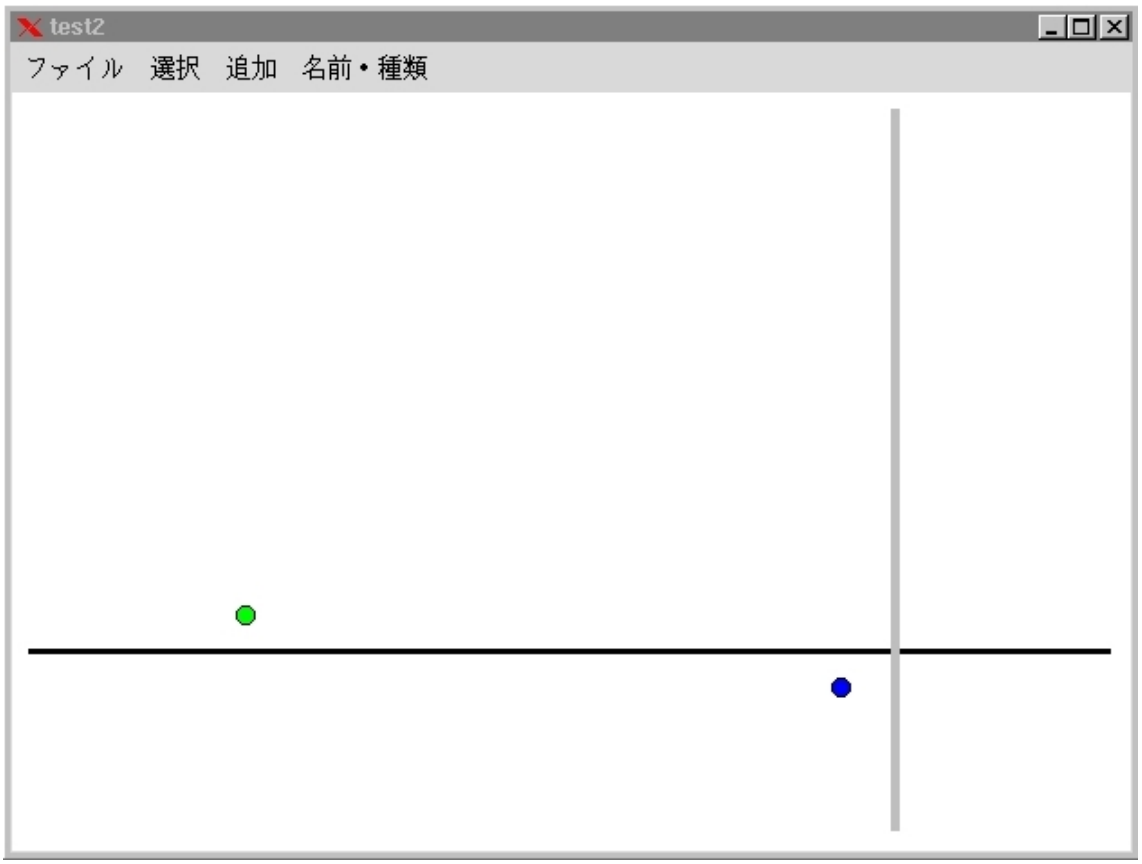

Fig. 5.1 例の入力ファイルを読み込んだ場合の画面

新規に一からデータを作りたい場合は、size 行のみを書いたデータファイルを作成し、入力ファイルとすればよい。

次に、メニューは以下の通りになっています。

ファイル	作成したデータを保存する (ファイル名を指定して保存)
選択	図形を修正または削除する場合に、どの図形かを選択する
追加	新規に建物や道の作成をする
名前・種類	図形の名前、種類を設定、選択する。

6 データを SVG で読み込む

GUI のツールによって作成した地図データを加工するツールとして、SVG 画像形式にするツールを作成しようと思う。AWK でこのデータを読み込み、SVG 画像形式に変換しようと思う。

6.1 地図データ

GUI のツールで得た地図データは以下のように表示される。

size 行	画像のサイズ	minx miny maxx maxy
build 行	建物を表す	”名称”建物の位置 [x,y] 文字の位置 [x,y]
road 行	道路を表す	[x1,y1] [x2,y2] [x3,y3]

今回のテストで使用した地図データを以下に示す。

```
size 0 0 400 385
build s "ゆきむら亭" 340 192 -5 15
build s "一翔" 315 215 -5 15
build s "おとぎ話" 340 215 -5 15
build s "ガスト" 19 133 -18 -8
build s "ミスタードーナッツ" 40 125 -7 15
build s "担々麺屋" 20 110 -18 -8
build s "スキ屋" 80 150 -5 15
build s "とんかつ太郎" 133 133 -18 -8
build s "くいしん坊" 145 133 5 3
build s "廻鮮日本海" 145 150 -5 15
build s "ほなみ" 50 222 -5 15
build s "かわたに" 140 210 -5 15
```

```
build s "無尽蔵" 220 150 -10 15
build s "姥姥ラーメン" 250 150 -5 15
build s "翁" 280 260 -5 15
build s "椿亭" 240 235 -15 -8
build s "かまや食堂" 270 240 -18 -8
build s "かねこ" 285 175 -5 15
road 3 "" 50 70 300 50 398 123
road 2 "" 2 142 272 142 398 100
road 1 "" 300 132 300 350
road 1 "" 280 180 390 180
road 1 "" 390 103 390 350
road 1 "" 200 30 200 350
road 1 "" 390 267 2 207
road 1 "" 30 80 30 212
```

6.2 プログラムと実行結果

地図データを読み込み SVG に変換する AWK のプログラムについて説明する。地図の基本情報はプログラムに直接書き込んでおき、建物の位置や道、駅などはデータを読み込んで表示するようにする。

「build」は建物と判断し、「s」がついている場合は日本語入力される。最初の 2 つの数値は、x 座標と y 座標になっており、後ろの 2 つは建物の名前を x 座標または y 座標からいくつ変えるかを示したものになっている。建物の場合、基本情報に建物は大きさ 4 の円で表す、というふうに書いてあるので、これで円とそこから数値だけ離れた場所に名前が表示される。

地図データの「build s ”ゆきむら亭” 340 192 -5 15」を読み込むと

```
<circle cx="340" cy="192" r="4" fill="blue"/>
<text x="335" y="207" font-size="10" font-family="serif" fill="black">
 ゆきむら亭
</text>
```

と変換される。

「road」は道と判断し、次の数字で道の基本情報が判断される。1 は普通の道で黒で表示され、2 は少し大きい道で太く表示される。3 は線路で色が灰色となる。そのあとの数値は x 座標、y 座標と並んでおり、点を取って線で結ぶ直線が表示される。

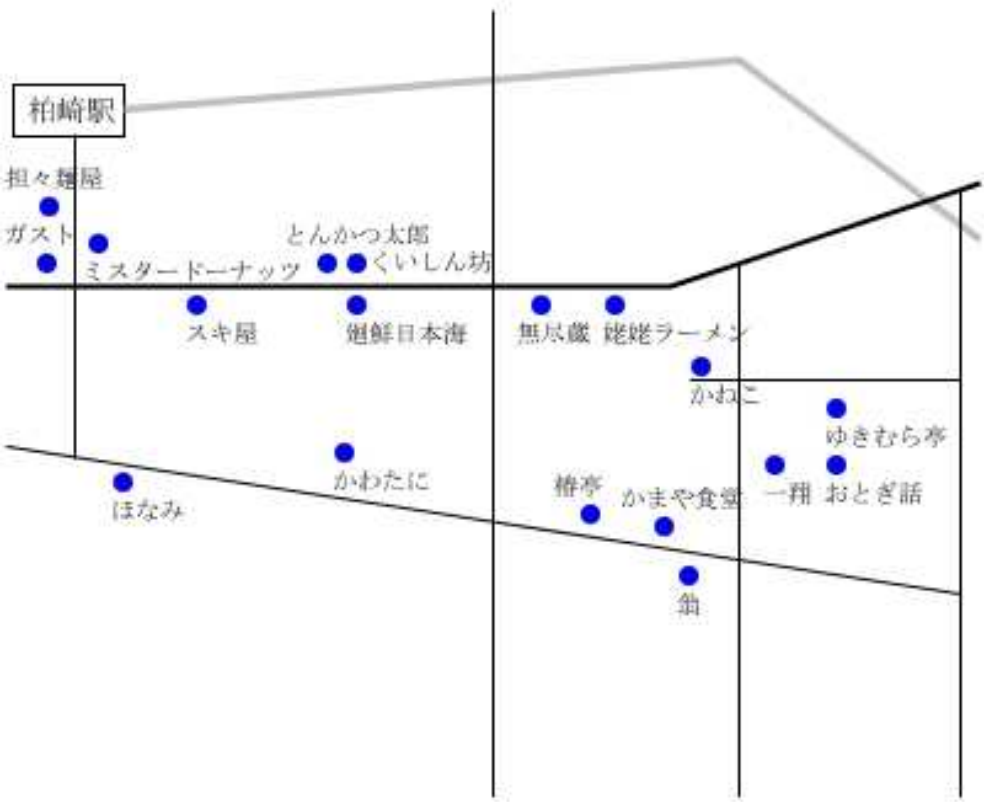

Fig. 6.1 SVG で作成された地図を squiggle で表示させたもの

squiggle とは

batik という SVG のビューワソフトも含む java のソフト郡のコマンドの 1 つで、SVG ビューワソフトが squiggle である。batik は WWW サーバソフトの開発元として有名な Apatch の XML project が出しているそふとで、java で書かれています。

7 HTML 形式への変換

SVG で地図を作成することはできたが、SVG はあまり普及しておらず、対応するブラウザが少ない。これでは作成した地図も意味がない。そこで、SVG 画像形式を HTML 形式へ変換しようと思う。まず、SVG で作成した地図の画像形式を、squiggle で PNG 形式に変換し、AWK で HTML 形式にする。SVG では図に直接リンクを貼れたのに対し、

HTML 形式ではそれができない。そのため、座標を計算し、範囲を指定してクリッカブルマップを作らなければならない。しかし、座標は地図データにあるので問題ない。範囲は SVG で建物を表した円の大きさと同じにすれば、クリッカブルマップが簡単にできる。

7.1 プログラム

PNG に変換した画像に、AWK で HTML 形式にするプログラムについて説明する。HTML 形式の基本情報を AWK のプログラムに直接書き込んでおき、建物をクリッカブルマップにする。

建物にクリッカブルマップをつけるので、「build」の最初の 2 つの数値だけを読み込む。2 つの数値は x 座標、y 座標とし、範囲は基本情報に入力してある数値を出力してクリッカブルマップの座標と範囲にする。PNG に変換した画像を表示させるので、形や範囲は SVG で指定したときと同じにする。

8 地図の一部分切り取り

SVG で述べた通り、SVG のビューポートの範囲指定で、一部分の切り取りも可能になるだろうと考えた。実際に実行してみたところ、切り取った地図の中に、切り取って範囲の外になった店の名前の半分など、余分なものも表示されてしまった。これでは綺麗に切り取れていないので、範囲外になった建物は表示しないようなプログラムを AWK 作ろうと思う。

8.1 プログラムと実行結果

ビューポートの座標指定の範囲外の建物を、表示しない AWK のプログラムを説明する。SVG 画像形式では始めにビューポートサイズの指定を行なう。ビューポートのサイズ指定は左上の x 座標と y 座標、右下の x 座標と y 座標を入力することで四角に表示される。一部分を切り取る場合、この数値の指定をしなければならない。そこで、その数値の範囲外の建物をデータエラーとしてはじくことにした。

左上の座標を (minx,miny) とし、右下の座標を (maxx,maxy) とし、その範囲外の建物をはじくプログラムを以下に示す。

```
($1 ~ /build/){
    if($2 == "s"){ # 店
```

```
        if(NF != 7) next; # データエラー
        if($4<minx || $5<miny) next; # データエラー
        if($4>maxx || $5>maxy) next; #データエラー
name = $3; gsub(/\"/,"",name); # " を削除
putshop($4,$5,$6,$7,name);
    }
    next;
}
```

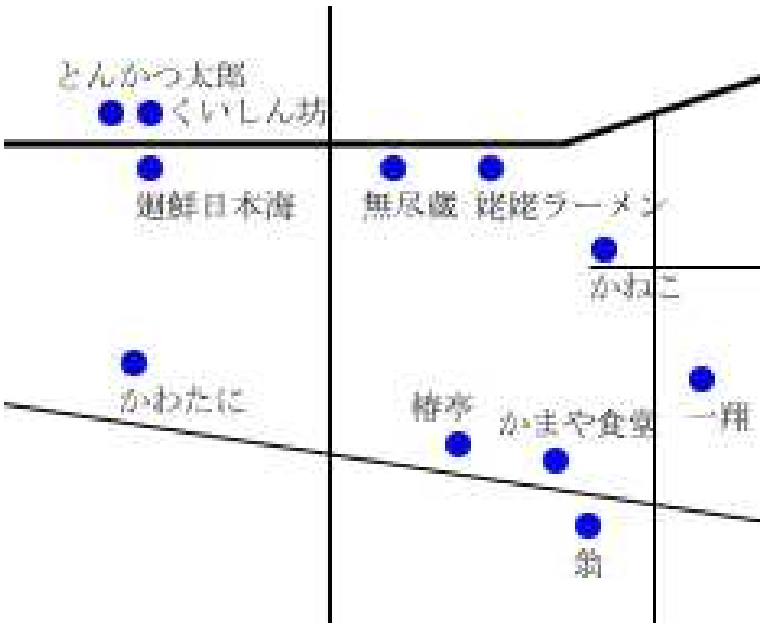

Fig. 8.1 SVG で作成された地図の一部分を切り取り squiggle で表示させたもの

9 まとめ

地図を作成するために様々な描画ソフトや画像形式について考察した。本研究は、座標をもつ単純な地図データからの WWW 用のファイル等の作成だったので、クリッカブルマップに重点を置き直接リンクを貼れる SVG 画像形式を使った地図作成をすることにした。地図データから awk を使用し、SVG 画像形式に変換し、squiggle で表示した。

しかし、SVG 形式はまだ広まっておらず、対応するブラウザも少ない。SVG 形式に対応するブラウザにするには、Adobe 社の SVG viewer というツールをブラウザにプラグインするという方法がある。しかし、このツールを確認してみたところ、リンクされている場所をクリックしても何も起こらなかった。このツールは、リンクに対して完璧にはサ

ポートされてないようである。そのため、SVG 形式のまま WWW ページに公開しても、見れる人は限られてしまうし、リンク先にジャンプすることはできない。そこで今回は、従来の HTML 形式に変換するために、SVG から PNG 画像と、地図データから HTML データを出力するツールを作成した。

まず、画像変換は squiggle を使って対話的に画像変換をした。だが、Unix の利点である、コマンドツールでの自動的に画像変換を行なえるように改良する必要があるが、squiggle に画像変換のコマンドツールがあるが、今回はそこまで至らなかった。

次に、HTML データは awk を使って作成した。クリッカブルマップに関しては、地図データに座標が入っているので、位置の計算をすることなくできた。範囲は、使用する画像が SVG から PNG 画像に変換したものなので、SVG 形式画像の建物の図の大きさとあわることで、簡単に指定できた。

再利用に関しては、SVG の機能を利用し、地図の表示したい部分の座標を入力し、範囲外の建物を表示させないという方法で、一部分を切り取ったように表示することができた。

今後の課題は、上に挙げたコマンドツールで自動的に画像変換を行なうことや、例えば、私が今回作った地図は駅の南側だが、駅の反対側の地図を同じように他の誰かが作ったとする。それを、一緒にして一枚の地図にすることができたら便利だと思う。しかし、縮図をあわせる、道をあわせる、それぞれ作成された地図の重なっている部分はどうするか、など色々な問題がある。だが、このような機能があれば、複数人で地図を分担して作成できるようになり、便利である。

参考文献

[1] アドビの SVG に関するページ http://www.adobe.co.jp/svg/main.html

[2] 無料チュートリアル「SVG 実習マニュアル」http://www.summa.jp/tutorial/

[3] 須栗歩人 「Tcl/Tk」 (昭和システム)

[4] 磯野康孝、蔵守伸一 「HTML タグリファレンス」(ナツメ社)