

3.2 減災情報共有データベースツール

3.2.1 研究の概要と目標

本プロジェクトでは、減災に関わる各種情報システムを連携させるために、共通のプロトコル(MISP)を用い各情報システムを接続する枠組みとなっており、このMISPに従うシステムであれば情報共有を行なうことが可能である。しかしながら既存の情報システムがすでに存在する場合、このシステムを接続するためにはこのプロトコルに従って通信するよう変更する必要がある。このような変更を行なったもの、もしくは新規作成し、システムが直接共通プロトコルに従った通信を行なうもの(native client)でなくとも、簡易にMISPに従った通信を行なえるよう接続補助ツール群を整備した。

本年度は、この接続ツールを含め既存システムとの連携が容易になるよう、以下のような各種連携ツールを作成した。

- 基盤データ変換ツール：プロジェクトで定めた標準の都市基盤データを MISP に対応した XML 形式に変換するためのツール。
- CSV および XML 接続ツール群：DaRuMa と CSV、XML 形式でデータをやり取りするシステム群。これにより、広く利用されている CSV 形式や XML 形式の入出力を行なうシステムを DaRuMa と連携させることができる。
- スキーマエディタ：DaRuMa/MISP で用いる情報の XML 表現の形式定義(スキーマ)をグラフィカルに設計するためのツール
- DaRuMa 管理用ツール：ネットワーク上の DaRuMa あるいは MISP 準拠のデータベースを管理するためのグラフィカルなインターフェースを提供する。

以下では、おのおののツールについて解説を行う。

(1) 基盤データ変換ツール

本プロジェクトにおいて建築研究所が作成した「見附市基盤データ」は見附市を例題として、地方自治体建築物における建築物やその他の構造物のデータを記録し、自治体の地域情報の基盤となるデータである。この基盤データは地形情報の格納形式として古くから用いられている ESRI Shape 形式の地形データと、この地形データに関する補助的な情報を記述した CSV 形式のデータからなる。これらの形式のデータは歴史が古く、各方面で広く用いられているが、その歴史の古さゆえに現在のデータの運用のあり方を反映できず、運用面での柔軟性が不足している。そのため、DaRuMa ではこれらの形式ではなく、より柔軟性の高い XML のマークアップ形式を用いた MISP 用いることにした。また、Shape 形式や CSV 形式ではファイル形式上の制限から項目名等が省略して記述され、別途文書でその内容が記述されている。しかし MISP 形式では情報を直接文書に含めることができるようになったため、MISP 形式への変換時に省略された項目名ではなく本来の項目名を用い、可読性を向上させることとした。

a) Shape 形式ファイルの変換

Shape 形式のファイルを直接 MISP 形式に変換することは出来ない。これは互いのファイル構造が

全く異なっていること、および Shape 形式ファイルを扱うソフト間のファイルの内部表記の揺れから来る非互換性のために単純な変換が出来ないためである。本ツールでは変換工程を分割し、いくつかの段階を経て変換していくことにより、精度の高い変換と問題発生時の対処の容易さを共に実現した。

実際に作成されたプログラムの大まかな機能は図 3. 2-1 に示すとおりである。このプログラムは入力に Shape 形式ファイルを与えると、出力に MISP 形式ファイルとそれに即したスキーマファイルを生成する。

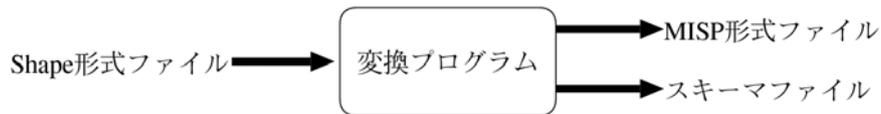


図 3. 2-1 Shape 形式から MISP 形式への変換プログラム

このプログラムの内部の詳細を図 3. 2-2 に示す。この中で Shape 形式ファイルの変換作業の中心となるプログラムは OGR である。これを用いて Shape 形式から、MISP 形式に最も近い GML 形式に変換し、GML 形式のファイルから MISP 形式のファイルを作成する。プログラムは大きく分けて「前処理」、「OGR による Shape 形式から GML 形式への変換」、「後処理」、「GML 形式から MISP 形式への変換」で構成される。

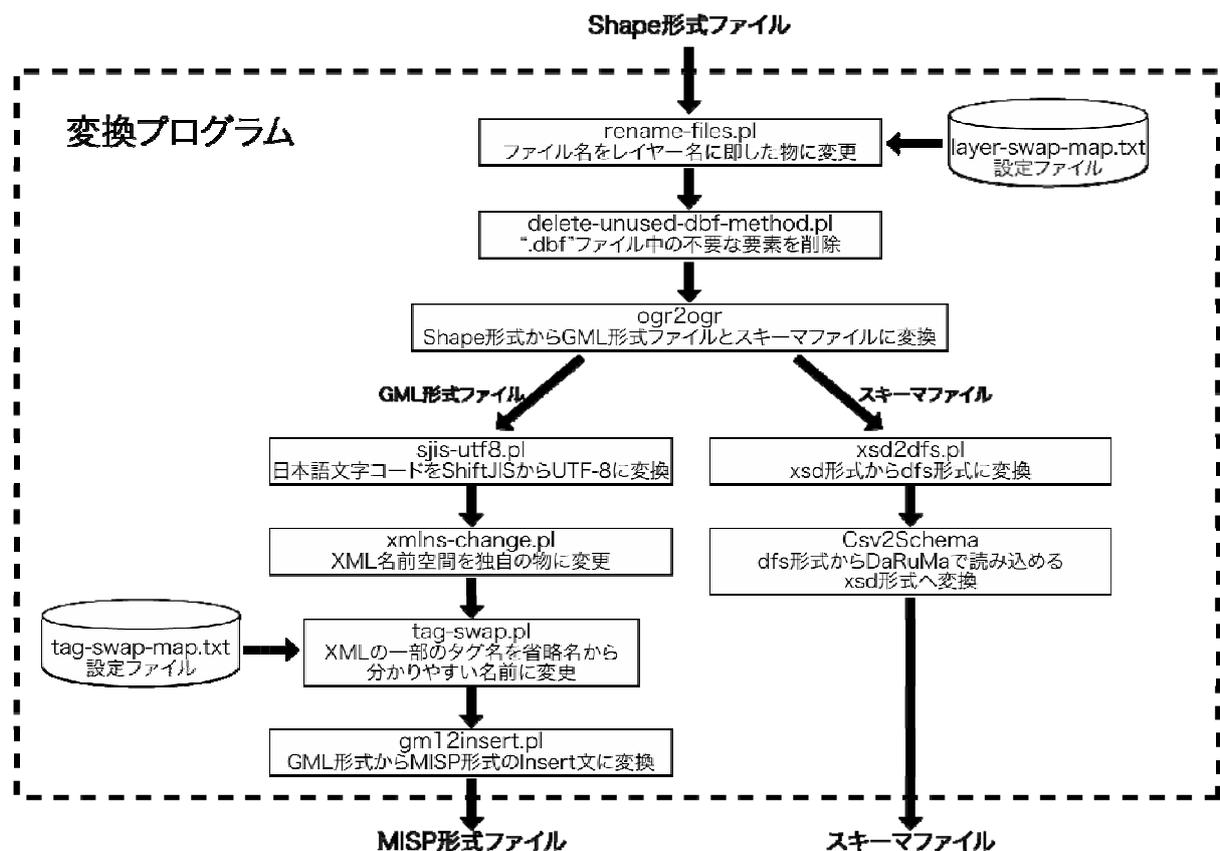


図 3. 2-2 Shape 形式から MISP 形式へ変換するプログラムの内部構造

以下にその詳細を示す。

- 前処理

変換表に従いファイル名の変更を行なう。これは OGR が Shape 形式では表現できないが GML 形式、そして MISP 形式では必要となるレイヤ名をファイル名から推測するためである。さらに Shape 形式ファイルに含まれる不要なデータをあらかじめ削除する。

- OGR による Shape 形式から GML 形式への変換

OGR を用いて GML 形式のファイルとそれに対応するスキーマファイルを作成する。

- 後処理

Shape 形式ファイルでは漢字コードとして Shift JIS を用いているが、MISP 形式では基本的には UTF-8 を用いるため、漢字コードを UTF-8 に変換する。そして GML 形式の文書に含まれる OGR 特有の記述(名前空間表記など)を DaRuMa 用の表記に改める。さらに Shape 形式特有の制限から来る GML 形式文書中の XML タグ名の表記を変換表に従い分かりやすいものに変換する。

- GML 形式から MISP 形式への変換

GML 形式の文書からデータの部分を集めて MISP 形式で出力する。

データの登録に必要なスキーマファイルについてはデータ変換プログラムの OGR が変換時に自動作成する正確だが複雑なスキーマファイルを DaRuMa はそのまま受け付けられないため、DaRuMa が理解できるように単純化することにした。そのため将来的なデータの可搬性を考え、スキーマファイルを直接出力するのではなく、一旦スキーマファイルをスキーマエディタが理解できるデータ形式である DFS 形式のファイルに変換し、更にここからスキーマエディタを用いてあらためてスキーマファイルを逆変換して作成するようにした。この逆変換作業はスキーマエディタの他、コマンドライン版の DaRuMa クライアントツール群に含まれる “Csv2Schema” を用いても同様のことが行なえる。本プログラムでは作業の自動化のために実際にはスキーマエディタではなく、この Csv2Schema を用いている。

b) GSV 形式ファイルの変換

CSV 形式のファイルは、DaRuMa クライアントツール群の中に CSV 形式、MISP 形式の変換を自動的に行い通信するツールが含まれており、本来ならばそのまま手を加えずとも良いはずである。しかし、実際には CSV 形式ファイルの公式な定義の曖昧さから来る各プログラムが受け付ける CSV ファイルの仕様の違いを吸収したり、データ自体に含まれる間違いを訂正するために若干の変換作業が必要となる。

この訂正作業を行なうプログラムは図 3.2-3 のようなものである。このプログラムは入力から CSV ファイルを与えると DaRuMa クライアントツール群が読み込める形式の CSV 文書を出力する。



図 3. 2-3 CSV 形式から DaRuMa クライアントが読み込める CSV 形式への変換プログラム

このプログラムの内部は図 3. 2-4 のようになっている。

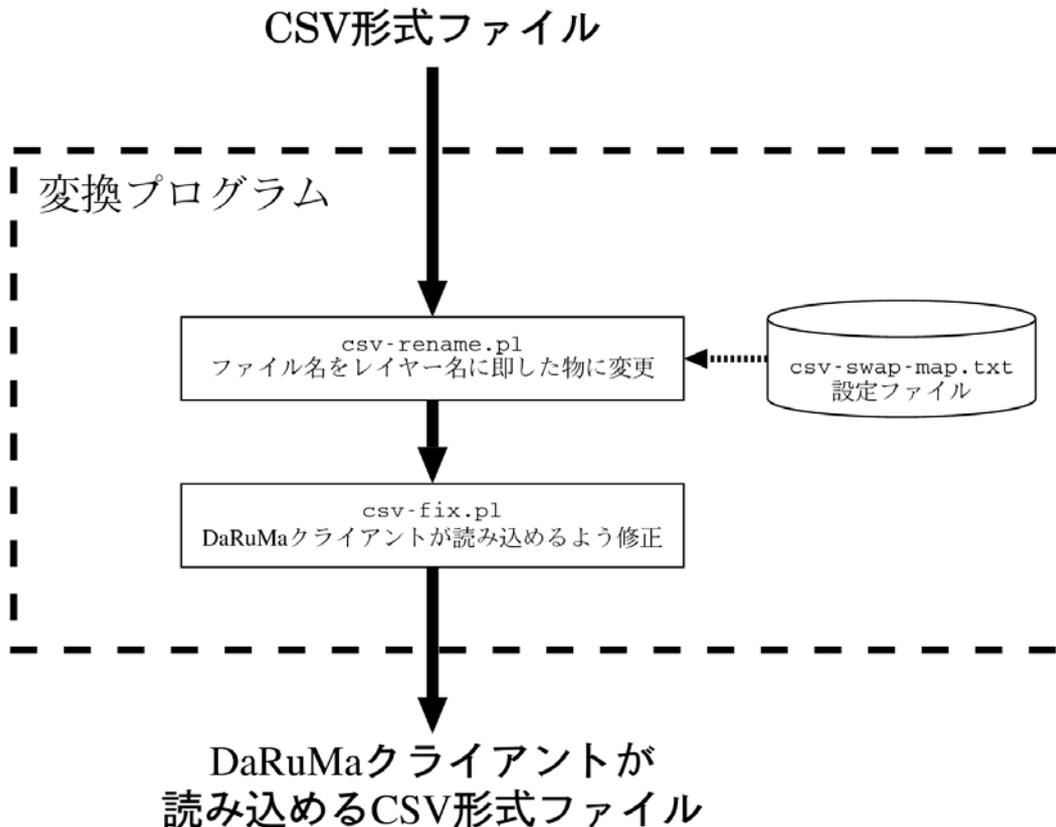


図 3. 2-4 CSV 形式から DaRuMa クライアントが読み込める CSV 形式へ変換するプログラムの内部構造

また、変換した CSV ファイルを DaRuMa クライアントツール群を用いて DaRuMa に登録する時に必要となるスキーマファイルや変換方法を記述した XML ファイルは、スキーマエディタを用いて手作業で作成する必要がある。これらのファイルの作成の際には Shape 形式由来のファイルのように全ての項目が有効なデータで埋まっているとは限らないので、必要に応じて一つのデータに含まれる各項目の最小個数を 0 にする等の必要がある。スキーマエディタは標準で項目の最小個数を 1 個にするのでこの部分を直さないと思わぬデータ登録ミスの原因になることがある。

(2) XML および CSV 接続ツール群

XMLおよびCSVクライアント群は、DaRuMa が使用している減災情報共有プロトコルであるMISP

を使用するクライアントツール群である。

本ツールで提供するコマンド群は、GetFeature, Insert, Update, Delete 等の基本リクエストに加え、登録されているFeatureType のリストや指定したFeatureType の登録データ数を取得するリクエストを処理することができる。

基本機能アクセスツールが提供するコマンドは、以下のとおりである。

a) DarumaClient

MISP に準拠したXML リクエストを送信し、レスポンスを受信するクライアントとして動作する。XML リクエストは、ファイルまたは標準入力を使用して指定できる。DaurmaClientコマンドは、汎用コマンドであるため、GetFeature やInsert, Update, Delete 等のMISP のリクエストファイルを生成することによって、様々な用途に使用できる。

b) RegisterFeatureType

RegisterFeatureType コマンドは、指定したFeatureType のスキーマを登録するクライアントとして動作する。RegisterFeatureType は、MISP のRegisterFeatureType プロトコルを使用しており、MISP のRegisterFeatureType と同様の結果となる。

c) DescribeFeatureType

DescribeFeatureType コマンドは、指定したFeatureType のスキーマを出力するクライアントとして動作する。DescribeFeatureType は、MISP のDescribeFeatureType プロトコルを使用しており、-xpath オプションをしていない場合は、MISP のDescribeFeatureType と同様の結果となる。

d) CountFeatureType

CountFeatureType は、指定したFeatureType の登録件数を返すクライアントとして動作する。出力結果として、登録件数の数字のみを表示する。カウントされるFeatureType は、サーバに登録されている同一のFeatureType の全データである。

e) ShowFeatureTypes

現在、サーバに登録されている全FeatureType を取得するクライアントとして動作する。ShowFeatureTypes は、MISP のGetCapabilities プロトコルを使用しており、このリクエストのレスポンスからFeatureType の名前を取得している。

f) XmlGetFeature

XmlGetFeature は、MISP のGetFeature プロトコルを用いて、データを取得するクライアントとして動作する。

g) XmlInsert

XmlInsert は、MISP のInsert プロトコルを用いて、XML ファイルのデータを挿入するクライアントとして動作する。

h) CsvGetFeature

CsvGetFeature は、MISP のGetFeature プロトコルに従って取得したデータを、CSV に変換するクライアントとして動作する。CSV への変換ルールは-csvcfg オプションで指定された

ファイルに従って変換を行う。

i) CsvInsert

CsvInsert は、MISP のInsert プロトコルを用いて、CSV ファイルのデータを挿入するクライアントとして動作する。CSV の変換ルールは-csvcfg オプションで指定されたファイルに従って変換を行う。

(3) スキーマエディタ

スキーマ編集ツールは、DaRuMaのサービスが使用しているスキーマを編集するためのツールである。本ツールで提供するプログラムはスキーマの編集、スキーマの妥当性検証、XSD ファイルのエクスポートができる。また、(3)での述べるCSVクライアント用の設定ファイルも同時に編集・作成できる。これにより、XML形式とCSV形式を統一的に管理することができ、現状で汎用性の高いCSV形式ファイルを利用するシステムを容易に減災情報プラットフォームに接続することができるようになる。

図3.2-5はスキーマエディタのスクリーンショットである。ここでは、豊橋での実証実験で用いた道路リンクの形式を編集している画面を表示している。

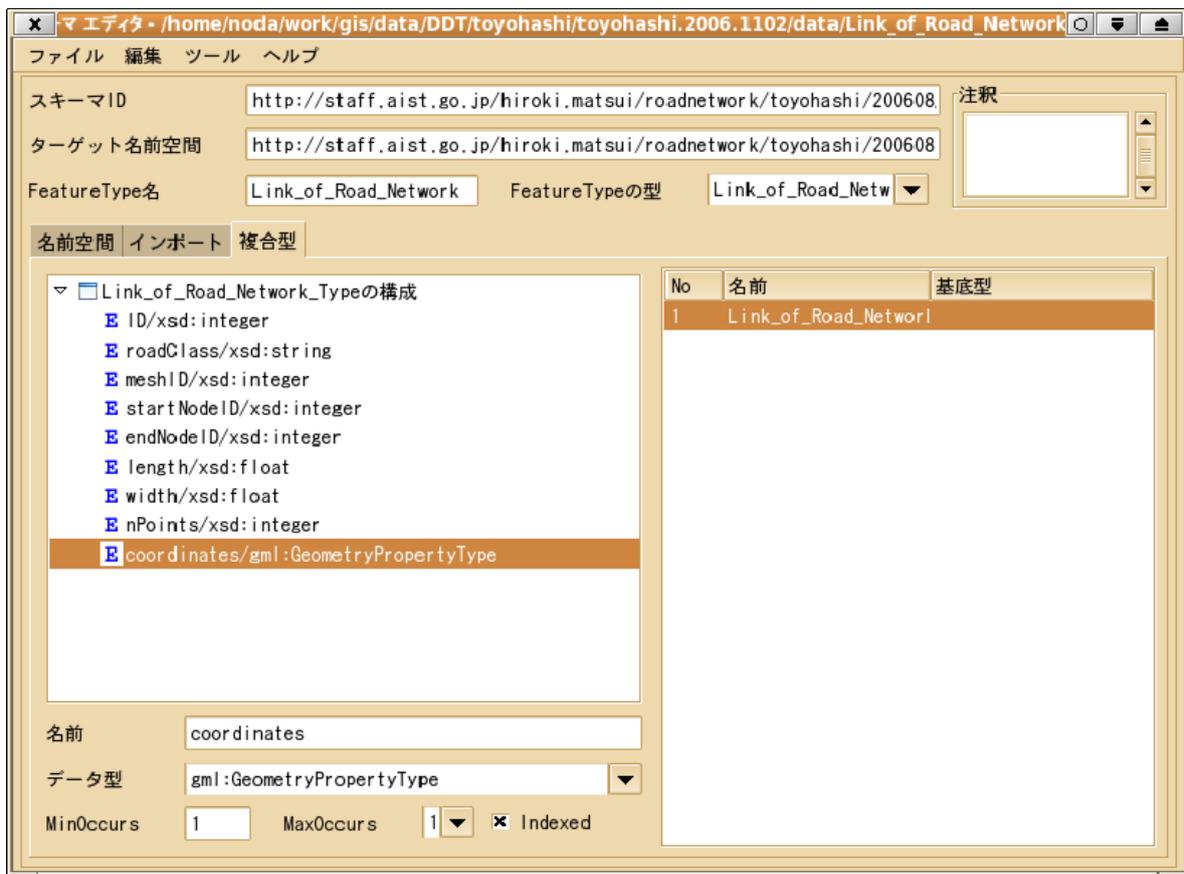


図 3.2-5 SchemaEditor

(4) DaRuMa 管理用ツール

DaRuMa管理用ツールは、DaRuMa を管理、サーバからFeatureType を取得、DaRuMa 用のクライアントコマンドの起動を行なうことが可能な高機能なDaRuMa 用のグラフィカルなクライアントツールである。主な機能には以下のものがある。

- a) サーバ管理
データベースの追加，削除機能を提供する。
- b) FeatureType 管理
FeatureType リストおよびFeatureType 情報の取得. ターゲット名前空間，データカウント，スキーマの表示等を行う。
- c) コマンド実行
クライアントコマンドツールにある下記のコマンドを実行できる。コマンドの内容については、(3)節を参照のこと。
 - DarumaClient
 - RegisterFeatureType
 - XmlGetFeature
 - XmlInsert
 - CsvGetFeature
 - CsvInsert

3.2.3 まとめ

減災情報共有データベースを用いた情報共有を容易とするツール群の作成を行った。これらツール群は、基盤データ整備や既存の情報システム，新規に構築する情報システムを，減災情報共有プラットフォームに参画させるための開発を支援するものである。これらツール群を整備したことにより，減災情報共有プラットフォームを利活用する情報システムが，容易に開発できる環境が整えた。