階層構造グラフを用いた半構造化データの構造化手法

上島紳一[†] 森下淳也[††]
大月一弘[††] 杉山武司[†††]

本稿では，半構造化データを段階的に構造化するための枠組みとして階層構造グラフモデルを導入する．ここでは，半構造化データの中でも特に試行錯誤的な構造化作業を必要とするデータを対象に，データの集合化と属性の付与の2つの構造化作業について議論する．このモデルは利用者の視点を陽に扱い，視点をデータとして取り扱うという特徴がある．階層構造グラフは，有向グラフを基本としており，グラフ構造を表すオブジェクトとその要素を表すオブジェクトのオブジェクトの2段構造で構成することで，複数の視点から多様な構造を持つデータを柔軟に構造化することができる．また，視点に応じた半構造化データの多重な表現を与える目的で，仮想オブジェクトの概念を導入し，利用者の仮説や直感に基づく上記のデータの構造化作業を支援する．仮想オブジェクトは実行時に生成でき，利用者はデータとして取り扱うことができる．最後に本モデルのプロトタイプシステムを示す．

Incremental Data Organization of Semi-structured Data Using Hierarchical Graph Model

SHINICHI UESHIMA,[†] JUN-YA MORISHITA,[††] KAZUHIRO OHTSUKI[††]
and TAKESHI SUGIYAMA[†††]

In this paper, we propose the hierarchical graph model that enables flexible structuring of semi-structured data incrementally. Here we focus on the ways to collect data and add attributes to them in a trial and error manner. In this model, hierarchical grpah is defined as a directed graph. We treat "viewpoints" as data explicitly, and treat both semi-structured data and viewpoints as nodes on the graph. The graph consists of component objects and a graph object. The former possesses the attributes of data and "viewpoints." The latter holds their relations. By this model, users can give multiple representations of semi-structured data based on user's viewpoints. We introduce virtual objects in order to support the structuring according to user's ad-hoc intuitions or hypotheses. Users can generate virtual objects at run-time, and treat them as data in our model. We also show a prototype system.

1. はじめに

最近，科学技術データ，Webデータ，構造化文書データなどの半構造化データの扱いが注目されている．半構造化データは厳密にはスキーマに制約されていないデータである．このようなデータに表現を与え，利用することが重要な課題の1つである[1)~4)]．これまで半構造化データの表現法の1つとしてラベル付きグラフモデルがいくつか提案されている．たとえば，異種構造を持つ情報資源の情報を統合し，相互に交換するための共通の書式としてOEMがあり[5)]，ビデオデータ，生物学データなどの自己記述的なデータをエッジにラベルを持つ木構造で表現するラベル付きエッジグラフモデルなどがある[6),7)]．また，各々に対して任意長のエッジの列を検索できる言語としてLOREL，UnQLが提案されている[6),8)]．また，Rufusは，部分的に書式が既知であるデータ群をあらかじめ定義したクラス階層に分類する機構を提供している[10)]．これらのモデルは半構造化データに柔軟に構造を与えることができ，ラベル集合が持つ構造に対して問い合わせることができるという特徴を持つ．

一方，半構造化データの中でも構造化のための書式やスタイルを容易に見つけ出すことのできないデータがある．たとえば，様々な情報資源から得られたマルチメディアデータ[3)]，研究作業のための基礎資料となるデータ[7),11)]，計測機器で観測されたデータ[12)]など

† 関西大学総合情報学部
Faculty of Informatics, Kansai University
†† 神戸大学国際文化学部
Faculty of Cross-cultural Studies, Kobe University
††† 姫路獨協大学情報科学センター
Information Science Center, Himeji Dokkyo University

の例が考えられる．

このようなデータを対象とする場合は，利用者の意図に基づいた直感や経験的な知見などの情報を与えながら，試行錯誤的にデータを集約することが望ましい．つまりデータの集約化とデータへのアドホックな属性の付与が重要な手段であると考えられる．たとえば，WWW（World Wide Web）で用いられているリンクは，この方法で構造化されている．しかし利用者がデータを利用する手段は原則としてリンク航行操作に限られており，データを機能的に活用しにくい[9]．また，フィールドワークで収集されたデータや考古学データなどは，研究者がそれらをもとに研究作業を行う1次データであり，これらのデータを2次利用し，目的に応じた形に加工する必要がある，このような場合も試行錯誤的な集約作業が不可欠である[11]．言い換えれば，データを多様な角度から表現でき，かつ効率的に集約できる枠組みを提供することが重要であると考えられる．前述のモデルはこのような角度からは議論されていない．

ここでは，この種の半構造化データを対象として構造化作業をデータの集合化作業，属性の付与作業に限定して議論する．また，作業の多様性を表現するため“視点”の概念を導入する．ここで，視点の役割は，利用者の興味に応じたデータの範囲を規定し，データへのアドホックな属性の付与と集合化作業における操作単位となるものと仮定する．視点を軸にした上記のデータの取扱いとして以下を考える．

- 視点の実体化，
- 実行時の視点に応じたデータへの属性の付与，
- 実行時の視点に応じたデータの集合化．

本稿では，視点に基づいて多様な構造を持つデータを柔軟に構造化することができるモデルとして階層構造グラフモデルを提案し，上記の3つの操作が簡便に行えることを示す．本モデルはノードとエッジにラベル付けした有向グラフを基本としており，複数の視点から半構造化データを構造化することができる．グラフの各ノードとエッジにはオブジェクトを配置しており，複雑な構造を持つデータや異種構造を持つデータ群をオブジェクトとして収容でき，グラフを用いて構造化できる．また，構造化作業がグラフを単位として行われるため作業結果の再利用性が高い．

さらに，本モデルでは，構造化作業において仮想オブジェクトの概念を導入して視点に応じたデータの多重表現を与えることができる．

本手法により，利用者の視点に基づく半構造化データの構造化，異種データベースのオブジェクトの構造化，未整理なデータの試行錯誤的な整理や分類，既存のデータベースを利用した個人データベースの作成，などを行うことが可能となり，既存のデータベースの利用法が広がるものと考えられる．

2. 基本概念

2.1 対象データと視点

本稿では，データベースの枠組みや利用の目的を規定せずに作成した半構造化データを対象とする．このようなデータとしては，様々な情報資源から得られたマルチメディアデータ，研究作業のための基礎資料となるデータ，計測機器で観測されたデータなどの例が考えられる．これらのデータは，データの型や属性構造に統一性があるとは限らず，利用者は構造化のための書式やスタイルを見つけることが困難である[11]．

ここでは，構造化作業をデータの集合化作業とデータへの属性の付与作業に限定し，議論する．上記のデータに対する利用者による構造化のアプローチは以下のようなものと仮定する．

- 曖昧な動機，直感，仮定などに基づく試行錯誤的な構造化を行う．
- 多様な角度からデータを扱うため，複数の目的からの構造化を行う．データに対する発散的な思考を行う場合に行われる．

これらを考慮すると，システムは各々のアプローチによる構造化を区別できる必要がある．これらの多様性を実現するため“視点”の概念を導入し，“視点”を実体化する．

2.2 視点による集合化と多重表現

視点には次の2つの役割を与えている．

- **視点によるデータの集合化**

  同一の視点から複数の半構造化データを集合化する．つまり，利用者が視点を1つ指定することで，利用者の興味の対象とするデータの範囲を限定する役割である．この場合，スキーマなどに基づいて集合化されるとは限らず，互いにスキーマの異なるデータを集合化する必要がある．データの集合化はデータの分類や，様々なスキーマを持つデータをまとめる手段となる．

  さらに，1つの視点による集合をさらに集合化する．複数の視点による集合をさらに集合化したり，視点をより上位の視点に集合化したりすることで視点間の関係が階層になる．

  図1で，視点 V_1，V_2 はそれぞれデータ D_1，D_2 と D_2，D_3 を集合化している．V_1，V_2 は，上位の視点 V_4 に包含され，D_1，D_2，D_3 が V_4 に間

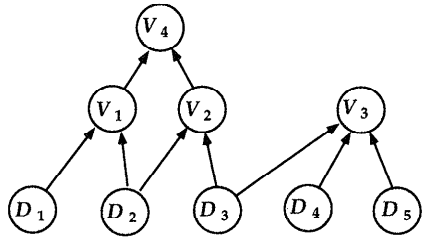


図1 視点による集合化（D_i:半構造化データ，V_i:視点）
Fig. 1 Collecting semi-structured data by viewpoints.

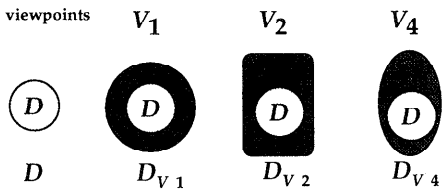


図2 視点に依存したデータの多重表現
Fig. 2 Multiple representations of semi-structured data by viewpoints.

接的に集合化されている．

- **視点によるデータの多重表現**
  半構造化データの多様な構造化を実現するため，視点に基づいたデータの多重な表現を生成できることが必要である．特に，半構造化データに対しては視点に基づいた属性を実行時に付与でき，付与した属性を用いた表現が適切に抽出できることが必要である．**図2**で，D を原データとするとき，2つの異なる視点 V_1，V_2 からの D の表現が D_{V_1}，D_{V_2} のように異なることである．さらに V_4 からも V_1，V_2 を経由して異なる D が見える．これを D の**多重表現**という．
  属性の付与は，データへの属性の追加や補完，また，属性の書換えなどの多様な目的が考えられる．

これらの視点による集合化作業とデータの多重表現は段階的に繰り返し行われる．

2.3 階層構造グラフ

ここでは集合化における視点とデータの多重表現における視点を同一のものとして扱い，同一な構造のもとで取り扱う．これによりモデルが有効性を保持しながら単純化される．

定義1 階層構造グラフは，ノードとエッジからなる非巡回的な有向グラフ (N, E) である．葉ノードは半構造化データを表し，その他のノード（カテゴリと呼ぶ）は利用者の視点を表す．ノードとエッジはそれぞれ性質を表すラベルを持ち，ノードのラベルはそれ自身の性質を表し，エッジのラベルは2つのノード間の関係を表す．有向エッジは2つのノードの上下関係を表し，方向は下位に位置づけられたノードから上位のスーパーノードへ向かうものとする． □

階層構造グラフを用いて，視点によるデータの表現は両者の間の経路に与えられた属性を集めることで得られる．カテゴリとデータの連結は，その視点からのデータの集合化をも表す．一般に，階層構造グラフは独立な連結有向グラフの集合である．連結有向グラフは特別な場合として，単一ノードでエッジを持たないグラフをも含む．

例1 **図3**に電子メールに対する階層構造グラフの例を示す．図では，メーリングリストを用いた授業に対するメールを構造化している．ここでは，大きな視点（大目的）として

- 討議項目ごとのメールの整理
- 成績処理
- 授業分析

を想定し，階層化を行っている．

図において，メール78の発した「パケット通信」に関する授業のまとめに対して，メール79以下で2種類の項目について質疑応答が行われている．各メールは，利用者（=グラフ作成者）の興味に応じてカテゴリに集合化されている．各エッジには，元のデータ（メール）には書かれていなかった属性がそれぞれの視点に応じて付与されている（**属性の付与**）．

また，図の右端上の部分は，講義に対してどのような質問が発せられたかを調べるために，カテゴリ「質問」を作成し，質問を発したメールを集めようとしていることを示す（**段階的集合化**）．

利用者が注目している視点からデータを見る場合，元データの属性だけでなく，視点とデータとの間に付与された属性をも含めて，データに対する属性と見なして取り扱う．たとえば，メール78は，「パケット通信」という視点から見ると「授業のまとめ」を書いたメールであり，「討議項目」という視点からは，「11/5に講義されたパケット通信に関するまとめ」であることが示されている．一方，同メールは「成績処理」に関しては，「佐藤の書いたメールで評価は5点」であることが示されているが，「成績処理」の視点からこのメールを読む場合は，「講義日」や「内容」には興味がないのでそのような属性は見えないことを表している（**視点によるデータの多重表現**）．

本グラフにおいては，1つの視点からデータに至る経路が複数存在する場合がある．利用者は，視点というものをキーにして元のデータに対する属性を読みとろ

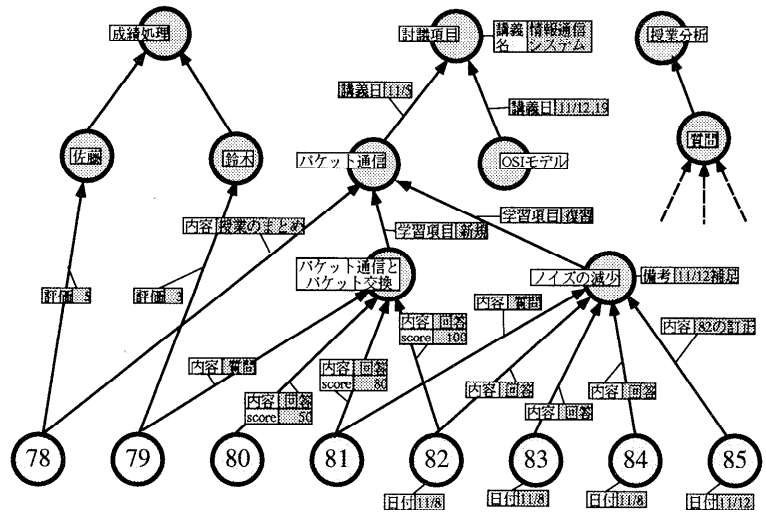


図3 メールの構造化作業における階層構造グラフの適用例
Fig. 3 A hierarchical graph to organize email data from multiple viewpoints incrementally.

うとするので，1つの視点を通したデータの表現は一意である必要がある．つまり，視点によるデータの表現は，視点とデータ間に存在する経路ごとの個別の表現ではないものとする（**視点からのデータ表現の一意性**）．たとえば，視点「パケット通信」からみたメール82は，「パケット通信とパケット交換に対する回答，ならびに，ノイズの減少に対する質問」がなされているメールであると考える． □

3. 階層構造グラフモデルの基本要素

本節では，オブジェクト指向の表現に基づいて階層構造グラフの定式化を行う．

階層構造グラフは，システムの利用者により段階的に構築される．すなわち，グラフの構造はあらかじめ準備されておらず，利用者の手によって構造が生成されていく．

一般にデータの構造はスキーマを用いて定義されるが，構造が試行錯誤的に更新される場合には，システムの再コンパイルが頻繁となるため，この方法は効率的でない．本システムでは，インスタンスベースモデルを用い，グラフでスキーマの役割を代行させる．

階層構造グラフは1つのグラフオブジェクトとそれから参照される要素オブジェクト群から構成する．グラフオブジェクトと要素オブジェクトはそれぞれ独立なオブジェクトとして格納される．グラフオブジェクトの全体を Γ で表し，要素オブジェクトの全体を Ω で表す．このように階層構造グラフはこの2種類のオブジェクト集合のレイヤー構造で構成される（図4）．

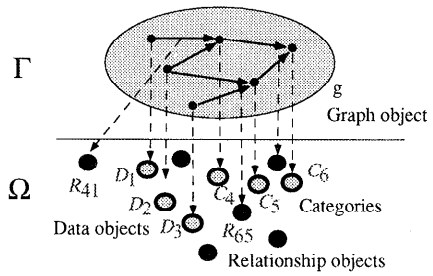


図4 階層構造グラフモデルのレイヤー構造
Fig. 4 Two-layered structure for hierarchical graphs.

3.1 要素オブジェクト

要素オブジェクトはグラフ上の個々のノードやエッジに1対1に対応しており，それぞれのラベル（コンテンツ情報）を格納する．グラフオブジェクトはノード間の接続関係を表す．すなわち，コンテンツ情報と構造情報をそれぞれ要素オブジェクトとグラフオブジェクトに分離して持つ．グラフの構造化作業は，要素オブジェクトに対するコンテンツ属性の追加・削除，ならびにグラフオブジェクトの更新により行われる．

要素オブジェクトは階層構造グラフ上のノードやエッジの位置によって次の3つの種類に分類できる．

- **データオブジェクト（*D*）**
  データオブジェクトは，葉ノードに対応するオブジェクトである．構造化の対象となる半構造化データはデータオブジェクトに格納される．
- **カテゴリ（*C*）**
  下位のノードからのエッジを持つノードに対応す

るオブジェクトをカテゴリと呼ぶ．視点はカテゴリとして格納される．

- **関係オブジェクト（R）**
  エッジに対応するオブジェクトを関係オブジェクトと呼ぶ．関係オブジェクトは，上位のノードに置かれた視点を表すカテゴリと下位のノードに置かれたオブジェクトの間に固有の属性集合を保持する．

要素オブジェクトは次のように定義される．

$$O \equiv \langle \mathtt{oid}, \{a_1 : v_1, \ldots, a_n : v_n\}\rangle. \qquad (1)$$

ここで oid はオブジェクト識別子を表し，a_i は属性，v_i はその値を表す．a_i と v_i の定義は以下のように ⟨attribute⟩ と ⟨values⟩ で与えられる．

$$\begin{aligned} \langle \mathtt{attribute}\rangle &::= \mathtt{symbol},\\ \langle \mathtt{values}\rangle &::= \langle \mathtt{value}\rangle\\ &\quad\ |\ \langle \mathtt{value}\rangle, \langle \mathtt{values}\rangle,\\ \langle \mathtt{value}\rangle &::= \mathtt{string}|\mathtt{int}|\ldots|\mathtt{oid}. \end{aligned} \qquad (2)$$

式 (2) で定義されているように，要素オブジェクト O の属性 a_i は多値を許すため，一般には値の集合である．値は ⟨values⟩ で与えられているように，オブジェクトの参照（oid）を含む通常のデータ型を扱える．要素オブジェクトの保持するデータは自由形式である．すなわち，(1) 本モデルでは属性名と属性値の双方をデータとして扱い，(2) 要素オブジェクトの内部には，データを属性集合として，多値の場合も含めてどのような属性の組合せでも格納できる．したがって，様々な異なる形式のオブジェクトが与えられた場合でも，本モデルではデータオブジェクトに収納できる．

3.2 グラフオブジェクト

グラフオブジェクトは要素オブジェクト間の接続を行うグラフである．今，データオブジェクト，$D_1, \ldots, D_m$ とカテゴリ，$C_{m+1}, \ldots, C_n$ があり，図 4 のようにこれらのオブジェクトが接続されている場合，グラフオブジェクト g はこれらの参照される要素オブジェクトを用いて，以下のように表現される．

$$\begin{aligned} g \equiv \langle &\mathtt{oid},\\ &\{D_1, \ldots, D_m, C_{m+1}, \ldots, C_n\},\\ &\{\ldots, R_{ij}, \ldots\}\rangle. \end{aligned} \qquad (3)$$

ここで添え字はノードの番号として付けられている．R_{ij} は i 番目と j 番目のノードの間に張られたエッジから参照される関係オブジェクトを表している☆．本モデルを半構造化データの構造化作業に適用する際は，まず半構造化データをデータオブジェクトとして Ω に格納し，グラフオブジェクトはエッジを持たないノードの集合とする．これが構造化作業の初期状態となる．このグラフオブジェクトに対して，カテゴリの生成や属性付与などの操作を施すことで，グラフが成長する．このように本モデルでは階層構造グラフは単一の永続グラフオブジェクトとして収納されている．以下では，階層構造グラフは，Γ と Ω のそれぞれのオブジェクトを更新することで行われるが，記述の簡潔さのために明示する必要のない場合は省略する．

3.3 特　　徴

本モデルでは，Γ と Ω を分離して持つことにより，同一のデータオブジェクトの集合に対して Γ 上にまったく異なるグラフオブジェクトを定義し，異なる構造を表すことが容易である．つまりデータオブジェクトを共有して複数の互いに異なる構造を生成できるという特徴がある．さらに，g に対する部分グラフを別のグラフオブジェクトとして定義することにより，階層構造グラフの部分構造を自在に取り出すことができる．また，カテゴリや関係オブジェクトをすべて同一の Ω に保持することができるため，構造化作業においてそれらを共通利用することもでき，Ω のオブジェクトの検索も効率的に行える．

本モデルと同じインスタンスベースモデルとして Obase モデルが提案されている[17]．Obase モデルではインスタンス間の上位/下位の関係をインスタンス自身が持ち，レイヤー属性により関係の属性を表すことができる．Obase モデルを用いて階層構造グラフを表すことができるが，同モデルでは，構造情報とコンテンツ情報を分離せずに，両者を同じオブジェクトに持たせているため，インスタンス間の複数の構造を表現しにくい．

4. 階層構造グラフによる多重表現

階層構造グラフ上では，視点は多段に階層化されているため，1 つの視点からデータオブジェクトに至る経路が複数存在する場合がある．2.3 節で述べたように，視点に依存したデータオブジェクトの表現は，指定された視点とデータオブジェクトに対して一意に定められる．すなわち，1 つのデータオブジェクトに対して，グラフ上で到達可能なカテゴリそれぞれに対応した表現が与えられる．視点に依存したオブジェクトは，指定された視点からデータオブジェクトに至る経路上に付与されたすべての属性をもとに表現される．任意のデータオブジェクトに対して視点を切り替える

☆ 見やすさのため，式 (3) ではオブジェクトの識別子に対する添字は省略している．

ことにより，グラフ上に配置された属性が選択的に継承され，伝播される属性が変化する．各視点に対応したデータオブジェクトの表現は，

(1) グラフオブジェクト g から表現に必要な経路（部分グラフ）の抽出

(2) 抽出されたサブグラフ上に付与された属性から必要な属性の導出

により実現される．

4.1 部分グラフの抽出

ここでは，抽出するサブグラフを仮想オブジェクトと呼び，以下のように定義する．

定義 2 グラフオブジェクト g 上のオブジェクト O が上位のカテゴリ C に連結されているとき，C から O に至るすべての経路からなる g の部分有向グラフ g' を視点 C に依存した O の**仮想オブジェクト**といい，$\tilde{O}[C]$ と表記する．この場合，O はカテゴリでもよい．

定義2では g' は Γ 上で g とは独立に生成される．O に対して，異なるカテゴリを指定することで複数の仮想オブジェクトを定義することができる．

例 2 図 **5** で $\tilde{D}_2[C_6]$，$\tilde{D}_2[C_4]$ を表すグラフオブジェクト g'，g'' は次のようになる．

$$\begin{aligned} g' &\equiv \langle \mathtt{oid}, \{C_4, C_5, C_6, D_2\}, \\ &\quad \{R_{64}, R_{65}, R_{42}, R_{52}\}\rangle. \\ g'' &\equiv \langle \mathtt{oid}, \{C_4, D_2\}, \{R_{42}\}\rangle. \end{aligned} \tag{4}$$

また，$\tilde{D}_1[C_6]$，$\tilde{C}_4[C_6]$ なども同様に定義できる．□

仮想オブジェクトを得るにはグラフ上の2点を結ぶ経路を抽出しなければならない．このような場合，通常，木探索アルゴリズムにより実行時に経路をたどる方法が用いられるが，この方法で g 上の経路を抽出する計算はNP完全となる．このため木探索アルゴリズムを用いる前に，まず仮想オブジェクトを構成するノード集合を部分グラフとして抽出し，次に部分グラフを対象として経路を抽出する方法を考える．この手法は，階層構造グラフの構造情報 g とは別に部分グラフを Γ 上に定義することで可能となっている．

ここでは，有向グラフにおける到達可能なノードを調べるアルゴリズムを用いる．この方法では，グラフのノード集合と隣接関係を用いることで，仮想オブジェクトの経路上のすべてのノードを抽出することができる．

g から仮想オブジェクト $\tilde{D}[C]$ を構成する経路を導出する場合を考える．g のノード集合を Σ とし，$\tilde{D}[C]$ を構成するノード集合を $\Sigma(C,D)$ とすると，

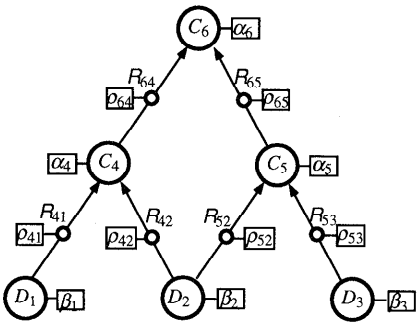


図 5 階層構造グラフ
Fig. 5 Hierarchical graph.

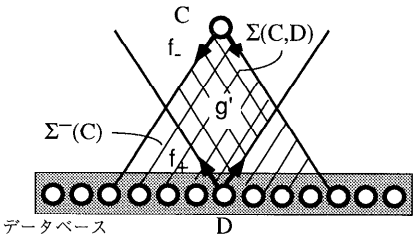


図 6 上と下からの探査を用いた g' の導出
Fig. 6 Generation of g' from g by f_- and f_+ algorithms.

$$\begin{aligned} \Sigma^-(C) &= f_-(\Sigma, C), \\ \Sigma(C,D) &= f_+(\Sigma^-(C), D). \end{aligned} \tag{5}$$

により得ることができる（図 **6**）．

ここに f_- と f_+ は，それぞれグラフ g 上で下向きと上向きに到達可能なノード集合を取り出す手続きであり，両方とも向きの違いを除いて同様に実現することができる．

[f_- アルゴリズム]

ノード集合 $V_0 = \{C\}$，$V' = \{C\}$ とする．

(1) V' から出るエッジに下向きに接続されたノードで，V_0 に含まれないすべてのノードを探す．なければ終了．

(2) (1) で得られたノードを V' とする．

(3) $V_0 \leftarrow V_0 + V'$

(4) (1) へ戻る． □

f_+ はエッジの矢印の向きを逆に考えることで同様に行える．

このアルゴリズム f はシステムで経路を記憶することなく，多項式時間で計算可能である．また，導出された仮想オブジェクトのノード集合から経路を探索し，仮想オブジェクトを構成することは妥当な計算時間で行える．次に，階層構造グラフオブジェクト g を構成するノード集合の中から上記のノード集合 $\Sigma^-(C)$，

D に対応するノードとエッジを抽出し，仮想オブジェクト $\tilde{D}[C]$ とすればよい．

4.2 仮想オブジェクトの属性

仮想オブジェクト $\tilde{D}[C]$ の属性，すなわち C から見た D の表現を導出する手順は以下のとおりである．

(1) 仮想オブジェクト $\tilde{D}[C]$（部分グラフ）に対して，C から D に至る経路の部分グラフをすべて導出する．

(2) 各経路に対して経路上のノードとエッジに付与された属性を伝播する，ただし端点を除く．

(3) 経路ごとに得られた属性ならびにデータオブジェクト D の属性の集合和をとる．

各経路に対する属性の伝播では，基本的には経路に付与された属性の集合和をとる．このため，視点に依存したデータオブジェクトの属性表現は，基本的には仮想オブジェクトに含まれるすべてのノードとエッジに付与された属性の集合和となる．

このような集合和をとる場合，グラフ上のノードやエッジで同一の属性が重複されて定義される可能性が生ずる．同一経路上の多重定義に対しては，グラフ上の位置関係にルールを持たせることなどよりその取扱い方法の指定が可能となる[21]．しかしながら別経路上の多重定義に対しては，重複属性間の関係をグラフ構造から判断するのは困難となる．現在のプロトタイプシステムにおいては，経路の違いを表示して属性を別記している．

5. 集合化と検索

本章では，階層構造グラフの段階的な集合化と検索について述べる．

5.1 集合化における基本操作

利用者は仮想オブジェクトを用いて，データオブジェクトを集合化する．その結果，新しい仮想オブジェクトが生成され，新しいオブジェクトの表現が生成される．これを繰り返すことで階層構造グラフが段階的に更新される．集合化は次の操作を組み合わせて行われる．

- create category（新規カテゴリの生成）
- add attributes to objects（属性の付与）
- create relationship objects（関係オブジェクトの生成）
- register object (s) to category（カテゴリへのオブジェクトの登録）
- select objects（登録候補オブジェクトの選択）

上記の第5の操作で，カテゴリに登録すべき候補オブジェクトを選択する際は，データオブジェクトや仮想オブジェクトを用いてマニュアル操作で選択する場合と検索を用いて効率的にオブジェクトを選択する場合がある．

5.2 検　　索

本モデルにおける検索機構は，探索機構とオブジェクト特定機構からなる．

探索機構は，階層構造グラフの部分グラフを対象として，属性名/属性値に関する検索条件を与え，グラフを返す．まず，Ω 上を探索して条件に適合する要素オブジェクトを取り出す．次に，得られたオブジェクトから部分グラフを生成する．形式は次のようである．

$$\begin{array}{ll} search & \text{要素オブジェクト} \\ from & *, *[C] \\ where & a:v, a:*, *:v. \end{array} \qquad (6)$$

式 (6) で，$from$ 句において階層構造グラフ上の検索領域を指定する．たとえば，階層構造グラフ g の全体（$*$）を指定する．この場合，Ω 上のすべてのオブジェクトが探索の対象となる．また，g の部分グラフ g' を指定することもできる．$*[C]$ は視点 C に属している仮想オブジェクトが検索対象となっていることを示す．カテゴリ C を指定することで，4.1 節で述べた f^- アルゴリズムにより Γ 上で g から部分グラフ g' が抽出される．探索は g' が参照している Ω 上のオブジェクトが対象となって行われる．

where 句では，通常の属性名の固定した属性値の検索パターン $(a:v)$，属性値に依存しない属性名の検索 $(a:*)$，属性名に依存しない属性値の検索 $(*:v)$ に関する条件を指定する．これらは g' のノードから参照される Ω 上の要素オブジェクトに対する条件であり，適合するオブジェクトは Ω に属す．このオブジェクトを Γ に返すことで，探索機構を Γ 上の first class 演算として実現している．

探索機構で得られたオブジェクトを含む仮想オブジェクトは一意ではなく多数存在する．本モデルでは利用者の意図や目的に依存して利用者に仮想オブジェクトを与えるため，オブジェクト特定機構は，探索機構で得られたオブジェクトに対して利用者が見やすいそれらの表現形式を選択するため，オブジェクトを特定する機構を提供する．

オブジェクト特定機構では，まず探索機構で得られたオブジェクトから探索対象の部分グラフ g を上向き/下向きにたどる．これによってヒットしたオブジェクトのすべての上位/下位のオブジェクトを抽出できる．これらを組み合わせることで得られたオブジェクトを含む仮想オブジェクトのすべてを容易に特定できる．仮想オブジェクトの指定には次の2つの指定子 [],

() を用いる.

$$VO[a_1, a_2], \quad VO(a_1, a_2) \tag{7}$$

ここで

$$a_i \ = \ 整数 \ | \ オブジェクト \ | \ *. \tag{8}$$

式 (7) で, [] は端点を表し, () は範囲を指定する. a_i に整数を与えた場合はヒットしたオブジェクトからのエッジ数で相対位置を示し, オブジェクトを与えた場合はグラフ上のオブジェクトの絶対位置を示す. また, * はグラフの端点を示す.

たとえば, 図5で R_{42} がヒットした場合, $VO[C_6, *]$ は仮想オブジェクト $\tilde{D}_2[C_6]$ が特定され, $VO(+1, -1)$ は3つの仮想オブジェクト $\tilde{C}_4[C_6]$, $\tilde{D}_2[C_6]$, $\tilde{D}_2[C_4]$ を表す. 4.1, 4.2節のアルゴリズムにより部分グラフを構成し, 仮想オブジェクトの属性を見ることもできる.

仮想オブジェクト以外にもオブジェクト特定機構を用いて適合した要素オブジェクトをそのまま出力したり, 式 (7) で特定される仮想オブジェクトを構成する要素オブジェクトを出力することもできる.

式 (6) の *from* 句, *where* 句と上記の *select* 文を合わせたものが, 利用者が用いる検索形式である.

例3 例1に述べた「電子メールに対する段階的集合化」における検索機能の適用例を示す.

$$\begin{array}{ll} select & データオブジェクト \\ from & カテゴリ「討議項目」 \\ where & 内容：質問 \end{array} \tag{9}$$

とすることにより, 79ならびに81番のメールを選択することができる. あるいは,

$$\begin{array}{ll} select & VO(+1, *) \\ from & カテゴリ「討議項目」 \\ where & 内容：質問 \end{array} \tag{10}$$

とすると,「パケット通信とパケット交換」に関する「質問」のメールである79番のメールと「ノイズの減少」に関する「質問」である82番のメールを取り出すことができる.

選択したオブジェクトをカテゴリに登録する場合, その間のエッジ (関係オブジェクト) に付与する属性に関しては, 利用者の判断に委ねられるが, 検索時に表示された仮想オブジェクトの属性を利用する場合も多いと考えられる. たとえば, カテゴリ「質問」とデータオブジェクト「82」の間に属性「ノイズの減少」と付与する. □

このようにオブジェクト特定機構では, 階層構造グラフ上でヒットしたオブジェクトから複数の仮想オブジェクト, (実) オブジェクトを自由に取り出すことができる. これによりヒットしたオブジェクトに対しても多様な見方を与えることができる.

本モデルの検索機構は, 通常の OODBMS における構造 (クラス) を指定した検索機能とは異なり, まずインスタンスを検索し, 次に構造情報を抽出することができる. これにより発見的作業においてインスタンスから多様なオブジェクトの表現や集合を探す場合にも有効である.

6. プロトタイプシステム

本稿で提案している半構造化データの構造化におけるシステムの動作を確認するため, 階層構造グラフモデルの一部の機能を実現したプロトタイプシステムを作成した. データベースエンジン部の実装には Elk[13] を用い, 結果の表示とデータの入力部に Tcl/Tk を用いている.

エンジン部分に Lisp システムを用いた理由は, 以下の点である. (1) インタプリタであるため, 変更をすぐにシステムに反映させ, 即座に結果が得られる点でプロトタイプとして優れている. (2) Lisp 言語は内部構造をすべてリスト形式で保持していて, ほとんどすべての要素がリファレンスであるため, 本モデルのオブジェクトを擬似的に表現するのに適している. (3) インタプリタとしてメモリ, ディスクの区別なくリファレンスを表現できる OODB を容易に真似られる.

本システムは画面による入出力部分を除いて, Elk のみで動作しており, グラフの構造化, 仮想オブジェクトの生成, ならびに検索機構などを容易に実現できることを確認した.

図7 に, プロトタイプシステムを用いた電子メールの構造化例を示す. 上部のウインドウは視点「パケット通信」に対する部分グラフを表示している. 視点を指定することにより, 複雑なグラフから利用者が欲する部分のみの構造を抽出して見ることができる. また, 左中のウインドウを用いて, 視点の指定を変更するだけで, 簡便にグラフの表示部分の切替えが行える. 右中のウインドウは,「パケット通信」を視点に集合化されているオブジェクトの一覧を示す. 利用者がオブジェクトを指定することにより, その属性が見える (下奥).

プロトタイプシステムを実現することで, 本モデルがデータに種々の属性を付与し多様な属性構造を実現できること, ならびにデータが持つ意味に従った構造化に適していることを確認できた. 同時に GUI 部で

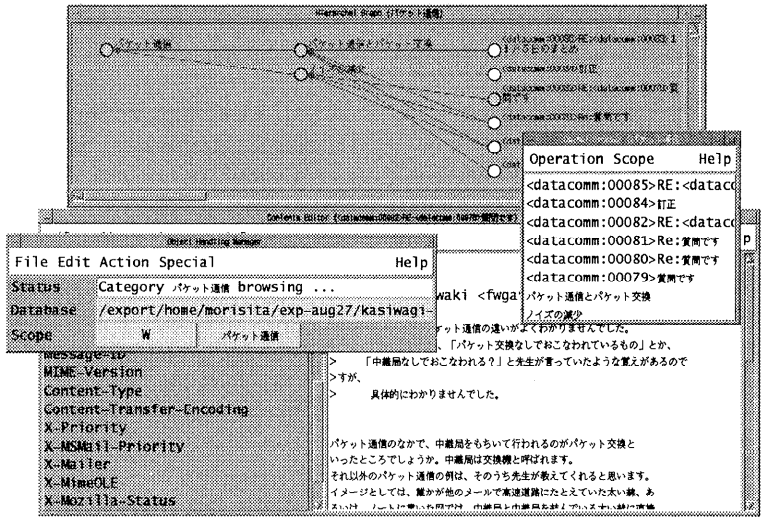


図7 プロトタイプシステム
Fig. 7 Prototype system.

は，操作の簡便性を図るだけでなく，仮想オブジェクトの表示形態，グラフの見えかた，視点の簡便な切替えなどの構造化作業に適したGUIの作成が重要であることが確認された．

7. 関連研究との比較

半構造化データに対してグラフ表現を行うアプローチとして，Bunemanらの研究とPapakonstantinouらの研究がある．Bunemanらは，エッジにラベルの付いたグラフを用い，エッジのラベルにデータの意味を記述し，treeが1つのデータとして取り扱われる[6),7)]．tree以下に対する意味付けは上位のエッジのラベルで行える点は本モデルと同じである．また，Papakonstantinouらは異種のオブジェクトを統合化するモデルとして，OEMモデルを提案している[5)]．しかし，半構造化データに柔軟な表現を与えるためには，アドホックな属性付けが必要であり，本モデルのようにエッジにデータ構造を持たせることが望ましい．本モデルでは，エッジにオブジェクトを配置し，その点を可能にしている．さらに，OEMとUnQLのどちらもノード間の関係は，ノード自身が持つ点が本手法と異なる．

一方，オブジェクトに多面的な表現を生成する手法として，代行オブジェクトモデルがある[14)]．代行オブジェクトは基底オブジェクトに対して個別的にオブジェクトビューを与えることができ，地図データのようにデータの利用法に応じてビューを必要とするデータに有効である．しかし，代行オブジェクトはそれぞれ対応する1つの代行クラスで集約されるのに対し，本モデルでは，オブジェクトがカテゴリにより次々に集合として集約され，カテゴリが階層化される．この違いは，本モデルが半構造化データに対して試行錯誤的に多重な表現を与えながら構造化することを目的とするためであり，両者の目的が異なることに起因する．**表1**に本手法と代行オブジェクトモデルを比較する．

表1 代行オブジェクトモデルと本モデルの比較
Table 1 Comparison with deputy object model.

	代行オブジェクトモデル	本モデル
多重表現	代行オブジェクト	仮想オブジェクト
属性	属性の組合せ	アドホックな属性
生成	マッピングとスイッチ機構	属性伝播
集約	代行クラス	カテゴリ
利用法	プロキシー，フィルタリング	データ構造化

Suciuは複数のサイト上のWebデータをラベル付きエッジグラフモデルで表現し，Webネットワークの論理的な分割をビューとして定義し，実行時に生成している[9)]．

インスタンス単位の属性継承機構として大本ら[15)]による伝播ビューがある．継承は，インスタンス間の包含関係に基づいて行われ，ビデオデータや図面データなどのように領域を持つ場合に有効である．これに対して，本モデルでは，あらかじめ構造が確定されていないデータに対して，新規にオブジェクトを生成し，その属性を恣意的に継承させることで，半構造化データを修飾し，次々に新しいデータの表現を生成している点が異なる．

問合せを用いたオブジェクトの有効なビューの生成法が提案されている．Abiteboul ら[16]は既存のクラス階層へ問い合わせて仮想属性，仮想クラスを定義することで複数の属性を組み合わせる手法を提案している．また Tanaka ら[18]は問合せを用いてクラス階層を仮想化する手法を提案している．しかし，これらのアプローチでは，利用者が新しく生成するアドホックな属性を取り扱うことはできない．また，Rundensteiner ら[19]により検索結果を仮想クラスに格納する手法が提案されている．本モデルでは5章で述べたカテゴリの生成と検索を用いて，検索の結果のオブジェクトの集合をカテゴリの中に入れることで同様の操作を実現できる．

Gutierrez ら[20]は，ノードとエッジにラベルの付いたグラフを用いてデータベース内の既存のオブジェクトの関係付けを graph view として構成する手法を提案している．既存のオブジェクトに対してグラフ構造を生成・更新することでビューを次々に導出できるが，本手法のように1つのオブジェクトの多重表現を段階的に生成・更新するわけではない．また，データベースに新規にオブジェクトを生成したり，それに対応してビューを生成・更新することはできない．

8. おわりに

本稿では，階層構造グラフモデルを導入し，半構造化データを利用者の視点に基づいて段階的に構造化するための枠組みについて述べた．本モデルでは，視点をデータとして陽に扱うため，利用者の視点に基づいて集合化，属性付与などの構造化作業を実行時に行える特徴がある．また，仮想オブジェクトの概念を導入し，データの多重表現を実行時に生成する方法，仮想オブジェクトの導出法などについて述べた．さらに，階層構造グラフに対する動的な集合化と検索について議論した．最後に本稿で提案している半構造化データの構造化におけるシステムの有効性を確認するため，電子メールを例にプロトタイプシステムを示した．階層構造グラフに対しては，上記以外にも階層構造グラフの再構成を行う基本的な操作の実現が重要である．これらについては今後の課題とする．

謝辞 本研究の一部は，文部省科学研究費補助金重点領域研究「高度データベース」(課題番号09230101)，ならびに基盤研究（C）(2)（課題番号07680443）の助成により行われた．

参考文献

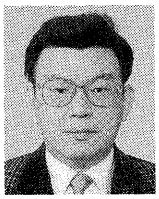
1) Buneman, P.: Semi-structured data, http://www.cis.upenn.edu/db/tutorials/semistructured-paper.ps.
2) Abiteboul, S.: Querying semi-structured data, *Proc. ICDT* (Jan. 1997).
3) IEEE Computer Society: Special Issue on Scientific Databases, *Bulletin of the Technical Committee on Data Engineering*, Vol.16, No.1 (1993).
4) 上島紳一，森下淳也，大月一弘，杉山武司：階層構造グラフを用いた半構造化データの段階的な構造化手法の提案，情報処理学会研究報告，DBS-111, pp.9–16 (1997).
5) Papakonstantinou, Y., Abiteboul, S. and Garcia-Molina, H.: Object Exchange Across Heterogeneous Information Sources, *IEEE Proc. International Conference on Data Engineering*, pp.251–260 (1995).
6) Buneman, P., Davidson, S., Hillebrand, G. and Suciu, D.: A Qucry Language and Optimization Techniques for Unstructured Data, *Proc. 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, pp.505–516 (June 1996).
7) Buneman, P., Davidson, S., Hart, K. and Overton, C.: A Data Transformation System for Biological Sources, *Proc. 21st VLDB Conference*, Zurich, Switzerland, pp.158–169 (1995).
8) Abiteboul, S., Quass, D., McHugh, J., Widom, J. and Wiener, J.: The Lorel Query Language for Semistructured Data, *Journal on Digital Libraries*, Vol.1, No.1 (1996).
9) Suciu, D.: Query Decomposition and View Maintenance for Query Languages for Unstructured Data, *Proc. 22nd VLDB Conference*, India, pp.227–236 (1996).
10) Shoens, K. Luniewski, A., Schwarz, P., Stamos, J. and Thomas, J.: The Rufus System: Information Organization for Semi-structured Data, *Proc. 19th VLDB Conference*, Dublin, Ireland, pp.97–107 (1993).
11) Ueshima, S., Ohtsuki, K., Morishita, J., Qian, Q., Oiso, H. and Tanaka, K.: Incremental Data Organization for Ancient Document Databases, *Proc. 4th International Conference on Database Systems for Advanced Applications (DASFAA'95)*, Singapore, pp.457–466 (Apr. 1995).
12) Zdonik, S.: Incremental Database Systems: Databases from the Ground Up, *Proc. 1993 ACM SIGMOD International Conference on Management of Data*, Washington DC, USA, pp.408–412 (May 1993).
13) Laumann, O. and Bormann, C.: ELK: The

Extension Language Kit, *USENIX Computing Systems*, Vol.7, No.4, pp.419–449 (1994).
14) Kambayashi, Y. and Peng, Z.: Object Deputy Model and Its Applications (Keynote Paper), *Proc. 4th International Conference on Database Systems for Advanced Applications (DASFAA'95)*, Singapore, pp.1–15 (Apr. 1995).
15) Oomoto, E. and Tanaka, K.: OVID: Design and Implementation of a Video-Object Database Systems, *IEEE Trans. Knowledge and Data Enginerring* (Aug. 1993).
16) Abiteboul, S. and Bonner, A.: Objects and Views, *Proc. 1991 ACM SIGMOD, International Conference on Management of Data*, Denver, Colorado, USA, pp.238–247 (Feb. 1991).
17) Tanaka, K., Nishio, S., Yoshikawa, M., Shimojo, S., Morishita, J. and Jozen, T.: Obase Object Database model: Towards a More Flexible Object-Oriented Database System, *Proc. International Symposium on Next Generation Database Systems and Their Applications (NDA'93)*, pp.159–166 (Sept. 1993).
18) Tanaka, K. and Yoshikawa, M.: Schema Design, Views and Incomplete Information in Object-Oriented Databases, *Journal of Information Processing*, Vol.12, No.3, pp.239–250 (1989).
19) Rundenstciner, E.A.: Object-Oriented View Technology: Challenges and Promises, *Proc. International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'96)*, Kyoto, Japan (Dec. 1996).
20) Gutierrez, A., Pucheral, P., Steffen, H. and Thevenin, J.M.: Database Graph Views: A Practical Model to Manage Persistent Graphs, *Proc. 20th VLDB Conference*, pp.391–402, Chile (Apr. 1994).
21) 森下淳也，上島紳一，大月一弘，杉山武司：階層構造グラフにおける属性の取り扱い方に関する検討，信学技報，DE-96, pp.31–36 (1997).

(平成 9 年 9 月 3 日受付)
(平成 10 年 2 月 2 日採録)

上島 紳一（正会員）

1955 年生．1978 年京都大学工学部数理工学科卒業．1983 年同大学大学院工学研究科博士課程単位取得退学．1986 年関西大学文学部専任講師，同助教授を経て，1994 年同大学総合情報学部教授，現在に至る．1985 年システム制御情報学会より樫木記念賞論文賞受賞．京都大学工学博士．システムモデリング，柔構造データベースなどの研究に従事．電子情報通信学会，システム制御情報学会，情報考古学会，ACM 等各会員．



森下 淳也（正会員）

1956 年生．1979 年神戸大学理学部卒業．1984 年同大学大学院自然科学研究科博士課程後期修了．同年，神戸大学総合情報処理センター助手．1988 年姫路獨協大学外国語学部助教授を経て，1997 年神戸大学国際文化学部助教授，現在に至る．神戸大学学術博士．オブジェクト指向データベースシステム等の研究に従事．システム制御情報学会，米国物理学会，AAAS 等各会員．

大月 一弘（正会員）

1958 年生．1981 年京都大学工学部数理工学科卒業．1986 年同大学大学院工学研究科博士課程単位取得退学．同年 神戸大学教養部助手．現在，同大学国際文化学部助教授．京都大学工学博士．情報ネットワーク，通信方式，科学技術データベースの研究に従事．電子情報通信学会会員．

杉山 武司（正会員）

1951 年生．1974 年 早稲田大学理工学部機械工学科卒業．1977 年同大学大学院理工学研究科修士課程修了．現在，姫路獨協大学一般教育部助教授．数値解析，関数型プログラミングの研究に従事．電子情報通信学会会員．