

地理情報標準に基づく屋内地図モデル定義とデータ生成 Indoor-map Model Definition and Instance Generation Methods based on Geographic Information Standards

情報工学専攻 高野 佑介

Information and System Engineering TAKANO Yusuke

概要: これまでに行われている構内 3 次元経路案内システムの開発では、システム独自のデータフォーマットを用いて地物情報を表現していた。本研究では、データモデルを地理情報標準に則った形式で定義し直し、データ自体の再利用性を高めた。また、新たなデータモデルを既存システムでも利用できるように、既存のデータフォーマットと相互に変換できるデータトランスレータを開発した。

キーワード: 地理情報標準, データモデル, XML Schema

1 背景

近年、経路案内や位置情報関連のサービスが発展し、より身近なものになりつつある。以前より、国や企業によって地理情報関連サービスの整備が勧められており、自動車に対してはカーナビゲーションシステムが、歩行者に対しては Google マップをはじめとするさまざまなサービスが普及している。

しかし、これらの既存サービスでは、大学や施設などのプライベートな場所においては、案内が未整備である場所も多く存在する。これらの場所の案内を行うためには、業者等に案内の作成を依頼するか、独自に経路情報や案内方式を作成する必要があるが、手間やコストがかかってしまう。

また、平成 31 年 3 月に国土交通省より提示された屋内地図・屋内測位環境構築ガイドラインの案 [3][4] によると、屋内電子地図は各施設管理者や民間地図事業者が、必要に応じて必要なエリアを個別に整備している状況であり、統一的な仕様での基盤地図の整備が必要不可欠であるとされている。

2 過去研究

2.1 写真による案内表現

写真には膨大な情報量があり、その場所の情報を直感的につかむ媒体として有用である。このことに注目し、写真を主体とした歩行者ナビゲーションの提案が大平落 [1] によって行われた。この研究では、写真間にリンクをつけて仮想空間を作成するソフトウェアである

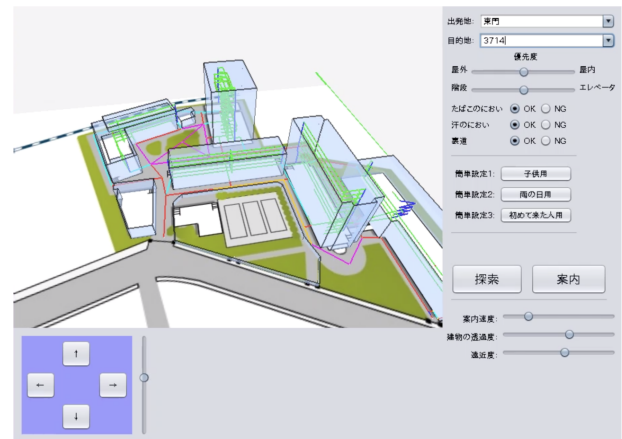


図 1 構内 3 次元経路案内システムの GUI

PhotoWalker[5] を利用し、写真のみで空間を構成し、経路案内を試みた。

この研究において、写真のみによるナビゲーションは撮影方法の制限から必要な写真の数が膨大になることが報告された。また、元々 PhotoWalker には経路探索の機能はなく、経路案内に特化されているソフトウェアであるわけではないため、案内を行う上での良し悪しは、データの作成者の技量次第となっている。

2.2 写真と簡易 3DCG による案内表現

3 次元データのみで経路の空間を構成すると、手間が膨大になり、写真のみを用いる案内方法では空間構造の把握に問題が生じる。そのような問題点を考慮し、過去研究 [2] では、案内表現方法に焦点を絞り、写真と簡易 3 次元 CG を用いて構内の案内を行う“構内 3 次元経路案内システム”が考案された。

このシステムでは、簡易的な 3 次元 CG を使用して空間構造を作り、適所に写真を用いて経路の紹介を行うことで、空間構造の把握のしやすさと写真の情報量の保持を同時に補うことを目指している。

2.3 問題点

ここまでで紹介した過去研究では、大きく分けて、写真とリンクにより構内を表現するものと、写真と簡易 3D により構内を表現するものの 2 種類の描画システム

が存在した。これらのシステムにおけるデータ構造は、それぞれ別々に定義されており、同じ構内の案内データであっても互換性なく設計されていたため、データの再利用が困難であるという問題点があった。

3 目的

本研究では、あるシステムで生成したデータを他の地理情報システムで再利用可能とするよう、地理情報標準にのっとった形のデータモデルを作成する。また、既存システムで用いられているデータ構造と新たに作成したモデルとの相互変換を可能にするプログラム、および作成したモデルに基づき実在する建物からデータを作成するためのプログラムの開発を行う。

4 提案システム

4.1 概要

これまで、内部のデータ構造が描画システムに依存する部分であったため、それぞれのシステムごとに内部のデータ構造を定めていた。これらのデータ構造を統一することは、全てのシステムの内部仕様を変更しなければならないため無理がある。そこで、統一のデータ構造の構造スキーマを定め、それぞれのシステムに応じたデータ構造を何らかの形で記述し、それらを相互に変換することができれば、統一のデータ構造からシステム内部のデータ構造に基づくデータを生成できる。

そこで本研究では、統一のデータ構造を新たに定義する。ここでは、統一のデータ構造を記述するためのスキーマとして地理情報標準プロファイル (JPGIS) を用いる。

また、Java には JAXB (Java Architecture for XML Binding) と呼ばれる XML ソースと Java オブジェクトの相互変換を可能にする API 仕様が存在する。ここでは、データ構造を表すためのアプリケーションスキーマを Java のクラスとして記述する。

4.2 データフォーマットの相互変換

既存のシステムにおいて使用される内部のデータ構造と、統一のデータ構造との間でデータの各要素を対応づけすることで、データの変換が可能となる。

本研究では、“構内 3 次元経路案内システム” で使用されているデータ構造を用いて、統一のデータ構造との相互変換を可能にした。

5 地理情報標準

地理情報標準は、地理情報システム (GIS) の基盤となる空間データ (地理空間情報) を、異なるシステム間で相互利用する際の互換性の確保を主な目的として、データの設計、品質、記述方法、仕様の書き方等の規則を定めたものである。

本研究では、国内の地理情報標準の実用規格である日本地理情報標準プロファイル (JPGIS) に基づくフォーマットでデータモデルを定義する。

5.1 地理情報標準プロファイル (JPGIS)

JPGIS (Japan Profile for Geographic Information Standards) は、地理情報標準に関する国際規格である ISO 19100 シリーズ、またそれに準拠する日本産業規格の JIS X 7100 シリーズから、必要最低限の要素を抽出したものである。

JPGIS は、国際規格や国内規格の内容を実利用に合わせて体系化し、より実用的な規格としたものであり、国土院では地理情報標準を用いる場面で JPGIS の利用の推奨を図っている。

5.2 JPGIS を利用するメリット

JPGIS の利用が進むことで、データの相互利用がしやすい環境が整備され、異なる整備主体で整備されたデータの共用や、システム依存性の低下、重複投資の排除等の効果を期待することができるとされている。

JPGIS の活用例として「データ・ソフトウェアの標準化」が挙げられている。例えば、相互運用性が確保されたデータは、様々な GIS ソフトウェアで使うことができる。GIS で利用する場合には、各々のシステムの内部形式に変換して使用し、データの交換時には標準の形式で交換する。これにより、データの相互運用性や再利用性が向上することが期待できる。

6 データモデルの定義

過去研究では、建物データ、地点 (node) データ、接続関係 (edge) データ、写真と node の関係性データの 4 つをそれぞれ個別のテキストファイルとして保存していた。しかし、これらのデータファイルは非常に扱いづらいため、XML を用いて新たにデータモデルを定義した。

新たなデータモデルは、JPGIS に則ったものとし、標準規格に従った地理情報の応用スキーマを用いることに

より汎用性を高めた。

データモデルは次の要素から構成される。

6.1 navidata (ルート要素)

navidata は、全ての要素の基底となる要素である。子要素として、建物を表現する building のリスト、地点を表現する node のリスト、2つの地点の接続関係を表現する edge のリストを持つ。

6.2 building (建物)

building は、1つの建物を表現するための要素である。子要素として、建物の名称、最下階および最上階の階数、建物の領域情報を持つ。また、属性として building の識別番号を持つ。

6.3 node (地点)

node は、1つの地点に関する情報を持つ要素である。子要素として、名称、座標、node の属性情報、その地点に紐付けられている写真の識別番号のリストを持つ。また、属性として node の識別番号を持つ。

6.4 edge (接続関係)

edge は、2つの地点の接続関係を表現するための要素である。子要素として、両端点の node の識別番号、経路案内のための経路探索条件を持つ。経路案内のための経路探索条件は、経路に特性を与えるための要素で、例えば、施設の関係者以外の通行ができないといった情報を保持することが可能である。また、属性として edge の識別番号を持つ。

6.5 picture (写真)

picture は、node で使用する1枚の画像データの情報を持つ要素である。子要素として、ファイルパス、配置パターン、写真が撮影された方向の情報、どの edge に向かって撮影されたかの情報を持つ。また、属性として picture の識別番号を持つ。

7 データモデルの実装

定義したデータモデルにより作成した XML スキーマの構造を Java 言語のクラスとして記述し、javax.xml.bind.annotation パッケージに含まれるアノテーションをソースコード中に記述することにより、JAXB を用いて Java オブジェクトと XML ソースを相互に変換できるようにした。

これにより、Java プログラム中で生成したオブジェクトを XML に変換することで、テキストファイルを直

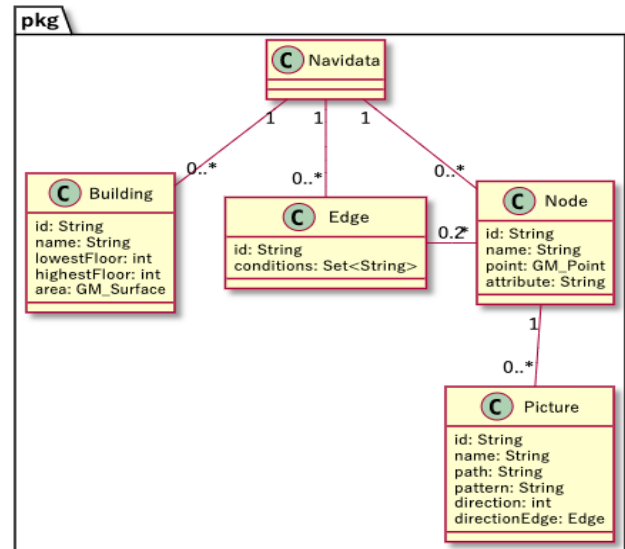


図2 定義したデータモデル

接編集することなくデータの生成を可能とした。

7.1 JAXB

JAXB (Java Architecture for XML Binding) は、Java オブジェクトと XML ソースを相互に変換するための Java の API 仕様である。

JAXB に含まれる “schemagen” ツールを利用することで、アノテーションが付加されたクラスのソースコードから XML Schema を生成することができる。

7.2 javax.xml.bind.annotation パッケージ

javax.xml.bind.annotation パッケージには、要素を表す XmlElement や、属性を表す XmlAttribute といったアノテーションが含まれる。

これらのアノテーションをクラス定義中に記述することにより、XML Schema の作成や、Java オブジェクトと XML ソースの相互変換を容易に行えるようにした。

8 実データの生成

定義したデータモデルから実際の建物についてデータを生成する方法について説明する (ソースコード 1, 2)。

8.1 Java によるデータの生成

データモデルの実装において実装したクラスを利用し、Java のプログラム内で新たにインスタンスを生成する。JAXB を用いて Java オブジェクトを XML に変換することにより、XML 化されたデータを得られる。

8.2 既存システムからのデータ変換

過去研究において開発されたシステムで使用しているデータモデルのクラス構造を、新たに定義したデータモデルのクラス構造と対応付けを行った。これにより、

既存システムでのデータフォーマットと新たに定義したデータフォーマットを相互に変換することが可能となった。

8.3 新たなデータの生成

新たにデータを生成する場合には、過去研究で開発されたオーサリングツールでの手順をもとに、出力されるデータフォーマットが新たに定義したものとなるよう改良したツールを使用する。

9 結論

本研究では、過去研究でのデータモデルを JPGIS に基づくフォーマットでデータモデルを再定義し、既存のデータフォーマットを本研究でのデータフォーマットに変換するためのトランスレータを作成した。

10 今後の課題

今回は敷地内の建物を対象とし、座標を敷地内でのローカルな座標としてデータを生成した。今後、敷地外とのシームレスな経路案内を実現するにあたり、ローカル座標だけではなく、必要に応じて緯度経度情報を付加してデータの生成を行えるようにする必要があると考える。また、JPGIS で提供されている空間スキーマを用いることで、JPGIS に則った形のデータモデルを定義したが、今後、実際に生成されたデータがどの程度正しいかを評価するための方法を検討する必要がある。

参考文献

- [1] 大平 落 優, “携帯電話上の写真表現による歩行者ナビゲーションシステム”, 中央大学工学部情報工学科卒業論文, 2006 年.
- [2] 矢澤 章, 久保田 光一, “経路特性を考慮した構内 3 次元経路案内”, 情報処理学会第 69 回全国大会予稿集, Vol.3, pp.221-222, 2007.
- [3] 国土交通省 国土地理院, “地理情報標準プロファイル (JPGIS)”, <https://www.gsi.go.jp/GIS/jpgis-jpgidx.html> 最終アクセス日 2020 年 2 月 13 日.
- [4] 国土交通省 国土政策局, “屋内地図／屋内測位環境構築ガイドライン (案)”, <https://www.mlit.go.jp/common/001317136.pdf>, 最終アクセス日 2019 年 12 月 16 日.
- [5] 田中 浩也, “PhotoWalker”, <http://web.sfc.keio.ac.jp/~htanaka/earthwalker/photowalker/>, 最終アクセス日 2020 年 2 月 10 日.

ソースコード 1 データモデルの Java クラス (抜粋)

```
@XmlElement(name = "navidata")
public class Navidata {
    private List<Node> mNodes;
    // (途中省略)
    @XmlElementWrapper(name = "nodes")
    @XmlElement(name = "node")
    public List<Node> getNodes() {
        return mNodes;
    }
    // (以下省略)
}

@XmlType(name = "node")
public class Node {
    private String mId;
    private GM_Point mPoint;
    // (途中省略)
    @XmlAttribute(name = "id")
    @XmlID
    public String getId() {
        return mId;
    }
    @XmlElement(name = "point")
    public GM_Point getPoint() {
        return mPoint;
    }
    // (以下省略)
}
```

ソースコード 2 Xml Schema (抜粋)

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<xs:schema elementFormDefault="qualified"
version="1.0" targetNamespace="http://num-
info.ise.chuo-u.ac.jp/indoormap" xmlns:tns
="http://num-info.ise.chuo-u.ac.jp/
indoormap" xmlns:xs="http://www.w3.org
/2001/XMLSchema" xmlns:jps="http://www.
gsi.go.jp/GIS/jpgis/standardSchemas2014">
<xs:element name="navidata" type="tns:navidata
"/>
<xs:complexType name="node">
<xs:sequence>
<xs:element name="point" type="jps:
GM_Point"/>
<!-- 途中省略 -->
</xs:sequence>
<xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
<!-- 以下省略 -->
```