

版管理機能をもつデータベースシステムにおける 視覚的問い合わせ言語の開発

井伊克益* 北川博之†

* 筑波大学理工学研究科 † 筑波大学電子・情報工学系

CADシステムやソフトウェア開発を支援するデータベースにおいて、その設計データの版管理を行なうことは重要である。当研究グループでは版管理データモデル TVDM (Temporal Version Data Model) を提案し研究を進めている。本稿では、版の導出関係や時間情報などの履歴情報に関する問い合わせの条件を図式表現を用いて視覚的に記述できる問い合わせ言語として、TVQE (Temporal Version Query by Example) を提案する。TVDM の概要について触れ、次に、TVQE の設計と実装について報告する。

Development of a Visual Query Language for Database Systems with Version Management

Yoshimi Ii* and Hiroyuki Kitagawa†

*Master's Degree Program in Sciences and Engineering, University of Tsukuba

†Institute of Information Sciences and Electronics, University of Tsukuba

In database management of engineering design applications, version management is necessary. TVDM (Temporal Version Data Model), which we have proposed, is a version management data model which manages version derivation relationship and temporal information. In this paper, we propose TVQE (Temporal Version Query by Example), in which the user can describe historical conditions visually. We mention an outline of TVDM, then we report design and implementation of TVQE.

1 はじめに

CAD システムやソフトウェア開発等において、設計データベースの重要性が認識されている。ここで扱われる設計データは、設計の進展、製品系列の拡充などの要因によって、時間の経過に伴い様々な版(バージョン)が作成される。従って、設計データの複数の版を記録し、必要な版の検索を可能とするための版管理は、設計データベースに必要不可欠な機能の一つである [1]。

当研究グループでは、版管理データモデル TVDM (Temporal Version Data Model) を提案し研究を進めている [2]。TVDM はリレーショナルモデルの枠組の中で版管理を行うことを目的として提案されている。リレーショナルモデルの枠組での版管理を行う方法でまず考えられるのが、各版の生成時刻、削除時刻、親の版、子の版等を表す基本的な属性を付加させる方法である。版管理に関する問い合わせ記述では、SQL や QUEL 等の通常のリレーショナルデータベース言語記述の中でこれらの属性に対する検索条件を記述する。しかし、この基本的属性を付加する方法では、導出関係や時間的生成順序関係に関して版の間を辿るような検索条件や、時刻を指定する時制データベース的な条件指定は、付加された属性に関する基本的条件の組み合わせとして記述され、問い合わせ記述が複雑化しがちである。また、ある版から導出された全ての子孫の版を求めるような推移的閉包 (transitive closure) の検索も版管理では必要であるが、通常のリレーショナルデータベース言語では記述することが出来ない。

TVDM では、上記のような版管理用の基本的属性を付加する代わりに、抽象データ型である「履歴型」として版の導出関係と時間情報を一括して管理する。TVDM では各種データ操作を行なうためのデータベース言語として、TV-Quel (Temporal Version Quel) を提案している。「履歴型」に付随する各種オペレータを TV-Quel 中で用いて、版の導出関係と時間情報を組み合わせた検索条件をより直接的に記述することを可能とする。また、このような履歴型のオペレータとして、版管理上必要な推移的閉包を実現できる。

TV-Quel では、導出関係・時間情報といった履歴情報に関する条件記述がより直接的に可能である。特に、使用する履歴型のオペレータの少ない時に有効である。しかし、オペレータの数が多くなり、オペレータを数段に入れ子にする時に、条件記述が複雑になる場合がある。このような場合に、導出関係・時間的生成順序関係等を 2 次元的な図式を用いて記述し、視覚的な理解を促すことが有効であると考えられる。そこで、履歴情報に関する問い合わせの条件を視覚的に記述できる言語として、TVQE (Temporal Version Query by Example) を提案する。TVQE はリレーショナルデータベース言語である QBE (Query by Example) をベースに、履歴情報に関する条件記述を行う機構を拡張する形で設計している。

本稿の第 2 章では、この研究の背景に当たる TVDM と TV-Quel について説明する。第 3 章では、TVQE を提案し、その概要について述べる。第 4 章では、TVQE のプロトタイプの実装について述べる。第 5 章では、今後の課題を述べる。

2 TVDM (Temporal Version Data Model)

2.1 TVDM

一般に、設計データベース中の設計データは、設計の進展などにより、1 つの設計対象に複数の版が生じる。以下では、ある設計対象に対応する版の集合を版集合 (version set) と呼ぶ。図 1 は、ある実体型「システム」の“Sys-A”に関する版導出の課程を時間軸に沿って表したものである。図中のノードが版を表し、上へ向かう矢印が版の導出関係を表す。例えば、“Sys-A”第 2 版は時刻 t_2 の時に第 1 版から導出されている。各版集合は版集合識別子 (version set identifier) により一意に識別される。また、ある版集合中の各版は版番号 (version number) により識別される。従って、版集合識別子と版番号により、各版を識別することが出来る。

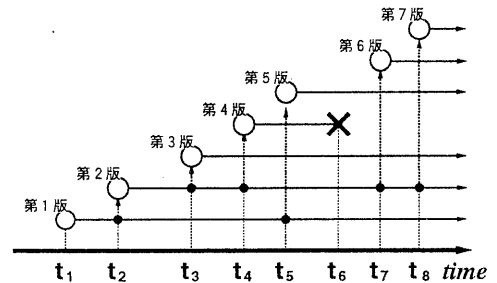


図 1: システム “Sys-A” の版導出関係

このような複数の版を持つ設計データにおいては、1 つの版に対し異なった複数の改訂を行なうことで、その導出関係は枝分かれを持った木構造をとる。これらの版管理では、「昨年 6 月以前に設計された版を元に改訂した版は？」などの時間情報と導出関係を組み合わせたデータ操作が要求される。これらの要求を満たすため、TVDM では抽象データ型である「履歴型」を導入し、これを用いて版の導出関係と各版の生成時刻・削除時刻等の時間情報を管理する。

TVDM でのデータ表現では、単純値の集合を直接属性値として記述可能な入れ子型リレーションを用いる。版とその履歴に関する情報は版リレーション (version relation) と履歴リレーション (history relation) のペアでモデル化する (図 2)。版リレーションは各版に付随する情報を 1 タプルとして表現する。版リレーションには各版の属性記述のためのユーザ定義の属性に加え、各版の属する版集合の版集合識別子を表す属性 ($_vsetid$) と版番号を表す属性 ($_vno$) を追加する。履歴リレーション

は、1つの版集合中の版の導出関係と時間情報を1タプルとして表現する。履歴リレーションは、版集合識別子のための属性(.vsetid)と履歴情報を格納するための「履歴型」の属性(.history)からなる。

版リレーション SYSTEMS

_vsetid	_vno	sname	spec	designer
1	1	Sys-A	spec-a1	Tom
1	2	Sys-A	spec-a2	Tom
1	3	Sys-A	spec-a3	Tom
1	4	Sys-A	spec-a4	Marry
1	5	Sys-A	spec-a5	Jone
1	6	Sys-A	spec-a6	Marry
2	1	Sys-B	spec-b1	Michael
2	2	Sys-B	spec-b2	Michael
:	:	:	:	:

履歴リレーション _SYSTEMS

_vsetid	.history
1	→
2	→
3	→
:	:

図2: 版リレーションと履歴リレーション

履歴型は1つの版集合に属する版の導出関係と時間情報を表すデータ構造とその操作のためのオペレータを一体化した抽象データ型(ADT)である。履歴情報の参照用オペレータとしては、CREATED_TIME(版の生成時刻)、DELETED_TIME(削除時刻)、PARENT(親の版)、CHILD(指定した時刻での子の版)等がある [2]。また、PARENT、CHILD等では、それぞれ推移的閉包をサポートしている。

2.2 TV-Quel

TVDMでは、データ検索やデータ更新を記述するデータベース言語としてTV-Quelを提案する。TV-QuelはPOSTQUEL[3, 4]をベースに設計されている。TV-Quelでは、導出関係と時間情報を組み合わせた問い合わせの記述を行うために、検索用コマンドのretrieveのターゲットリストや条件節中に履歴型の参照用オペレータを記述可能である。例えば、「“Sys-A”第6版の生成時刻において存在する、“Sys-A”第6版の兄弟の版の版番号とその版の設計者を求めよ」という問い合わせをTV-Quelで記述すると以下ようになる。

```
retrieve(S._vno,S.designer)
from S in SYSTEMS, H in _SYSTEMS
where S.sname="Sys-A"
and S._vsetid=H._vsetid
and S._vno ∈ CHILD(H._history,PARENT(
H._history,6),CREATED_TIME(H._history,6))
and DELETED_TIME(H._history,S._vno) >
CREATED_TIME(H._history,6)
```

また、版リレーションと履歴リレーションの更新のためのコマンドとしてvcreate(版リレーションの生成)、

vfirstversion(初版の登録)、vderive(新しい版の導出)等が定義されている [2]。

3 TVQE(Temporal Version Query by Example)

TV-Quelでは履歴型のオペレータを用いて、時間情報と導出関係を組み合わせたい問い合わせを可能とした。しかし、時間情報・導出関係の複雑に入り組んだ条件を伴う問い合わせを記述するためには、履歴型のオペレータを数段に入れ子にする必要があり、記述が複雑になることがある。そこで、時間情報・導出関係に関する検索条件を、図式を用いて視覚的に記述することで問い合わせを行うことを可能とする言語TVQE(Temporal Version Query by Example)を現在設計開発中である。TVQEは、関係データベース言語であるQBE [5, 6]をベースにしている。

3.1 TVQEの構成

TVQEは、QBEをベースに、時間情報・導出関係を組み合わせたい検索条件を視覚的に記述するHistorical Condition Boxを加えた拡張として設計されている。TVQEの構成を、図3に示す。Output Skeleton、Relation Skeleton、Condition Box、Historical Condition Boxを画面構成要素と呼ぶ。この画面構成要素中で、変数、定数、比較演算子、論理演算子を記述して、問い合わせを行う。

Output Skeleton

Output	属性名	...	属性名
--------	-----	-----	-----

Relation Skeleton

リレーション名	属性名	属性名	...	属性名
---------	-----	-----	-----	-----

Condition Box

--

Historical Condition Box

Version Condition Area
Temporal Condition Area

図3: TVQEの画面構成要素

画面構成要素のそれぞれの機能は以下の通りである。Output Skeleton、Relation Skeleton、Condition Boxの機能は、QBEに準ずる。

- Output Skeleton
問い合わせの結果となるリレーションのスキーマ


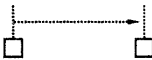
を指定する。上段に属性名、下段に変数、定数、あるいは、それらを含んだ式を記述する。

- Relation Skeleton
リレーション名とその版リレーションの属性名が表示されており、属性名の下に、定数、変数、または、それぞれに比較演算子を適用した式を記述することで、版リレーションの検索条件を指定する。
- Condition Box
論理演算子 \wedge, \vee を用いたやや複雑な検索条件の指定に用いる。
- Historical Condition Box
版の間の時間的生成順序関係や導出関係に関する検索条件の指定に用いる。表示されている時間軸の上を Version Condition Area、下を Temporal Condition Area という。Version Condition Area には、版を表す記号と、導出関係などの版と版との関係に関する条件を記述する。Temporal Condition Area では、時刻を表す定数、変数や、版に関するイベントの時間的順序関係に関する条件を記述する。

Historical Condition Box で記述する記号

Historical Condition Box で履歴情報に関する条件記述に用いる記号を表 1 に示す。

表 1: Historical Condition Box 記述用の記号

1	(1,a)
2	(1,3) → (1,a)
3	(1,3) - - - - -> (1,a)
4	(1,3) → * → (1,a)
5	(1,3) - - * - -> (1,a)
6	(1,a) → ×
7	(1,a) → △
8	
9	

1. 版を表す記号

Relation Skeleton で記述した版リレーションのテーブルに関しての履歴情報に関する条件を記述するため、そのテーブルの版集合識別子と版番号の組として記号で表す。表の例は、版集合識別子が 1、版番号が変数 a の版を表す。

2. 親子関係を表す記号

版の間の親子関係は、対象となる 2 つの版の間に矢印を引くことで表す。表の例では、版集合識別子 1 の版集合の第 3 版から第 a 版を生成したことを表す。

3. 時間的生成順序関係を表す記号

親子関係と同様に、対象となる 2 つの版の間に矢印を引くことで表す。上記の例では、第 a 版が第 3 版の直後に生成されたことを示す。

4. 親子関係の推移的閉包を表す記号

推移的閉包は矢印と "*" を組み合わせて表す。表の例では、第 a 版は第 3 版から幾度かの導出を行って出来た版であることを示す。

5. 時間的生成順序関係の推移的閉包を表す記号

記号 4 と同様、矢印と "*" を組み合わせて表す。表の例は第 a 版は第 3 版よりも後に導出された版であることを示す。

6. 版が削除されたことを表す記号

記号 8 と組み合わせて版が削除された時刻を表す。図 4 に、削除時刻の表現方法を示す。この例では、記号 8 が版集合識別子 1 版番号 a の版の削除時刻を示す。

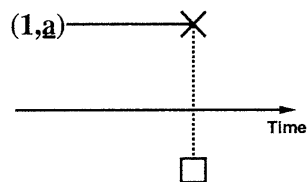


図 4: 削除時刻の表現方法

7. 版がカレントでなくなったことを表す記号

記号 8 と組み合わせて版がカレントでなくなった時刻を表す。図 5 に、カレントでなくなった時刻の表現方法を示す。この例では、記号 8 が版集合識別子 1 版番号 a の版のカレントでなくなった時刻を示す。

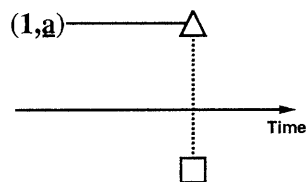


図 5: カレントでなくなった時刻の表現方法

8. 時刻を表す記号

記号の上端には版に関するイベントを表す記号(記号 1、6、7)を、下端の矩形には時刻を表す定数・変数を置き、版に関するイベントの起きた時刻やある特定の時刻を表す。記号 1 を組み合わせた時、記号 8 は版の生成時刻を意味する。図 6 に、生成時刻の表現方法を示す。この例では、記号 8 が版集合識別子 1 版番号 a の版の生成時刻を示す。

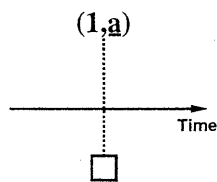


図 6: 生成時刻の表現方法

記号6を組み合わせた時、記号8は版の削除時刻を意味する(図4)。記号7を組み合わせた時、記号8は版がカレントでなくなった時刻を意味する(図5)。特定のある時刻、例えば1994年7月20日等、を表すには、上端には何もつなげずに、矩形にその時刻を表す定数を記述する(図7)。

9. 時刻の順序関係を表す記号

時刻を表す記号8同士をつないで矢印の指す先の時刻の方が時間的に後であることを示す。図7に、時刻の順序関係の表現方法を示す。この例では、版集合識別子1版番号aの版の生成時刻が1994年7月20日より早いことを表す。

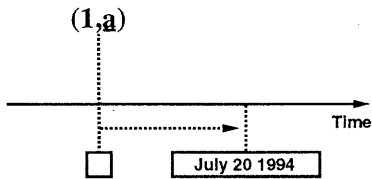


図7: 時間の順序関係の表現方法

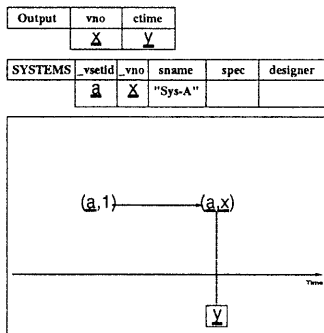
記号8の矩形と記号9は Temporal Condition Area に記述する。

3.2 TVQE による問い合わせ

以下では、TVQE による問い合わせ記述の例を示し、TV-Quel による問い合わせ記述と比較する。

Q1. “SYSTEMS” の “Sys-A” の版番号1の版の子供の版の版番号とその生成時刻を求めよ。

• TVQE による記述

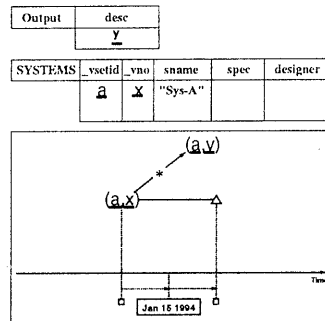


• TV-Quel による記述

```
retrieve(vno=S._vno,ctime=
  CREATED_TIME(H._history,S._vno))
from S in SYSTEMS, H in _SYSTEMS,
where S.sname="Sys-A" and
  S._vsetid=H._vsetid and
  S._vno ∈ CHILD(H._history,1,now)
```

Q2. 1994年1月15日においてカレントであった “Sys-A” の版の子孫の版の版番号を求めよ。

• TVQE による記述

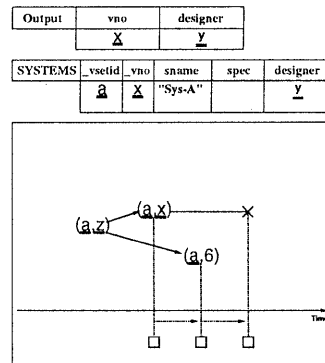


• TV-Quel による記述

```
retrieve(desc=RCCHILD(H._history,
  CURRENT(H._history,
    "Jan 15 1994"),now))
from S in SYSTEMS,H in _SYSTEMS
where S.sname="Sys-A" and
  S._vsetid=H._vsetid
```

Q3. “Sys-A” 第6版の生成時刻において存在する、“Sys-A” 第6版の兄弟の版の版番号とその版の設計者を求めよ。

• TVQE による記述



• TV-Quel による記述

```
retrieve(S._vno, S.designer)
from S in SYSTEMS, H in _SYSTEMS
where S.sname = "Sys-A" and
  S._vsetid = H._vsetid and
  S._vno ∈ CHILD(H._history,
    PARENT(H._history,6),
    CREATED_TIME(H._history,6)) and
  DELETED_TIME(H._history,S._vno) >
    CREATED_TIME(H._history,6)
```

以上の例のように、TVQEとTV-Quelによる問い合わせ記述を比較すると、時間情報と導出関係を組み合わせた複雑な検索条件が、TVQEではより直観的な理解のしやすい形で表現できることが分かる。

4 プロトタイプシステムの実装

現在、TVQEのプロトタイプシステムの実装を行っている。図8にTVQEプロトタイプシステムの構成を示す。

今回の実装では、拡張可能関係DBMS POSTGRES [3, 4]を用いてTVDMに基づくデータ管理を行う。POSTGRESでは、組み込みのデータ型に加え、新たにユーザが定義したデータ型を属性のドメインとして用いることが可能である。これを用いて「履歴型」のデータ構造を定義する。またPOSTGRESには、関数の定義の機能があるので、これを用いて「履歴型」専用のオペレータを作成した。また、POSTGRESのインターフェイスである関数ライブラリLIBPQをベースとして、TVDMのインターフェイスとなるライブラリ関数LIB-TVQを作成した。TVDMに関するデータ定義・操作は、このLIB-TVQ関数を介する形で実行される。

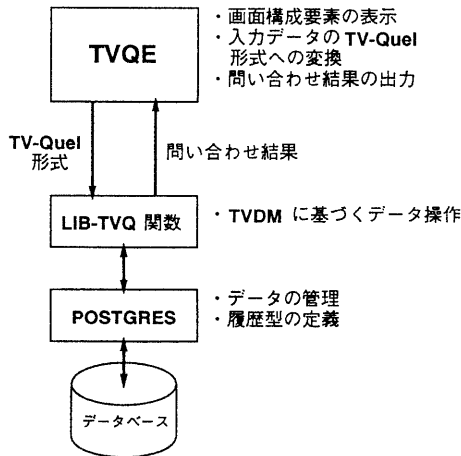


図8: TVQEプロトタイプシステムの構成

また、TVQEの4つの画面構成要素を実現するために、Xウィンドウ・システムのツールキットであるTcl/Tk [8]を用いている。LIB-TVQ関数をTcl/Tkのコマンドとして実行できるように、Tcl/Tkを解釈実行するインタプリタを拡張してある。4つの画面構成要素をTcl/Tkのコマンドで記述した中で、LIB-TVQ関数のコマンドを呼び出す。このLIB-TVQ関数はTV-Quelの構文に沿った設計がなされているため、TVQEの画面構成要素に記述された問い合わせを等価なTV-Quelの問い合わせに変換しなければならない。変換によって作成されたTV-Quelの問い合わせはLIB-TVQを使って実行され、その結果をTcl/Tkにより作られた出力部が画面に表示する。

この章では、Tcl/Tkによる画面構成要素の作成と、TVQEからTV-Quelへの変換について述べる。

4.1 画面構成要素の作成

TVQEの4つの画面構成要素はTcl/Tkを用いて実装した。Tclはアプリケーションに独自のコマンドを追加できるインタプリタ言語、Tkはその上に構築されたXウィンドウ・システムのツールキットであり、これらは、UC Berkeleyで開発された。Tclのプログラムは、複数のコマンドから構成され、各コマンドは以下の形式をとる。

コマンド 引数・・・ オプション・・・

以下で示すTkのコマンドはwidgetと呼ばれるGUIオブジェクトを生成するものの一例である。画面構成要素を構成するメニューやボタンなどはこれらを使って作成する。

- **button**
ボタンを生成する。オプション -command で指定した手続きを呼び出すことができる。
- **entry**
キーボードから文字列を入力する entry widget を生成する。
- **canvas**
図形や文字 (item) を記述・表示する canvas widget を生成する。この item は単に表示するだけでなく、生成・削除や、移動などの修正が出来る。
- **menubutton**
メニューボタンを作成する。
- **menu**
メニューを作成する。

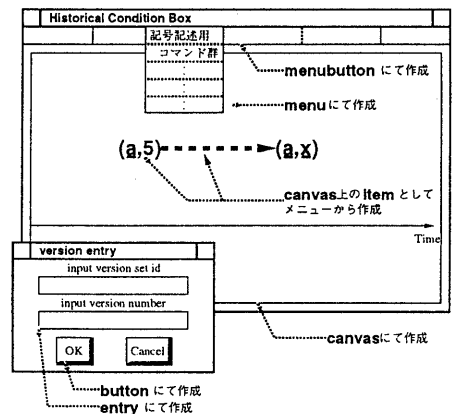


図9: Historical Condition Boxの構成

これらのコマンドを組み合わせることで4つの画面構成要素を作る。図9は、Historical Condition Boxを作成するのに、どのような widget を用いているかを示している。版を表す記号や関係を表す記号は canvas 上の item として描く。これらの記号の記述は、menu と menubutton で作成したメニューからコマンドを選ぶことで行なわれる。版の記号などで文字列の入力が必要な時は、entry widget を含むウィンドウが新たに開き、これを用いて入力する。版や時間の関係の記号はメニューを選んだから、対応する item をクリックすることで矢印が記入される。

Tcl/Tk を解釈実行するインタプリタはアプリケーション独自のコマンドが組み込み可能なので、TVDM のデータ定義・操作を行うインタフェースであるライブラリ関数 LIB-TVQ の各関数を Tcl のコマンドとして組み込んだ。問い合わせの実行を行う button widget に対応する手続き中で、この LIB-TVQ 関数を実行するコマンドが呼び出されている。

図10に、TVQEの画面構成要素の概観を示す。

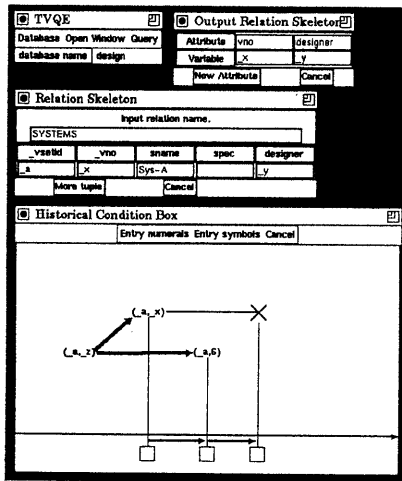


図10: TVQEの画面構成要素の概観

4.2 TVQE から TV-Quel への変換

TVQE で書かれた問い合わせから TV-Quel への変換は以下の規則にしたがって行われる。

1. Output Skeleton から retrieve 節に必要な “属性名” = “変数名” の形の式を生成する。
Q 3 を例に式を生成する。Q 3 の Output Skeleton を以下に示す。

Output	vno	designer
	<u>x</u>	<u>y</u>

上記の Output Skeleton から次の2つの式が生成される。

$vno = x, designer = y$

2. Relation Skeleton から from 節に必要な “tuple 変数” in “リレーション名” の形の式を生成し、where 句の条件に用いる版リレーションと履歴リレーションの対応をつける式 “tuple 変数”._vsetid = “tuple 変数”._vsetid を生成する。Q 3 の Relation Skeleton を以下に示す。

SYSTEMS	_vsetid	_vno	sname	spec	designer
	<u>a</u>	<u>x</u>	"Sys-A"		<u>y</u>

上記の Relation Skeleton から次の3つの式が生成される。

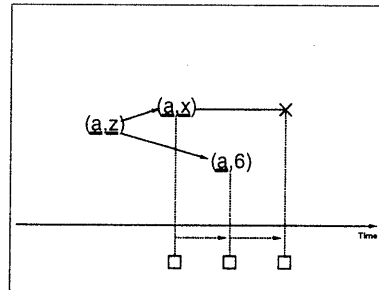
R1 in SYSTEMS, H1 in _SYSTEMS,
R1._vsetid = H1._vsetid

3. Relation Skeleton から、 “tuple 変数”._属性名 = “element 名” の形式の条件式を生成する。

上記の Relation Skeleton から次の4つの式が生成される。

R1._vsetid = a, R1._vno = x,
R1.sname = "Sys-A", R1.designer = y

4. Historical Condition Box 中の版の記号の中でも、2と3の変換を行う。Q 3 の Historical Condition Box を以下に示す。



上記の Historical Condition Box から、版 (a,z) について次の5つの式が生成される。

R2 in SYSTEMS, H2 in _SYSTEMS,
R2._vsetid = H2._vsetid,
R2._vsetid = a, R2._vno = x

また、版 (a,6) についても同様に次の5つの式が生成される。

R3 in SYSTEMS, H3 in _SYSTEMS,
R3._vsetid = H3._vsetid,
R3._vsetid = a, R3._vno = 6

5. Historical Condition Box では、導出関係・時間関係の記号を履歴型のオペレータを用いた条件式に変換する。上記の Historical Condition Box からは次の4つの式が生成される。

```
z = PARENT(H1._history,x),
z = PARENT(H3._history,6),
CREATED_TIME(H1._history,x) <
    CREATED_TIME(H3._history,6),
CREATED_TIME(H1._history,6) <
    DELETED_TIME(H3._history,x)
```

6. ここまでの変換で、TVQEとして記述された内容は TV-Quel の形式に変換されている。しかし、変数として記述されたものがそのまま残っているため、残っている変数を式の置換えで取り除く。

この操作の後、以下の式が完成する。これを LIB-TVQ 関数のコマンドの引数として渡すことで実行される。

```
retrieve(vno=R1._vno, designer=R1.designer)
from R1 in SYSTEMS, H1 in _SYSTEMS,
     R2 in SYSTEMS, H2 in _SYSTEMS,
     R3 in SYSTEMS, H3 in _SYSTEMS
where R1._vsetid = H1._vsetid and
     R2._vsetid = H2._vsetid and
     R3._vsetid = H3._vsetid and
     R1._vsetid = R2._vsetid and
     R2._vsetid = R3._vsetid and
     R1.sname = "Sys-A" and
     R2._vno=PARENT(H1._history,R1._vno)and
     R2._vno = PARENT(H3._history,6) and
     CREATED_TIME(H1._history,R1._vno) <
         CREATED_TIME(H3._history,6) and
     CREATED_TIME(H3._history,6) <
         DELETED_TIME(H1._history,R1._vno)
```

5 おわりに

本稿では、導出関係・時間情報に関する検索条件を視覚的に記述できる問い合わせ言語 TVQE を提案した。また、TVQE の POSTGRES 上での実装についても述べた。

今後の課題としては、次のものが挙げられる。

- データ定義・操作機能の追加
TVQE の現時点での実装では、問い合わせのみ可能である。TV-Quel はスキーマ定義、版の導出・削除等の機能も有する。TVQE にもデータ定義・操作の機能の追加が必要であると考えられる。
- TVQE の表現力についての評価
TVQE の問い合わせは TV-Quel の問い合わせに変換できることから、TV-Quel は TVQE の表現力を包含する。しかし、TV-Quel においては記述

可能な「生成時刻が1992年より前か、あるいは、1995年より後である版を求めよ」といった OR 条件は TVQE では記述できない。このような TVQE では記述できない条件の詳細について検討し、今後の TVQE の改良に利用したい。

- プロトタイプシステムを用いた TVQE と TV-Quel との条件記述の容易さの比較検討
TVQE は TV-Quel の履歴情報に関する条件記述の複雑さを解消するために設計された。そこで、両問い合わせ言語の記述の容易さの比較を行い、これを1つの評価としたい。具体的には、実装したプロトタイプシステムを用いて、実際に被験者に問い合わせを記述させ、使い心地について調査する予定である。
- 時間に関する条件記述の検討
TVQE、TV-Quel 共に現在の実装では、「同じ時刻である」という条件は秒単位まで等しくなければならず、実際の問い合わせにおいて意味のあるものとは思えない。また、「ある期間における」のような time interval は2つの時刻の比較条件を組み合わせて行われており、さらなる記述の簡略化が期待できる。このような、時間に関する条件記述についてもさらに検討が必要である。

参考文献

- [1] R. H. Katz, "Toward a Unified Framework for Version Modeling in Engineering Databases," ACM Comput. Surv.,22(4), 1990.
- [2] 北川博之, 田中 肇, 大保信夫, 鈴木 功, "履歴データ型を用いた版管理データモデルの提案," 情報処理学会論文誌, 34(5), 1993.
- [3] M. R. Stonebraker and L. A. Rowe, "The POSTGRES Data Model," Proc. of 13th VLDB, 1987.
- [4] J. Rhein and G. Kemnitz, "The POSTGRES User Manual," EECS Dept., University of California, Berkeley.
- [5] M. M. Zloof, "Query by Example," AFIPS Conference Proceedings, National Computer Conference 44, 1975.
- [6] M. M. Zloof, "Query-by-example:A database language," IBM Syst.J.,1977.
- [7] G. Özsoyoğlu and V. Matos and Z. M. Özsoyoğlu, "Query Processing Techniques in the Summary-Table-by-Example Database Query Language," ACM Trans. on Database Systems, 14(4), 1989.
- [8] J. K. Ousterhout, "An Introduction to Tcl and Tk," EECS Dept., University of California, Berkeley.