

2.1 時空間データベース構造の開発に関する研究

2.1.1 目的

災害対応の場に関与する機関間において、時々刻々と変化する災害情報の管理および共有に利用されるデータベース構造には、以下のような特徴が求められる。

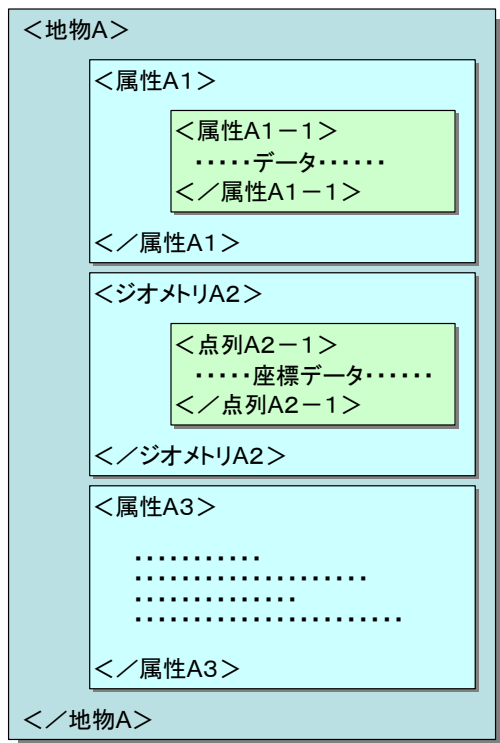
- 1) 汎用的であること
- 2) 公開形式であること
- 3) 情報の時間管理ができること
- 4) 差分等による効率的なデータ交換が可能であること
- 5) 検索, 追加, 更新が効率的に行えること
- 6) 特定のコード体系によらない普遍的なキーにより情報を管理すること

機関間における情報共有は、大きく、同一機関内、協定機関間、一般との間に分類することができる。情報交換・共有の量・頻度は、同一機関内、協定機関間、一般との間の順に大きい。自治体の本所・本部、支所・出先のような同一機関内では、同一地域を対象とした同一の情報管理体系に基づいて情報共有が行われるため、一般的な情報記述を導入するより差分による情報共有の効率・実効性が求められる。市町村と県等のような協定機関間では、事前に情報共有する内容を取り決めておくことができるが、対象地域が包含関係にあたり別の領域であったり異なるため、最小限の、代表点、属性、領域のような情報構造による共有と位置整合が求められる。その他一般との間においては、一般に対する個人情報の保護等を考慮したある範囲の情報公開が必要となる。また、各機関は、その目的・分掌に基づき主体的に情報を収集し関係先と交換し活用するため、各機関特定の ID (識別子) をキーとした情報管理を行うのが一般的である。これらを住民票コードの全国統一のような標準化は事実上不可能である。

時空間データベース構造 (KIWI+) ^{1), 2)} は、普遍的な位置座標と時間をキーとして情報を管理することができ、上記のような情報共有に求められるデータベース構造の要件を満たすものである。また、複数の自治体において全庁利用され継続的に運用されているなどの実績を有している。しかし、情報共有は、異なる情報システム間で共有するプロトコル、具体的には本研究で開発した「減災情報共有プロトコル (2.4 節)」を用いて機械による会話を成立させることが前提となる。減災情報共有プロトコルは、一般によく認知されている XML (Extensible Markup Language) 系の記述と添付ファイル (MIME : Multi-Purpose Internet Mail Extension) を用いて汎用性と実効性を狙ったもので、データベース操作やデータ記述に WFS (Web Feature Service), GML (Geographical Markup Language) を用いている。GML は、実世界の地物 (フィーチャー) を記述している。地物は、データベース構造の応用スキーマや論理スキーマの記述、事象の記述に対応する。一方、KIWI+ は、地理データの物理スキーマを中心として記述したものである。両者は、スキーマの階層が異なるため、そのままでは対応しないという問題がある (図 2.1-1)。

本節の目的は、減災情報共有によりよく適合するよう時空間データベース構造を高度化することであり、具体的な目標は、GMLとKIWI+との一意的で双方向対応を保証する変換アルゴリズムを開発することである。

(a) GML:フィーチャー(地物)の記述
(応用スキーマ, 論理スキーマ)



(b) KIWI+:地理情報の記述
(論理スキーマ, 物理スキーマ)

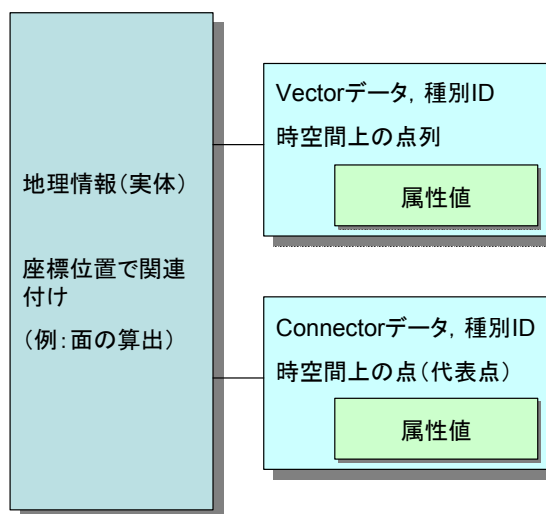


図 2.1-1 GMLとKIWI+のデータモデル

2.1.2 システム概要

(1)時空間データベース構造の位置づけ

被災自治体をはじめとして各機関は、主体的に必要な情報を収集・交換し行動して、自律分散的な災害対応の場を形成する。プロトコルを共有すれば、ITにより情報共有が迅速化し、意思決定の迅速化や被害の拡大抑止など減災効果の向上が期待できる。データベースも自律分散的なものである。時空間データベース構造 KIWI+は、情報の時間記述が可能なデータベース・フォーマットである。それは、公開形式であるため競争原理を導入して高額な地図データ作成や保守費用の低減が期待できる。また、KIWI+は、減災情報共有プラットフォーム（共通プラットフォーム）の時空間データベースのフォーマットとして用いられる（図 2.1-2）。

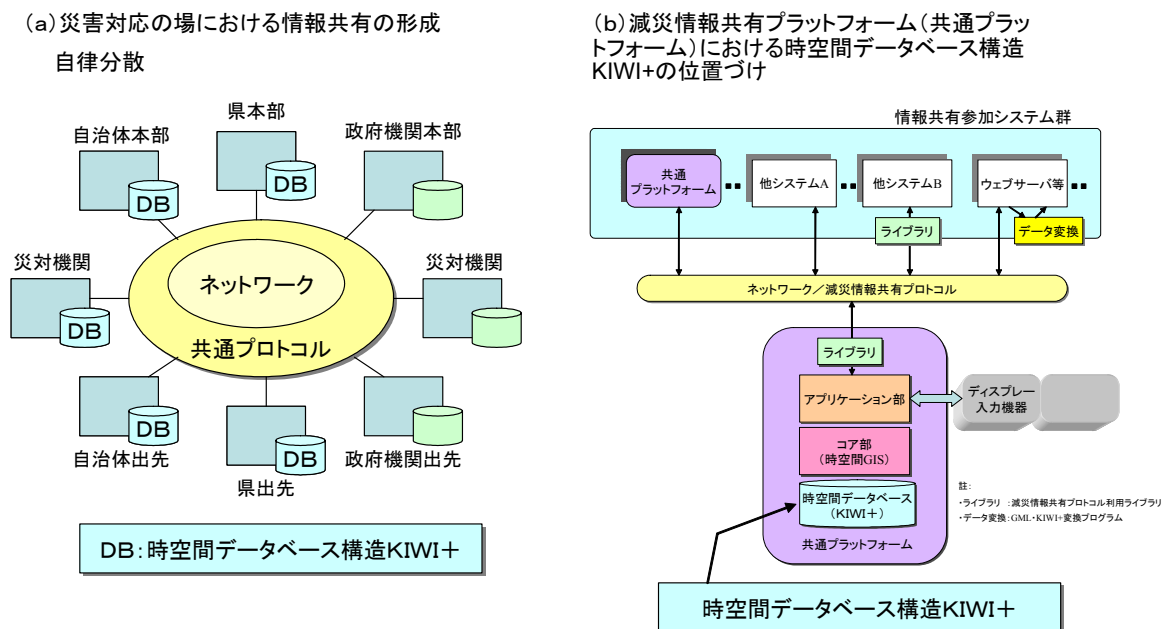


図 2.1-2 時空間データベース構造 KIWI+の位置づけ

(2) GML・KIWI+変換アルゴリズムの概要

GMLとKIWI+の変換は、関係モデル・データベース形式を中間形式とし、GML⇔関係、関係⇔KIWI+の2段階方式をとる。両者のスキーマ対応には、データベース管理者、または、プログラムで作成した事象テーブルを用いる（図2.1-3）。

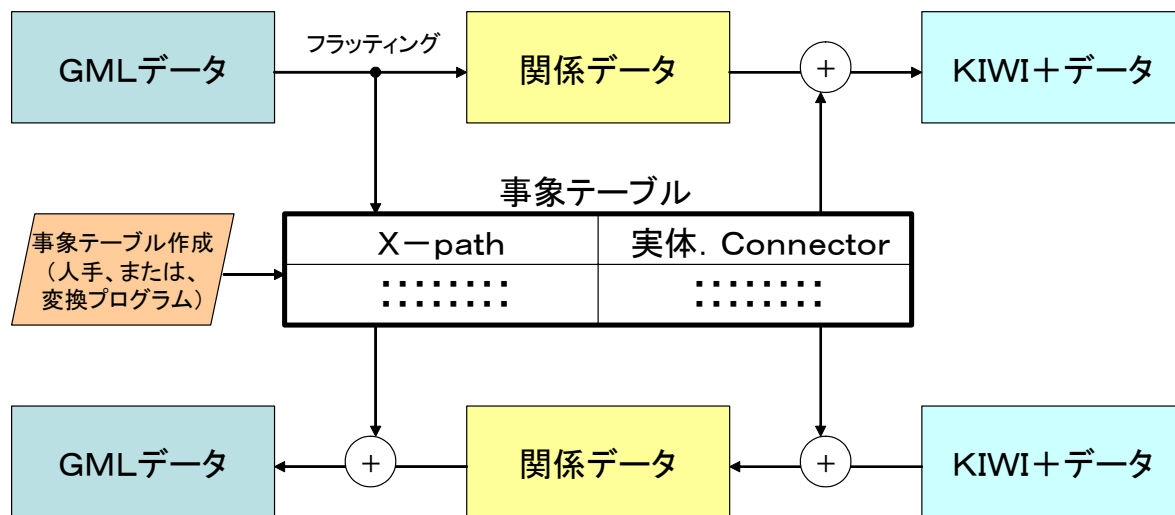


図 2.1-3 GML・KIWI+変換アルゴリズムの概要

2.1.3 研究の内容

(1) 課題

GML・KIWI+変換アルゴリズムは、GML⇔関係モデル、関係モデル⇔KIWI+の2段階に分けて考えることができる。中間に関係モデル（関係、または、テーブル）を置くことで、データベース管理システム(DBMS: DataBase Management System)の実装において、関係モデル DBMS (Relational DBMS, RDBMS) との両立性を得ることができ、利用者の選択性・利便性を高めることができる。また、共同先である産業技術総合研究所・野田らのチームは、GML と関係モデルとの双方向変換を RDBMS・MySQL を用いて試作している。GML⇔関係モデルの変換は、GML の木構造を一筆書きでたどり（トリーウォーク）線形にして（フラットニング）、関係のタプル（レコード、レコードタイプについても同様。）を得るものである。関係のテーブル名および項目名に X-path（タグ名称をフルパスで表現したもの）を用いることにより、GML と関係との一意的な対応を保証する。GML の繰り返し構造については、その部分を別の関係とし、繰り返し分のデータをタプルで表現する。

上記により、本稿の課題を関係モデル⇔KIWI+の変換アルゴリズム開発に絞り込むことができる。

(2) 関係（テーブル）の構造

関係 R の名称 RN, R に所属する項目 F の名称 FN および F の型 FT, 項目の並び順を<F*>としたとき、関係の構造は、図 2.1-4 のように表すことができる。

$$R = \langle RN, F^* \rangle, F = \langle FN, FT \rangle$$

または,

$$R = \langle RN, (\langle FN, FT \rangle)^* \rangle$$

<a, b> : 順序付けられた集合

* : 繰り返し

図 2.1-4 関係の構造

MySQLのようにジオメトリ(図形または形状)を表現できるRDBMSでは、FTとしてPOINT, LINE, POLYGON, 等のジオメトリクラスが指定できるようになっている。

(3) KIWI+の構造

KIWI+の構造を模式的に表すと、そのデータの基本要素は、Connector（点の物理レコード、名称 CN, 型名 ConnectorID, 属性 CA（時空間情報を含む）.）、Vector（線の物理レコード、名称 VN, 型名 VectorID, 属性 VA（時空間情報を含む）.）であり、時空間上の位置を占める。Connector,

Vector のどちらにも属性値を記述できるが、関係モデルと異なりジオメトリの存在を前提としている。Vector は、一部か全部の地物輪郭形状、または、一部か全部の地物間境界線を表す。データは、空間を矩形で切り取ったパーセルに対応したファイルに格納されるため、パーセルの境界を横切る Vector は、パーセル境界との交点で切断されている。すなわち、物理レコードを得るための2次元的なブロッキング処理である。論理的な線の形状は、同一 VectorID で検索することにより得ることができる。なお、論理的な形状を厳密に議論したい場合は、パーセル境界上の点がデータ管理上のブロッキング処理により便宜的に生成されたものなのか、交差点のようなノードや道路形状を補間する点(形状補間点)のような正規の形状データなのかを区別する必要があり、KIWI+の規定に加えることにしている。

KIWI+はデータベース構造の物理スキーマを中心に記述しているが、KIWI+の実体定義を用いて論理的なジオメトリ(点, 線, 面, 体)を表現することもできる。実体定義の模式的な構造は、実体 e の名称 eN, 型 eT (点'P', 線'V', 面'F', 体'B'), その形状の構成候補となる Vector 種別 (VectorID) のリスト, 実体に所属する Connector 種別 (ConnectorID, 実体の代表点.) の集合からなる。Vector は、複数実体の形状要素となり得るが、Connector は、たかだか1つの実体に所属する。したがって、Vector の属性値は、Connector の属性値と異なり、実体の属性値を一意的に構成することは保証されていない。実体定義により、KIWI+で記述されたデータベースの直接的な管理プログラムである時空間 GIS (GIS : Geographical Information System) エンジン は、自動的に実体形状を生成することが可能となる (図 2.1-5)。

$$\begin{aligned} \text{Vector} &= \langle \text{VN}, \text{VectorID}, \text{VA} \rangle \\ \text{Connector} &= \langle \text{CN}, \text{ConnectorID}, \text{CA}, \text{eN} \rangle \\ \text{e} &= \langle \text{eN}, \text{eT}, \text{VectorID}^* \rangle \end{aligned}$$

図 2.1-5 Vector 定義, Connector 定義, 実体定義

しかし、実体の属性については、時空間 GIS エンジンで一意的に求めることができない要因が存在し、従来の時空間 GIS エンジンでは、直接、Connector の属性値、または、Vector の属性値をアクセスする仕様となっている。以下は、実体定義が示す実体形状と実体属性の範囲である。実体属性を一意的に決めるためには、まず実体属性の範囲が一意的に決まらなければならない。このためには、適切な規制と用法が必要である。

1) (eT = 'P') のとき

実体に所属する Connector が、実体のインスタンス（形状と属性）となる。

2) (eT = 'V') のとき

実体定義の Vector が実体形状となる。実体属性は、実体に所属し実体形状上の Connector の属性値と、実体形状を構成する Vector の属性値によって表される。しかし、Vector の属性値が一意的に実体属性を構成するかどうかは、利用者がデータベース設計時に保証する必要がある。

3) (eT = 'F') のとき

実体定義の Vector ID 群、および、指示された代表点から生成される面が実体形状となる。実体属性は、実体に所属し実体形状・面内の Connector の属性値と、実体形状を構成する Vector の属性値によって表される。Vector の属性値が一意的に実体属性を構成するかどうかは、利用者がデータベース設計時に保証する必要がある。

4) (eT = 'B') のとき

実体定義の Vector ID 群と代表点から生成される面を代表点の比高位置まで引き伸ばしてできる体が実体の形状となる。実体属性は、実体に所属し実体形状・体内の Connector の属性値と、実体形状を構成する Vector の属性値によって表される。Vector の属性値が一意的に実体属性を構成するかどうかは、利用者がデータベース設計時に保証する必要がある。

(4) 実体属性を一意的に定めるための条件

関係と KIWI+ との変換アルゴリズムは、双方向で一意的な対応を保証する必要がある。しかし、次のような問題がある。実体の型が点の場合を除き、実体属性の範囲を決定する手順は、煩雑で一意性が保証されていない。このため、実際のデータベース設計では、属性を Connector にしか与えない、Vector を 1 実体にだけ所属させる等の制約を加えている。関係モデルで頻繁に用いられる View 指定（必要な属性項目の選択と並び順の指定）も、KIWI+ は記述しない。また、ジオメトリを記述できる RDBMS では、複数の項目型にジオメトリを指定できるし、全く指定しないこともできるが、実体型では、1 つのジオメトリを指定する必要がある。

上記の問題は、関係と KIWI+ の用法に、以下のような制限と工夫を加えることで解決することができる。

- 1) 関係で使用するジオメトリを実用上充分と考えられる点、線、面に制限する。
- 2) 実体 e の順序付き集合で関係に対応する事象 E を導入する。E=<e*>.
- 3) ジオメトリを持たない事象 E は、対象領域外の空間（虚空間）に存在するものとして取り扱う。
- 4) 事象 E の定義（事象定義、または、事象テーブル）には、View も記述する。
- 5) 事象テーブルは、関係項目と同一順で以下のような構造と持つものとする（図 2.1-6）。

事象テーブル = <RN(X-path), (<FN(X-path), FT, eN[.ConnectorID[#]]>)*>

[a] : 選択可能な記述

: 属性項目 ID

図 2.1-6 事象テーブルの構造

- 6) 実体属性は、CA だけを用い、VA は無視する。
- 7) 実体型と所属 Connector の関係を以下のように制限する。
 - ・実体型=点：その Connector.
 - ・実体型=線：両端点上の Connector, ノード上の Connector.
当面は、前者を採用する。
 - ・実体型=面：面内の Connector. 面の輪郭線上の Connector は含めない。
- 8) 繰り返し構造を持つ関係項目に対応するため、Connector は、複数の属性項目を記述できるととする。データが溢れる場合は、同一座標・同一種別の Connector に溢れたデータを格納する。当面は、データ形式として CSV 形式を採用する。

(5) 変換アルゴリズム

a) 事象テーブルの作成

変換処理は、事象テーブルに従って実行されるため、前処理として事象テーブルを作成する必要がある。この場合、以下の3つの利用ケースが考えられる。

1) 事前にデータベース管理者が作成できる場合

同一機関内、協定機関間でデータ変換を行う場合は、データベース管理者が、対象の GML のスキーマおよび KIWI+のスキーマを知っているか知り得ることができたため、事前に事象テーブルを作成し、変換アルゴリズムに与えることができる。

2) 事前にデータベース管理者が作成できない場合

前述 1) 以外の場合、データベース管理者は、GML のスキーマか KIWI+のスキーマの一方を知っているか、両方とも知らないと考えられるため、事前に事象テーブルを作成することができない。この場合、変換アルゴリズムを実装したプログラムは、入力データを分析して、上記の一方を作成し、未知のもう一方を仮に作成する機能（事象テーブル作成支援機能）を持つ必要がある。自機関または相手機関のデータベース管理者は、その結果を確認し、事象テーブルを修正・完成して、再変換を実行する。

仮作成においては、形状（ジオメトリ）とそれに所属する属性（ジオメトリ以外の項目）を判別する必要がある。また、仮の名称をユニークに生成する必要がある。以下に、変換の方向に応じた形状とそれに所属する属性の判別方法を示す。

i) 関係→KIWI+変換の場合

関係のジオメトリ項目は、KIWI+の実体に対応し、次に出現するジオメトリ項目の手前か終了までの属性を、当該実体の属性とする。先行したジオメトリ以外の項目があった場合は、これらも最初に出現したジオメトリ項目に対応する実体の属性とする。

ii) 関係←KIWI+変換の場合

KIWI+の実体は、関係のジオメトリ項目に対応し、実体に所属する属性を項目として関係を生成する。

3) 事象テーブルに未定義の事象を追加する場合

未定義の事象について、前述 2) と同様の処理を行う必要がある。

以下、完成した事象テーブルを与えた場合の変換アルゴリズムを示す。記述は、構造化日本語による。

b) 関係→KIWI+変換アルゴリズム

1. 各関係について以下を繰り返す。

1.1 各項目について以下を繰り返す。

1.1.1 項目の X-path をキーに事象テーブルと照合する。

1.1.2 項目に eN が対応する場合、実体 e の形状を出力する。

注：面の場合、実体定義の最初の VectorID で Vector を出力する。属性がない場合は、最初に見つけた所属 ConnectorID で Connector を面内位置に出力する。

1.1.3 項目に eN.ConnectorID が対応する場合、その ID の Connector を属性に加えて生成する。項目が繰り返し構造の場合は、属性項目を複数持つ属性とする。Connector のデータレコードサイズがオーバーした場合は、同一 ID（種別）の Connector を同一位置に生成して溢れたデータを格納する。

注 1：ジオメトリの型に応じて、生成する位置座標が異なる。

- ・点：その位置
- ・線：始点の位置
- ・面：面内の位置

注 2：虚空間に関する処理は省略しているので注意。

c) 関係←KIWI+変換アルゴリズム

1. 事象テーブルの各事象について以下を繰り返す。
 - 1.1 事象名 (X-path) を関係名として検索する。
 - 1.2 事象の各項目について以下を繰り返す。
 - 1.2.1 項目名 (X-path) を関係項目名として検索する。
 - 1.2.2 項目が eN の場合、実体定義から形状を算出し座標(列)を関係のジオメトリ項目値として出力する。
 - 1.2.3 項目が eN.ConnectorID の場合、実体 e に所属する Connector の属性値を項目値として出力する。Connector 属性値の項目数が 2 以上の場合は、繰り返し構造を持つ属性値として出力する。この場合、溢れた属性値を格納した Connector (同一 ID, 同一座標) の属性値も出力する。

2.1.4 本年度の成果

異機種間、自律分散環境下での災害情報共有の観点から、時空間データベース構造を見直し、本来の代表点および差分による情報共有利用に加え、ジオメトリ、属性、添付ファイルからなる事象スキーマの記述方法を拡張した。これにより、GML、関係モデル、時空間情報モデルの対応関係を明確に整理し、GML・時空間データベース構造 (KIWI+) の変換アルゴリズムを開発した。また、時空間データベース構造仕様書に改良を加えた²⁾。

2.1.5 平成 17 年度の計画

時空間データベース構造の開発は、平成 16 年度でほぼ完了したため、更なる改良および変換アルゴリズムの実装を、平成 17 年度より減災情報共有プラットフォームの開発に統合して進めることとする。また、改良した時空間データベース構造仕様書を広く認知してもらうための活動を行う。

参考文献

- 1) KIWI+事務局：時空間地理情報システム (ST-GIS) データベース構造仕様－KIWI+フォーマット－第 0.74 版，2001.1
- 2) リスク対応型時空間情報システム研究委員会：リスク対応型時空間情報システム研究委員会報告書，付録 (6) 時空間データベース構造KIWI+フォーマット規定 (第 0.9a版)，防災科学技術研究所地震防災フロンティア研究センター川崎ラボラトリー，2004.11 (<http://www.kedm.bosai.go.jp/>)