

シミュレーションデータウェアハウスにおける災害情報の統合分析

趙 菁[†] 石川 佳治[†] 杉浦 健人[†] 王 元元^{††} 佐々木勇和[†]
 瀧本 祥章[†]

[†] 名古屋大学大学院情報科学研究科 〒464-8603 愛知県名古屋市千種区不老町

^{††} 山口大学大学院理工学研究科 〒755-8611 山口県宇部市常盤台 2-16-1

E-mail: †{zhao,sugiura,yuya,takimoto}@db.ss.is.nagoya-u.ac.jp, ††ishikawa@is.nagoya-u.ac.jp,

†††y.wang@yamaguchi-u.ac.jp

あらまし 地震、津波など災害の対策に関する研究では、さまざまなシミュレーションが行われ、大量のデータが生み出されている。計算機能力の向上に伴い、シミュレーションデータの大規模化がさらに進んでおり、データの利活用において大きな課題となっている。このような背景を踏まえ、本研究では大規模シミュレーションデータの対話的分析を実現するため、シミュレーションデータウェアハウス (simulation data warehouse) の概念を提案し、その実現方式について述べる。また、探索的な分析を行うため、異なるシミュレーションデータを統合し、事前に集約処理を行い、高速な応答を実現する。さらに、開発したプロトタイプシステムのアーキテクチャや利用例などを述べる。
 キーワード 災害データ, データウェアハウス, 対話的分析

1. はじめに

科学の領域では、情報技術の進展やコンピュータやストレージの高性能化・大容量化を受けてデータが急速に増大している。大規模データを扱うことを前提とした科学の取り組みは、しばしばデータサイエンス (data science) と呼ばれ、新たな潮流として大いに着目を浴びている。本研究では科学のさまざまな分野で行われるシミュレーションに着目する。シミュレーションにより生み出されるデータは膨大であり、そのような大規模なデータの活用に焦点を当てたデータ集約的なコンピューティング (data-intensive computing) は、科学の「第4のパラダイム」として大いに着目されている [11]。

地震・津波などの災害に関しても多くのシミュレーションが行われる。さまざまな条件やパラメータの設定のもとで多数のシミュレーションが行われることから、膨大なデータが発生する。シミュレーション結果をそのままにしておくのではなく、異なるシミュレーションデータを比較・統合してより高度な分析を行うため、さらに事後的に活用することが考えられる。このような考え方は、上述の「第4のパラダイム」に対応するものである。今日、シミュレーションにより生み出されるデータは増加を続ける一方で、ビッグデータとしてのシミュレーションデータを効果的に利活用するための技術開発が求められている。科学分野に特有である探索的な解析の支援と、大規模なデータに対する分析要求に瞬時に応答できるような対話性の支援が求められる。本研究では、このような背景を念頭においてシステム技術の開発を行う。

本研究では特に、時間情報・空間情報を伴う時空間的なシミュレーションを想定し、大量のシミュレーションデータを管理し、分析処理を行うデータウェアハウスシステムの技術開発を行う。具体的な対象領域として地震・津波の災害シミュレ

ーション [10] を取り上げ、この分野における要求を踏まえたシステム的设计・構築を図る。本稿では、システムのアーキテクチャおよび実装技術について述べ、構築したプロトタイプに基づいて、分析のためのインターフェースや機能について説明する。

本稿の構成は以下のようになる。2章では、関連研究と本論文の位置付けについて説明する。3章では、シミュレーションデータの分析に関する要件を整理する。4章では、プロトタイプシステムのアーキテクチャなど、システムの実現手法を述べる。5章では、対象のシミュレーションデータに対して、どのようにシミュレーションデータウェアハウスを構築したかを述べる。6章では問合せと可視化の事例を示し、7章では、議論と今後の課題について説明する。最後に8章でまとめを行う。

2. 関連研究

データウェアハウスおよび OLAP に関する研究は、主としてビジネス分野を対象になされてきたが、本研究と関連の深い時空間データを対象としても開発されてきた [28] の11章や [5] に解説・サーベイがある。本稿で述べたような、空間的な領域の包含関係により概念階層を構築する手法自体は、従来の空間データウェアハウスでも行われている。実装技術に関しては、空間索引の効率的利用などの点で空間データベースの技術が導入され、現在まで発展している [20], [21]。一方で、地理空間に関するセマンティクスを取り込むためのデータモデルの拡張 [6] や、より柔軟な空間の分割方式の導入 [7] も行われている。

空間データウェアハウスに関する取り組みの多くは地図データを対象にしており、地図上での統計情報 (例: 人口分布) などを解析することが目的である。空間情報に加え時間情報も活用される場合もある。このような観点では [14] などの、移動軌跡データに対するデータウェアハウスは、連続的な時空間情報を集約表現する点では本研究と技術的に関連が深い。一方、

シミュレーションデータについては、空間情報を用いることはもちろんであるが、シミュレーションの性質上、時間情報も不可欠である点が特徴である。また、本論文における浸水深のように、次元として扱うべきか、ファクトとして扱うべきかの判断が難しい情報がある。さらに、シミュレーション時にはさまざまな実験パラメータや条件設定がなされるため、異なるパラメータでのシミュレーション実験の統合分析という新たな課題が発生する。これは従来の研究にはない、新たな問題である。

本研究では、ユーザによる探索的な分析の支援を目的としている。データウェアハウスおよび OLAP におけるデータキューブの探索に関しては、主にビジネス分野で研究がなされてきた [23], [24], [25]。その一例として、ユーザによる探索を助けるため、データキューブから着目すべき例外 (exception) 箇所を発見することが挙げられる。[23] では、例外の指標は前もって計算され、データ探索の際に分析者をガイドするのに活用された。別のアプローチ [25] では、興味深さ (interestingness) がユーザのコンテキストに基づいて計算される。さらに、[24] では、ユーザによる探索のためのいくつかの先進的な演算子が提案されている。そこでの目的も、以前の研究と同様、人手による過度な探索をせずに例外的な箇所を発見しようというものである。

ビジネス分野の一般的状況を考慮して定義された「例外」を発見しようとするこれらの研究とは異なり、本研究では地震・津波に関するシミュレーションデータを対象としており、ユーザは「被害の大きさ」などの、対象ドメインに依存した指標に興味を持つという点が異なっている。また、時間と空間に伴い、興味深いメジャーの変化やパターンを発見することも目指している。今回の開発では、利用可能なシミュレーションデータの内容を考慮して、時空間情報に伴う浸水深と避難者数の分布や組合せに着目し、特にその可視化における活用に着目した。今後は、これらの指標に加え、ユーザの探索を助けるための、地震・津波の被害解析のためのより高度な指標を導入したいと考えている。

今回のシステム開発では、Microsoft SQL Server の先進的なデータウェアハウス機能を活用して実装を進めたが、今後は、前節で述べたように動的な要素が増えていくことも予想される。データキューブでは前処理を行い事前に集約を行うことで実行時の高速化を実現しているが、近年開発が進んでいる配列 DBMS と処理を分担する、ハイブリッドなアーキテクチャをとることも考えられる。SciDB [26], [27] に代表される配列 DBMS (array DBMS) では、大規模な配列データを効率良く扱うことができ、浸水深データのようなグリッド表現のデータの格納と問合せに適している。

3. シミュレーションデータの分析

災害シミュレーションでは、しばしばある特定の領域が対象とされ、災害の発生後に、時間につれそれぞれの場所においてどのような変化が生じるかが計算される。たとえば津波シミュレーションの場合には、各地点の浸水深 (flood depth) がどの程度になるかが各時点ごとに求められ、人の避難シミュレ-

ションの場合には、それぞれの避難者が各時点にどこに存在するかが求められる。得られるシミュレーションデータは、発生時刻などの時間情報と、座標値などの空間情報と結び付けられることになる。

災害シミュレーションデータを活用した分析の例として、分析者が地震の規模や場所を指定して、地震に関する複数のシミュレーションデータを統合して被害状況を分析することを考える。「4月のウィークデイの正午に、千葉県沖の震源地 x を中心とするマグニチュード y の地震が起きたとき、東京都心部における震度、津波の被害、人の避難の状況を統合的に分析したい」という要求があるとする。この要求に応えるためには、条件に合ったシミュレーションデータを、データベース内で即座に特定し抽出できる必要がある。

一方、分析作業は一般的には探索的に行われる。詳細なシミュレーションデータをいきなり分析するのではなく、大ざっぱに粗くデータをさまざまな観点から眺めて注目すべき箇所を発見し、その後より詳細な分析のために詳細度を上げてデータを調べていく。たとえば「1km × 1km の粗いメッシュで、地震発生後の指定領域内の人の避難状況を集計し、メッシュの各セル内にいる人の人数を求めよ」といった問合せが瞬時に実行でき、可視化などの手段で各時点における各セル中の人数の分布を大まかに捉える機能があれば、分析に大いに役立つものと考えられる。

ビジネス分野を中心として、探索的・対話的に大規模データの分析を支援する技術としてデータウェアハウス (data warehouse) [9], [12], [28] がある。これは「データの倉庫」を意味しており、データベース中のデータを分析用に事前に加工しておくことにより、対話的な分析が効率よくできるように工夫されたシステムである。通常のデータベースと異なり、データの更新には対応する必要がないため、分析に特化してデータの組織化を行うことが可能となる。

本研究では、このような要求と背景を踏まえ、シミュレーションデータの統合的・対話的分析のためにデータウェアハウスの技術を活用する。特に、災害情報などに見られる時空間データに焦点を当て、この分野に特有のデータや分析の性質を考慮したシステム技術を開発する。シミュレーションデータに特化したという立場を踏まえ、本研究ではシミュレーションデータウェアハウス (simulation data warehouse) という新たな用語を用いることにする。時空間的なシミュレーションでは時間情報と空間情報の双方が関わってくるため、それらに応じた対応が必要となる。また、科学の領域はビジネス分野と比べてより探索的な側面があることから、対話的に試行錯誤しながらデータを分析することが求められる。

4. プロトタイプシステムの構成

4.1 システムのアーキテクチャ

今回構築したプロトタイプシステムのアーキテクチャを図 1 に示す。すべてのシミュレーションデータはデータウェアハウスに格納される。データウェアハウスの管理には、商用の RDBMS である Microsoft SQL Server [17] を用いる。同シス

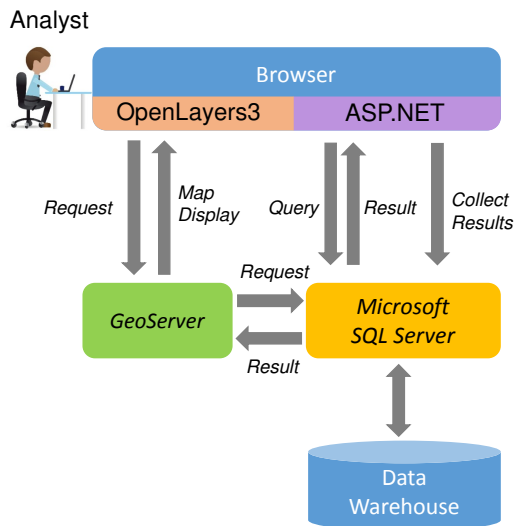


図 1 Prototype system architecture

テムは多次元データキューブを含むデータウェアハウスの基本機能が実現されているため、効果的に活用することで対話的な分析を効率よく行うことができる。

GeoServer [4] は、地理情報の共有や編集を行うオープンソースのサーバソフトウェアである。要求に応じて空間データベースにアクセスし、データベースに格納された地理情報をベクタ形式やラスタ形式の図として配信できる。その他にも、さまざまな地理・空間データに対する機能を備えている。

分析者はウェブブラウザをインターフェースとして用いて分析を行う。ブラウザ上で地図を表示するため、ブラウザ上で幾何データを扱うための JavaScript および CSS (Cascading Style Sheets) のライブラリである OpenLayers [19] を用いる。ブラウザ上の地図以外の情報管理には Microsoft の ASP.NET フレームワークを利用し、バックエンドでデータベースへの問合せを行い、データベースからの問合せ結果をもとに集計をし、GeoServer が参照するための集計結果のデータベース格納も担当する。

4.2 Microsoft SQL Server のデータウェアハウス機能

プロトタイプシステムを実現する上で利用した Microsoft SQL Server のデータウェアハウス機能について簡単に紹介する。SQL Server におけるデータウェアハウス機能は、多次元モデリング (multidimensional modeling) 機能と呼ばれており、SQL Data Server Tools に含まれている [18]。このような基本機能は他の商用 RDBMS でも提供されているので、本研究の実装のアプローチは他の RDBMS 上でも成立する。

多次元データキューブを構築するには、対象となるデータベースを指定した後、集計情報本体を表すファクトテーブル (fact table) と、分析の各次元を表すディメンジョンテーブル (dimension table) を構築する必要がある。ファクトテーブルは対象データを決めれば一意に決まるので、概念階層を考慮したディメンジョンテーブルの構築が主な作業となる。その後、キューブウィザードを用いてデータキューブの構築を行った後、各次元の属性などを追加・修正などして望むデータキューブの

構造を構築する。さらに、配置 (deployment) を行うことで、実際にデータキューブを用いた分析が可能となる。

データキューブを用いた分析処理については、SQL の標準で定められている OLAP 機能を用いることが一つの策である。しかし、この機能は SQL/OLAP として標準化自体はされているものの [2], [3], [16], その拡張機能を実装した RDBMS は少ない。そこで、プロトタイプシステムの実装においては、SQL Server で問合せ言語として提供されている MDX (MultiDimensional eXpression) [15], [28] を用いる。

また、本プロトタイプにおける問合せでは、地図上のデータの絞込みなどが多く発生することから、必要に応じてデータに SQL Server の機能である空間索引を設定する。これにより、特に狭い領域に対する分析処理について、応答速度の大幅な向上が見込める。

5. シミュレーションデータウェアハウスの構築

5.1 対象とするデータセットとシステム要件

今回のプロトタイプシステム構築にあたっては、当該分野より、地震・津波に関する二種類のシミュレーションデータの提供を受けた。データセットの一つは、東北大学 越川研究室より提供を受けた、地震発生時の高知市における津波の浸水深のデータである。もう一つのデータセットは、大規模地震の発生時の、同じく高知市における人流の避難シミュレーションデータである。こちらのデータは東京大学 関本研究室からの提供による。今回実験で用いるサンプル人流は、地震発生後に約 4 万人の人が避難する状況を、パーソントリップデータに基づいてシミュレーションしている。

両者のデータセットでは、対象の領域やシミュレーションの長さなどに違いがあるため、データセットの前処理と調整を行った。浸水深は $3,504 \times 2,364$ のグリッドで 30 秒ごとにシミュレーションされており、地震発生から 3 時間までのデータが記録されている。一方、人流データは、9 時に地震が発生し、避難開始のピークが地震発生の 60 分後にあるという条件のもとで作成された 6 時間分のデータであり、10 秒ごとにデータが記録されている。今回は、領域に関しては、浸水深データの領域をカバーするように、データキューブの最粒度のグリッドを $4,096 \times 4,096$ と設定した (このため、データが含まれないグリッドが存在する)。各グリッドは、実空間ではおよそ $10m \times 10m$ のサイズに対応する。なお、人流データの座標値のグリッドへの対応付けは容易に求めることができる。時間については、人流データに合わせ、9 時に時間が発生したと想定し、浸水深データが存在するその後の 3 時間までを対象とする。また、時間的な最小の刻み幅は 10 秒ごととした。このため、浸水深は 3 回ずつ同じ値が繰り返される。

地震・津波の分析に関する要求を把握するため、これらのデータセットに対して以下のように要件を整理した。

(1) 複数のデータの統合解析が行えること。今回は、2 種類のシミュレーションデータセット (浸水深および避難人流) を活用するが、被害の予想などの分析には複数のデータを総合して判断する必要があるため、本質的な要求である。

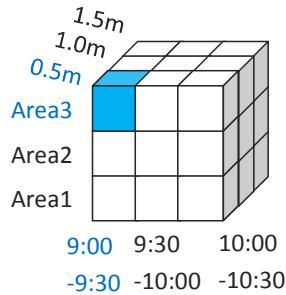


図 2 Data cube for the prototype system

(2) 可視化機能を含むユーザインタフェースにより、対話的な分析が行えること。特に、試行錯誤によるシミュレーションデータのパラメータや制約条件の調整が行える必要がある。可視化においては、地図表示上でのズームイン・ズームアウト機能も必要である。

(3) 対話的な分析を可能とするため、大規模なデータに対しても応答時間が妥当な範囲にあること。今回のシステム実装においては、具体的には最大でも 1 秒の応答時間であることを目標とする。

本研究では、統合解析のシナリオを想定してデータキューブの構築を行うことにより、上記の要件 1 に対応する。要件 2 については、地理・空間データの表示等のためのソフトウェア (GeoServer) を活用する。要件 3 については、データキューブ構築時にデータの集計を行っておくことで、コンパクトな統計データを保持しておくことができ、想定する基本的な処理については短時間での応答が可能となる。

5.2 データキューブの構築

上記のデータを統合利用した分析にはさまざまなものが考えられるが、本プロトタイプにおいては、時間、領域、および津波の浸水深を三つの軸として、避難者の数と浸水深の値を分析するシナリオを想定する。これを踏まえて構築したデータキューブの概念図を図 2 に示す。データキューブの各セルには、時間、領域、津波の深さの三つの次元をキーにしてアクセスでき、セル中にはファクトとして避難者数が入っているというイメージである。なお、分析の際には三つの次元をすべて用いる必要はなく、たとえば時間と領域の次元を用いた解析なども可能である。

データキューブの各次元に対する概念階層をどのように設定したかについて説明する。構築したデータベーススキーマの詳細は付録 A に示す。まず、時間軸については、最大の時区間を 1 時間、最小の時区間を 10 秒として、その間に 30 分、10 分、5 分、1 分という時区間の階層を設けた。図 3 に概念図を示す。概念階層のそれぞれのレベルにおいて、0 から開始するシーケンス番号を振っている。

領域については、空間を格子状に分割し、シーケンス番号を振ることにした。概念階層は、全体を一つのセルとするのが最上位であり、 2×2 に分割するのが第 2 階層である。これを $4,096 \times 4,096 = 2^{12} \times 2^{12}$ まで各レベル分割するので、13 個のレベルからなる階層となる。図 4 にその概念図を示す。各レ

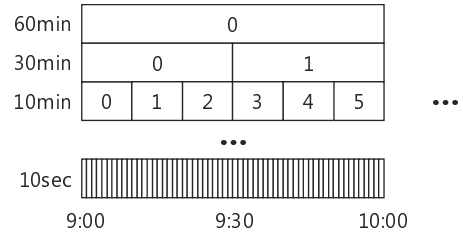


図 3 Concept hierarchy for time dimension

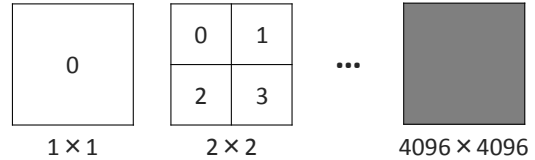


図 4 Concept hierarchy for area dimension

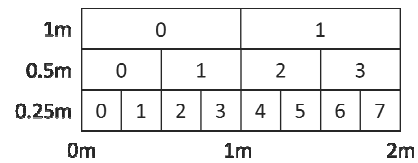


図 5 Concept hierarchy for flood depth dimension

ベルにおけるセルの番号は z-order 方式 [22] を用いる。z-order では、2 次元平面で近いセルに比較的近い番号を与えることができ、番号が容易に計算できるという利点がある。

津波の浸水深については、図 5 に示すように、3 段階のレベルでの概念階層を構築した。ここでは単純に 1m, 0.5m, 0.25m 単位での分割を行っている。

以上のような次元の設定のもとで、次元のレベルをさまざまに設定して MDX 言語を用いた問合せを発行することができる。たとえば、「時間を 10 分刻みで、領域を 16×16 分割で、浸水深を 0.5m 単位としたときの、各セルに入る避難者数の合計人数をもとめよ」といった問合せが可能となる。一部の次元のみ使用したり (例: 浸水深は用いない)、次元に範囲を設定したり (例: 9 時から 10 時までのみ計算)、集計関数を変更したり (例: 平均や最大値など) することもできる。

データウェアハウス機能を持つ RDBMS では、問合せ時の応答処理を高速化するため、あり得る問合せのパターンに応じて一部の集計処理を事前に済ませておく機能がある [9], [28]。問合せ時には部分的に計算済みの値を組み合わせることで、高速な問合せ処理を実現する。

6. 問合せと可視化の事例

シミュレーションデータウェアハウスにおける対話的分析を行うため、プロトタイプに示したような可視化技術を提案した。動機例としては、次の 3 種類の問合せを考慮した。

- Top-k 問合せ: 指定された粒度、時刻や空間領域において、top-k の避難人数 (浸水深) を持つ場所を示す。さらに、粒度の変更も行われる。

- 多数のメジャー: 指定された粒度、時刻と空間領域にお

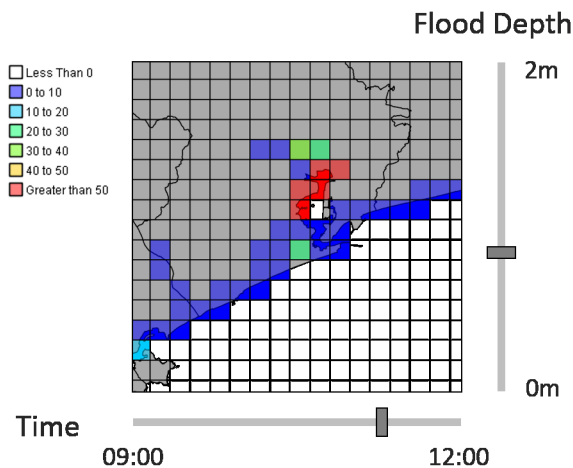


図 6 Image of the user interface

いて、避難人数と浸水深の組合せの分布を表示する。

- 時空間情報に伴う変化：指定された粒度，時間と空間範囲では，避難人数（浸水深）の変化を表す。

これらの動機例に基づき，被害状況，メジャーの関係及び時間と空間による変及を考慮した可視化技術を提案する。

プロトタイプにおける分析用のユーザインターフェース（ウェブブラウザ）では，簡易的にパラメータや絞り込み条件（開始時刻，終了時刻など）の指定を行うことができる。ユーザインターフェースのイメージを図 6 に示す。表示は，高知県の行政区域に関する地図の上に，問合せ結果である人流データの集計値がヒートマップで表示される 2 レイヤの構成となっている。二つのスライダーで時間と浸水深の指定を対話的に変更しながら，各セル内の避難者数の大まかな状況を調べることができる。空間領域に関する次元レベルの変更はデータウェアハウスの roll-up と drill-down 操作に対応し，ズームイン/ズームアウトの機能を用いて実現する。指定が変わるごとに，MDX 言語による新たな問合せがバックエンドの SQL Server に発行され，新たな集計値を計算して画面上に表示する。分析者は，これらの機能を活用して，広域な地図から段階的に地域を絞りこみながら分析を行うことができる。

図 7 ではズームインの機能を例示している。この例では，指定された開始時刻と終了時刻の間の各領域内の避難者の最大値をヒートマップ表示しており，浸水深の次元は用いていない。ここでは，分析者が上の表示をまず眺め，中心部に興味をもったという状況を考えている。分析者は左上の「+」アイコンを用いてズームインを行い，下の表示結果を瞬時に得る。領域の大きさのレベルは 1 段階細かくなっているが，この調整はシステムが自動的に行う。応答時間については，今回のシステムでは，問合せ発行から表示にいたるまで，0.7 秒程度の応答時間を達成できた。

これら以外にも，プロトタイプで実現した機能としては，着目した領域における避難者数の経時的な変化を可視化する，動画生成の機能がある。どのように状況を変化するかを把握するには，動画による提示は有効であると考えられる。

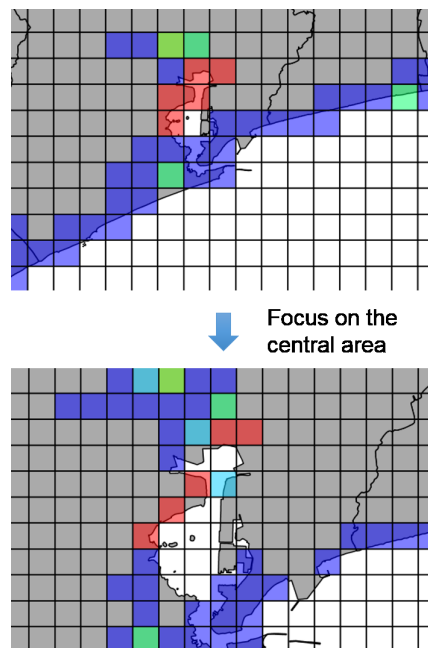


図 7 Zoom-in feature

5.1 節で述べたシステム要件について，実験結果を踏まえて検証する。1 の「複数のデータの統合解析が行えること」については，データキューブを用いての構築により，実現することができた。2 の「可視化機能を含むユーザインターフェースにより対話的な分析が行えること」については，比較的単純な分析要求についてであるが，システム上で実証することができた。3 の「応答時間が妥当な範囲であること」については，1 秒という設定した条件を下回る応答時間を達成できている。このような点から，本プロトタイプシステムは，今回の要件をクリアしていると判断できる。本システムが試作段階であり，まだ機能が十分ではないため，今回の要件は比較的対応しやすいものであった。今後機能を拡充するに際し，要件をさらに見直して検証を精緻化したいと考えている。

7. 議論と今後の課題

今回のプロトタイプ構築において，今後取り組むべき多くの課題が明らかになった。以下にいくつかの問題を挙げる。

- 次元とファクトの選択：ビジネス分野では，たとえば「期間」や「地域」を次元とし，「売上高」をファクトにするなど，次元とファクトの選び方にさほど難しさはない。しかし，今回の災害データの分析においては，論文の中で述べたように，浸水深を次元として用いたいこともあれば，浸水深自体をファクトとして扱い，「ある条件のもとで，浸水深データを可視化せよ」といった問合せを発行したいこともある。次元とファクトの組合せが必ずしも固定的に決まらないことから，シミュレーションデータウェアハウスでは柔軟に次元とファクトの設定ができる必要がある。また「避難者数と浸水深の組合せにより決まるある被害指標の値により，被害状況を可視化したい」といった要求もありえる。このような場合，ユーザ定義関数を組み込む機能が必要となる。

- 多数のパラメータへの対応：今回のプロトタイプではデータセットが固定であったため問題とはならなかったが、一般にはシミュレーションでは多数のパラメータが用いられるため、それらをどのように扱うかが問題となる。たとえば「震源地の場所を x から y まで少しずつ移動させたときの、地点 p における浸水深の変化を知りたい」という要求があったとする。この場合、異なる震源地パラメータで生成された複数のシミュレーションデータを統合利用する必要がある。データウェアハウスの枠組みでどのように表現するかが問題となる。

- 対話的分析のための実時間応答：今回の実験においては、問合せ発行から表示まで約 0.7 秒程度の応答時間となっており、対話的分析には支障はなかった。たとえば対象のシミュレーションの時間が延びた場合であっても、今回と同様の可視化処理に関しては、実際にアクセス対象となるデータキューブの領域にはさほど変化がないことから、同等の応答時間が期待できる。また、人流データが増加した場合は、データキューブ構築時に集計してしまうため、影響はない。一方、より動的で高度な問合せについては、データ量の増加が影響を及ぼすことも考えられる。支援する問合せの拡充とともに、効率的な処理技術の開発を行いたい。

- 高度な分析機能：より高度な分析機能として、たとえば「震源地以外のパラメータを指定された値に固定した上で、地点 x の浸水深が 1m 以上となるような震源地の場所をすべて求めよ」という問合せを考える。この問合せに答えるためのデータはデータウェアハウス上に存在しているため、この問合せに答えることは可能である。しかし、それをどのように表現し、いかに効率よく処理するかが課題となる。このような問合せを簡潔に記述するための問合せ言語などが必要となるかもしれない。

- システムアーキテクチャ：シミュレーションデータウェアハウスのため、今回は商用 RDBMS を用いて、それが提供するデータウェアハウスの機能や空間データベースの機能を活用した。この方針は、本稿で例示したような分析事例については有効であり、瞬時の応答が可能であったが、先に述べたような柔軟な次元やファクトの組合せについては十分ではない。Google BigQuery [8] に見られるように、多数のマシンで並列実行することで前処理なしに実時間の応答を可能とするアプローチもあるが、計算機資源や費用の面で問題がある。動的なデータキューブの問合せ処理はホットな研究課題であり [13]、研究の余地が多い。一方、SciDB [27] に見られる、大規模配列データの利活用に特化した DBMS を用いることも考えられる。

- 柔軟な可視化機能：本稿でも見たように、問合せ結果を分析に適した形でどのように可視化するかは、対話的な分析を支援する上で重要な要素である。今回のプロトタイプシステムでは、分析の種類ごとに簡易的なインタフェースを構築したが、分析や問合せの種類が増えるごとにインタフェースを構築するのはオーバーヘッドが大きい。そのため、柔軟な可視化技術の開発が必要である。また、逆の発想として、可視化システムの構成要素として DBMS を埋め込むアプローチ [1] も選択肢として考えられる。

8. ま と め

本稿では、災害シミュレーションデータの対話分析を想定した、シミュレーションデータウェアハウスの概念、プロトタイプシステムのアーキテクチャ、および分析の事例について述べた。7章に述べたとおり、本研究には多くの課題が残されている。新たなシステム技術の開発のみならず、地震・津波などに関する専門家たちとの共同研究をさらに深め、他のシミュレーションデータの活用や、統合的な分析における要求の分析やユースケースの蓄積などを進めたいと考えている。

文 献

- [1] A. Aiken, J. Chen, M. Stonebraker, and A. Woodruff. Tioga-2: A direct manipulation database visualization environment. In *Proc. ICDE*, pages 208–217, 1996.
- [2] A. Eisenberg, K. Kulkarni, J. Melton, J.-E. Michels, and F. Zemke. SQL:2003 has been published. *ACM SIGMOD Record*, 33(1):119–126, Mar. 2004.
- [3] A. Eisenberg and J. Melton. SQL standardization: The next steps. *ACM SIGMOD Record*, 29(1):63–67, Mar. 2000.
- [4] GeoServer. <http://geoserver.org/>.
- [5] L. Gómez, B. Kuijpers, and B. Moelans. A survey of spatio-temporal data warehousing. *International Journal of Data Warehousing and Mining*, 5(3):28–55, 2009.
- [6] L. Gómez, B. Kuijpers, and A. Vaisman. A data model and query language for spatio-temporal decision support. *GeoInformatica*, 15(3):455–496, 2011.
- [7] S. I. Gómez, L. A. Gómez, and A. A. Vaisman. A generic data model and query language for spatiotemporal OLAP cube analysis. In *Proc. EDBT*, 2012.
- [8] Google BigQuery. <https://cloud.google.com/bigquery/?hl=en>.
- [9] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition, 2011.
- [10] S. Hayashi and S. Koshimura. The 2011 Tohoku tsunami flow velocity estimation by the aerial video analysis and numerical modeling. *Journal of Disaster Research*, 8(4):561–572, 2013.
- [11] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.
- [12] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 3rd edition, 2002.
- [13] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed and interactive cube exploration. In *Proc. ICDE*, pages 472–483, 2014.
- [14] L. Leonardi, G. Marketos, E. Frentzos, N. Giatrakos, S. Orlando, N. Pelekis, A. Raffaetà, A. Roncato, C. Silvestri, and Y. Theodoridis. T-warehouse: Visual OLAP analysis on trajectory data. In *Proc. ICDE*, pages 1141–1144, 2010.
- [15] MultiDimensional eXpressions. http://en.wikipedia.org/wiki/MultiDimensional_eXpressions.
- [16] J. Melton. *Advanced SQL:1999 – Understanding Object-Relational and Other Advanced Features*. Morgan Kaufmann, 2003.
- [17] Microsoft SQL Server. <http://www.microsoft.com/en-us/sqlserver>.
- [18] Multidimensional modeling (Adventure Works tutorial). [https://msdn.microsoft.com/en-us/library/ms170208\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/ms170208(v=sql.120).aspx).
- [19] OpenLayers 3. <http://openlayers.org/>.
- [20] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao. Efficient OLAP operations in spatial data warehouses. In *Proc. SSTD*, pages 443–459, 2001.

- [21] D. Papadias, Y. Tao, P. Kalnis, and J. Zhang. Indexing spatio-temporal data warehouses. In *Proc. ICDE*, pages 166–175, 2002.
- [22] H. Samet. Object-based and image-based object representations. *ACM Computing Surveys*, 36(2):159–217, June 2004.
- [23] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In *Proc. EDBT*, pages 168–182, 1998.
- [24] S. Sarawagi and G. Sathe. I³: Intelligent, interactive investigation of OLAP data cubes. In *Proc. ACM SIGMOD*, 2000.
- [25] S. Sarawagi. User-adaptive exploration of multidimensional data. In *Proc. VLDB*, pages 307–316, 2000.
- [26] M. Stonebraker, P. Brown, A. Poliakov, and S. Raman. The architecture of SciDB. In *Proc. SSDBM*, pages 1–16, 2011.
- [27] M. Stonebraker, P. Brown, D. Zhang, and J. Becla. SciDB: A database management system for applications with complex analytics. *IEEE Computational Science & Engineering*, 15(3):54–62, 2013.
- [28] A. Vaisman and E. Zimányi. *Data Warehouse Systems: Design and Implementation*. Springer, 2014.

付 録

1. シミュレーションデータウェアハウスのスキーマ

今回のプロトタイプにおいて構築したシミュレーションデータウェアハウスのスキーマ記述を図 A.1 に示す。RecordID は一意に付与される主キーの ID である。EvacuationRecordTable がファクトテーブルにあたる。TimeKey, PlaceKey, DepthKey はそれぞれ、時間、領域、浸水深の次元に対応する外部キー (foreign key) である。Number と Depth はそれぞれ該当するセル内の避難者数と浸水深である。つまり、このテーブルでは、時間、領域の観点から避難者数と浸水深を捉えている。TimeTable はディメンジョンテーブルにあたる。TimeKey は ID であり、主キーである。hour, min30, min10, min5, min, sec10 は、概念階層のそれぞれのレベルで見たときのシーケンス番号を格納している。PlaceTable, DepthTable もそれぞれ空間と浸水深のディメンジョンテーブルである。詳細は省略する。

```
CREATE TABLE TimeTable(
    TimeKey int NOT NULL PRIMARY KEY,
    hour int,
    min30 int,
    min10 int,
    min5 int,
    min int
)

CREATE TABLE PlaceTable(
    PlaceKey int NOT NULL PRIMARY KEY,
    AreaID1 int,
    AreaID2 int,
    AreaID4 int,
    AreaID8 int,
    AreaID16 int,
    AreaID32 int,
    AreaID64 int,
    AreaID128 int,
    AreaID256 int,
    AreaID512 int,
    AreaID1024 int,
    AreaID2048 int,
    AreaID4096 int,
)

CREATE TABLE DepthTable(
    DepthKey int NOT NULL PRIMARY KEY,
    cm100 int,
    cm50 int,
    cm25 int
)

CREATE TABLE EvacuationRecordTable(
    RecordID int NOT NULL PRIMARY KEY,
    TimeKey int FOREIGN KEY
REFERENCES TimeTable(TimeKey),
    PlaceKey int FOREIGN KEY
REFERENCES PlaceTable(PlaceKey),
    DepthKey int FOREIGN KEY
REFERENCES DepthTable(DepthKey),
    Number int,
    Depth int
)
```

図 A.1 Schema of simulation data warehouse