

骨格地図実用のためのシステム設計

川端 秀和[†] 久保田 光一[‡]中央大学大学院 理工学研究科 情報工学専攻^{†‡}

要約: 本稿では、骨格地図記述による略地図 [3] を WWW 上の地図 (以下、実地図) と結びつけ、地下鉄の駅や建物内部などの 3 次元構造と実地図の連携システムを設計・開発した。骨格地図記述の XML 表現を導入し、実地図との連携には GoogleMaps API を用いて、WWW 上で実地図から骨格地図へ、骨格地図から実地図への連携システムを構築した。

キーワード: 骨格地図, 略地図, 実地図, GoogleMaps API

1 背景

詳細な地図としての能力を残しながら、道案内などのランドマーク等の位置関係を把握できるように特化した略地図を骨格地図と呼び、その骨格地図を記述する言語を骨格地図記述言語と呼ぶ。骨格地図記述言語は、基本的には場所と場所との繋がりを記した言語だが、図 1 のように簡単な図を描画することもできる。

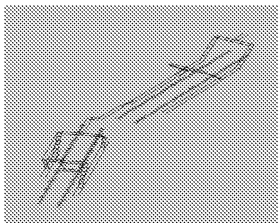


図 1 骨格地図ビューワで描画された浅草駅

文献 [1, 2, 5, 6] では、新しい命令の追加や、データ量の最適化等が行われた。実際にサンプルデータを作り、データ量の評価や記載時間についての研究も行われたが、実地図との連携による広域探索などの利用方法に関する研究は途上である。

2 目的

本研究では、過去に作成された地下鉄各駅構内のデータを利用し、GoogleMap の地理データとの連携を実現させ、骨格地図記述言語の実用性を高めることを目的とする。具体的には、地理データとの連携により、位置情報の共有、出入口の対応付け、マウスクリックによる次の駅へのリンクの実現と、スペース・ランドマーク等の表示・非表示の任意指定の機能の実装である。

3 骨格地図記述言語

骨格地図記述言語とは、位相関係の記述を重視した言語である。以下にその概略を示す。

3.1 アロー

アローとは、ある地点間を結ぶ道を示す命令である。その道の間には幾つもの地点を含むことができ、いわゆるグラフネットワークの枝とは異なる。アローは必ず向きを持っている。ま

たアローは、道 (road)、階段 (stair)、線路 (rail) といった属性を持たせることができる。

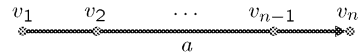


図 2 アロー命令の例

図 2 では、 v_1 から v_n に向いている、 n 個の地点を含むアロー a の例を示す。これを実際に記述すると、

(arrow a ($v_1 v_2 \dots v_{n-1} v_n$) road)

のように記述することができる。

3.2 ジョイント

ジョイントとは、交差点や、道の曲がり角等の道の節目になる点を示す命令である。ジョイントはアローの出入りの情報を保持していて、基準となるアローを定めたらそのアローから反時計回り (右回り) の角度を同時に保持している。

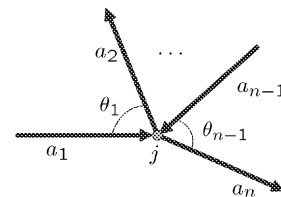


図 3 ジョイントの例

図 3 では、 n 本のアローが出入りしているジョイント j の例を示す。これを実際に記述すると、

(joint j ((- $a_1 \theta_1$) \dots (- $a_{n-1} \theta_{n-1}$) (+ a_n)))

のように記述することができる。

4 描画プログラムの仕様変更

文献 [2] で作成された描画プログラムは、簡易的に描画することのみを目的として作成された。ユーザが操作できる部分は、ズーム・回転だけである。本研究におけるシステムでは、描画することで動的に既存の実地図と連携することができ、ユーザによる操作から反応を返すことで実用性を見出すものを想定している。よって描画プログラムの抜本的見直しと改良が必要である。

また、文献 [2] での描画プログラムは、Z 軸を 2 次元平面に落とし込んだ疑似 3 次元表示によるものであり、その後の研究 (文献 [1, 5, 6]) もまたこの描画プログラムで事足りていた。この描画プログラムのまま本システムを実装しようとした場合、ランドマークの対話的追加や、ジョイントからの情報取得、位置情報の共有等を別途実装する必要がある。そこで本研究では Java3D を用いて描画プログラムを再実現した。それにより、ジョイントを Shape として定義し、マウスクリックによる情報取得、位置情報の共有を可能にした。

Practical system for Skelton Map

[†] Hidekazu KAWABATA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

[‡] Koichi KUBOTA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

5 XML フォーマットへの変換

GoogleMap との連携を効率的に行うため、また今後の拡張性を鑑みて、今まで S 式フォーマットであった骨格地図記述言語を XML フォーマットへ変更した。基本的には今までのものと記載されていることは変わらないが、第 6 章で説明するとおり、各種描画部品はプラグインとして切り分けた。

6 描画部品のプラグイン化

スペースやランドマーク等のような、略地図における部品の類を骨格地図記述言語から一度切り離すことで、略地図としての描画する際の柔軟性をあげることができると考えた。そこで、XML フォーマットに変更した際に、必要であれば描画するといった形式をとった。これにより本来の骨格地図記述言語が持つ“繋がり”を重視し、必要であれば作りこむこともできるといった拡張性を持った言語になった。XML 要素の階層構造を簡易的に記すと、図 4 のようになる。

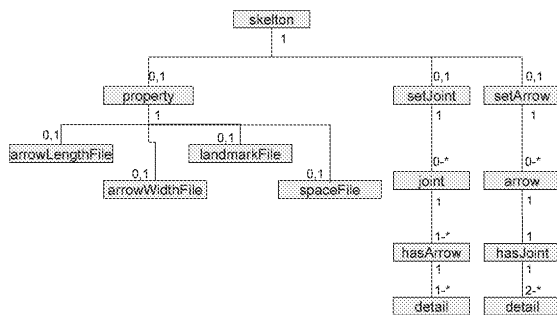


図 4 XML 要素の階層構造

setJoint 要素, setArrow 要素以下に今までの joint 命令, arrow 命令を保持し, property 要素以下で必要とあらば読み込むファイルを記載する。

7 システムの概要

本システムは、実地図では表現できていない建物内の繋がりをユーザに提供することを目的としている。今回は文献 [1, 2, 5, 6] で作成された地下鉄各駅構内のデータをもとに、地下鉄構内の繋がりを WEB ブラウザ上で実現した。建物内の繋がりは骨格地図記述言語を使用し、実地図は GoogleMap を利用する。現在は丸の内線を中心に作業している。

7.1 システム構成と動作の流れ

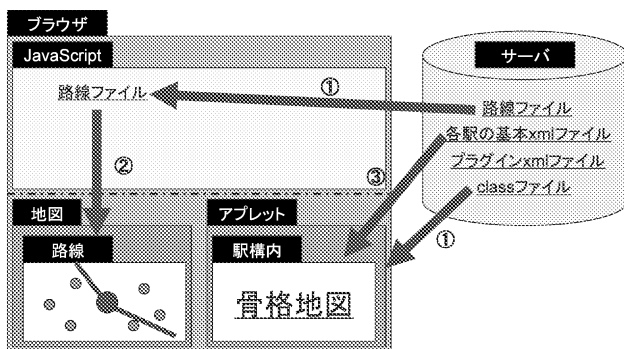


図 5 システムの構成

図 5 にシステムの構図を示す。サーバ上には丸の内線各駅の繋がりと出入口の緯度経度が記載されている路線ファイル, 各駅構内の繋がりを保持している xml ファイル, 各種プラグインファイル, そして描画アプレットの class ファイルが保管されている。丸の内線各駅と出入口の緯度経度は文献 [4] より手動取得したものである。ブラウザでページをロードした時に、路線ファイルがページ内の JavaScript に読み込まれ、ユーザの目に見える部分には GoogleMap とアプレットが表示される。実地図上には線路のラインが引かれ、駅にはマーカーが置かれている。ユーザはマーカーを選択すると、マーカーが保持している xml ファイルがアプレットに読み込まれ、Java3D による駅構内の 3 次元表示が現れる。

8 まとめと今後の課題

実地図と骨格地図との連携の実現と、骨格地図記述言語の記載方法の変更や描画方法の変更における拡張性の高さを実現できた。現在はまだ狭い範囲の対応付けのみであるが、今後はもっと多くのサンプルデータを実地図と結びつけることが課題である。

謝辞

本研究を進める上で、研究室の同輩、後輩の厚い手解きに心から感謝致します。

参考文献

- [1] 大森 光恵, 地下鉄の略地図表現, 中央大学理工学部情報工学科卒業論文, 2003 年
- [2] 小澤 政義, 略地図表現のための記述言語, 中央大学理工学部情報工学科卒業論文, 2001 年
- [3] 久保田 光一, G-XML について — 白地図・略地図表現の試み, 第 4 回統合型地理情報システムシンポジウム予稿集, 私立大学ハイテクリサーチセンター, 中央大学理工学研究所, 1999, pp.67-75
- [4] Geocoding, Geocoding, <http://www.geocoding.jp/>, 最終アクセス 2007 年 1 月 12 日
- [5] 田中 統一郎, 久保田 光一, 骨格記述言語設計とデータベース試作, 情報処理学会第 65 回全国大会予稿集, 2U-1, Vol.3, pp.103-104
- [6] 藤野 真一, 久保田 光一, 骨格地図記述の正規化, 情報処理学会第 67 回全国大会予稿集, Vol.3, pp.745-746