

対応データベースのためのグラフィカルエディタの開発

(平成16年11月30日 原稿受付)

工学研究科 桂 善 浩
工学研究科 唐 重 順 平
工学研究科 横 尾 徳 保
工学研究科 重 松 保 弘

Design and Development of a Graphical Editor for Correspondence Database

by Yoshihiro KATSURA
Junpei KARASHIGE
Noriyasu YOKOO
Yasuhiro SHIGEMATSU

Abstract

This paper presents the design and development of a Correspondence Database Editor (CDE), a graphical editor for Correspondence Database, and shows how to analyze the data graphically in the CDE. Furthermore, we discuss an ideal application interface for the semi-structured database comparing with similar applications. Correspondence Database is a kind of semi-structured database based on Correspondence Data Model which we have proposed. In Correspondence Data Model, data are represented with the labeled correspondence which is a set of three-tuples (label, initial vertex, terminal vertex), and its structure is essentially equivalent to a labeled directed graph. Therefore, Correspondence Database is unnecessary to define schema in advance. In Correspondence Data Model, "Correspondence Operation" enables users to manipulate data systematically. The CDE provides instinctive and visual manipulation such as a join operation, a selection operation or a cut-and-paste operation. The CDE also has a function to arrange nodes semiautomatically. These manipulations make users possible to analyze semi-structured data efficiently. In addition, the CDE is able to handle multimedia data such as images because the CDE attaches URL information to nodes in the graph.

1. はじめに

サイエンスやコンテンツアーカイブなどの分野では、構造が頻繁に変化する半構造データを対象とするデータベースの重要性が増しており、このようなデータに関する研究が盛んに行われている⁽¹⁾。半構造データベースの典型的な応用例としては、セマンティック Web の基礎技術である RDF⁽²⁾ が挙げられる。RDF は Web 上の多様な構造を持つデータを表現するため、ラベル付き有向グラフ構造のデータモデルを与える。

これまで本研究室では、半構造データのためのデータモデルとして、ラベル付き対応を用いた“対応データモデル”を提案し⁽³⁾、問合せ言語の設計や、データベース

管理システムの試作⁽⁴⁾などを行ってきた。対応データモデルは、現実世界の構造記述にラベル付き対応を用いる。ラベル付き対応とは、1 対多の対応関係にラベルを付けたものである。したがって、対応データモデルは実質的にラベル付き有向グラフ構造のデータモデルと等価である。そのため、現実世界の概念を理解したままの構造で記述することができる。また、対応という概念を導入することにより、グラフおよびグラフの構成要素（ラベル、始点、終点）の集合を代数として扱うことが可能となっている。加えて、データ操作の手段として対応演算が定義されており、体系的なデータ操作を行うことができる。

筆者らは、対応データモデルに基づくデータベースシステムの開発を行っている。このデータベースシステム

の設計目標は、スキーマに縛られない自由なデータの記述ができ、メモ感覚で手軽に構造化されたデータを蓄積できることである。主な用途として、携帯端末を利用した取材活動の支援、電子カルテ、アイデアプロセッサ等を想定している。このようなアプリケーションにおいて実世界を忠実に写生したデータベースを構築するためには、概念を直感的に記述できる GUI が不可欠である。

そこで本研究では、対応データベースシステムにおいて GUI が必要とする機能を考察するため、グラフデータを簡便に作成する GUI アプリケーション、CD エディタ (Correspondence Database Editor) を開発した。CD エディタには、対応データベースを編集するための基本的な機能のほかに、対応演算やカットアンドペーストを使ってグラフを再構成する機能などを実装した。これにより、ユーザは独自の視点でグラフを変形して眺め、分析することができる。

本稿では CD エディタの開発について述べ、視覚的なデータ操作ができることを示す。以降、2章で前提となる対応データモデルと対応データベースシステムの概要について述べる。3章でエディタの設計と実装について述べ、4章で使用例を示す。5章では、関連ツールとの比較を行いながら、対応データベースシステムにおける GUI のあり方について議論する。6章では、まとめと今後の課題について述べる。

2. 対応データベース

2.1 ラベル付き対応

A と B を集合とする。ある規則 f によって A の元 a に対してそれぞれ 1 つずつの B の部分集合 $f(a)$ が定められるとき、規則 f を A から B への対応と呼び、 $f: A \rightarrow B$ と書く。 $f(a)$ を f による a の像と呼ぶ。

対応 f にラベル集合 L を与え、ラベル $l (\in L)$ ごとに異なる対応として扱えるようにしたものを $f\{L\}$ をラベル付き対応と呼ぶ。また、ラベル l 、始点 a 、終点 b の組を対応要素と呼ぶ。ラベル付き対応を対応要素の集合と考え、式 1 のように定義する。

$$f\{L\}: A \rightarrow B = \{(l, a, b) \mid f\{l\}(a) = b, l \in L, a \in A, b \in B\} \quad (1)$$

ラベル付き対応は、式 1 の L, A, B をそれぞれ、ラベル付き有向グラフのラベルの集合、始点の集合、終点の集合と捉えることで、ラベル付き有向グラフと相互に変換できる。以降、ラベル付き対応をグラフ、対応要素をリンク、集合 A および B の元をノードと呼ぶ。

2.2 対応データモデル

対応データモデルは、実世界の概念を複数のグラフで

記述するモデルである。各ノードはそれぞれ異なるラベルのリンクが持てるため、実体ごとに異なる属性を記述できる。また、あるノードを始点 (終点) とする同じラベルのリンクが複数存在してもよいから、多対多の関係を記述できる。

対応データモデルでは、データの追加、削除、選択といった操作のために、対応代数が定義されている。対応代数には、表 1 に示す対応演算が規定されている。対応演算の定義を式 2 から式 11 に示す。なお、 θ は 2 項の関係演算子、 C は比較対象の値である。

(a) 結果がグラフである演算

対応和演算

$$f\{L_1\} \cup g\{L_2\} = \{(l, a, b) \mid (l, a, b) \in f\{L_1\} \vee (l, a, b) \in g\{L_2\}\} \quad (2)$$

対応差演算

$$f\{L_1\} - g\{L_2\} = \{(l, a, b) \mid (l, a, b) \in f\{L_1\} \wedge (l, a, b) \notin g\{L_2\}\} \quad (3)$$

対応共通演算

$$f\{L_1\} \cap g\{L_2\} = \{(l, a, b) \mid (l, a, b) \in f\{L_1\} \wedge (l, a, b) \in g\{L_2\}\} \quad (4)$$

domain 選択演算

$$f[\theta C\{L\}] = \{(l, a, b) \mid (l, a, b) \in f\{L\} \wedge a \theta C\} \quad (5)$$

range 選択演算

$$f[\{L\} \theta C] = \{(l, a, b) \mid (l, a, b) \in f\{L\} \wedge b \theta C\} \quad (6)$$

label 選択演算

$$f[\{L \theta C\}] = \{(l, a, b) \mid (l, a, b) \in f\{L\} \wedge l \theta C\} \quad (7)$$

表 1 対応演算

演算結果	分類	演算名	説明
グラフ	二項演算	対応和演算	リンクの和集合を求める
		対応差演算	リンクの差集合を求める
		対応共通演算	リンクの共通集合を求める
	単項演算	domain 選択演算	始点ノードが条件を満たすリンクの集合を求める
		range 選択演算	終点ノードが条件を満たすリンクの集合を求める
		label 選択演算	ラベルが条件を満たすリンクの集合を求める
ノードやラベルの集合	-	domain 演算	グラフから始点ノードの集合を取得する
		range 演算	グラフから終点ノードの集合を取得する
		label 演算	グラフからラベルの集合を取得する

(b) 結果がノードやラベルの集合である演算

domain 演算

$$dom[f\{L\}] = \{a \mid (l, a, b) \in f\{L\}\} \quad (8)$$

range 演算

$$ran[f\{L\}] = \{b \mid (l, a, b) \in f\{L\}\} \quad (9)$$

label 演算

- domain-label 演算

$$lab[=Nf\{L\}] = \{l \mid (l, a, b) \in f\{L\} \wedge a \in N\} \quad (10)$$

- range-label 演算

$$lab[f\{L\}] = N = \{l \mid (l, a, b) \in f\{L\} \wedge b \in N\} \quad (11)$$

対応データモデルを効率的に利用するには、適当な大きさにグラフを分割して管理し、必要に応じてそれらに対応和演算により結合するとよい。ここで“適当な大きさ”とは、次のような尺度の大きさを指す。

- 概念が把握しやすい大きさ。
- 計算機の処理速度や主記憶の容量、通信速度を考慮したとき、実用的にデータ操作が可能な大きさ。
- GUI でグラフを編集する際、画面に一度に表示される情報量が適切となる大きさ。

2.3 対応データベースシステム

筆者らは、メモを取る感覚で簡単に構造化されたデータを蓄積できるデータベースシステムの開発を目指している。そのためには、スキーマに縛られない自由なデータの記述ができ、時間の経過とともに明らかになる情報を即座に登録できる必要がある。

グラフ構造のデータベースの中には、データの不規則さや不完全さを排除するため、“この型のノードにはこの属性を記述すべきである”といったスキーマを設定し、スキーマに違反したデータの追加を禁止するものがある。しかし、あらかじめ設定したスキーマに適合しないデータを追加する場合、まずスキーマを修正し、その後データを追加する手間が生じる。そのため対応データベースシステムでは、このように柔軟性のないスキーマの管理機能を排除している。

3. エディタの設計と実装

3.1 エディタの概要

CD エディタはグラフデータを編集するエディタである。作成したデータはグラフごとにファイルに保存する。現在は主にシステムの動作確認に使うサンプルデータの収集に利用しているが、将来的にはユーザがGUI に対して行う操作を問合せ文に変換し、データベース管理サーバに送信することで、データベースシステムのフロントエンドとして利用する計画である。CD エディタ

は C# 言語で開発を行っている。

図-1 に CD エディタの概観を示す。画面上部のツールバーには、編集モードの切り替えやグリッド機能など、頻繁に使う機能をまとめた。画面右にはグラフの編集を行うビューアがある。ユーザはここでグラフデータを作成する。またビューアはデータを分析する際にも、グラフを変形しながら概念の間に存在する関係を確認するために利用する。画面左にはグラフリストがある。これは、ユーザがデータを分析する過程で中間生成したグラフを、一時的にメモリ中に保持しておくために使う。

CD エディタを起動した時点では、ビューアには何も表示されない。新しいグラフを作成するには、ツールバーからリンク作成モードを選択する。ビューア上で始点から終点に向かってマウスをドラッグすると、新しいノードとリンクを作成することができる。これを繰り返して、目的のグラフを描く。複数のグラフにデータを分割したいときには、新しいウィンドウを開き、グラフの一部をカットアンドペーストで移動する。複数のグラフを1つのグラフに結合して眺めることもできる。

3.2 対応演算

ユーザはデータを分析して情報を得るために、対応演算を組み合わせ使用。本節では、CD エディタに実装した対応演算の仕様について述べる。

基本的な対応演算 CD エディタには、対応和演算、対応差演算、対応共通演算、3つの選択演算を実装した。選択演算の関係演算子としては、 $>$ 、 \geq 、 $=$ 、 \leq 、 $<$ の5種類が使える。次に、結果がノードやラベルの集合である演算について述べる。対応データモデルでは、いず

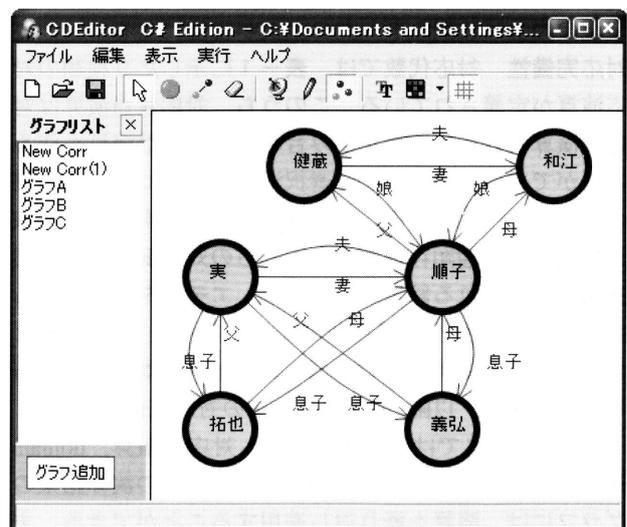


図-1 CD エディタの概観

れのリンクの始点でも終点でもない孤立したノードの存在を許すか否かを規定していないが、CD エディタは孤立したノードを便宜上許す。そこでCD エディタでは、リンクのないノードのみのグラフで集合を表現することにした。そこで、domain 演算、range 演算と等価な演算として、グラフから始点や終点のノードのみを抽出したグラフを得る演算を実装した。また label 演算と等価な演算として、グラフ内に存在するラベルと同じ名前のノードを持つグラフを得る演算を実装した。この演算と domain 選択演算や range 選択演算を組み合わせることによって、domain-label 演算、range-label 演算を行うことができる。

IN 演算 CD エディタでは、IN 演算を新たに定義して実装した。IN 演算には、SOURCE IN 演算、DEST IN 演算、LABEL IN 演算の3つがある。例えば SOURCE IN 演算とは、グラフ A のリンクのうち、始点がグラフ B に存在するノードまたはラベルの名前と同じものだけを抽出したグラフを得る演算である。IN 演算の定義を式12から式14に示す。

・ SOURCE IN 演算

$$f\{L_1\}.S.IN\ g\{L_2\} = \{(l, a, b) \mid (l, a, b) \in f\{L_1\} \wedge (a \in (dom[g\{L_2\}] \cup ran[g\{L_2\}] \cup L_2))\} \quad (12)$$

・ DEST IN 演算

$$f\{L_1\}.D.IN\ g\{L_2\} = \{(l, a, b) \mid (l, a, b) \in f\{L_1\} \wedge (b \in (dom[g\{L_2\}] \cup ran[g\{L_2\}] \cup L_2))\} \quad (13)$$

・ LABEL IN 演算

$$f\{L_1\}.L.IN\ g\{L_2\} = \{(l, a, b) \mid (l, a, b) \in f\{L_1\} \wedge (l \in (dom[g\{L_2\}] \cup ran[g\{L_2\}] \cup L_2))\} \quad (14)$$

domain 演算、range 演算、label 演算と IN 演算を交互に繰り返すことにより、グラフ中のパスをたどることができる。また、label 選択演算などと組み合わせれば、パス表現の間合せに似たデータの選択が可能になる。

対応完備性 対応代数では、表-1に示した9種類の対応演算が定義されている。このうち、対応共通演算は対応和演算と対応差演算を組み合わせると等価な演算を作ることができる。また、label 選択演算は式1にラベルの集合 $L' (C L)$ を与えることと等価である。すなわち、独立な演算は7種類存在する。これらの対応演算を有限回適用して得られる結果の集合を S_1 、エディタで許される操作を有限回使って得られる結果の集合を S_2 とするとき、 $S_2 \supseteq S_1$ が成り立つならば、エディタは対応完備であると定義する。以降、CD エディタの対応完備性を示す。

CD エディタでは、対応和演算、対応差演算、domain 選択演算、range 選択演算が行える。これらの演算結果のグラフには、演算を繰り返し適用することができる。また domain 演算、range 演算、domain-label 演算、range-label 演算と等価な演算が行える。これらの演算結果は、

集合をリンクのないノードのみのグラフで表現したものである。このグラフに対応和演算、対応差演算、対応共通演算を適用して、集合の和、差、積をシミュレートすることができる。IN 演算は選択演算の定義(式5から式7)の比較対象の値 C をノードやラベルの集合に、関係演算子 θ を \in または \ni に置き換えたものに相当する。したがって、関係演算子 θ を $>$, \geq , $=$, \leq , $<$, \in , \ni のいずれかと仮定すれば、CD エディタは対応完備であるといえる。

対応演算の実行手順 対応演算を実行するには、図-2に示すダイアログを開く。そして演算対象のグラフを、ビューアで編集のグラフ、クリップボード上のグラフ、グラフリストのグラフの中から選択する。グラフリストから複数のグラフを選択すると、演算に先立ってそれらが対応和演算により結合される。次に、演算の種類を選択し、選択演算の場合はラベル、始点、終点に加える条件を指定する。最後に、演算結果の出力先を決定してOKボタンを押す。

カットアンドペースト CD エディタでは、グラフの一部をカット(コピー)アンドペーストできる。これにより、大きなグラフを分割したり、意味上のまとまりを考慮してグラフを再構成したりすることができる。カットアンドペーストは、“切り取り”の操作が対応差演算に、“貼り付け”の操作が対応和演算に相当することから、対応演算の一部であるといえる。

3.3 その他の機能

グラフの編集 グラフ編集の操作性は、ユーザが概念を直感的に記述するために、最も工夫が求められる部分である。CD エディタでは断片的な知識を記述できるよう、グラフの任意の部分から任意の順番で編集可能な操作性を考案した。

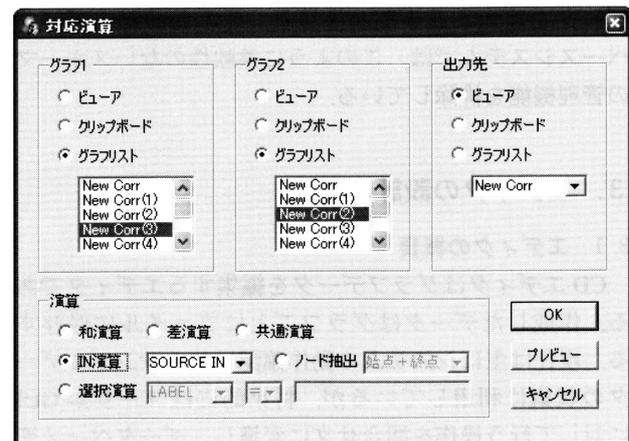


図-2 対応演算ダイアログ

CD エディタには、ポインタ、ノード作成、リンク作成、消しゴムの4つのモードがある。ポインタモードは、ノードやリンクを選択したり、ドラッグによって移動したりするモードである。ノード作成モードでは、クリックした場所にノードが新規に作成される。グラフの作成には、主にリンク作成モードを用いる。このモードでは、ドラッグした2点間にリンクが作成される。ドラッグを開始した位置に既にノードが存在する場合は、そのノードを始点とするリンクが作られる。ノードが存在しない場合（すなわちドラッグを開始した位置が背景だった場合）は、始点ノードとリンクを一度に作成できる。終点のノードも同様である。ノードまたはラベルをダブルクリックするとテキストボックスが開くので、正しい名前に変更すればリンクの作成が完了する。消しゴムモードでは、クリックしたノードおよびリンクが削除される。

リンクに付けられたラベルをドラッグすると、ラベルがその方向に移動し、リンクが曲線となる。双方向のリンクが設定されている場合や、複雑なグラフの場合に、見やすくレイアウトすることができる。

レイアウトの半自動化 グラフを見やすいレイアウトに修正する作業を支援するため、マグネティックスプリング法⁶⁾を用いた自動レイアウト機能を実装した。ただし“このノードをここに置きたい”というユーザの意図を尊重するため、対話的に支援が受けられるよう工夫した。ノードが移動する様子はアニメーションで確認できる。ノードはCtrlキーとShiftキーを同時に押したときだけ移動する。また、あらかじめマウスで選択したノードのみが移動する。そのためユーザは、ノードの位置を手動で変更する操作をあわせて行うことで、意図するレイアウトに短時間でたどりつける。

スタイルの保存 対応データベースは、本質的にノード間にどのようなリンクが存在するかという情報のみを持つ。しかし、CD エディタはノードやラベルの表示位置に関する情報もあわせてファイルに保存できるようにした。このため、ユーザはノードを発見しやすく、編集作業を中断した後もスムーズに作業を再開できる。

ファイルとテキストの関連付け 対応データベースシステムでは、マルチメディアデータをノードに関連付けて管理する。CD エディタはこの仕様に準じ、ファイルパスやURLをノードに関連付けて保存できるようにした。ノードをクリックすると、関連付けられたファイルを開くことができる。また、関連付けたファイルがグラフデータの場合には、CD エディタでそのファイルを開くため、グラフのハイパーリンクを表現できる。同様に、テキストをノードに関連付けて管理することもできる。

ファイル形式 CD エディタは、グラフデータを独自のXML形式にシリアルライズして保存する。この他に、CSV形式に似た3つ組形式での保存と読み込みができる。RDF

ファイルを読み込むこともできる。

4. エディタの使用例

本章では、CD エディタを用いて年賀状のデータベースを構築し、分析する例を示す。

4.1 年賀状データベースの設計

CD エディタでは、データの分析に対応演算を用いる。複数のデータに対して同じ操作を適用し、効率的にデータの選択を行うためには、基本的なデータの構造が統一されている必要がある。そのためこの例では、まず年賀状データベースの構造の設計を行う。

それぞれの年賀状について、年賀状の画像、年度、差出人、その年賀状に対して返事を書いたかどうかを記録することにする。ある年度に届いた年賀状のデータを1つのグラフにまとめて記述することにする。年賀状を表すノードには3桁の通し番号を付ける。年賀状の画像はこのノードに関連付ける。差出人を表すノードのノード名として人名を用いると、同じ人名を持つ差出人の区別ができないので、差出人を表すノードのノード名は2桁の通し番号とし、差出人の名前はこのノードの属性として記述する。リンクの始点と終点の集合を表-2のように仮定する。

表-2 データの構造

ラベル	始点	終点
年度	年賀状を表すノード	西暦の4桁の数字
差出人	年賀状を表すノード	差出人を表すノード
名前	差出人を表すノード	人名
返信	年賀状を表すノード	“あり”または“なし”

4.2 データベースの作成

仮定した構造に従いデータを作成する。異なる実体を同じ名前のノードで表したり、同じ実体を表すノードを違う名前で作成したりすると、期待した演算結果が得られない。そのためノードを追加する際にユーザは、同じ実体を表すノードが既にデータ中に存在していないかを常に確認しなければならない。図-3にグラフ“2003”と“2004”を示す。

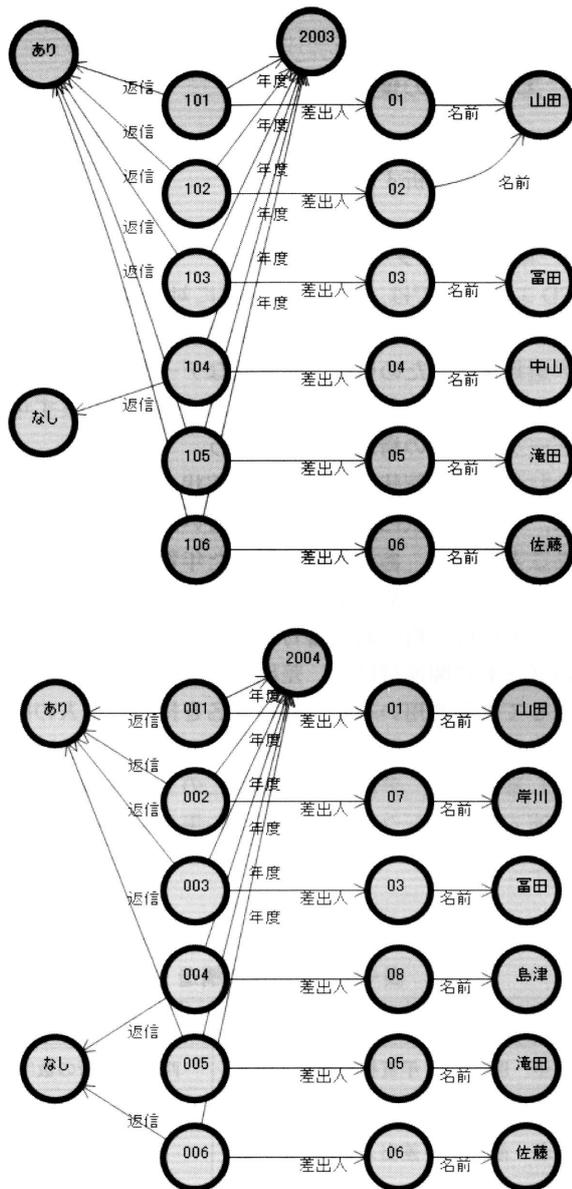


図-3 グラフ“2003”(上)とグラフ“2004”(下)

4.3 データの分析

ここでは2003年度、2004年度に年賀状を受け取っており、どちらの年賀状にも返信した相手を見つける。

- (1) まずグラフ“2003”と“2004”の対応和演算を行う。差出人を表すノードのうちいくつかは両方のグラフに含まれているので、それらが接点となって2つのグラフは結合される。これをグラフAとする。
- (2) 次に、返信した年賀状のみを選択するため、グラフAに対して終点が“あり”のリンクのみを抽出するrange選択演算を行う。さらに、domain演算で始点ノードを抽出する。これをグラフBとする。

- (3) 返信した年賀状の差出人を選択するため、グラフBに含まれるノードを始点に持つリンクのみをグラフAから抽出するSOURCE IN演算を行う。これをグラフCとする。グラフCの差出人を表すノードのうち、2つ以上のリンクの終点となっているものが目的とする差出人である。グラフCから、その他の差出人のノードを削除する。これをグラフDとする。
- (4) 差出人の名前を得るため、グラフDに含まれるノードを始点に持つリンクのみをグラフAから抽出するSOURCE IN演算を行う。その結果のグラフに対して、ラベルが“名前”のリンクのみを抽出するlabel選択演算を行う。分析結果のグラフを図-4に示す。

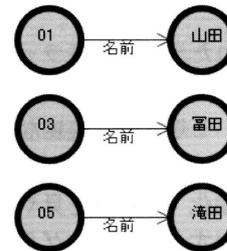


図-4 分析結果のグラフ

この例のようにユーザの仮定した構造に沿って規則正しくデータが作成されている場合は、対応演算を用いて計画的にデータの分析を進めることができる。特にデータベース管理システムでは、問合せ文を作成することで、対応演算を複雑に組み合わせたデータの操作を比較的行うことができる。しかし実際には、ユーザは不規則なデータや不完全なデータを作ることができる。そのため、データ操作が正しく行われたことを確認したり、対象のデータの特徴を把握したりしながらデータの分析を進める必要がある。

5. 関連ツール

本章ではグラフデータを扱う関連ツールを紹介し、定性的な比較を行いながら、対応データベースシステムのGUIに必要な機能について論じる。

5.1 関連ツールの紹介

TouchGraph TouchGraph[®]は、意味ネットワークの分析やWebサイト間の関連の可視化などに用いられている。TouchGraphではノード同士が自然な距離を保つように自動的に分散して配置される。TouchGraphでは、中心と

なるノードを1つ定める。そのノードから指定したホップ数でたどれるノードのみが画面に表示され、その他のノードは自動的に隠れる。この方式は、中心のノードからの1対多の関係を把握するには適しているが、多対多の関係を読み取る目的には向かない。また、再表示のたびにノードの配置が変化し、ノードを発見しづらい。

CDエディタのグラフ編集の操作性はTouchGraphを参考に設計したため、大きな影響を受けている。ただしTouchGraphでは、ノードに隣接するノードを次々に作成し、グラフを拡張する方式でデータを作成する。そのため、グラフを任意の部分から作成することができず、グラフ構造データベースの作成には適さない。

iEdit iEdit⁽⁷⁾はアイデアプロセッサの一種である。iEditの特徴はツリーとグラフを組み合わせたデータの構造にあり、グラフを階層的に整理できる。iEditでは、グラフを任意の部分から作成できる。また、ノードにはテキストやURLをつけることができる。データの分析のため、ノードを自動的に分散してレイアウトする機能がある。

iEditでは、ノードやリンクをカットアンドペーストすることができるが、ノードが接点となってグラフが結合することはない。スキーマのサポートや問合せの機能はなく、データベースとしての性格は薄い。メモを取るようにグラフデータを作成できる点で設計思想が似ている。

Protégé Protégé⁽⁸⁾⁽⁹⁾はドメインオントロジーを作成するツールである。定義したクラスのインスタンスをGUIで作成し、オントロジーに基づいて問合せを行うことができる。Protégéではデータの型や値の範囲について厳密な定義を与える。また、オントロジーを構築すると同時に、属性値にアクセスするためのフォームを作成できる。そのためユーザは、自然なインタフェースでスキーマに沿ったデータの作成ができる。ノードの作成時には、予め定義したクラスから作成したいものを選んで配置する。リンクの作成時には、始点と終点のノードの型がオントロジーの制約に違反しないか自動的にチェックされる。Protégéは、データを構築する過程でスキーマを変更する必要が生じた場合や、スキーマが不明確なデータを記述する場合に、スキーマの修正のための操作が複雑になる。

anacubis anacubis⁽¹⁰⁾は、グラフ構造データの分析のための商用アプリケーションである。1つのドキュメントの中には複数のシートが作成でき、シートに実体の属性と実体間の関連をラベル付き有向グラフ構造で記述していく。必要に応じて、複数のシートに記述されたデータをドラッグアンドドロップ操作により1つのシートに統合し、眺めることができる。これはデータを適当な大きさに分割して管理するという点で、対応データベースとよく似ている。ノードには型が用意されていて、記述すべき属性が定義されている。ノードの型はユーザが新しく

定義することもできる。このスキーマ管理機能はProtégéと同様に、語彙と構造の統一に役立つ。

グラフィカル問合せ 問合せ文の代わりに、問合せのための図を作成してデータを検索するシステムが開発されている。例えばDUO⁽¹¹⁾はグラフィカル問合せ言語の一種である。DUOはパス表現だけでなく、不完全なデータに対する問合せに有効な正規表現をサポートする。

対応データベースは性質上、スキーマをもとに問合せ文を作成することができない。また、リレーショナルデータベースではユーザが直接SQL文を作成することは稀であるが、対応データベースでは対象のデータに合わせて問合せ文の構造を変える必要があり、ユーザ自身が問合せ文を作成してデータの分析に用いることが予想される。したがって問合せ言語は、ユーザがデータベースから取得したい概念を直感的に表現できるものでなければならない。このような理由から、CDエディタはグラフィカル問合せをサポートすることが望ましい。

Focus+Context Focus+Context⁽¹²⁾は、ユーザの関心が強い部分を詳細に、周辺に向かうにしたがい概略的に情報を表示する可視化手法であり、グラフデータへの適用例⁽¹³⁾もある。ユーザがデータの構造を知らない場合には、このような技術を利用しながらグラフ内を探索し、所望の情報に近づいていく必要があると考えられる。

5.2 GUIに対する要求

対応データベースシステムのGUIに対する要求を以下にまとめる。

グラフ編集の操作性 時間の経過とともに明らかになる情報を即座に記述できる必要がある。また、ユーザは現実世界の概念により近い操作モデルでグラフを作成することが望ましい。例えば、グラフ作成のためのインタフェースを複数提供し、状況に応じて使い分けるなどの工夫が考えられる。

スキーマ管理 ユーザはスキーマを仮定し、自主的にデータの構造を統一しなければならない。そのため、より複雑なデータベースを構築する場合には、ユーザを支援する機能が必要となる。例えば、ラベルの一覧をユーザに提示する機能や、ラベルを決定すると終点ノードの候補を提示する機能が考えられる。

データの分析機能 データの絞り込みや再構成といった分析機能が必要である。また、その分析機能は半構造データの性質を反映した柔軟なものでなければならない。例えば、厳密に条件にあてはまるものだけでなく、構造や語句の類似したものを返すあいまいな検索ができるとよい。また、グラフィカル問合せは問合せ文の作成を容易にするので、ユーザは問合せ文を修正しながら対話的に検索を進めることで、データ中の局所的な構造を利用した検索が可能になる。

ブラウジングの支援 Focus+Contextは、ユーザをより興

味の深い情報に誘導する。また、グラフの自動レイアウト機能は、ユーザの作業を軽減して情報の発見を促す。

6. おわりに

本稿では、対応データベースをグラフィカルに作成するアプリケーション、CD エディタを紹介した。また、対応演算を用いて結果をグラフィカルに確認しながらデータを分析できることを示した。最後に、対応データベースの GUI に必要な機能について考察した。

これまで対応データベースの作成は、データベース管理システムにコマンド文を入力しながら行っていた。しかし CD エディタの開発によって、直感的なデータの記述が可能になり、実世界の概念を忠実に写生したデータベースの構築ができるようになった。これは、対応データモデルによる情報の整理と発見のプロセスに関わる、本質的な問題の改善といえる。

対応データベースシステムは、スキーマに違反したデータの追加を禁止するような、柔軟性のないスキーマの管理機能を排除している。しかし、データの共有性や再利用性を確保するためには、基本的なスキーマを統一する必要がある。そのため筆者らは、より柔軟にスキーマの統一を支援する機能を設けたり、不規則なデータからスキーマが明確なデータへの変換を補助するツールを提供したりする予定である。今後は、このようなデータの作成と分析のための機能について検討を進め、CD エディタに実装する。また、データベース管理サーバや Web との連携を図り、多人数がデータベースにアクセスすることを想定したシステムの構築を行う。

参考文献

- (1) 田島 敬史：半構造データのためのデータモデルと操作言語、情処論データベース、Vol. 40, No. SIG3 (TOD1), pp.152-170, 1999.
- (2) G. Klyne and J. Carroll : Resource Description Framework (RDF) : Concepts and Abstract Data Model, W3C Working Draft, 2003.
- (3) 横尾 徳保, 重松 保弘：半構造データのための対応データモデルの提案と問い合わせ言語の設計, DEWS 2003 論文集, ISSN 1347-4413, 8-B-04, 2003.
- (4) 横尾 徳保, 唐重 順平, 桂 善浩, 重松 保弘：半構造データのための対応 DBMS の開発 - 問い合わせ言語の設計とデータベースエンジンの実装 -, 平成 16 年度電気関係学会九州支部連合大会論文集, pp.505, 2004.
- (5) 三末 和男, 杉山 公造：マグネティック・スプリング・モデルによるグラフ描画法について, 情処論ヒューマンインタフェース, Vol. 55, No. 055-003, pp.17-24, 1994.
- (6) TouchGraph : <http://www.touchgraph.com/>
- (7) iEdit : <http://homepage3.nifty.com/kondoumh/>
- (8) Protégé : <http://protege.stanford.edu/>
- (9) J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy and S. W. Tu : The Evolution of

Protégé : An Environment for Knowledge-Based Systems Development, International Journal of Human-Computer Studies, vol. 58(1), pp.89-123, 2003.

- (10) anacubis : <http://www.anacubis.com/>
- (11) 宝珍 輝尚：グラフィカル問合せ言語 DUO の問合せ能力, 情処論, Vol. 36, No. 4, pp.959-970, 1995.
- (12) G. W. Furnas : Generalized fisheye views, Proc. of ACM CHI'86, pp.16-23, 1986.
- (13) M. Sarkar, M. H. Brown : Graphical Fisheye Views of Graphs, Proc. of ACM CHI'92, pp.83-91, 1992.