ユビキタス環境における情報変化過程の トレーサビリティのためのデータベースモデル

大木 康幸† 有澤 博‡

†(株)日立製作所 ビジネスソリューション事業部 〒212-8567 川崎市幸区鹿島田 890
‡横浜国立大学 大学院 環境情報研究院 〒240-8501 横浜市保土ヶ谷区常盤台 79-7
E-mail: †ookiyasu@itg.hitachi.co.jp, ‡arisawa@ynu.ac.jp

あらまし ユビキタス環境では無数の自律機器が動的な構成の下で計算を行う。この際、その計算が、いつどこで誰が生産・加工・流通した情報を用いたのかという、情報変化過程に対する記録・経路検索・復旧が難しくなる。本稿では、情報変化過程の蓄積・検索・復旧を行うデータモデルの素案を述べる。本モデルでは、情報が変化していく様を関係鎖という二項関係のリスト構造で表す。複数の情報による相互変化を、関係鎖の同期関係として表す。検索には並行同期型の正規表現を用い、復旧には逆関係による打消しを用いる。また、*traceable*, *recoverable* という変化記録の分析概念を述べる。本モデルにより、通信・流通経路、計算過程、時間経過、メディア列など、様々な情報変化を一元的に蓄積・検索・復旧するための形式的分析が可能となる。

キーワード マルチメディア DB, ユビキタス・コンピューティング, アクティブ DB, 時間 DB, 半構造データ

A Trace Data Model of Ubiquitous Information Morphing

Yasuyuki OOKI† and Hiroshi ARISAWA‡

† Business Solution Systems Division, Hitachi, Ltd. 890 Kashimada, Saiwai, Kawasaki, Kanagawa, 212-8567 Japan
‡ Graduate School of Environment and Information Sciences Yokohama National University
79-7 Tokiwadai, Hodogaya-ku, Yokohama, 240-8501 Japan
E-mail: † ookiyasu@itg.hitachi.co.jp, ‡ arisawa@ynu.ac.jp

Abstract

We propose a trace data model of information morphing for ubiquitous computing. We use a binary relation as various kind of information modification, a chain of the relations as a sequence of changes, and synchronized chains as changes of interrelated information. Application of inverse relation makes simple rollback of change sequence, and matching of continuous value such as time and media information. We define a method of path-finding from trace of information morphing, as composite operation between automaton-like relation graphs and inverse relation. We also introduce *traceable, recoverable* properties so as to optimize the trace recording and monitoring.

Keyword Multimedia DB, Ubiquitous Computing, Active DB, Temporal DB, Semi-structured Data

1. はじめに

1.1. ユビキタス環境での情報変化トレーサビリティの重要性

ユビキタス・コンピューティング環境(以下、ユビキタス環境)が現れつつある[1]。次のような特徴がある。

(1) 多様な自律機器が環境中に多数埋没している。
(2) 複数の機器が総体となって一つの計算を行う。
(3) これら機器の状態や構成は時々刻々と変化する。

ユビキタス環境の実現のため、Grid や Web Service [2]、センサ・ネットワークやストリーム・データ管理システム[3]などの技術が提案されている。

無数の機器が動的構成の下に計算を行うようになると、どの情報が、いつどの機器により計算され、どのネットワークを経由したかという計算過程が複雑となる。特に、機器故障やウィルス発生の際、その問題原因の発見、経路の特定、さらに状態の復旧は困難となる。本稿では、このような複数機器による一連の情報変化過程を記録・追跡(検索)・復旧を可能にする トレーサビリティ管理用データモデルを提案する。

1.2. 既存手法の限界

これに関する既存手法としては、以下に示すものがある。しかし、これらの手法はトレーサビリティ管理用データモデルとしては不十分である。

まず実行時ログ機能がある。しかし、この機能は通常、単体機器の動作記録や、それらの単純な寄せ集めである。どの機器がどの機器の計算結果を用いたかという情報が明示

的ではない。このため迅速な問題経路発見、復旧の用途には向かない。複数機器にまたがる計算履歴を矛盾なく追跡・復旧できるログの設計手法が必要である。

次に RDB のトランザクション＆ロールバック機能や、GUI のコマンド履歴＆アンドゥ機能[4,5]が挙げられる。これらは、いつどの計算機から何の操作をしたかを記録し、操作の取消しに用いる。しかしトランザクションの場合、その復旧単位はRDBテーブルに限定されており、多様な情報の変化を扱えない。コマンド履歴は多様な情報を扱えるが、開発技法[4]としての性格が強く、情報変化の依存関係分析や検索、集約演算などの、形式的手法が確立されていない。

その他、情報が変化していく様を表現し分析する手法としては、古くからペトリネット[6]やオートマトン[7]が研究されている。これらでは、機器のネットワーク構成(設計図)をまず人手で設計してから、その上で変化を分析する。しかし、ユビキタス環境では、ネット構成自身が、環境の変化に合わせて時々刻々と変化する。つまり、機器のネット構成(の変化)は、環境自身(の変化)との相互作用として表される。オートマトンやペトリネットには、ネット構成自体(の変化)を演算対象とし、かつ、それらを集合的に記録・検索・再現・復旧するという、データベース用モデルとしての観点が弱い。

1.3. 情報変化過程のデータモデルの提案

ユビキタス環境でのトレーサビリティ実現には、動的構成の複数機器による多様な情報演算にも対応できる、情報変化過程のデータモデルが必要である。このモデルの諸性質を整理できれば、複数の情報変化過程の効率的な記録や検索、および、障害復旧における影響範囲の最小化などの応用が期待できる。従来の実行時ログ技術に対する設計・評価手法の一つとしても活用できる。

我々は、情報変化過程を、情報の変化差分に着目してモデル化する。多様な情報加工による変化を二項関係として抽象化し、一連の情報変化を、関係鎖と呼ぶ二項関係のリストで表現する。二項関係での抽象化により、数値や座標、画像、音声波形などの情報加工に対する経路検索や復旧を形式的に統一できる。複数機器の連動は、複数関係鎖の並行同期関係で表す。変化過程の復旧は、関係鎖と逆関係の合成による打消しにより定義する。変化過程の経路検索は、関係鎖の集合に対する並行同期型の正規表現による照合として定義する。また変化過程の基本的性質として *traceable, recoverable* という分析概念を導入する。

1.4. 論文構成

以降、二章では関係鎖のモデルと、モデルが持つ基本的性質を述べる。三章では、関係鎖のデータベースへの記録や取得・再現、および、並行同期型の正規表現による関係鎖の検索手法を概説する。四章では関係鎖の演算の定義とプログラムを示す。五章では考察として別研究分野との関係を述べる。六章では、まとめと残課題を述べる。

2. 情報変化過程のデータモデル

2.1. データモデルに対する要件とアプローチ

前章をまとめると、ユビキタス環境の情報変化過程のためのデータモデルの実現には、次が必要であると考える。

(1) ユビキタス環境中の一連の変化を総体として表現
  (a) 多様な情報の変化の形式的表現
  (b) 複数機器の連動による情報変化の表現
  (c) 機器構成の変化の表現
(2) 情報変化過程のデータモデル同士の代数演算
(3) 情報変化過程の記録、再現、検索(追跡)、復旧
(4) データベースとしての集合的操作(変化の一元管理)
(5) 効率の良い実装方法の提示、または示唆

我々のアプローチは、多様な情報の変化を、二項関係として抽象化し、代数的な操作を可能とする点にある。一連の変化を二項関係のリストとして表し(要件 1a)、機器連動を同期関係として(1b)、機器構成変化をこれらの集合の変化として(1c)表す。この代数構造を用いて、(2)(3)(4)を実現する。実装方法(5)は、検討中ではあるが、考察として分散記録の手法を述べる。

本章では、本モデルの基本となる(1a)から(1c)を述べる。

2.2. 多様な情報の変化のモデル化

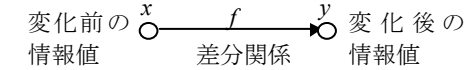
本モデルでは、情報の変化を、図 1a のように、変化前の情報(x)と変化後の情報(y)の間の差分(f)として抽象化する。本稿では、(f)を二項関係と考え、$x\ (f)\ y$ や $f(x)=y$ と表す。情報変化過程は、図 1b のように、この二項関係を鎖状につないだ $x\ (f)\ y\ (g)\ z$ というデータ構造として表現する(関係鎖と呼ぶ)。関係鎖のうち、自明な項は省略できる。

なお以降、説明用として、加算による差分関係を($+dx$)と、「変化なし」の関係を加算($+0$)や関数(1)と記述する。

これら演算は情報の種類ごとに定める(後述)。また、二項関係(f)に対しては逆関係(f^{-1})を定義できる。

図 1b は温度センサ機器 X の温度値の変化である。温度値は連続的に変化するが、本例では最小変化量 3℃で量子化して記録している。このように、差分 (f)は値の連続性[12]やその変化量や補完関係を表現するものとなる。

(a) 情報変化 $M = (x\ (f)\ y) \Leftrightarrow y=f(x)$



(b) 温度センサ $X=(18\ 12\ 15\ 19) = (18\ (-6)\ (+3)\ (+4)\ )$

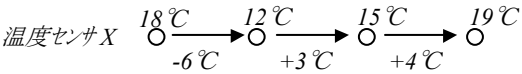


図1 関係鎖の定義と例

2.3. 複数機器が連動した場合の情報変化過程

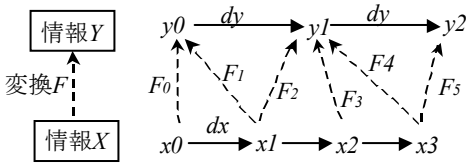
本モデルでは、複数機器が連動した場合の情報変化過程は、二つの関係鎖 X,Y の間の二項関係(F)として表す。

ユビキタス環境の自律機器 X,Y はそれぞれ関係鎖 $X,\ Y$

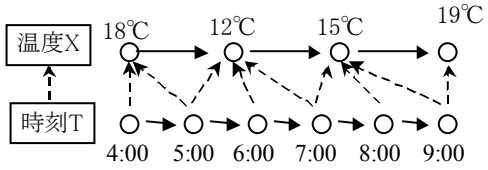
を持つ。機器 *X,Y* の情報の間に、*Y=F(X)*なる連動関係(または参照関係)がある場合、本モデルでは次のように表現する。機器 *X,Y* のどちらか一方の情報が変化した場合、関係 *Fi=(xi,yi)*というリンク、つまり関係を追加する。これを繰り返すと図 2a のような並行同期型の関係鎖ができる。なお、可能なら、*F* に逆変換 F^{-1} を定義しても良い($Fi^{-1}=(yi,xi)$)。

図 2b にセンサ *X* とクロック *T* とによる時間サンプリング例を、図 2c にマウス操作 *M* と計算過程 *P* による連動例を示す。

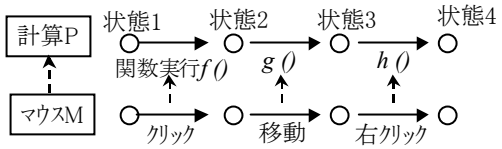
(a) 機器の連動関係 *X(F)Y* ⇔ 変換 *Y=F(X)*



(b) 同期関係鎖: 温度センサ値の時間サンプリング



(c) マウス操作Mと計算処理過程Pのログ記録



(d) 集合表記*A (R) {X,Y}* ⇔ *A (R) X ／ A (R) Y*

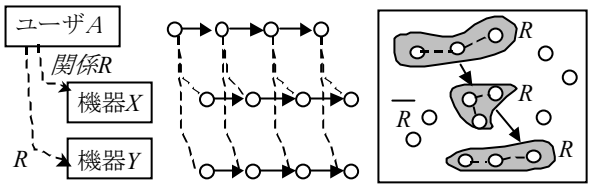


図2 複数情報の相互変化過程のモデル

ここで、関係 *F* 自体の変化として、*F0* から *F1* へ変化を *F0 (dF) F1* と表記する。たとえば *F0=(x0,y0), F1=(x1,y0)*の場合 *dF=(+dx,+0)*とする。*dF* が取りうる値は*(+0,+0), (+dx,+0), (+0,+dy), (+dx,+dy)*がある。

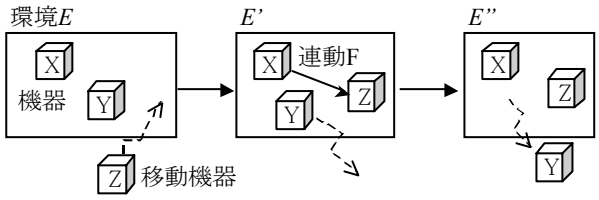
図 2d は 3 機器以上への一般化である。この場合、ユーザ *A* との関係 *R* を満たす機器 *X,Y* が変化していくことを表している。(説明のため *A,X,Y* が完全に同期して動く例を用いた)

2.4. 機器構成自体の動的変化過程

ユビキタス環境では、機器の増減や、機器の接続構成の変更が動的に起こる。本モデルでは、ユビキタス環境そのものを、関係鎖 *X* や関係鎖間の関連 *F* からなる集合 *E* として表し、関係鎖の追加や削除を差分*(dE)*で表す。

図 3はユビキタス環境 *E* 中の機器の構成が変化する様を表す例である。初期状態 *E* には機器 *X,Y* のみ存在する。次に移動体機器 *Z* が環境に侵入し、機器 *X* と通信を行う(*E'*)。*X* と *Z* の通信が終わると同時に機器 *Y* が環境から退出する(*E''*)。本モデルでは、*Z* の侵入と *X* との通信開始は*(dE')= E'-E = { Z, F }*と表す。同様に *Y* の退出は*– {Y}*として表す。

(a) ユビキタス環境での機器構成の動的変化



(b) 動的構成変化の変化過程モデル

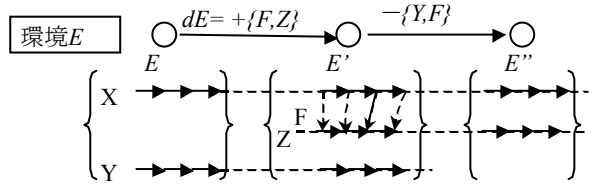


図3 機器構成の動的変化過程

2.5. 情報変化過程の定義

関係鎖の定義を以下に示す。

(0)表記方法 ***x (f) y ⇔ f(x)=y***

$f: X \to Y, \quad x \xrightarrow{f} y, \quad x \xrightarrow{f} y$

(1)関係合成(推移律)

f:X→Y , g:Y→Z のとき、g f : X→Z , g(f(x))=z

(2)結合律 ***(h g) f = h (g f)***

(3)恒等関係 ***f 1_X = 1_Y f = f*** ***1_X:X→X, 1_Y:Y→Y***

なお、二項関係には、逆関係を定めても良い。

(4)逆関係 ***f f^{-1} = f^{-1} f = 1***

特に誤解がない場合、説明用に次の略記を用いる。

(i)加算表記 $x+dx=y \Leftrightarrow x\ (+dx)\ y$

$x,dx,y \in G,\ (+dx): G \to G,\quad 1_G=(+0),\quad (+dx)^{-1}=(-dx)$

(ii)無名関係 *(a,b) ⇔ a (f) b , f:A→B, a∈A,b∈B*

(iii)集合表記 *a (f) {x,y} ⇔ a (f) x ／ a (f) y*

(iv)値 *x,y* と関係*(f)*の略記と、関係鎖への自動変換

(関係略記) *(a b) ⇔ a (f) b, f=(a,b)*

(値略記 1) *(a (f)) ⇔ a (f) f(a)*

(値略記 2) *((f) b) ⇔ f^{-1}(b) (f) b*

2.6. 情報変化過程の基本的性質

ここでは情報変化過程の基本的性質について述べる。

(1) 情報変化の二項関係の定義例

表 1 のように、様々な情報の加工について、変化の二項関係と逆関係を定義することができる。

変化 *a(f)b* を記録する実装方法としては、変化前の情報の場所 a_{id} と、変化前の値 a_{val}、変化後の値 b_{val} を用いることが多い[4,8]。本モデルでは、変化鎖の性質を分析できるよう、便宜的に変化後の変数位置 b_{id} や

変化差分情報 f_{Δ} も示せることにする。

(i) 変化記録レコード $T(f) = [a_{id}, a_{val}, f_{\Delta}, b_{id}, b_{val}]$

ここで、差分関係(*f*)に関する性質として、次を定める。

(ii) ***(a-)追跡可能(a-traceable)*** ⇔ a_{id} ***が非ヌル***

(iii) ***(a-)復旧可能(a-recoverable)***⇔a_{val} ***を導出可能***

⇔ ***(a-)追跡可能*** ∧

(a_{val} ***非ヌル*** ∨(b_{val} ***非ヌル***∧f_{Δ}^{-1} ***定義あり)***)

*(a-)*追跡可能とは、現在の値 *b* からその情報源 *a* の位置が分かることを示す。*(a-)*復旧可能とは、更に *a* の値を復旧できることを示す。これらはペトリネットの可達性(reachability)の概念[6]を、トレーサビリティ管理用に細分化したものと考えてよい。

表1 本モデルで適用可能な情報変化過程の例

情報	二項関係	逆関係	モデルの用途
数値	＋*x*(加算)	○ *–x*	時間,温度センサ,音声波形,株価
	×*p*(乗算)	○ *1/p*	確率(ﾏﾙｺﾌ過程)
情報全般	関数適用 *f*	△ f^{-1}	Turing 機械, ﾌﾟﾛｸﾞﾗﾑ, ﾘﾀﾞｸｼｮﾝ
	二項関係 *r*	△ r^{-1}	Is-a 継承,XML
ﾃｰﾌﾞﾙ	RDB 更新	○rollback	ﾄﾗﾝｻﾞｸｼｮﾝ
集合	∪集合和	○-*A 削除値*	DB ｸｴﾘ, ｽｷｰﾏ
文字列	連結	×	文字列,ｵｰﾄﾏﾄﾝ
条件式 *θ*	∧論理積	○¬ *θ*	属性文法(後述)
画像	画像合成	×	MPEG 映像
座標 P	+ (ﾍﾞｸﾄﾙ)	○ *–P*	GPS 軌跡
IP ｱﾄﾞﾚｽ	転送(hop)	△ 逆 hop	IP traceroute
関係鎖 X	*F* 同期	△ F^{-1}	同期,置換関係

(2) 変化鎖への追加と 関係の合成演算

本モデルでは、変化鎖*(a (f))*の後に変化*(g)*を加える操作を、関係の合成演算と考える。つまり *a(f)(g)= a (f g)*と考える。また、合成関係の追跡可能性を定める：

(iv) ***a(fg)b* での*(a-)traceable*** ⇔ ***f,g* 共に*(a-)traceable***

ユビキタス環境では、様々な機器が計算に参加してくる。情報変化過程の *traceable* を保証するには、新たに計算に参加してくる演算 *f* や *g* が *traceable* であることを確認する必要がある。

(3) 逆関係(f^{-1},*-dx*)に関する基本的性質

変化関係 *f* が逆関係 f^{-1} を持つ場合、種々の情報に対して RDB ロールバックのような情報復旧をモデル化できる $(f^{-1}(f(x))=x)$。一般に、合成関係の逆関係は、$(f\ g)^{-1}=g^{-1}f^{-1}$ であるので、関係鎖 *A=(a (f)(g) b)*の逆関係は、$A^{-1}=(b\ (g^{-1})(f^{-1})\ a)$ となる。

また、*(f g)*の *recoverable* も *traceable* と同様に定義できる。

(v) ***a(fg)b* での*(a-)recoverable*⇔*f,g* 共に*(a-)recoverable***

traceable や *recoverable* などの変化鎖の諸性質は、ログの圧縮などに役立つ。情報変化過程の *traceable* は保証するが *recoverable* は保証不要な場合、変化前の値 a_{val} の記録は不要である。また図 1b のセンサ値変化の場合、変化前の情報の場所 a_{id} と変化後の場所 b_{id} は同一であり、*traceable* の保証だけならば、複数の変化記録レコード ***T(f)*** は必要なく、一つで十分である。他にも、ユーザ限定や *traceable* 期間限定など、様々な*'X'-traceable* を定義すれば、より効率的な実行時モニタリングを実現できると考える。

3. 情報変化過程の記録、再現、検索

ここでは、情報変化過程の記録、再現、検索の方法について、直感的な説明を行う。本モデルでは、記録、再現、検索を、差分と積算によって統一的に表現できる。

3.1. 情報変化過程の記録

図 4は、情報変化過程をユビキタス環境 *UW* 中から計算機 *DB* に記録する手法の概念図である。ユビキタス環境 *UW* には無数の情報機器 *A,B,...X,Y,Z* が埋没し、自律的に情報を変化させている。本モデルでは、情報変化過程の記録を、「環境 *UW* のうち、ある範囲 *Q* を観察し、その中の情報変化を切り出し、転写媒体 *DB* に転写記録する」ことと考える。

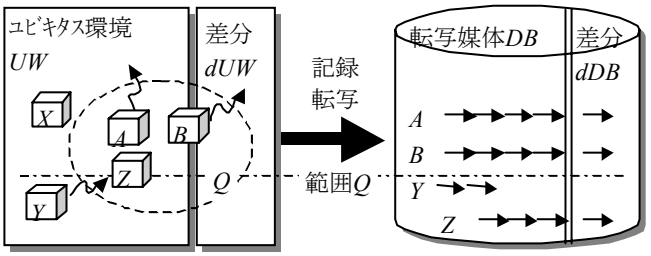


図4 ユビキタス環境 *UW* から転写媒体 *DB* への記録

まず、*UW* のうちの観察範囲 *Q* を切り出し、*DB* に入れることを、従来手法のように関数合成として定義する[9]。

$$DB = Q \cdot UW$$

本モデルでは、環境 *UW* の変化 *dUW* に応じて、*DB* に差分 *dDB* を記録する。

$$(DB + dDB) = (Q+dQ) \cdot (UW+dUW)$$

今回は、環境変化時 *dUW* に対して、範囲 *Q* は変動しないとする。すると *dQ=0* となり、

$$= Q \cdot (UW+dUW) = Q \cdot UW+Q \cdot dUW$$
$$= DB + Q \cdot dUW$$
$$\therefore \quad dDB = Q \cdot dUW$$

となる。 この差分を次のように積算すると *DB* を得られる。

$$DB = \int Q \cdot dUW = 0+dDB+dDB+\ldots+dDB = \int dDB$$

=プログラム： On イﾍﾞﾝﾄ UW 変化(dUW) {

dDB=Q(dUW); DB.更新(dDB); }

このように、環境変動 *dUW* の列が入力として定まれば、*DB* への転写記録を実現できる。

現在、実装方式として分散記録方式を検討中である。この場合、*DB=DB1+DB2*、*UW=UW1+UW2* と分割し、

dDB=dDB1+dDB2、*dUW=dUW1+dUW2*として分割記録する。しかし、*dDB1,dDB2* 連携における *traceable* 保証方法など、分割における諸性質を検討する必要がある。

3.2. 情報変化過程の再現

図 5は、媒体 *DB* に記録された情報変化過程の再現の例である。*DB* 中の検索範囲 *Q* に属する *A,B* を切り出し、関数 *F* で加工し、それを新しい環境 *UW'*の機器に反映させている。記録の場合と同様に、*UW'=F・Q・DB* と定め、*DB* の差分 *dDB* により *UW'= ∫F・Q・dDB* として積算処理を行う。

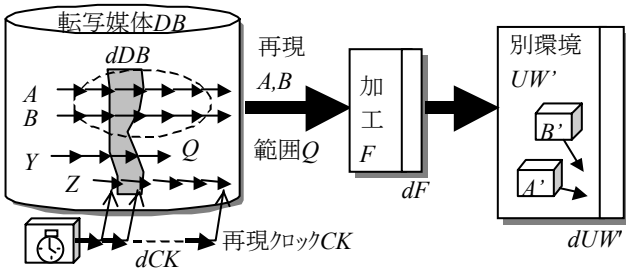


図5　転写媒体 *DB* からの再現，検索，加工，出力

再現が記録と異なる点は、再現クロック *CK* の存在である。転写媒体 *DB* は、前述の *UW* とは異なり、指示がない限り状態は変化しない(*dDB=0*)。従って再現の際は、*DB* 外部に再現クロック *CK* を用意し、再現クロックの変化 *dCK* を用いて *DB* の入力刺激 *dDB* を発生させている。

直感的には、次のような操作となる。まず、関係鎖 *CK* と *DB* を繋げるために、二項関係(/)を用意し、

dDB/dCK　『ある *dCK* 発生による、*DB* に起こる変化 *dDB*』

と表記する。また、二項関係(・)を

(dDB/dCK)・dCK = dDB

『「*dCK* なら *dDB*」において *dCK* が発生』=『*dDB*』

と定義する。これにより、先の *UW'*は次のように導出できる。

UW' = ∫F・Q・dDB　= ∫F・Q・(dDB/dCK)・dCK

= ﾌﾟﾛｸﾞﾗﾑ: for(CK=0;終了迄;次の dCK) {

dDB =「dDB/dCK」表の dCK 欄;

UW' += F(Q(dDB)); }

次章にて、この *dDB/dCK* の性質を具体的に定義する。

3.3. 情報変化過程の検索

本モデルにおける情報変化過程のデータベース、およびその検索は、次のように考える。図 6のように、データベース *P* は、複数の関係鎖 *pi* が集まった集合として表す(*P={pi}*)。次に検索とは、次の3段階により定義される。

(1) 1関係鎖 *p,q* の照合　*(p=q)*　(注:同期関係鎖を含む)

(2) 1関係鎖 *I* と、集合 *P*(データベース)の照合　*(I∈P)*

(3) 複数の関係鎖の集合 *Q, P* の照合　*(Q⊆P)*

1 関係鎖 *p,q* の照合は、次のように定義する。

p=q　⇔　a(f)b = x(g)y　⇔　(a=x)∧(f=g)∧(b=y)

つまり、同じ始点 *a,x* から始まり、同じ経路 *f,g* を通り、同じ終点 *b,y* に辿りついた物を同一と考える。

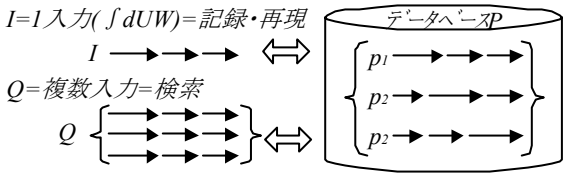
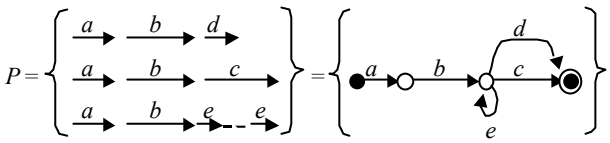


図6　本モデルによるデータベース検索

次に、1 関係鎖 *I* と、関係鎖の集合 *P*(データベース)の照合を考える。本モデルでは、関係鎖の集合は、図 7のように、データベース内で統合されて、一種のオートマトンを構築する。入力関係鎖 *I* は、このオートマトン *P* と照合される。

本手法が既存のオートマトンと異なる点は、照合対象が文字列だけではなく、抽象化された情報変化の列である点である。つまり、連続数値や、条件式、画像、DB 更新ログなどが対象となる。また、同期関係鎖を用いて、音声と映像など、複数種類の情報が連動しあう環境変化も照合できる。



P={ (a)(b)(d), (a)(b)(c), (a)(b)(e)(e)...} = { a b (d|c|e) }*

図7　関係鎖の集合によるデータベース表現

我々は、この関係鎖オートマトンのために、新しい照合方式を提案する。図 8は画像と時間の同期関係鎖からなる集合 *P* と、入力関係鎖 *I* の照合手順を示す。

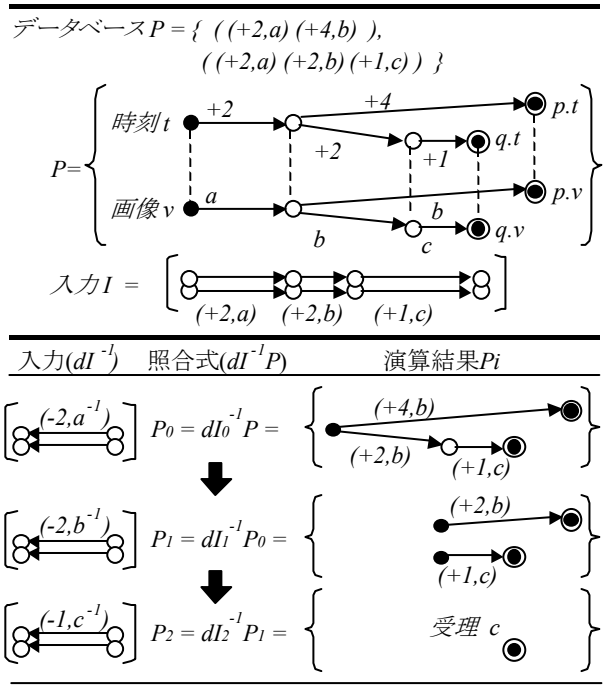


図8　連続値変化と並列同期を含む関係鎖の照合

本照合方式では、入力関係鎖 I の先頭要素 dI を取り、それと集合 P の先頭要素と照合する。照合は dI の逆関係 dI^{-1} を $dI^{-1}P$ のように左に掛け合わせることにより行う。もしこれらが同一なら、$f^{-1}(f\ g\ h)=(1\ g\ h)=(g\ h)$のように、関係鎖の先頭要素がなくなる。これを入力 dI が続くまで行う。もし照合が成功すれば、$(h^{-1}g^{-1}f^{-1})\ (f\ g\ h)=1$ となり、入力が受理される。これを直感的に書けば、

$$I=P \quad \Leftrightarrow \quad 1=I^{-1}P \quad \Leftrightarrow \quad 0=\int(-dI+dP)/dCK \cdot dCK$$

となる(dCK は I と P を走査するための外部クロック)。最後に、これを図 8のように集合 P との演算に拡張すれば、データベースの照合 $I \in P$ が可能になる。

同様の手法により、複数の関係鎖の集合 Q と、データベース P の照合($Q \subseteq P$)も可能となる。

なお、本モデルでは実用のための補助として、関係 R を用いた曖昧照合も定義する。照合対象が音声や画像の場合、一般に完全一致 $dI=dP$ は難しい。このため「大体似ている」という関係 R を用意し、$dI(R)dP$ として照合を行う。この場合、$R(dI)=dP$ であるので、先の照合式を$(dI^{-1}R^{-1})\ P$ と変更するだけでよい。なお、このような値の曖昧照合 R に対し、時間伸縮など、関係鎖の長さの曖昧照合は、後述する正規表現の繰り返し表記 p^*により行うことができる。

3.4. 検索例

図 9に検索例を示す。本モデルの検索言語は現在開発中であるため、図式で表記する。

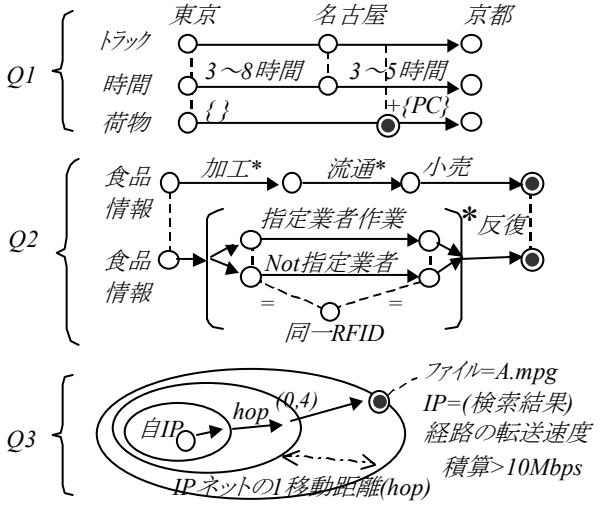


図9　本モデルでの検索例

問合せ Q1 は、「東京～名古屋～京都に至るトラックの中で、荷物としてPCを積み込んだ位置と時間を求める」である。問合せ *Q2* は、「食品の加工・流通・販売において、各過程は、指定業者の作業か、または、指定業者でなくとも加工前後の無線タグ ID 値が同一である」ものである。問合せ *Q3* は、P2P 型ファイル検索の例である。ネットワークを 0 から 4 回移動(hop)しながら、その先に「A.mpg」というファイルを持つ端末を探している。検索の結果、端末の IP アドレスだけでなく、そのネットワーク経路と、その経路の転送速度が最低 10Mbps であることを保証している。

4. 情報変化過程における演算操作と比較･検索

4.1. 関係鎖上の演算の定義

関係鎖同士の演算や関係を、以下のように定義する。

(1)関係鎖の一致 (または受理)

$$a(f)b = x(g)y \quad \Leftrightarrow \quad (a=x) \wedge (f=g) \wedge (b=y)$$

(2)関係 R による関係鎖の一致

$$(a(f)b)\ (R)\ (x(g)y) \quad \Leftrightarrow \quad R(a)=x \wedge (R\ f)=g \wedge R(b)=y$$

ここで、情報同士の比較 $a=x,\ b=y$ は各情報の上で定めた定義、つまり、従来のデータモデルを用いる。

(1')関係同士の一致 $f=r \Leftrightarrow fr^{-1}=1$

$$(f\ g\ h)=(r\ s\ t) \Leftrightarrow (f\ g\ h)\ (t^{-1}s^{-1}r^{-1})=1$$

(3)(関係鎖中の)関係連結 $(f)\ (g\ h)=(f\ g)\ (h)=(f\ g\ h)$

関係削除(左削除, 走査) $(f^{-1})\ (f\ g\ h)=(g\ h)$

関係削除(右削除, 復旧) $(f\ g\ h)\ (h^{-1})=(f\ g)$

関係削除(一般形) $(f\ g\ h)\ (k^{-1})=(f\ g\ (h\ k^{-1}))$

また、慣用表記として次の演算を定める。

(4)微分作用 ∂ $P=a\ (f)(g)(h)(i)\ c$ のとき、 $\partial P=f$

(5)積分作用 $\int$ $P=a\ (f)(g)(h)(i)\ c$ のとき、 $\int \partial P=c$

4.2. 関係鎖の集合、データベース、検索パターン式

前述のように、本モデルでは、関係鎖の集合は一種のオートマトンであると同時に、データベースや検索パターン式でもある。次のように定義する。

(6)関係鎖の集合 $P=X(F)Z=E \cup F$

X = 開始点(情報)の集合、 Z = 受理点(情報)の集合

F = 二項関係の集合、 E = 情報の集合 = $X \cup Z$

(7)パターン式 = 関係鎖集合 P の代数的表現

ワイルドカード型パターンも含め、表 2 のように定める。

(1')照合一致(または受理)

$$x(f)z \in X(F)Z \quad \Leftrightarrow \quad (x \in X) \wedge (1 \in f^{-1}F) \wedge (z \in Z)$$

(3')関係連結(集合に対する)

$fP = \{ f\ g \mid \forall g \in F$,かつ$(f\ g)$が定義されている時 $\}$

その他は表 2 のように定める。

(4')微分(集合に対する)

$\partial P = \{ \partial f \mid f \in F \}$とし、その他は表 2 のように定める。

同様に、関係鎖の集合同士の演算を定義する。

(1")包含 (または DB クエリ)

$Q=A(G)B$, $P=X(F)Z$ のとき、

$Q \subseteq P \Leftrightarrow A \subseteq X \wedge \{1\} \subseteq G^{-1}F \wedge B \subseteq Z$

(3")関係連結(集合同士) $PQ = \cup fQ \quad (\forall f \in P)$

(3"')関係右削除 (P の P_2 による情報変化過程の復旧)

$$PP_2^{-1}=(P_1P_2)P_2^{-1}=P_1$$

表2 検索パターン式に対応する各種演算

#	意味	パターン式	集合表現	オートマトン表現	関係連結 $f\,P$	微分 ∂P	逆関係 P^{-1}
1	*常遷移*	1 または P^0	{ 1 }	◉ または $\xrightarrow{1}$◉	{ f }	*1*	*1*
2	*連結*	$P\,Q$	$P\,Q$	$\xrightarrow{1}$ P $\xrightarrow{1}$ Q $\xrightarrow{1}$◉	$(f\,P)\,Q$	∂P	$Q^{-1}\,P^{-1}$
3	*選択*	$P\|Q$	$P \cup Q$	$\xrightarrow{1}$ P $\xrightarrow{1}$◉, $\xrightarrow{1}$ Q $\xrightarrow{1}$◉	$(f\,P) \cup (f\,Q)$	$\partial P \cup \partial Q$	$P^{-1}\|Q^{-1}$
4	*反復*	P^n	$PP\cdots P$ (*n 個*)	(*省略*)	$(f\,P)\,P^{n-1}$	∂P	$P^{-n}=P^{-1}\cdots P^{-1}$ (*n 個*)
5	*任意反復*	$P^{(n,m)}$ $P^*=P^{(0,\infty)}$	$\cup P^i$ ($i=n \sim m$)	◉→ P ($n \sim m$回) →◉	$\cup (f\,P^i)$	$\cup\ \partial(P^i)$	$P^{(-n,-m)}=(P^{-1})^{(n,m)}$ $P^{-*}=(P^{-1})^*$
6	*同期,置換*	$P(Q)R$	{ Q,P,R }	P →◉ 常時 Q Q Q成立 R →◉	$(f\,P)\,(Q)\,(f\,R)$	$\partial P\,(Q)\,\partial R$	$R^{-1}(Q^{-1})P^{-1}$
7	*任意*	_ (*下線*)	{ $f \mid \forall f \in F$}	(省略) 常に遷移	{ $f\,g \mid \forall g \in F$}	_	$_{}^{-1}=\{ f^{-1} \mid \forall f \in F\}$

(8)積集合 (*P,Q* で一致する経路の切り出し)

$$\underline{\boldsymbol{P \cap Q}} = \underline{\boldsymbol{P_X(P_F)P_Z \ \cap\ Q_X(Q_F)Q_Z}}$$
$$= \underline{\boldsymbol{P_X \cap Q_X\,(P_F \cap Q_F)P_Z \cap Q_Z}}$$

4.3. 記録・検索の方法

DB に蓄積されている関係鎖の集合 P と、入力される関係鎖 I との照合方法は図 10のようになる。

```
照合状態 Rest;  照合記録 Rec;
受理状態  S = enum {未開始,照合中,受理,未受理};
OnInit ( ) {
   Rec←{1};    Rest←DB;     S←未開始;
}
OnInput(dI ) {
    if (S=未開始)
        if ( ∫dI⊆DB.開始 entity 集合) S←照合中
    else if ( ∂(Rest)=dI ) {
        Rest←dI^-1 Rest;    Rec←Rec dI;
        if ( {1}⊆Rest ) { S←受理 として終了; }
   } else
          S←未受理 として終了;
}
```

図10 蓄積関係鎖 *DB* と入力関係鎖 *I* の照合方法

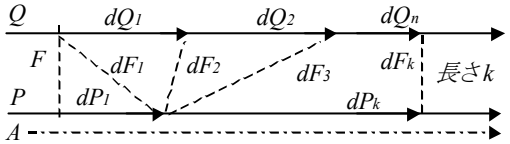
これを関係鎖データベースへの記録手法に用いるには、照合記録 *Rec* を記録媒体に蓄積し、*DB* の照合パターンを「_*」(全てに照合可能)とすればよい。同様に、関係鎖データベースの検索に用いるには、入力関係鎖 *I* に検索したいパターン式を指定し、*DB* に検索対象となる関係鎖の集合を指定すればよい。この場合、検索結果は *Rec* に入る。

4.4. 再現用クロックの選定方法

前章のように、蓄積済みの関係鎖 *DB* を外部クロック *CK* で再現する際、1 クロック発生時 *dCK* における *DB* の変化量 *dDB/dCK* に留意する必要がある。図 11は、二つの関係鎖 *P,Q* が *F* により連動している例である。このとき、*dQ/dP* を、

$$dQ/dP = F(dP) = \{\, dQ \mid \forall dF=(dP,dQ) \in F \,\}$$

のような、*dP* により定まる関数とする。この関数は通常、図11の dQ/dP_1 のように多値関数となる。



$dF_1=(dP_1,+0)$ × $dQ/dP_1 = \{+0,\ dQ_1,\ dQ_1dQ_2\}$
$dF_2=(dP_1,dQ_1)$ ◎ $dQ/dP_k = \{dQ_n,\}$ 一意に決定
$dF_3=(dP_1,\ dQ_1dQ_2)$

図11 並行関係鎖の対応

このため再現に用いる外部クロックには、すべての $dP \in P$ において$|dQ/dP|$=1、つまり、*dP* に対し *dQ* が常に一意となる必要である。例えば、タイマー機器などの時刻値を *P* とし、機器 *Q* を完全同期させる等の方法がある。

その他の解決方法としては次がある。この *dQ/dP* の非決定性の問題は、(厳密ではないが)言語の構文解析における生成規則 $dP_1 \rightarrow \varepsilon \mid dQ_1 \mid dQ_1dQ_2$ の曖昧性の問題と考えることができる。したがって、解決策としては、*P* にて長さ *k* の先読みを行い、$dQ/dP_k=\{dQ_n\}$のように一意に決定させる方法(*LL(k)*文法[7])、および、別の関係列 *A* を *P* と組合せて *dQ* を一意に決定させる方法がある(属性値主導文法[16])。

本モデルは形式文法と異なり、文字列以外にも、時刻などの連続値や、確率積算値、画像メディアなども照合できるため、応用用途が広い。例えば、*P* を時間軸、*dP* を 1 秒サンプリングとした場合、1 秒経った(dP_1)だけでは *Q* の次の動作(*dQ*)は決定できないが、しばらく *k* 秒間、各機器の動きを見れば($dP_1dP_2\cdots dP_k$)、次の行動($dQ_1dQ_2\cdots dQ_n$)を決定できる、などの柔軟な機器連動の記述が可能になる。

5. 考察

ここでは、先に挙げなかった別研究分野との関係を述べる。

(1) 関係モデル、XML、半構造データでの検索[10]

本モデルは、関係や正規表現検索を用いる点では関係モデルや XML、およびその検索手法[10]と同様である。だ

が、本モデルでは数値や画像列の連続性や補完を扱うために、関係同士の合成演算を用意している。また、本モデルでは並行同期関係についての検索ができる点も異なる。

(3)時間に関するモデル[11,12]

時制モデル[11]や移動情報モデル[12]などの時間モデルは、時刻や座標などの数値列、またはユーザ操作などのイベント列のどちらか一方をモデルの基盤とする。本モデルでは、数値列もイベント列も統一して「関係」として扱う演算系を用意している。これにより、音声波形(連続)と音声文法(離散)を混在させた検索パターンを自然に記述できる。

(4)Active DB[13]などの変化イベント監視機構

本モデルは Active DB と同様に、既存のデータモデルに別の側面(動的挙動)を提供する。本モデルでは、Active DB のように一回の更新操作だけでなく、複数機器の一連の更新を対象としたパターン検索ができる。また本モデルでは Active DB とロールバックの概念を自然に融合できる。

(5)ストリーム・データ管理システム(DSMS)[3]

各機器が連鎖的にデータ処理を行う点は DSMS と同様である。しかし、更に本モデルは、その計算過程を記録し、必要に応じて、経路を辿る検索やその復旧を可能にする。経路の動的変化の記述、および、経路自体の性質の保証や分析を行う点が DSMS と異なる。また DSMS では一定のウィンドウ幅をバッファとして用意するが、本モデルでは *traceable* など経路の持つ性質(目的)を用いて、必要最低限のバッファを用いた効率的な動作モニタリングができる。

(6)並行文法[14,15]、属性文法[16]

複数の入力を持つ並行文法として[14,15]などが提案されている。しかし、これらは主に文字列を対象とした照合である。一部数値計算は可能だが、それは計算実行用モデルである。本モデルのように蓄積や検索、復旧などのデータベース用モデルとしての演算操作はできない。本モデルは、また、文法と属性値の連動という意味では属性文法とも関係が深い。しかし、属性文法の解析進行軸は文字列だけである。本モデルでは、外部クロックの導入により、任意の情報変化を基軸にした照合操作が可能である。今後は、並行文法で研究が進んでいる通信プロトコルの仕様記述や分析の手法を、本モデルにも取り入れていきたい。

6. まとめ

本稿では、ユビキタス環境中の多様な情報変化の過程をありのままに記録・検索・復旧できる基本的なデータモデルを提案した。情報変化過程を表現するために、情報の変化を差分関係のリストである変化鎖を用いて表現する。また、複数の変化鎖の並行同期関係や、変化鎖の追加・削除などの基本演算を定義した。最後に、変化鎖の集合をオートマトン形式のデータベースとして表現し、複数の変化鎖の検索や、その記録・再現方法について述べた。

本モデルは、従来のオートマトンや XML 半構造データなどとは異なり、文字列やタグ名の一致照合だけでなく、離散値・連続値を交えた検索、画像や音声波形の類似照合、および複数の同期データ列の伸び縮みや反復・抜けなどを検索できる。また、本モデルでは、記録・検索と DB ロールバックとを自然に統合するため、これらを組合せた最適化を考え易い。また簡単にではあるが、既存の実行時ログに対する設計手法への活用として *traceable, recoverable* という分析・設計の概念を示した。

本モデルはまだ基本モデルであり、検討課題も多い。今後は、記録・検索方法の最適化、変化過程の復旧の影響範囲の特定、および、完全性・健全性の検証などに取り組む予定である。

参考文献

[1] M. Weiser, "The Computer for the 21st Century," Scientific American, 265 (3), pp.94-104, 1991.

[2] I. Foster, K. Kesselman, et al. "The physiology of the grid: An open grid services architecture for distributed systems integration," Technical report, Globus Project, 2002.

[3] D. Abadi, D. Carney, et al., "Aurora : a new model and architecture for data stream management," VLDB Journal, Vol. 12, No. 2, Aug. 2003

[4] E. Gamma, R. Helm, et al., Design Patterns, Addison-Wesley Pub. Co., USA, 1995.

[5] 田幡, 有次, 金森, 半構造データモデルによる画像処理履歴の管理, 情報処理学会論文誌 データベース, Vol.41, No.SIG01(TOD5), pp.64-75, 2000.

[6] 村田, ペトリネットの解析と応用, 近代科学社, 東京, 1995.

[7] J.E.Hopcroft, J.D.Ulman, 言語理論とオートマトン, サイエンス社, 東京, 1971.

[8] ISO/IEC 15938-1 Information technology - Multimedia content description interface (MPEG-7) - Part 1: Systems, 2002

[9] P. Buneman, R. Frankel, et al., "An Implementation Technique for Database Query Languages," ACM ToDS, Vol.7, No.2, pp.164-186, Jun. 1982.

[10] S. Boag, D. Chamberlin, et al., "XQuery 1.0: An XML Query Language," W3C Working Draft 12, Nov. 2003 http://www.w3.org/TR/xquery/

[11] E. Bertino, E. Ferrari, "Temporal Synchronization Model for Multimedia Data," IEEE ToKDE, Vol.10, No.4, pp.612-631, Jul. 1998.

[12] P. Agarwal, L. Guibas, et al., "Algorithmic Issues in Modeling Motion", ACM Computing Surveys, Vol.34 No.4, pp.550-572, 2002.

[13] N. Paton, O. Diaz, "Active Database Systems," ACM Computing Surveys, Vol 31, No 1, pp.63-103, 1999.

[14] 山下,中田, "文脈自由文法の拡張とそれに基づく計算モデル", 情報処理学会 ソフトウェア基礎研究会報告 No.15-5, 1985

[15] G. Paun, L. Polkowski, A. Skowron, Parallel communicating grammar systems with negotiation, Fundamenta Informaticae Vol. 28, No.3-4, 1996

[16] 大木,平見,中川,山下,中田, "拡張1パス型属性文法に基づくコンパイラ生成系の実現," 情報処理学会プログラミング 研究会, No.17-2, pp. 9-16, 1994.