

明 細 書

発明の名称：

データ解釈装置、方法及びプログラム、データ統合装置、方法及びプログラム、並びにデジタル都市構築システム

技術分野

[0001] 本発明は、データ解釈装置、方法及びプログラム、データ統合装置、方法及びプログラム、並びにデジタル都市構築システムに関する。

背景技術

[0002] 科学技術基本計画のSociety5.0が前提とするサイバー空間とフィジカル空間の高度な融合を現実的なコストで実現していくためには、新規構造物に対して新しい試みを実施していくだけでなく、古いインフラ構造物に対して膨大に蓄積されてきた設計図面等の資料を活用していくことが不可欠である。しかし、古い資料は基本的に人間が目で見えて解釈するように表現されているため、広域かつ高解像度なシミュレーションへの活用には莫大なコストが必要となる。

[0003] 1980年代以降に、設計図面から3次元モデルを自動構築する研究は数多くなされてきたが（非特許文献1）、図面の要素（線や文字）をボトムアップに組み上げていく手法に頼っていたために、部分の解釈失敗が全体に波及して3次元化が破綻し、複雑な図面に対応できない困難があった。

[0004] 近年は、図面をCADシステムに取り込んで半自動で3次元化を行う技術が主流となっている。しかし、この技術の利用にはCADシステムと設計図面の両方の知識をもつ技術者が必要であり、多数の構造物の3次元化にはやはり膨大なコストを要する。

先行技術文献

特許文献

[0005] 特許文献1：特開2011-123644号公報
特許文献2：特開平04-231514号公報

特許文献3：特開平04－030265号公報

特許文献4：特開平04－275684号公報

特許文献5：特開2018－109977号公報

非特許文献

[0006] 非特許文献1：「機械図面自動認識システム」、精密工学会誌、Vol.60, No.4, p524-529, 1994

非特許文献2：「地震応答解析モデルの堅牢な自動構築のための床形状判読手法の開発」、土木学会論文集A1（構造・地震工学）、Vol.70, No.4（地震工学論文集第33巻）, I_1124-I_1131, 2014

非特許文献3：「特集 第2章 画像理解」、電気学会雑誌、105巻、5号、409－411頁、昭和60年

非特許文献4：「画像・映像の認識と理解のこれまでとこれから」、情報処理、Vol.56、No.7、628－633頁、2015年7月

非特許文献5：「異種GISデータに記録された構造物の3D形状と属性情報の自動関連付け」、土木学会論文集A2（応用力学）、Vol.70, No.2（応用力学論文集Vol.17）, I_631-I_639, 2014

発明の概要

発明が解決しようとする課題

[0007] 設計図面を自動解釈して3次元モデルを自動構築する技術は古くから開発されてきているが、完全な情報抽出が難しく3次元モデルの汎用的で堅牢（ロバスト）な自動構築は実用化されていない。不完全な図面の自動解釈結果に対しても柔軟かつ堅牢に3次元モデルが構築できる汎用的な方法論が求められている。さらに、その3次元モデルは、多目的な用途、例えば高解像度な数値シミュレーションに使用可能であることが求められている。

[0008] 基本的に人間が目で見えて解釈するように表現されていて、コンピュータがそのままでは情報を抽出できないデータを、「非構造化データ」と称する。例えば、2D-CAD図面は非構造化データであり、線や曲線、文字列等の要素から構成されているが、必ずしも要素の集まりがもつ「図」であるとか「表」

であるなどの意味が明示的に記述されておらず、人間は目で見て「図」や「表」であると解釈し情報を読み取る。非構造化データには、2D-CAD図面の他に、ピクセルを要素とする画像や、点を要素とする点群データがある。また、一般に、非構造化データと呼ばれないデータについても必要な情報が直接抽出できない場合には非構造化データとみなしてよい。非構造化データを、その要素のデータと要素間の関係（位置関係を含む）に基づいてコンピュータが意味を自動的に解釈して断片的な情報を読み取る技術が求められている。

[0009] 2D-CAD図面に含まれる図や表は、2D-CAD図面で表現しようとするフィジカル空間の対象（例えば、ある構造物）の断片的な情報を表現したものであり、2D-CAD図面の表現対象そのもの（すなわち、フィジカル空間の対象）ではなく、表現対象の模型でもない。ここで、「模型」は表現対象のバーチャル空間における対応物を称している。また、「断片的な情報」は、必ずしも表現対象の模型を作成するために十分でなく、他の情報と統合して表現対象の模型を作成してもよい情報のことを称している。ある対象（例えば、構造物）について、低いコストでその対象の模型を作成するためには、個別の目的に応じて作成される、新規なデータから古い資料（データの種類）までの複数かつ異種のデータを活用することが求められ、複数のデータに分散している断片的な情報を統合して表現対象の模型を作成する技術が必要とされる。

[0010] データ爆発といわれる急激なデータの増加と、数十年続いている計算機性能の指数関数的な向上に対応して、データを活用する技術を継続的に高度化していくためには、プログラムの高い再利用性が必要である。しかし、一般に、個別開発されるプログラムは個々の目的に最適化された独自の形式のデータを読み書きし、プログラムの間の連携のために、素朴にはプログラムの組み合わせの数だけのデータ変換プログラムが必要となる。これは再利用のコストが著しく低いことを意味し、素朴な対処で継続的に技術を高度化することは実現不可能である。この問題に対しては、標準の表現形式を定める対処が一般的であるが、表現形式を一律的かつ固定的に定めると、データ提供

者およびデータ利用者が、データ形式を独自に工夫することを制限する。また、固定的な表現形式にすべてのプログラムが依存するために、技術が硬直化する。

- [0011] 本発明は、データを活用するための技術を継続的に高度化するために、標準と定められた形式で表現されたデータと、個々の目的に最適化された独自形式のデータの両方が同時に利用可能となる疎結合の方法を提供する。また、新規なデータから古い資料までの幅広いデータから情報を抽出するためのデータ解釈装置、方法及びプログラムを提供する。更に、複数のデータに分散している断片的な情報を統合してデータの表現対象のモデルを作成するデータ統合装置、方法及びプログラムを提供する。また、デジタル都市プラットフォームも提供する。

課題を解決するための手段

- [0012] 本開示においては、課題を解決するための手段の一例として、2次元設計図面を堅牢に自動解釈し、解釈結果の精度に応じて異なる詳細度の3次元モデルを柔軟に自動構築するための手法を提示する。
- [0013] 本願においては、図面の要素をボトムアップに組み上げるだけでなく、図形のコンテキストを自動的に特定することで、図形に適合する意味付けをトップダウンに推定しその意味付けに応じて図面から情報を補う図面解釈方法を開示している。これは、具体的には、図面の要素がノードに関連付けられた木構造を、全体と部分の関係を親子関係として自動成長（自動構造化）させていくことで実現する。
- [0014] この方法は、もし解釈に失敗しても、木構造が成長しない、つまり解釈の精度が向上しないだけで全体が破綻することのない堅牢な手法である。本研究では、この自動構造化手法を3次元モデルの自動構築でも採用する。これにより、図面の解釈の精度に応じて柔軟に詳細度の異なる3次元モデルを自動構築することが可能となる。本研究の手法で自動構築される3次元モデルは、全体と部分の関係に基づいて情報が整理されており、その形状だけでなく、構造物の内部構造や物性など多岐にわたる情報が保持され得るため、多

目的な用途への応用が期待できる。

[0015] また、本発明では、オブジェクト間の外延・内包関係に基づく自動的な型変換を実行するオブジェクト指向プログラミングが可能なインタプリタを構築し、データの解釈時およびデータの統合時に、適宜、より詳細な外延側のオブジェクトを内包側のオブジェクトから作成する（ダウンキャスト）。一般的なオブジェクト指向プログラミングにおいては、情報が足りないために適切な実行ができないダウンキャストが、本発明の手法では可能である。

[0016] すなわち、本発明は以下のとおりである。

[1] オブジェクトの型変換を自動実行するプラットフォームを備えたデータ解釈装置であって、前記プラットフォームは、前記データ解釈装置の制御部に設けられ、獲得部及び解釈部を有しており、前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得し；前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行う；ように、制御部を動作させる、データ解釈装置。

[2] 前記解釈部は、前記オブジェクト間の外延・内包関係に基づいて前記オブジェクトの外延側への型変換であるダウンキャストを実行し、前記ダウンキャスト後のオブジェクトに関連付けられるノードを前記グラフに追加させることで、前記グラフを自動成長させる、[1]に記載のデータ解釈装置。

[3] オブジェクトの型変換を自動実行するプラットフォームを用いたデータ解釈方法であって、前記プラットフォームは、コンピュータの制御部に設けられ、獲得部及び解釈部を有しており、前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得するステップ；前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行うステップ；

を実行する、データ解釈方法。

〔４〕コンピュータが実行するプログラムであって、当該コンピュータの制御部に備えられ、獲得部及び解釈部を有し、オブジェクトの型変換を自動実行するプラットフォームにおいて、前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得するステップ；前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行うステップ；を行うように、制御部を動作させる、プログラム。

〔５〕オブジェクトの型変換を自動実行するプラットフォームを備えたデータ統合装置であって、当該プラットフォームは制御部に設けられ、獲得部及び解釈部を有しており、前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得し；前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行い、１以上の前記オブジェクトを統合した統合データを構築する；ように、制御部を動作させる、データ統合装置。

〔６〕オブジェクトの型変換を自動実行するプラットフォームを用いたデータ統合方法であって、前記プラットフォームはコンピュータの制御部に設けられ、獲得部及び解釈部を有しており、前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得するステップ；前記解釈部が、オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行い、１以上の前記オブジェクトを統合した統合データを構築するステップ；
を実行する、データ統合方法。

〔７〕コンピュータが実行するプログラムであって、当該コンピュータの制御部に備えられ、獲得部及び解釈部を有し、オブジェクトの型変換を自動実

行するプラットフォームにおいて、前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得するステップ；前記解釈部が、オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行い、1以上の前記オブジェクトを統合した統合データを構築するステップ；を行うように、コンピュータの制御部を動作させる、プログラム。

〔8〕データの表現形式を自動変換するプラットフォームであって、所定の要素プログラムを有し、任意の表現形式の入力及び出力をもつ複数のプログラムが、表現形式の自動変換によってデータを介して疎結合されており、統合データの表現対象が都市又は構造物である、デジタル都市構築プラットフォーム。

発明の効果

〔0017〕 本発明は、データの表現形式をデータの意味に基づいて自動変換することでプログラムの適用範囲を拡大させ、プログラムの再利用性を向上させることが可能なインタプリタであるデータ処理プラットフォームを、実現することができる。データ処理プラットフォームは、標準と定められた形式で表現されたデータと、個々の目的に最適化された独自形式のデータの両方が同時に利用可能であり、データを活用するためのプログラムを蓄積して技術を継続的に高度化することができる。

また、データ処理プラットフォームを用いて、新規なデータから古い資料までの幅広いデータから情報を抽出するためのデータ解釈装置、方法及びプログラムを、実現することができる。

更に、データ処理プラットフォームを用いて、複数のデータに分散している断片的な情報を統合してデータの表現対象のモデルを作成するデータ統合装置、方法及びプログラムを、実現することができる。

図面の簡単な説明

〔0018〕〔図1〕実施の形態1に係るデータ解釈装置の物理的な構成を示す図である。

[図2A]「ソース1」、「ソース2」、「ソース3」の表現形式のデータを、「ターゲット1」、「ターゲット2」、「ターゲット3」の表現形式のデータに変換することを模式的に示す図である。

[図2B]「ソース1」、「ソース2」、「ソース3」から、「ターゲット1」、「ターゲット2」、「ターゲット3」への変換の媒介として、「媒介データ」を設定することを模式的に示す図である。

[図3A]自動データ変換による媒介データの別の状況を示す図である。

[図3B]本発明のインタプリタにおけるデータ自動変換の例を示す図である。

[図4]CADデータにより構成されている、橋脚構造を示す橋脚構造一般図の例である。

[図5]橋脚構造一般図における「表題欄」を示す図である。

[図6]「橋脚構造一般図」において線で構成されるオブジェクト（クラス名「LineBuf2D」）を示す図である。

[図7]線の集まりのオブジェクト（クラス名「LineBuf2D」）からサブクラス「CellSet」でないものが除外され、6個のオブジェクトが「CellSet」であると解釈されたことを示す図である。

[図8A]クラス「CellSet」のオブジェクトに対して、CADデータから抽出した文字列等を入力し、「Table」のクラスに解釈されることを示す図である。

[図8B]クラス「Table」のオブジェクトにおいて、「表題欄」の項目が存在するか分析及び分類し、存在すればクラス「表題欄」のオブジェクトに解釈されることを示す図である。

[図9]線の集まりのオブジェクト（クラス名「LineBuf2D」）からサブクラス「View」でないものが除外され、8個のオブジェクトが「View」であると解釈されたことを示す図である。

[図10]図9に示す8個のクラス名Viewのオブジェクトのうち、主構造物（D-STR）レイヤに属していないものが除外されて、残りの7個のセルを示す図である。

[図11]クラス「View」のオブジェクトに対して、CADデータから抽出した、表

題となる文字列を入力し、「橋脚正面図」特有の項目を持つものを、クラス名が「橋脚正面図」であるオブジェクトであると解釈することを、示す図である。

[図12A]正面図、平面図、及び側面図から、構造物の投影面が自動作成される様子を示す図である。

[図12B]構造物の投影面から3次元モデルが自動作成される様子を示す図である。

[図13]2D-CAD図面に含まれる「橋脚正面図」が段階的に認識されることを示す図である。

[図14]2D-CAD図面に含まれる「下部工座標表」が段階的に認識されることを示す図である。

[図15]本開示に係るデータ解釈装置及びデータ解釈方法では、2D-CAD図面認識のための段階が階層化していることを示す図である。

[図16]本開示では、2D-CAD図面の認識手法として、主として（１）「情報の入力」と、（２）「分析・解釈」との、二通りが設定されていることを示す図である。

[図17]デジタル橋梁自動構築手法の流れを示す図である。

[図18]図面解釈のための初期木構造を示す図である。

[図19]図面解釈における木構造の成長例を示す図である。

[図20]2次元CAD図面に基づくデジタル橋脚の自動構築を示す図である。

[図21]表題欄の認識過程を示す図である。

[図22]実施の形態2に係るデータ統合装置の物理的な構成を示す図である。

[図23A]個別に開発する場合のデータ処理の実装方法を示す図である。

[図23B]標準形式を利用する場合のデータ処理の実装方法を示す図である。

[図23C]自動変換を利用する場合のデータ処理の実装方法を示す図である。

[図24A]表現形式と変換経路がつくるネットワークのトポロジーの一例として標準の表現形式を中心とする星形のネットワークを示す図である。

[図24B]表現形式と変換経路がつくるネットワークのトポロジーの一例として

表現形式の自動変換に基づく自由なネットワークを示す図である。

[図25A]標準形式による固定化されたデジタル都市の形態を示す図である。

[図25B]開発持続可能性を備えるデジタル都市構築システムの形態を示す図である。

[図26]表現形式の自動変換の例をL1の一項関数 ϕ の場合について示す図である。

[図27]2D-CAD図面「006_P2橋脚構造一般図.dxf」を解釈するためのスクリプトの例である。

発明を実施するための形態

[0019] 以下、適宜図面を参照しながら、実施の形態を詳細に説明する。但し、必要以上に詳細な説明は省略する場合がある。例えば、既によく知られた事項の詳細説明や実質的に同一の構成に対する重複説明を省略する場合がある。これは、以下の説明が不必要に冗長になるのを避け、当業者の理解を容易にするためである。

[0020] なお、発明者は、当業者が本開示を十分に理解するために添付図面および以下の説明を提供するのであって、これらによって特許請求の範囲に記載の主題を限定することを意図するものではない。

[0021] 1. データ処理プラットフォームの研究

はじめに、データ処理プラットフォーム（Data Processing Platform：DPP）について、「1. 1. 研究の目的と内容」、「1. 2. データの表現形式が自動変換されることの意義と重要性」、「1. 3. データ間の論理的な同等性の定義」、「1. 4. DPPの要件と実装」、及び、「1. 5. 開発した理論および実装に対する結論」の節（それら節を、以下「本研究及び考案に係る開示」と称する）にて説明する。当該データ処理プラットフォームは、本発明者の研究に係るものであり、デジタル都市を構築するためのインタプリタとして機能する。

[0022] 1. 1. 研究の目的と内容

1. 1. 1. デジタル都市の必要性

地震や津波が都市にもたらす被害をスパコンで数値化する試みは、交通や経済に与える影響の評価など、学際的な研究に進展している。そして、その成果となるプログラム群は、高性能計算機とともに利用され、総合的な防災に役立てられることが期待されている。しかし、計算機とプログラムがあっても、災害を模擬する場となるデジタル都市がなければ実際の都市には適用できない。

[0023] 1. 1. 2. デジタル都市の要件

デジタル都市は、どのようなものであるべきか。少なくとも、被害の数値化に必要な情報を自在に抽出できなければならない。また、「データ爆発」や「計算機性能の指数関数的向上」が生じている中、すぐに陳腐化する固定的なものではなく、柔軟に新データ・新プログラムに対応して、継続的に発展可能なものであるべきである。

[0024] 1. 1. 3. デジタル都市構築の困難：異種性

デジタル都市構築の課題は、多種多様なデータから自在に情報を抽出することと、デジタル都市の継続的な発展を可能とすることの両立であるが、これは実は容易ではない。従来、自在な情報抽出のためにはデータの表現形式を標準化して統一する方策がとられるが、その表現形式は固定的であるからである。また、実情として、データは作成時の目的に応じた最適な表現形式で記録され、既存のデータの表現形式は統一されていない。

[0025] 1. 1. 4. デジタル都市構築、従来手法と予測される困難

それでも、当面の目的のために必要な標準化を行い、デジタル都市を構築することはできる。しかし、この方法は、目的が変わるたびに、デジタル都市とその構築のためのプログラムを初期段階から開発する必要が生じる非効率なものであり、デジタル都市の複雑さが再開発に耐えられない段階になった時点で、デジタル都市構築技術の発展の頭打ちをもたらすことが予想される。

[0026] 1. 1. 5. 目的

本来、標準化された機械部品の形状などと異なり、データの表現形式はそ

のデータの意味・内容が同等であれば柔軟に変更可能であり、特定の表現形式を入力として仮定したプログラムであっても、自動的に表現形式を変更する仕組みがあれば異なる形式で表現されたデータに適用が可能である。例えば、直交座標系で表現された二次元平面上の2点間の距離を計算するプログラムに対して、極座標系など他の座標系で表現された点を与えられた場合に、座標系の自動変換を行うことで一つのプログラムを複数の表現形式に対して共通利用可能とすることができる。もしこの仕組みが汎用的に実現できれば、表現形式に依存しない情報抽出プログラムが容易に作成可能となり、また、データやプログラムの入れ替えも容易となる。

[0027] 本研究及び考案に係る開示では、表現形式を自動変換する仕組みをもつインタプリタ（これをデータ処理プラットフォーム（DPP）と称する）を構築し、デジタル都市を構築するための要素プログラムをそのライブラリとして蓄積することを提案する。「プラットフォーム」とは、要素プログラムを集約して連携させる土台として機能するものを指す。本明細書に開示する「データ処理プラットフォーム（DPP）」についても、プラットフォームの語は、DPPが要素プログラムを集約して連携させる土台として機能することを意味している。

[0028] DPPとそのライブラリからなるシステムにおいて、新しいデータや新しいプログラムを取り込むことは、プログラムの単純な追加や置き換えによって可能であり、システムの高度化がデジタル都市の高度化に対応する。また本研究及び考案に係る開示では、このシステムを実現するため、表現形式の異なるデータ間の論理的な同等性を定義し、その定義に従う表現形式の自動変換方法を提示することを目的とする。

[0029] 1. 1. 6. 内容

本研究及び考案に係る開示の内容は次のように要約される。デジタル都市の構築においてデータの表現形式が自動変換されることの意義と重要性について「1. 2.」で強調し、その変換のための理論、特にデータ間の論理的な同等性について「1. 3.」で説明する。そして、その理論に対応する実

装について「1. 4.」で説明する。「1. 5.」では、開発した理論および実装に対する結論を述べる。

[0030] 1. 2. データの表現形式が自動変換されることの意義と重要性

1. 2. 1. この節の目的

この節「1. 2.」では、デジタル都市の材料となるデータの表現形式の多様性とその多様性がもたらす利用の際の課題について説明する。また、データの表現形式をデータの意味に基づいて自動変換することによって、通常はデータの表現形式によって制限されるプログラムの適用範囲が拡大し、プログラムの再利用性が向上することを説明する。そして、この再利用性の向上によって、デジタル都市を自動構築するシステムが、要素プログラムを柔軟に追加・変更できる疎結合なシステムとして実現可能であることを説明する。

[0031] 1. 2. 2. データの表現形式の多様性と利用の課題

1. 2. 2. 1. 各分野における表現形式に対する工夫

デジタル都市の材料となるデータは、通常、何らかの特定の目的のために個別に作成されたデータであり、その表現形式は、それぞれ目的や使用方法に応じて工夫されている。

[0032] 例えば、一般に、データの共有は標準化機構が定めた形式で行われ、人間も読めるテキスト形式のデータであることが多いが、高性能計算ではバイナリ形式の採用が通常であり全体を高速に読み書きできるよう同種のデータが一行に並べられることが多い。また、長期間保存するデータの検索では一部へのアクセスが通常であり、検索対象の意味・内容のまとまりに基づいたデータ構造が選択され得る。同等な意味・内容をもつデータであっても、多様な処理目的に応じて多様な表現形式が計算の主体やデータの管理主体によって定められ利用されている。

[0033] 1. 2. 2. 2. デジタル都市構築における困難1：プログラム開発における組み合わせ爆発（図23A）

デジタル都市の構築と利用は、複数のデータを入力として目標のデータを

出力する処理の体系的な適用と捉えることが可能であり、各々の処理は入力データと出力データによって特徴づけられる。一般に、同等な意味・内容をもつ処理に対しても、データの表現形式が異なれば異なるプログラムを実装する必要がある、表現形式が多様であることは著しく開発効率を低下させる恐れがある。例えば、二つの物体の距離を出力する簡単な処理に対しても、図23Aのように入力データの表現形式の組み合わせごとに実装が必要であり、容易に組み合わせ爆発が生じる。

[0034] 図23A～図23Cは、3通りのデータ処理の実装方法を示す図である。矢印はデータの表現形式の変換を表す。図23Aは、個別に開発する場合であり、図23Bは、標準形式を利用する場合であり、図23Cは、自動変換を利用する場合である。なお、図23A～図23Cでは、 $F(X_i, Y_j)$ は表現形式 X_i のデータ x_i と表現形式 Y_j のデータ y_j を入力して、所定のデータ処理を行う関数を表している。また、 $i \neq i'$ であれば X_i と $X_{i'}$ の表現形式は異なり、同様に、 $j \neq j'$ であれば Y_j と $Y_{j'}$ の表現形式は異なるものとする。

[0035] 1. 2. 2. 3. デジタル都市構築における困難2：標準フォーマットの問題

もし、図23Bのように、利用するデータの表現形式を統一し、処理プログラムの実装を標準の表現形式に対してのみ行えば、類似の実装を繰り返すことを避けることができる。しかし、この方法論は、目的に適合した表現形式を設計して利用することを阻害するため、特に高性能計算の分野では実現可能ではない。高性能計算の分野で開発されるプログラムは、多くの場合に、計算性能の向上を目的として開発者が独自に設計した表現形式のデータを入力とするからである。標準化機構によって標準の表現形式が定められることの有用性は言うまでもないが、高性能計算分野で開発された高性能なプログラムを応用した数値シミュレーションをデジタル都市に適用するためには、多種多様な表現形式のデータが取り扱える、データとプログラムの連携に関する新しい方法論が必要である。

[0036] 1. 2. 3. 自動データ変換による処理プログラムの再利用

1. 2. 3. 1. 自動変換による処理プログラムの再利用（経路探索含む）

図23Cに示すように、データの表現形式をデータの意味に基づいて自動変換することで、プログラムの適用範囲を拡大させ、プログラムの再利用性を向上させることが可能である。通常、プログラムの適用範囲はデータの表現形式に依存するが、あらかじめ表現形式間にデータの意味・内容を保つ変換経路を定義し、この変換経路をたどる経路探索が自動で実施される機能を実装しておけば、プログラムの適用範囲は意味・内容に依存する形に拡大する。また、経路探索の方法を適切に定義することで複数の処理プログラムの実装から最適なものを選択して実行することも可能と考えられる。

[0037] 1. 2. 3. 2. 媒介データ・標準の相対化

図24のように、表現形式をノード、変換経路をリンクとするネットワークを考えることができる。ここで、図24は、表現形式と変換経路がつくるネットワークのトポロジーである。図24Aは、標準の表現形式を中心とする星形のネットワークであり、図24Bは、表現形式の自動変換に基づく自由なネットワークである。

[0038] 標準の表現形式に基づく方法論は、図24Aのように、このネットワークのトポロジーを標準の表現形式を中心とする星形に制限するものであり、標準の表現形式は、データとプログラムを仲介する中間データ形式の役割を果たす。一方で、表現形式の自動変換に基づく方法論では、ネットワークのトポロジー構成が自由であり、データとプログラムが結合する位置も自由である（図24B参照）。また直接プログラムと結合していないデータも自動変換によって間接的に自動結合されるため、データとプログラムの入れ替えが容易である。そして、このネットワークは新しい表現形式や処理プログラムの追加により成長が可能である。ネットワークの成長の結果として、星形のトポロジーが構成されることもあり得るが、中心に位置する表現形式は、トップダウンの形で定義される標準の表現形式と異なり、ボトムアップの形で自然に定まるデファクトスタンダードな表現形式である。

[0039] 1. 2. 3. 3. データの利用者負担の低減

近年では、デジタル都市の材料として有用なオープンデータが、クリアリングハウスを通じて数多く公開・共有されている。しかし、オープンデータの表現形式は多種多様であり、応用プログラムの開発者が複数のデータを組み合わせる際には、多くの場合に、データ一つ一つの表現形式を理解して、データから情報を抽出するプログラムを開発することから始める必要がある。もし表現形式の自動変換に基づく方法論を実現するシステムがあり、情報抽出プログラムが共有できれば、オープンデータと既開発の処理プログラムが自動的に結合され、応用プログラムの開発が容易になる。また、そのシステムでは、データの本質的でない詳細、すなわち特定の表現形式に固有の実装を意識せずにデータを取り扱うことができるため、データの利用者の学習負担も低減される。

[0040] 1. 2. 3. 4. 処理プログラムのラップによる再利用性の向上

表現形式の自動変換がデータの実装の詳細を隠蔽する一方で、プログラムの再利用性を高める別の仕組みであるアプリケーションプログラミングインターフェース（API）は、プログラムの実装の詳細を隠蔽する。Web APIによる都市情報の提供は、前述の情報抽出プログラムの共有にあたり、応用プログラムの開発コスト低減が期待される。このため、すでに都市の様々な情報がWeb APIとして提供されており、その数は今後も増加することが想定されるが、この状況は独自形式の有用なデータが増加することと類似している。APIを備えた応用プログラムも、その機能が入力と出力のデータの表現形式で特徴づけられるため、表現形式の自動変換を適用することが可能である。この場合に、入力データはAPIに対する指示を表現したものとなる。

[0041] 1. 2. 4. デジタル都市構築システム

1. 2. 4. 1. デジタル都市の定義

都市の構成要素である構造物の耐用年数は長く、一方で研究開発の進展は速い。このため、設計、施工、維持管理、の各段階で生じるデータを蓄積しておき、後に開発される技術を適用して多目的に利用することが想定される

。実際に、既存の構造物に対して膨大なデータが保存されており、そのデータの有効活用が進められている。例えば、数値シミュレーションを用いた地震や津波の被害推定がその一例である。数値シミュレーションを実施するためには、相応の詳細度をもつ都市の模型をコンピュータ上に表現する必要があり、その模型は、複数のデータから、都市の情報を断片として抽出し、現実の都市の構成要素に対する知識を表現したデータへと統合することで構築される。本研究及び考案に係る開示では、この統合された知識としての都市の模型をデジタル都市と定義する。デジタル都市は、数値シミュレーションの入力データなど、目的のデータを作成するための情報源として利用する。

[0042] 1. 2. 4. 2. 標準形式に基づくデジタル都市

デジタル都市の構築と利用の素朴な形態は、図25Aのように、都市の構成要素をすべて表現可能な標準の形式を定めて、材料となるデータからその形式で表現されたデータを作成し、標準形式のデータに対して実装されたプログラムを適用して目的のデータを作成することである。しかし、この形態が基づいているデータとプログラムの連携に関する方法論は、前述の通り、実際の多様な表現形式に対応できない。また、すべてのプログラムが標準形式に依存するため、標準形式の変更に多大な労力を要する。新しい標準形式に合わせて既存の古い標準形式のデータを手作業で変換する作業も膨大である。その結果として、都市の数値的な表現は固定化され、技術の発展を阻害する。

[0043] 1. 2. 4. 3. デジタル都市構築システム

本研究及び考案に係る開示では、デジタル都市の構築と利用の形態として、図25Bのように、表現形式を自動変換する仕組みをもつ独自創作のインタプリタであるDPPを土台として、デジタル都市を構築するための要素プログラムを蓄積するデジタル都市構築システムを開発し、デジタル都市の材料データと目的データをそれぞれその入力と出力とすることを提案する。このシステムでは、プログラム間がデータを介して結合され、データとプログラムの間の結合がデータの表現形式ではなく、意味・内容に依存する。このこと

は、意味・内容が同じであれば表現形式の違いに依らず結合を柔軟に変更可能であることを意味する。要素プログラムを柔軟に追加・変更できる疎結合なシステムを利用することで、重複した開発を避けて技術の頭打ちを防ぐ設計を早期から採用することは、デジタル都市構築技術の発展にとってきわめて重要である。

[0044] 1. 3. データ間の論理的な同等性の定義

異なる形式で表現された二つのデータが同等であることは、あるデータを異なる形式で表現された別のデータで代用する正しい置き換えが相互に可能であることである。二つのデータが正しく置き換え可能であるための条件について、もし明確な定義がなければ、置き換えたデータから本来と異なる意味・内容が抽出され得る。この節「1. 3.」では、論理的に正しい置き換えが常に可能となるように、表現形式が異なるデータ間の論理的な同等性を、述語論理を用いて定義する。

[0045] 1. 3. 1. データ表現間の外延・内包関係

データの表現形式を自動変換する基礎となる、データ表現間の論理的な同等性を定義するため、データ表現全体の集合を議論領域D1とした述語論理L1を考える。ここで、データの各表現形式はL1の1項述語に対応し、その1項述語の値が真となるD1の部分集合がその形式で表せるデータ表現の全体とする。二つのデータ表現が論理的に同等であることは、D1の元であるデータ表現間の外延・内包関係を表すL1の2項述語 $\Rightarrow$ が双方向に成立することと定義する。なお、「外延」という用語の代わりに「下位概念」、「内包」という用語の代わりに「上位概念」、「外延・内包関係」という用語の代わりに「上位概念・下位概念の関係」という用語がそれぞれ用いられてもよい。

[0046] 2項述語 $\Rightarrow$ は、次の反射律および推移律を満たすものとして定義する。ただし、 $\Rightarrow$ は論理記号である。

[0047] $\forall x (x \Rightarrow x)$

$\forall x \forall y \forall z ((x \Rightarrow y) \& (y \Rightarrow z) \rightarrow (x \Rightarrow z))$

以降、 $x \Rightarrow y$ が成立していることを、xはyを内包にもつ、あるいは、yはx

を外延にもつということにする。また、2項述語 $\Leftrightarrow$ を次のように定義し、

$$\forall x \forall y ((x \Leftrightarrow y) \leftrightarrow ((x \Rightarrow y) \&\& (y \Rightarrow x)))$$

$x \Leftrightarrow y$ が成立しているとき、 x と y は同等であると定義する。

[0048] そして、L1の n 項関数 ϕ および n 項述語 ϕ は次の2条件を満たすとき正則であると呼ぶことにする。

[0049] $\forall x \forall y ((x \Rightarrow y) \rightarrow (\phi (\dots, x, \dots) \Rightarrow \phi (\dots, y, \dots)))$ (式1)

$\forall x \forall y ((x \Rightarrow y) \rightarrow (\phi (\dots, y, \dots) \rightarrow \phi (\dots, x, \dots)))$ (式2)

例えば x, y をそれぞれログハウスと家のデータ表現、 $\text{Roof}(a)$ を a に対して a の屋根のデータ表現を対応させる正則な1項関数とすると、一番目の式から、 $(x \Rightarrow y) \rightarrow (\text{Roof}(x) \Rightarrow \text{Roof}(y))$ が成り立つ。これは、「ログハウスが家であれば、ログハウスの屋根は家の屋根である。」ということの意味する。また、 $\text{HasRoof}(a)$ を「 a には屋根がある」という属性を表す正則な1項述語とすると、二番目の式から $(x \Rightarrow y) \rightarrow (\text{HasRoof}(y) \rightarrow \text{HasRoof}(x))$ が成り立つ。これは、「ログハウスが家であるならば、家に屋根があればログハウスにも屋根がある。」ということの意味する。

[0050] 恒等関数は、正則関数であり、常に真を返す述語と常に偽を返す述語は正則述語である。一方で、一般には、データ表現間の外延・内包関係について、反射律および推移律の成立を課しただけでは、どの関数・述語が正則関数・正則述語であるか定まらないし、二つのデータ表現間が外延・内包関係にあるかどうか定まらない。本研究及び考案に係る開示では、最初に基本的な正則関数・正則述語だけが定義されており、その後に正則関数・正則述語を逐次的に定義していく状況を考え、既定義の任意の正則関数および正則述語に関して式1と式2を満たせば、二つのデータ表現が外延・内包関係にあると定義する。これは、正則関数・正則述語の定義によって、対象の意味付けが行われることを意味する。本研究に係る開示での目的は、表現形式間の本質的でない差異を捨象することであり、表現形式に依らずに値が決定でき

る関数や述語を正則関数および正則述語と定義することで、データ表現間の外延・内包関係を決定する。

[0051] 一例として、2次元平面上の点を直交座標系で表現する表現形式Cartesian2Dと、2次元平面上の点を極座標系で表現する表現形式Polar2Dについて考える。Cartesian2Dのデータ表現は、x座標とy座標の組 (x, y) であり、Polar2Dのデータ表現は、動径 r と偏角 θ の組 (r, θ) である。2次元平面上の点の位置は、データ表現によらず、与えられた直交座標系に対してx座標とy座標を測ることができる。例えば、直交座標系と極座標系が $x = r \cos \theta$, $y = r \sin \theta$ と関係づけられているとき、Polar2Dのデータ表現 (r, θ) に対しては、この関係式からx座標とy座標を測ることができる。ここで、x座標とy座標の測量を、2次元平面上の点に対して実数の表現形式Realのデータ表現を対応させる正則関数と定義すると、Polar2Dのデータ表現がCartesian2Dのデータ表現の外延である場合には、式1より二つのデータ表現に対するx座標とy座標の測量結果が同等、すなわち二つのデータ表現が同一の点を表す。

[0052] 別の例のため、2次元平面上の色のついた点を、x座標とy座標、RGB値の組 (x, y, r, g, b) で表現する表現形式 Cartesian2D_RGB を考える。x座標とy座標の測量はCartesian2D_RGBのデータ表現に対しても可能であり、Cartesian2D_RGBのデータ表現がCartesian2Dのデータ表現の外延である場合には、先の例と同様に、二つのデータ表現に対するx座標とy座標の測量結果が同等で、二つのデータ表現が同一の点を表す。

[0053] 1. 3. 2. 表現形式間の外延・内包関係

前小節「1. 3. 1.」では、データ表現間の外延・内包関係を定義した。この小節では、L1の1項述語全体の集合を議論領域D2とする述語論理L2を導入し、L1の1項述語間の関係として外延・内包関係を捉えなおす。これは、1項述語名をラベル、各データ表現を指示対象とした記号間の関係としてデータ間の外延・内包関係を捉えることを意味し、データの性質とデータの表現方法を分離して扱うことを可能とする。具体的には、ある情報をデータとして記録する際に、先にそのデータの性質をL2において論じておき、その後にL

2で表現された性質をもつようにそのデータの具体的な表現方法を決定することを可能とする。

[0054] L1の1項述語A, BがL2において外延・内包関係にあることを、A, BがL1において次の関係式を満たすことと定義する。

[0055] $\forall x (A(x) \rightarrow \exists y (B(y) \ \&\& \ (x \Rightarrow y)))$ (式3)

この式より、L2においてもL1においてと同様に、外延・内包関係について反射律と推移律が成立することがただちにいえる。また、それぞれ条件 $\forall x (T(x))$, $\forall x (\neg F(x))$ を満たすL1の特殊な二つの1項述語T, Fを考えると、Tはすべての1項述語を外延にもち、Fはすべての1項述語を内包にもつ。

[0056] L1の各データ表現に対して、そのデータ表現に対してだけ値を真とするL1の1項述語を考えることが可能であり、その1項述語はD2の元である。本研究及び考案に係る開示では、表記を簡単にするため、L1のデータ表現をaと記したときに、このデータ表現aに対応するD2の元も同様にaと記すことにする。ただし、データ表現は小文字で表記し、大文字で表記するデータの表現形式と区別する。式3により、あるデータ表現a, bがL1で $a \Rightarrow b$ の関係にあるとき、L2においても $a \Rightarrow b$ の関係が成立する。また、ある表現形式Aとそのデータ表現aとの間には、式3より、L2において $a \Rightarrow A$ が成立する。

[0057] 式3の定義のもとで、L1の正則関数・正則述語は、次のようにL2の正則関数・正則述語に拡張することができる。まず、L2におけるn項正則関数 ϕ の値はL1の1項述語であり、その1項述語に対応するD1の部分集合を、L2における関数 ϕ の各引数、すなわちL1の1項述語、をそれぞれ真とする全てのL1の元の組に対して計算したL1における関数 ϕ の値全体の集合と定義する。また、L2におけるn項正則述語 ϕ の値が真であることを、その述語の各引数、すなわちL1の1項述語、をそれぞれ真とする任意のL1の元の組に対して、L1の対応する述語 ϕ の値が真となることと定義する。このとき、拡張されたL1の正則関数・正則述語に対して、L1における式1と式2に対応する式4と式5が、L2においても成立する。

[0058] $\forall A \forall B ((A \Rightarrow B) \rightarrow (\phi (\dots, A, \dots) \Rightarrow \phi (\dots, B, \dots)))$ (式4)

$\forall A \forall B ((A \Rightarrow B) \rightarrow (\phi (\dots, B, \dots) \rightarrow \phi (\dots, A, \dots)))$ (式5)

[0059] ある表現形式とそのデータ表現の関係は、オブジェクト指向プログラミングにおけるクラスとインスタンスの関係に対応し、外延・内包関係は継承関係に対応する。この観点から、式5を1項述語に関する式に特殊化したものは、オブジェクト指向プログラミングにおけるリスコフの置換原則に対応する。ただし、本研究及び考案に係る開示では、「クラス」と「データ型」（単に「型」とも記す）を同義語とする。

[0060] 真理値をデータとして表現する表現形式を定義すると、述語が関数として定義できることは重要である。このことは、正則述語を使わずに正則関数のみでデータを表現することができることを示す。また、真理値の定義の仕方によって、多値論理を表現することもできる。このことから、以降では正則関数に焦点を絞って議論する。

[0061] L2において $A \Rightarrow B$ であるとき、L1において

$$\forall x (A(x) \rightarrow (B(B \langle A \rangle(x)) \ \&\& \ (x \Rightarrow B \langle A \rangle(x))))$$

を満たすAからBへの写像 $B \langle A \rangle$ が少なくとも一つ存在し、この写像 $B \langle A \rangle$ を自動変換写像と呼ぶこととする。自動変換写像が実装されているとき、データの表現形式を自動変換することが可能となる。自動変換写像とD2の元は、圏論における射と対象に相当すると考えられ、一項正則関数は共変関手に相当すると考えられる。

[0062] 1. 4. DPPの要件と実装

この節「1. 4.」では、DPPの要件と実装について説明し、前節「1. 3. データ間の論理的な同等性の定義」で定義したデータ間の外延・内包関係に基づいて、DPPがデータの表現形式を必要に応じて自動変換（すなわち、型変換を自動実行）し、あらかじめ登録された処理プログラムに適用する方法について説明する。

[0063] 1. 4. 1. データ処理プラットフォームと、一般的なオブジェクト指向プログラミングとの特性の差異

以下に、データ処理プラットフォームと、一般的なオブジェクト指向プログラミングとの特性の差異を示す。

[0064] [表1]

	データ処理 プラットフォーム	一般的なオブジェクト 指向プログラミング
継承関係の 定義	外延・内包関係があることが 条件である。	外延・内包関係がなくても 定義できる。
	データ構造の共有がなくても 定義できる。	データ構造の共有が条件である。
データ抽象化	継承関係にある型の間では、リスコフの置換原則を拡張した式(※式5)に加えて、独自の式(※式4)を必ず満たす。	継承関係にある型の間で、置換可能性を保証するためのリスコフの置換原則を満たすことが推奨されている。
プログラムの 適用範囲	意味・内容に規定される形に 拡張される。	入力データの表現形式に 規定される。
ユーザ定義の 型変換の 回数の制限	型変換に回数制限はない。 型変換の経路が自動で探索され、 探索結果に沿って必要な回数の 自動変換を自動実行する。	一回までに制限されている。

[0065] 本開示に係るDPPは、表1に示すように、一般的なオブジェクト指向のプログラミングとは異なる考え方に基づくものである。一般的なオブジェクト指向のプログラミングでは、継承関係の定義において、データ構造の共有が条件とされている。それに対して、DPPでは、継承関係の定義において、データ構造の共有は要件とされていない。かわりに、外延・内包関係があることが、クラス間の継承関係が成立する条件とされている。

[0066] DPPにおいて、ユーザは、データ構造の共有という制約を受けずにクラス間の継承関係を定義することが可能である。自動的な（暗黙の）型変換の回数に制限はなく、型を変換する最適な経路が自動で探索され、探索された経路に沿って必要な回数の変換が自動実行される。DPPの型変換においては、論理的に同等な型の変換（同値キャストと称する）に加えて、外延側への型変換（ダウンキャストと称する）と内包側への型変換（アップキャストと称する）の両方が自動実行される。ただし、ダウンキャストはオブジェクトごとに変換先の型へ変換できる条件を定義しておき、その定義が満たされた場合にのみ実行される。

[0067] 本発明者は、インタプリタとして機能するDPPを、C++によって記述した。当然のことながら、当業者であれば、本明細書の記載に従い、同様のインタプリタを作成することができ、そのようなインタプリタによって実装される本開示のデータ処理プラットフォーム、並びに当該プラットフォームを備えた装置、方法、プログラムもまた、本発明の範囲に含まれる。

[0068] DPPを記述する言語も、C++に限らずとも、本明細書に記載する要件と同等の要件を満たし、同等の実装が可能である限り、任意の他の言語で記載されたものであってよい。

[0069] 1. 4. 2. 要件及び実装

本研究及び考案に係る開示におけるデジタル都市構築システムは、表現形式を自動変換する仕組みをもつ本発明開示のインタプリタ（DPP）を土台として、デジタル都市を構築するための要素プログラム群が蓄積されて構成されており、材料データからデジタル都市を構築し、目的のデータを作成する。

このDPPの実装にあたり、次の３点を要件として考慮する。

（要件１）デジタル都市構築システムに処理を指示するためのインタフェースを備えること。

（要件２）既存の処理プログラムのラッパーとして機能すること。

（要件３）処理プログラムを個別開発可能なライブラリとして蓄積可能であること。

[0070] 前節「１．３．」で触れたとおり、ある表現形式とそのデータ表現の関係は、オブジェクト指向プログラミングにおけるクラスとインスタンスの関係に対応し、外延・内包関係は継承関係に対応する。この対応に従って、デジタル都市構築システムの利用者は、DPPに一種のオブジェクト指向言語を用いて処理を指示することとする。実装の詳細は後述するが、DPPによって理解される言語は、通常と異なり、継承関係にあるクラス間でデータ構造を共有する必要がなく、個々のクラスのデータ構造を自由に設計可能であり、また、関数適用時に必要に応じて継承関係を辿る経路探索と型の自動変換が行われる。

[0071] 既開発のプログラムをDPPに組み込む際に、古いプログラムの機能を新しい言語で実装し直すのは非効率的である。DPPでは、C++言語のクラスと関数をDPPのクラスと関数としてラップして取り扱えるように実装する。これにより、DPPは、既開発のデータとプログラムを本研究及び考案に係る開示の方法論で疎結合に連携させることが可能である。

[0072] DPPでは、クラスの定義とそのクラスに関連する処理プログラムを個別にコンパイルして動的ライブラリにまとめられるよう実装する。ライブラリは必要に応じてロードされ、ライブラリ内で定義されている継承関係に応じた変換経路の探索と自動変換が実行される。DPPの利用者は、独自の表現形式や処理プログラムをライブラリ化してロードすることで、システムを自由に拡張することが可能である。

[0073] １．４．３．自動変換の実装と処理プログラムの適用方法

表現形式間に定義された自動変換写像を変換経路として辿ることで表現形

式の変換が可能である。多様な経路の変換が可能であるが、一般に、あるデータが異なる経路で変換された場合に、同等なデータ表現になるとは限らないことに注意が必要である。データを変換した結果が経路によらずに同等となるためには、変換先の表現形式がL1において次の条件をみたせばよい。

$$[0074] \quad \forall x \forall y ((A(x) \ \&\& \ A(y)) \rightarrow (\exists k ((k \Rightarrow x) \ \&\& \ (k \Rightarrow y)) \rightarrow (x \Rightarrow y)))$$

この条件を満たす表現形式を粒状であると定義する。粒状でない表現形式に変換を行う場合には、自動変換写像の定義によっては不要な情報の損失が生じる可能性がある。デジタル都市の材料データについて、既存の表現形式の多くは粒状であるが、そうでないものも存在すると考えられる。従って、DPPにおける表現形式の自動変換機能に対して粒状でない表現形式を考慮した実装を行う一方で、DPPのクラスの実装は原則として粒状とすることが適切である。

[0075] DPPのクラスは、前節「1. 3.」で定義された述語論理L2の対象、すなわちL1の1項述語を実装したものであり、データの表現形式である場合の他に、内部データを持たない単純な属性を表す場合がある。ここで、「属性」はL1の1項述語でデータの表現形式でないものを称している。表現形式から属性への自動変換写像は恒等写像とする。属性は、多様な表現形式間にまたがって定義されるため、一般的に粒状ではない。属性のみを取り扱い、属性の概念としての性質を表現する正則関数を適切に実装すれば、DPPは、オントロジー記述言語を定義することも可能と考えられる。

[0076] 二つの表現形式間に外延・内包関係が成立していても、その二つの表現形式間を結ぶ変換経路が用意されなければ、表現形式の自動変換はできない。利用者に自動変換可能か否かの誤解が生じないように、DPPは、自動変換写像が実装された場合にのみ、外延・内包関係が成立していると認識する。これは、一見、正則関数と正則述語の定義によって外延・内包関係が定まるとする理論に反するが、不都合は生じない。実際に、プログラム開発者が理論上の定義と実装上の定義を明確に区別し、理論上で外延・内包関係が成立して

いる場合にのみ実装上で外延・内包関係が認識されるようにすれば、外延・内包関係が理論上ではないにもかかわらず実装上ではあるとする誤認識を防ぐことができる。

- [0077] 表現形式をノード、自動変換写像をリンクとするネットワークにおいて、表現形式の自動変換は、コストが最小となる経路を探索して実行できるよう実装される。一例として、表現形式の自動変換は、例えば、ダイクストラ法に基づいてコストが最小となる経路を探索して実行し、自動変換写像が恒等写像である場合にはコストを0とし、その他の場合には1とすることができるが、これに限られるものではない。また、経路探索を効率化するため、外延・内包関係に基づいて、目標へと至らないノードとリンクはあらかじめ排除される。
- [0078] 図26に式4に基づく表現形式の自動変換の例をL1の1項関数 ϕ の場合について示す。 ϕ は、入力の実装形式ごとに処理プログラムが実装され得るが、 B を定義域とする正則関数 $\phi(B)$ が実装されているとき、DPPにおいて、 $A \Rightarrow B$ であれば、自動変換写像 $B(A)$ が定義されており、 A の任意のデータ表現 a に対して $\phi(B)(B(A)(a))$ を計算することができる。これは表現形式の自動変換により、 $\phi(B)$ の定義域が A を含むように拡大されているとみることができる。一方で、関数 $\phi(A)$ を実装すれば、式4より、 $\phi(A)(a) \Rightarrow \phi(B)(B(A)(a))$ である。二通りの $\phi(a)$ の計算が可能であるが、自動変換により情報を落としてから関数を適用するよりは、関数を適用してから必要に応じて情報を落とす方が適切であるため、DPPは、 $\phi(A)$ が定義されている場合には、後者の計算方法を採用する。これは一般の n 項関数についても同様であり、この実装された関数の選択方法は、一般的なオブジェクト指向プログラミングにおけるポリモーフィズムと同様である。

[0079] 1. 5. 開発した理論および実装に対する結論

異なる形式で表現された二つのデータが同等であるための条件を明確に定義し、その定義に基づいた実装を行うことで、データの表現形式を論理的な間違い生じさせずに自動変換することに成功している。DPPを土台として、デ

デジタル都市を構築するための要素プログラムを蓄積することで、デジタル都市構築技術を効率的かつ継続的に発展させることが可能と考えられる。

[0080] 古い表現形式を拡張した新しい表現形式のデータが特定の条件下で古い表現形式に変換でき、古いプログラムを活用できる場合がある。プログラムの再利用性を向上する観点からは、個々のデータ内容に依存して自動変換の可能性が定まる柔軟なデータ変換技術が必要である。このため、DPPは、ある形式で表現されたデータが特定の条件を満たすかを調べ、満たす場合には自動変換を実行することで関数の適用範囲をより拡大している。

[0081] 2. 実施の形態における目的

以下、発明の実施の形態に係る本開示は、データを活用するための技術を継続的に高度化するために、標準と定められた形式で表現されたデータと、個々の目的に最適化された独自形式のデータの両方が同時に利用可能となる疎結合の方法を提供することを、目的とする。即ち、本開示は、一律的な形式でデータを表現する標準化に替わり、表現形式の自動データ変換に基づいたデータの抽象化によって疎結合を実現し、異種データ・異種プログラム群を柔軟に連携させる方法を提供することを目的とする。

また、そのような疎結合の方法を用いる、新規なデータから古い資料までの幅広いデータから情報を抽出するためのデータ解釈方法を提供することを、目的とする。

更に本開示は、そのような疎結合の方法を用いる、複数のデータに分散している断片的な情報を統合してデータの表現対象の模型を作成するデータ統合方法を提供することを、目的とする。

[0082] 3. データ処理のためのプラットフォーム、及び、オブジェクト

上述の目的を達成するために、本発明者は、「1. データ処理プラットフォームの研究」にて説明した、データ処理のためのプラットフォーム、及び、当該プラットフォームに基づいて定義され構築されるオブジェクトの考案に到っている。本開示では、発明者の考案に係る当該プラットフォーム（インタプリタとして機能する）を、データ処理プラットフォーム（DPP）と称し

ている。又、DPPに基づいて定義され構築されるオブジェクトを、単にオブジェクトと称することとする。本発明者によるDPP及びオブジェクトの特性について、以下、整理して説明する。

- [0083] ある形式のデータを、目的の別の形式のデータに変換するには、通常、独自の変換プログラムが必要となる。例えば、図2Aに示すように、「ソース1」、「ソース2」、「ソース3」・・・「ソースM」の表現形式のデータ（即ち、M種類の表現形式のデータ）を、「ターゲット1」、「ターゲット2」、「ターゲット3」・・・「ターゲットN」の表現形式のデータに変換することを想定する。この場合、 $M \times N$ 通り（図2Aでは、具体的には $3 \times 3 = 9$ 通り）の変換プログラムが必要になる。
- [0084] ここで、図2Bに示すように、「媒介データ」の設定を導入する。つまり、「ソース1」、「ソース2」、「ソース3」・・・「ソースM」から、「ターゲット1」、「ターゲット2」、「ターゲット3」・・・「ターゲットN」への変換の媒介として、「媒介データ」を設定するとする。この場合、変換プログラムは $M + N$ 通り（図2Bでは、 $3 + 3 = 6$ 通り）で済むことになる。即ち、「 $M \times N$ 通り」から「 $M + N$ 通り」へ減少する。
- [0085] しかしながら、図2Bに示す「媒介データ」は、全てのコンポーネント（変換プログラム、データ）が、図2Bに示す「媒介データ」に依存することになり、標準化データである「媒介データ」に変更が生じれば、必ず $M + N$ 通りの変換プログラムの再開発が必要となってしまう。
- [0086] そこで、本開示の発明者は、前述のような、自動データ変換を行うDPPを考案している。
- [0087] DPPでは、一般的なオブジェクト指向プログラミングにおけるオブジェクトと異なり、型変換が最適な変換経路で自動実行されるオブジェクトを導入・定義している。オブジェクトにおけるクラスの継承関係については、以下の（条件1）～（条件3）のように定義される。
- [0088] （条件1）継承関係は反射律と推移律を満たす。
- [0089] （条件2）外延・内包関係が成立する時にクラス間の継承関係が認められ

る。

例えば、第1のオブジェクトが (x, y) 座標を持つ第1のクラスに属し、第2のオブジェクトが (r, θ) 座標（極座標）を持つ第2のクラスに属し、第1のクラスは第1のスーパークラスのサブクラスである、という状況を想定する。これらのクラスがC++により規定されるものである場合、座標表現が同じでないと継承関係は持ち得ないという原則から、第1のスーパークラスと第2のクラスとの間に継承関係は存在しない。これに対して、これらのクラスがオブジェクトを定義するDPPにより規定される場合、外延・内包関係が成立すれば継承関係が生じることから、第1のスーパークラスと第2のクラスとの間に継承関係が存在することになる。

[0090] （条件3）異なる内部構造を持つクラス間の継承関係に基づくポリモーフィズムが、クラス間に定義された変換処理を辿る自動経路探索と自動データ変換を実行することで実現される。

[0091] なお、前述の「外延・内包関係」は、オブジェクト指向プログラミングにおける「is-a関係」に当たる。また、ここで、外延・内包関係が成立すること、については、前述の「1. 3. データ間の論理的な同等性の定義」にて、説明している。

[0092] DPPでは、データは全て、オブジェクトとして扱われる。

[0093] 図3Aは、自動データ変換による媒介データの別の状況を示す図である。図3Aにおけるデータ、即ち、「ソース1」、「ソース2」、「ソース3」、「媒介データ1」、「媒介データ2」、「媒介データ3」、「ターゲット1」、「ターゲット2」、「ターゲット3」は、全てオブジェクト化されている。夫々の名称はクラス名（型名）である。

[0094] 図3Aにおいて、まず、「ソース1」、「ソース2」から、「ターゲット1」、「ターゲット2」への変換の媒介として「媒介データ1」が設定されており、「ソース3」から、「ターゲット3」への変換の媒介として「媒介データ3」が設定されているとする。また、「媒介データ1」と「媒介データ3」との間に、クラス「媒介データ2」が設定されているとする。この場

合、例えば、「ソース 1」「ソース 2」から、「ターゲット 3」への変換が、「媒介データ 1」「媒介データ 2」「媒介データ 3」を介して新たに実現され得ることになる。同様に、「ソース 3」から「ターゲット 1」「ターゲット 2」への変換が、「媒介データ 3」「媒介データ 2」「媒介データ 1」を介して新たに実現され得ることになる。

- [0095] ここで「媒介データ 1」と「媒介データ 2」との関係は等価関係であり、「媒介データ 3」は「媒介データ 2」を拡張したものである。媒介データの変更や拡張を古いコンポーネントを利用可能な状態で実施するためには、「媒介データ 1」と「媒介データ 2」との間に、2通りの変換が考慮されればよく、「媒介データ 2」と「媒介データ 3」との間でも、2通りの変換が考慮されればよい。これは、 $M+N$ 通りの再開発が2通りの変換の開発に効率化されることを意味する。
- [0096] なお、本開示のDPPにおいて、クラスであるデータ間の矢印（例えば、ソース 1 とターゲット 1 との間の、ソース 1 からターゲット 1 への矢印）が、継承関係を示すものであるように、各クラスが定義されている。ソース 1 とターゲット 1 との場合、矢印が示すように、「ソース 1」がサブクラスであり、「ターゲット 1」がスーパークラスである。
- [0097] 更に、DPPでは、自動データ変換のためのインタプリタも構築され得る。図 3 B は、インタプリタの動作例を示す図である。まず、「ソース 1」から、「ターゲット 1」への変換の媒介として「媒介データ 1」が設定されており、「ソース 2」、「ソース 3」から、「ターゲット 2」、「ターゲット 3」への変換の媒介として「媒介データ 2」が設定されており、更に、「ソース 3」から「ターゲット 3」への変換の媒介の設定は無い、とする。更に、「媒介データ 1」と「媒介データ 2」との間に相互の変換が設定されているとする。この場合に、インタプリタは、「処理Generate」で囲まれた部分 G の複雑な処理をユーザから隠蔽する役割を果たす。
- [0098] インタプリタは、その内部で必要に応じて媒介データを生成した上で自動データ変換を行う。例えば、図 3 B に示す場合の「ソース 3」から「ターゲ

ット1」の変換を考える。このとき「ソース3」→「媒介データ2」→「媒介データ1」→「ターゲット1」という経路が想定され得る。インタプリタでは、各クラスの変換関係（即ち、継承関係）を把握しており、実際の「ソース3」から「ターゲット1」への変換時にはそれに基づいて、必要な媒介データの生成を行う。

[0099] インタプリタは、実際の変換前に、複数の変換経路を想定し、それぞれに対して変換のコスト（例えば、矢印が示す変換（継承）が用いられた回数）を、計算するように構成されている。例えば、図3Bに示す場合の「ソース3」から「ターゲット3」の変換を考える。このとき「ソース3」→「媒介データ2」→「ターゲット3」という経路や、「ソース3」→「媒介データ2」→「媒介データ1」→「ターゲット3」という経路も想定されなくは無いが、「ソース3」→「ターゲット3」という経路が最も高速且つ効率的であることは明白である。このように、なるべく効率的であり高速である、変換経路を、インタプリタは自動決定しようとする。

[0100] 3. 1. DPPと、一般的なオブジェクト指向プログラミングとの特性の差異

DPPと、一般的なオブジェクト指向プログラミングとの特性の差異は、上記の表に示した通りである。

[0101] ここで「表現形式が異なること」について、説明する。「テキスト形式」と「バイナリ形式」は、表現形式が異なる一例である。

[0102] 更に、例えば、画像データであるならば、png, jpeg, bmp, tif、動画データであるならば、avi, mpeg, mov, wmv、などのように、無数の表現形式があり、それらが異なれば表現形式が異なる。

[0103] また、テキスト形式と言っても、改行コードの違いによって、

- ・改行コードをLFとする表現形式
- ・改行コードをCR+LFとする表現形式
- ・改行コードをCRとする表現形式

などの違いがある。

- [0104] C++言語のクラスでいえば、次のクラスA, Bは、データの格納順序が異なるので異なる表現形式である。

```
class A { double x; double y; };
```

```
class B { double y; double x; };
```

プログラミング言語もコンピュータへの指示を表現する表現形式の一つであり、C++、C、fortran、pythonなど、プログラミング言語が異なれば表現形式が異なる。

- [0105] 性別をM、Fと書くか、男性、女性と書くか、を決めても、表現形式を定めたこと（即ち、異なること）になる。

- [0106] 3. 2. オブジェクトがノードに関連付けられた木構造

本開示においては、データの解釈と、データの統合が、オブジェクトがノードに関連付けられた木構造（グラフの一例）を構築することで行われる。木構造は、一般に、木構造の構成要素であるノードと、二つのノードの間の親子関係からなり、各ノードが高々一つだけ親子関係の親にあたるノードをもつものである。親子関係の親にあたるノードをもたないノードは特別に根ノードとよばれて木構造にただ一つだけ存在する。本開示における木構造は、各ノードにオブジェクトが一つだけ関連付けられたものであり、また、木構造の異なるノードに同じオブジェクトを関連付けることを可能としている。本開示における木構造の親子関係は、データ処理プラットフォームの述語論理L2で定義された二項関係である。本開示において、ある木構造のノードに関連付けられたオブジェクトにダウンキャストが生じると、ダウンキャスト後の新しいデータ型に対して述語論理L2で定義されている親子関係から、そのノードに対して子ノードが新たに生じ得る。すなわち、木構造の各ノードに関連付けられたオブジェクトのダウンキャストは、木構造を成長させ得る。

- [0107] 4. [実施の形態1]

続いて、実施の形態1に係るデータ解釈方法及びデータ解釈装置を説明する。

[0108] 4. 1. データ解釈装置の構成

[0109] 図1は、本実施の形態に係るデータ解釈装置2の物理的な構成を示す図である。データ解釈装置2は、ハードウェアプロセッサに相当する制御部4と、メモリに相当するRAM(Random Access Memory)6と、メモリに相当するROM(Read Only Memory)8と、通信部12と、入力部14と、出力部16とを有する。これら各構成は、バス10を介して相互にデータ送受信可能に接続される。

[0110] 制御部4は、RAM6又はROM8に記憶されたプログラムの実行に関する制御やデータの演算、加工を行う。制御部4は、様々なプログラム(例えば、データ解釈のためのプログラム)を実行する演算装置である。制御部4は、入力部14や通信部12から種々の入力データを受け取り、入力データの演算結果を出力部16に表示したり、RAM6やROM8に格納したり、通信部12を通じて外部サーバに転送したりする。制御部4は、CPU(Central Processing Unit)などにより構成される。

[0111] RAM6は、データの書き換えが可能な記憶部であり、例えば半導体記憶素子で構成される。RAM6は、制御部4が実行するアプリケーション等のプログラムやデータを記憶する。

[0112] ROM8は、データの読み出しのみが可能な記憶部であり、例えば半導体記憶素子で構成される。ROM8は、例えばファームウェア等のプログラムやデータを記憶する。

[0113] 通信部12は、データ解釈装置2を外部ネットワーク20に接続する通信インタフェースである。

[0114] 入力部14は、ユーザからデータの入力を受け付けるものであり、例えば、キーボードやマウス、タッチパネル、スキャナーで構成される。例えば、入力データとして2D-CAD図面のような非構造化データを読み込む場合、スキャナーを使用して画像データ(ラスタデータ)を取得することができる。

[0115] 出力部16は、制御部4による演算結果を視覚的に表示するものであり、例えば、LCD(Liquid Crystal Display)により構成される。

- [0116] データ解釈のためのプログラムは、RAM 6やROM 8等の、コンピュータによって読み取り可能な記憶媒体に記憶されて提供されてもよいし、通信部 12により接続される外部ネットワーク 20を介して外部サーバ 24から提供されてもよい。CADデータや、CADデータに基づくオブジェクトは、通信部 12により接続される外部ネットワーク 20を介して外部サーバ 24等から提供されるのが好ましい。データ解釈装置 2では、制御部 4がデータ解釈のためのプログラムを実行することにより、獲得部 5aや解釈部 5bなどの様々な機能が実現される。なお、これらの物理的な構成は例示であって、必ずしも独立した構成でなくてもよい。例えば、データ解釈装置 2は、CPUとRAM 6やROM 8が一体化したLSI (Large-Scale Integration) や超LSIを備えていてもよい。
- [0117] 制御部 4にはプラットフォーム（ここでは、データ処理プラットフォーム）5が設けられる。当該プラットフォーム 5は、獲得部 5aと、解釈部 5bとを含む、機能ブロックを、備える。
- [0118] 獲得部 5aは、入力データをオブジェクトとして獲得する。入力データは、構造化データであってもよく、非構造化データであってもよい。
- [0119] 解釈部 5bは、獲得部 5aが獲得したオブジェクトに対して初期木構造を作成する。更に、解釈部 5bは、初期木構造から木構造を自動成長させた結果を入力データの解釈とする。解釈部 5bは、この解釈において、木構造の各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行する。
- [0120] 4. 2. データ解釈装置の動作
- 図 4 は、橋脚構造を示す橋脚構造一般図の例であり、CADデータにより構成されている。CADデータである橋脚構造一般図のデータは、線分や曲線、文字列等を構成要素として含み、構成要素の集まりは、平面図や正面図などの「図」と、各種表などの「表」とに解釈され得る。
- [0121] 4. 2. 1. データを解釈する過程
- 設計図面オブジェクトを入力として、図面の要素を表すオブジェクトがノ

ードに関連付けられた木構造を生成する過程であり、生成された木構造を図面の解釈結果とする。図18に示すように、図面に含まれる線分や文字列、円などの要素をすべて同一の階層に配した木構造を初期木構造とし、木構造を順次成長させる。ただし、図面データ中の要素が最初から構造化されている場合にはその構造を反映した木構造を初期木構造としてもよい。一例として図19に、引出線付きテキストバルーンを認識する過程を、木構造の成長の例として示す。図19では、図面内において円が文字列を包含している場合（上段）に、二つをまとめてテキストバルーンと解釈し（中段）、さらに、バルーンに折れ線が連結している場合にはその折れ線を引き出し線とみなしている（下段）。つまり、円要素と文字列要素の包含関係からテキストバルーンを認識し、テキストバルーンと線分要素の接続から引き出し線を認識する。引出線付きテキストバルーンの一部となった線分要素に「引出線」という役割が割り当てられるように、木構造の親子関係は、図面の要素の使われ方を明示したコンテキストを表している。

[0122] 4. 2. 2. 表である「表題欄」を解釈する動作

図4に示す橋脚構造一般図において、図5に示す「表題欄」を解釈する手順を説明する。「表題欄」は、CAD製図基準（国土交通省）において橋脚構造一般図の右下隅に記載することが規定されている。図4に示す橋脚構造一般図の例においても、右下隅に設けられている。更に、「表題欄」は、CADデータでは線と文字列から構成される。

[0123] まず、CADデータ「橋脚構造一般図」において、線の集まりであるデータが、オブジェクト化される（クラス名は、例えば、「LineBuf2D」とされる）。図6は、「橋脚構造一般図」において線で構成されるオブジェクト（クラス名「LineBuf2D」）を示す図である。オブジェクトとして、「A」から「P」まで、16個のものが示されている。

[0124] 次に、図6に示すように、線の集まりのオブジェクト（クラス名「LineBuf2D」）から、セルの集合と解釈できないものが「表題欄」の候補から除外される。ここで、セルの集合については「CellSet」というクラスが定義されて

おり、よって、線の集まりのオブジェクト（クラス名「LineBuf2D」）から、サブクラス「CellSet」へのダウンキャストが自動実行される。図7は、線の集まりのオブジェクト（クラス名「LineBuf2D」）がサブクラス「CellSet」にダウンキャストされ、「表題欄」の候補が残りの8個のオブジェクトであることを示す図である。「A」、「B」、「C」、「D」、「E」、「F」、「K」、及び「L」のオブジェクトが、「表題欄」の候補である。

[0125] なお、線の集まりのオブジェクト（クラス名「LineBuf2D」）から、セルの集合と解釈できないものを除外するに当たっては、サブクラス「CellSet」に対してクラス「LineBuf2D」からダウンキャストするための関数をあらかじめ定義しておく。ここで、スーパークラスからサブクラスへダウンキャストするためにあらかじめ定義しておく関数を、input関数と称することとする。ダウンキャストが失敗したものが「表題欄」の候補から外れることになる。ここでのサブクラス「CellSet」とクラス「LineBuf2D」の継承関係は、セルの集合「CellSet」と解釈されたならば線の集まり「LineBuf2D」とも解釈されなければならないという外延・内包関係に拠る（前述（条件2）参照）。

[0126] 更に、線の集まりのオブジェクト（クラス名「LineBuf2D」）から、セルの集合と解釈できないものを除外するに当たっては、スーパークラスからサブクラスへのダウンキャストがどの程度確からしいかを示す値（例えば、セルの集合と解釈されたオブジェクトが何%の確率で実際にセルの集合であるかを示す値）、確度をサブクラスへの変換時（input関数実行時）にオブジェクトに与えることができる。確度の計算の際に、スーパークラスに付与されている確度を参照することも可能である。

[0127] 次に、図8Aに示すように、クラス「CellSet」のオブジェクトに対してサブクラス「Table」へのダウンキャストが試行される。この場合のinput関数に対しては、スーパークラス「CellSet」のオブジェクトとともに、CADデータの要素（線や文字列等）が入力され、セルの集合の各セルに含まれる要素が抽出される。クラス「CellSet」のオブジェクトに対して、一定数の文字列等が入力された場合に、ダウンキャスト成功とするようにinput関数を定義す

ることが可能であり、「Table」のクラスにダウンキャストされたオブジェクトは、「Table」と解釈されることになる。

[0128] 次に、図8Bに示すように、クラス「Table」のオブジェクトにおいて、所定の項目、ここでは「図面名」の項目が存在するか調べる。存在すればクラス「表題欄」のオブジェクトにダウンキャストされ、「表題欄」として解釈されることになる。このとき、ダウンキャストは、input関数によらず、クラスにあらかじめ定義されたダウンキャスト可能かをクラスの内部データから判別する関数（is関数と称する）と実際のダウンキャストを実行する関数（cast関数と称する）に基づいて実行される。ここで、is関数とcast関数によるダウンキャストを分析・分類によるダウンキャストと称する。以上のようにして「表題欄」が解釈される。

[0129] ここでは、表である「表題欄」を解釈する動作について、木構造を成長させる観点から簡単に説明する。図21は、CAD製図基準（国土交通省）で定義されている「表題欄」を認識する過程を示している。「表題欄」の認識のため、まず、図21（1）のように、接続された線分の集まりが抽出され、線分の配置状況から「表の枠組み」であるかどうか判定される。図21（2）のように、「表の枠組み」と判定された場合には、それを表現するオブジェクトに表の各セルの情報が記録されている。このセル情報を使って、さらに図面からセル内に文字列が配置されているかを調べ、文字列が配置されている場合には「表」として判定し（図21（3））、セルの情報として文字列を格納する。CAD製図基準に従う図面の表に対して、セルの項目として“工事名”がある等の条件を満たせば、図21（4）のように、「表」は「表題欄」として認識することができる。

[0130] 4. 2. 3. 図である「橋脚正面図」を解釈する動作

続いて、図4に示す橋脚構造一般図において、図である「橋脚正面図」を解釈する手順を説明する。なお、図4では、中央上部に「橋脚正面図」が設けられている。

[0131] まず、CADデータ「橋脚構造一般図」において、線の集まりであるデータが

、オブジェクト化される（クラス名「LineBuf2D」）。図6は、「橋脚構造一般図」において線で構成されるオブジェクトを示す図である。オブジェクトとして、「A」から「P」まで、16個のものが示されている。

[0132] 次に、線の集まりのオブジェクト（クラス名「LineBuf2D」）から、寸法値が付いている図「View」と解釈できないものを除外するに当たっては、サブクラス「View」に対してクラス「LineBuf2D」からダウンキャストするための関数をあらかじめ定義しておく。ダウンキャストが失敗したものが「View」と解釈される候補から外れることになる。ここでのサブクラス「View」とクラス「LineBuf2D」の継承関係は、寸法値がついている図「View」と解釈されたならば線の集まり「LineBuf2D」とも解釈されなければならないという外延・内包関係に拠る（前述（条件2）参照）。図9は、線の集まりのオブジェクト（クラス名「LineBuf2D」）からサブクラス「View」でないものが除外され、残りの8個のセルを示す図である。「G」、「H」、「I」、「J」、「M」、「N」、「O」及び「P」のセルが、残されている。

[0133] なお、線の集まりのオブジェクト（クラス名「LineBuf2D」）から、寸法値が付いている図「View」へのダウンキャストに当たっては、スーパークラスからサブクラスへのダウンキャストがどの程度確からしいかを示す値（例えば、寸法値の付いた図と解釈されたオブジェクトが何%の確率で実際に寸法値の付いた図であるかを示す値）、確度をサブクラスへの変換時にオブジェクトに与えることができる。

[0134] オブジェクトが「橋脚正面図」であるかを調べるため、オブジェクトの内部データの所定の項目、ここではレイヤを参照し、主構造物（D-STR）レイヤに属しているオブジェクトの子要素があるか分析及び分類する。オブジェクトのすべての要素が主構造物（D-STR）レイヤに属していないとすれば、そのオブジェクトは「橋脚正面図」の候補から除外される。図10において、図9に示す8個のクラス名Viewのオブジェクトのうち、主構造物（D-STR）レイヤに属していないものが除外されて、「H」、「I」、「J」、「M」、「N」、「O」及び「P」を付して、7個が示されている。

[0135] input関数を用いたダウンキャスト（情報の入力によるダウンキャストと称する）では、オブジェクトが個々にもつ固有の内部データに加えて追加の情報をを用いてダウンキャスト可能か否かを判断することができる。図11に示すように、クラス「View」のオブジェクトに対して、CADデータから表題となる文字列が入力され、「橋脚正面図」であると解釈される。以上のようにして、図面である「橋脚正面図」が解釈される。

[0136] 4. 2. 4. 図面を解釈する動作のまとめ

本実施の形態に係るデータ解釈装置及びデータ解釈方法では、2D-CAD図面の意味及び内容を段階的に認識する。

[0137] 正面図や平面図などの各種の図に対しては、図13（1）及び図13（2）に示すように、線に着目する。即ち、線の集まりであるデータについてオブジェクト化（クラス名「LineBuf2D」）が行われる。次に、図13（2）、図13（3）及び図13（4）に示すように、寸法値などを図面から入力した上で、クラス名「View」のオブジェクトであると解釈し、更にクラス名「橋脚正面図」のオブジェクトであると解釈する。さらに、寸法値と「View」がもつ内部データの投影面から、橋脚の柱の高さが11mという情報が得られている。

[0138] 各種の表に対しては、図14（1）及び図14（2）に示すように、線に着目する。即ち、線の集まりであるデータについてオブジェクト化（クラス名「LineBuf2D」）が行われる。次に、図14（2）及び図14（3）に示すように、クラス名「CellSet」のオブジェクトであると解釈する。次に、図14（3）及び図14（4）に示すように、文字列などのセルの内容を図面から入力した上で、クラス名「Table」のオブジェクトであると解釈し、更に、クラス名「下部工座標」のオブジェクトであると解釈する。

[0139] 図15に示すように、各種の表において、線の集まりのオブジェクト（クラス名「LineBuf2D」）は、セルの集合のオブジェクト（クラス名「CellSet」）として解釈され、更に、複数のセルに内容が充填されているオブジェクト（クラス名「Table」）や一つのセルのみを含むオブジェクト（クラス名「

Cell」)として解釈される。同じように、図15に示すように、各種の図において、線の集まりのオブジェクト(クラス名「LineBuf2D」)は、寸法値が付いている図であるオブジェクト(クラス名「View」)として解釈され、更に、「橋脚正面図」特有の項目を持つオブジェクト(クラス名「橋脚正面図」)や「橋脚側面図」特有の項目を持つオブジェクト(クラス名「橋脚側面図」)として解釈される。

[0140] 図16に示すように、2D-CAD図面の認識手法として、本実施の形態では、二通り設定されている。(1)「情報の入力」と、(2)「分析・解釈」である。

[0141] 図16(1)に例として示しているように、セルの集合のオブジェクト(クラス名「CellSet」)に対して、入力データからの追加の情報として、図面上の情報が入力されることで、クラス名「Table」のオブジェクトとして解釈され得ることとなる。

[0142] また、図16(2)に例として示しているように、クラス名「Table」のオブジェクトにおいて、項目等を分析・分類してダウンキャストすることで、クラス名「表題欄」のオブジェクト、クラス名「構造高表」のオブジェクト、及び、クラス名「座標Table」のオブジェクト、が解釈され得ることとなる。

[0143] なお、図27は、2D-CAD図面「006_P2橋脚構造一般図.dxf」を解釈するためのスクリプトの例である。このスクリプトは、本開示に係るデータ処理プラットフォームに理解される言語の一例である。当然ながら、本開示のデータ処理プラットフォームは、これまでに述べた要件を満たす限り、本発明者の創作したインタプリタ以外の別のインタプリタであっても全く構わない。スクリプトも、当該別のインタプリタによって理解される言語であって構わない。

[0144] 5. [実施の形態2]

本実施の形態に係るデータ統合装置2'は、複数のデータに分散している断片的な情報を統合してデータの表現対象のモデルを作成する。

[0145] 図22を参照して、実施の形態2に係るデータ統合装置を説明する。

[0146] 5. 1. データ統合装置の構成

[0147] 図22は、本実施の形態に係るデータ統合装置2'の物理的な構成を示す図である。データ統合装置2'は、ハードウェアプロセッサに相当する制御部4と、メモリに相当するRAM(Random Access Memory)6と、メモリに相当するROM(Read Only Memory)8と、通信部12と、入力部14と、出力部16とを有する。これら各構成は、バス10を介して相互にデータ送受信可能に接続される。通信部12は、データ統合装置2'を外部ネットワーク20に接続する通信インタフェースであり、データ統合装置2'は、通信部12により接続される外部ネットワーク20を介して外部サーバ24とも接続する。これらの、RAM6、ROM8、通信部12、入力部14、出力部16、及び、バス10は、図1に示す実施の形態1に係るものと、同様のものである。

[0148] 制御部4にはプラットフォーム（ここでは、データ処理プラットフォーム）5が設けられる。当該プラットフォーム5は、獲得部5aと、解釈部5bと、及び統合部5cとを含む、機能ブロックを備える。

[0149] 獲得部5aは、入力データをオブジェクトとして獲得する。入力データは、構造化データであってもよく、非構造化データであってもよい。

[0150] 解釈部5bは、オブジェクトに対して初期木構造を作成する。更に、解釈部5bは、初期木構造から木構造を自動成長させた結果を入力データの解釈とする。解釈部5bは、この解釈において、木構造の各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行する。

[0151] 統合部5cは、獲得部5aが獲得したオブジェクトの表現対象ごとに、初期木構造を作成する。また、統合部5cは、獲得部5aが獲得したオブジェクトに対して、適宜、解釈部5bの解釈を実行させて表現対象の断片的な情報を読み取る。更に、統合部5cは、初期木構造に対して、1又は複数の断片的な情報を追加の情報として与えて、初期木構造から木構造を自動成長させた結果を統合データとして構築する。統合データは、表現対象のモデルである。統合部5cは、統合データの構築において、木構造の各ノードに関連付

けられたオブジェクトの外延側または内包側への型変換を適宜実行する。

[0152] 以上の構成により、本実施の形態に係るデータ統合装置 2' は、材料となる複数のデータから抽出した複数の断片的な情報を統合して所望のデータを構築する。

[0153] また、本実施の形態に係るデータ統合装置 2' は、表現形式が異なるオブジェクトのデータ構造を自動でデータ変換するものであり、型変換の経路を自動探索し、更に統合データを自動構成する。

[0154] 5. 2. データ統合装置の動作

本開示に係るデジタル橋梁自動構築手法の流れは図 17 に示される。入力データ群は個別に解釈され、その解釈結果から複数の情報が抽出される。情報がその表現対象ごとにまとめられ統合されることで、デジタル橋梁が構築される。デジタル橋梁自動構築の過程は、次の三つからなる。

(1) データを解釈する過程

(2) 情報抽出過程

(3) 同定・統合過程

なお、実施の形態 1 は (1) (2) を含む実施例であり、実施の形態 2 では、(1) ~ (3) までを含む。

[0155] 5. 2. 1. (1) データを解釈する過程

データを解釈する過程は、「4. 2. 0. データを解釈する過程」で説明している。

[0156] 5. 2. 2. (2) 情報抽出過程

図面の解釈結果から図面で表現している現実の対象を洗い出し、その対象に関する表現の集合として情報をまとめる。ここでは、この過程を簡略化し、対象を特定するためのオブジェクトとして、「P1橋脚」「A2橋台」など、その対象の名称を採用する。

[0157] 5. 2. 3. (3) 同定・統合過程

情報抽出過程で得られた複数の情報の統合は、複数の情報から一つのグラフを作成することで行う。このグラフは、情報から洗い出された表現対象（

例えば、「P1橋脚」、「A2橋台」）をグラフの一部として含み、各表現対象に関する情報もグラフの一部として集約され統合されている。この統合の過程で図面の解釈時と同様に木構造の成長による自動構造化を実施する。自動構造化の対象となる木構造はグラフの一部であり、木構造の親子関係はノード間の全体と部分の関係（例えば、橋脚とその基礎の関係）とすることができる。初期木構造は、表現対象を抽象的に表現した単一のノード（例えば、橋脚であることだけがわかっている場合のオブジェクトがノードに関連付けられたもの）であってもよい。木構造の成長は、オブジェクト間の外延・内包関係に従い、より詳細な外延側のオブジェクトを内包側のオブジェクトから作成することにより行う。これは、オブジェクト指向プログラミングにおけるダウンキャストに相当し、オブジェクトの内部データしか考慮しない通常のダウンキャストでは、情報が足りないために適切に実行できない。しかし、この過程においては、図面から抽出した情報を反映したグラフの一部を参照することにより適切に実行可能な場合がある。この場合に作成されたより詳細なオブジェクトは、グラフとして表現された情報を統合したものとみなすことができる。また、作成されたオブジェクトに応じて木構造（と、それを含むグラフ）が成長するため、その成長ごとにその時点でのグラフに応じたダウンキャストが試行される。なお、3次元モデルのオブジェクトを作成するためには、図面から抽出した情報だけでは不十分な場合がある。この場合には、例えば、工学知や何等かの推定技術によりグラフを修正すればよい。

[0158] 情報抽出過程では、情報を「名前が:"P1":である橋脚の高さが:Q("10m"):である;」のような自然言語風の文の集まりとして得る。複数の文の内容を統合するためには、各文をそれぞれグラフで表現し、各グラフのノードを同定して1つのグラフに統合する。ただし、文の中には、グラフで表現せず、統合されたグラフを修正するルールを記したものがあってもよく、このルールを適用してグラフを適宜修正してもよい。

[0159] 5. 2. 4. 2次元CAD図面への適用

本開示に係るデジタル橋梁自動構築手法の検討のため、線分や文字列等の図面の要素がベクター形式で記録されている2次元CAD（DXF形式）の図面を対象として自動解釈と表現対象の自動構築を試行した。図20は、試行により橋脚の3次元モデルが自動構築された例を示している。図面の解釈結果として、「橋脚正面図」、「橋脚平面図」、及び「橋脚側面図」が解釈される。それら「橋脚正面図」、「橋脚平面図」、及び「橋脚側面図」から、構造物の投影面が自動作成されて、更に構造物の投影面から3次元モデルが自動作成される。図12Aは、正面図、平面図、及び側面図から、構造物の投影面が自動作成される様子を示す図であり、図12Bは、構造物の投影面から3次元モデルが自動作成される様子を示す図である。

[0160] 6. 他の実施の形態

以上のように、本出願において開示する技術の例示として、実施の形態1及び2を説明した。しかしながら、本開示における技術は、これに限定されず、適宜、変更、置き換え、付加、省略などを行った実施の形態にも適用可能である。

[0161] また、実施の形態を説明するために、添付図面および詳細な説明を提供した。したがって、添付図面および詳細な説明に記載された構成要素の中には、課題解決のために必須な構成要素だけでなく、上記技術を例示するために、課題解決のためには必須でない構成要素も含まれ得る。そのため、それらの必須ではない構成要素が添付図面や詳細な説明に記載されていることをもって、直ちに、それらの必須ではない構成要素が必須であるとの認定をするべきではない。

[0162] また、上述の実施の形態は、本開示における技術を例示するためのものであるから、特許請求の範囲またはその均等の範囲において種々の変更、置き換え、付加、省略などを行うことができる。

[0163] 本願は、日本国に2019年7月29日に出願された基礎出願2019-139150号に基づくものであり、その全内容はここに参照をもって援用される。

符号の説明

[0164] ２・・・データ解釈装置、２'・・・データ統合装置、４・・・制御部、５・・・プラットフォーム、５a・・・獲得部、５b・・・解釈部、５c・・・統合部、６・・・ＲＡＭ、８・・・ＲＯＭ、１０・・・バス、１２・・・通信部、１４・・・入力部、１６・・・出力部、２０・・・外部ネットワーク、２４・・・外部サーバ。

請求の範囲

- [請求項1] オブジェクトの型変換を自動実行するプラットフォームを備えたデータ解釈装置であって、
- 前記プラットフォームは、前記データ解釈装置の制御部に設けられ、獲得部及び解釈部を有しており、
- 前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得し；
- 前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行う；
- ように、制御部を動作させる、データ解釈装置。
- [請求項2] 前記解釈部は、
- 前記オブジェクト間の外延・内包関係に基づいて前記オブジェクトの外延側への型変換であるダウンキャストを実行し、前記ダウンキャスト後のオブジェクトに関連付けられるノードを前記グラフに追加させることで、前記グラフを自動成長させる、請求項1に記載のデータ解釈装置。
- [請求項3] オブジェクトの型変換を自動実行するプラットフォームを用いたデータ解釈方法であって、
- 前記プラットフォームは、コンピュータの制御部に設けられ、獲得部及び解釈部を有しており、
- 前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得するステップ；
- 前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行うステップ；

を実行する、データ解釈方法。

[請求項4]

コンピュータが実行するプログラムであって、
当該コンピュータの制御部に備えられ、獲得部及び解釈部を有し、
オブジェクトの型変換を自動実行するプラットフォームにおいて、
前記獲得部が、入力データを前記プラットフォームのオブジェクト
として獲得するステップ；
前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グ
ラフの各ノードに関連付けられたオブジェクトの外延側または内包側
への型変換を適宜実行しながら、前記初期グラフからグラフを自動成
長させることで解釈を行うステップ；
を行うように、制御部を動作させる、プログラム。

[請求項5]

オブジェクトの型変換を自動実行するプラットフォームを備えたデ
ータ統合装置であって、
当該プラットフォームは制御部に設けられ、獲得部及び解釈部を有
しており、
前記獲得部が、入力データを前記プラットフォームのオブジェクト
として獲得し；
前記解釈部が、前記オブジェクトに対して初期グラフを作成し、グ
ラフの各ノードに関連付けられたオブジェクトの外延側または内包側
への型変換を適宜実行しながら、前記初期グラフからグラフを自動成
長させることで解釈を行い、1以上の前記オブジェクトを統合した統
合データを構築する；
ように、制御部を動作させる、データ統合装置。

[請求項6]

オブジェクトの型変換を自動実行するプラットフォームを用いたデ
ータ統合方法であって、
前記プラットフォームはコンピュータの制御部に設けられ、獲得部
及び解釈部を有しており、
前記獲得部が、入力データを前記プラットフォームのオブジェクト

として獲得するステップ；

前記解釈部が、オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行い、1以上の前記オブジェクトを統合した統合データを構築するステップ；

を実行する、データ統合方法。

[請求項7]

コンピュータが実行するプログラムであって、

当該コンピュータの制御部に備えられ、獲得部及び解釈部を有し、オブジェクトの型変換を自動実行するプラットフォームにおいて、

前記獲得部が、入力データを前記プラットフォームのオブジェクトとして獲得するステップ；

前記解釈部が、オブジェクトに対して初期グラフを作成し、グラフの各ノードに関連付けられたオブジェクトの外延側または内包側への型変換を適宜実行しながら、前記初期グラフからグラフを自動成長させることで解釈を行い、1以上の前記オブジェクトを統合した統合データを構築するステップ；

を行うように、コンピュータの制御部を動作させる、プログラム。

[請求項8]

データの表現形式を自動変換するプラットフォームであって、所定の要素プログラムを有し、任意の表現形式の入力及び出力をもつ複数のプログラムが、表現形式の自動変換によってデータを介して疎結合されており、統合データの表現対象が都市又は構造物である、デジタル都市構築プラットフォーム。