

版管理機能を持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設計, 実装, 評価

井 伊 克 益^{†*} 北 川 博 之^{††}

CAD, ソフトウェア開発等の設計データを管理するデータベースシステムでは版管理の機能が必須である。版管理においては、版の導出関係管理と各版の生成時刻・削除時刻等の時間情報管理の両者が重要である。我々は、このような要求に対応するための版管理データモデル TVDM およびその問い合わせ言語 TV-Quel の研究開発を進めてきた。TV-Quel では導出関係・時間情報に関する問い合わせ条件を、抽象データ型である「履歴型」に付随するオペレータを用いることにより記述可能である。しかし、「履歴型」のオペレータの利用規則に精通しているとは限らない末端利用者向けには、より直観的理解を促すような利用者インタフェースの提供が望ましい。本稿では、このような背景のもとに、リレーショナルデータベース言語 QBE 上に導出関係・時間情報に関する問い合わせ条件の 2 次元的図式表現を組み合わせた、視覚的問い合わせ言語 TVQE を提案する。本稿では、まず TVQE について述べた後、その処理系である TVQE インタプリタの設計と実装について述べる。また、TV-Quel との比較実験に基づく TVQE の有効性の評価について報告する。

Design, Implementation and Evaluation of a Visual Query Language for Database Systems with Version Management Function

YOSHIMI Ii^{†*} and HIROYUKI KITAGAWA^{††}

In management of engineering design databases, version management is indispensable. We have proposed a version management data model named TVDM, which features management of version derivation relationships and relevant temporal information with the history data type. We also proposed its database language TV-Quel. However, descriptions in TV-Quel sometimes become confusing for naive users who are not familiar with the usage of operators associated with the history data type. As an approach to this problem, we propose a visual query language TVQE. TVQE is based on QBE and features visual specification of historical conditions. We describe TVQE, and explain design and implementation of the prototype TVQE interpreter. We also discuss the result of an experiment which compares TVQE and TV-Quel from the viewpoint of user friendliness.

1. はじめに

CAD システム, 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¹⁾. 設計データにおいては, 設計の進展, 製品系列の拡充などの要因によって, 時間の経過にともない様々な版 (バージョン) が作成される. したがって, 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 (バージョン管理) は, 設計データベースに必要な不可欠な機能のひとつである. 版管理においては, ある版がどの版に基づいて生成されたかという版の導出関係や, ある版がいつ生成・削除されたかという時間情報の管理を一体として行うことが重要である.

我々は, このような要求に対応するため, 版管理データモデル TVDM (Temporal Version Data Model) を提案し研究を進めてきた^{2),3),4)}. TVDM は拡張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いる. TVDM は, 抽象データ型である「履歴型」を用いることにより, 版の導出関係と時間情報を一括して管理する点を特徴とする. データベースにおけるデータの時間変化を対象とした研究には時

[†] 筑波大学理工学研究科

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

^{††} 筑波大学電子・情報工学系

Institute of Information Sciences and Electronics, Uni-
versity of Tsukuba

^{*} 現在, シャープ株式会社

Presently with Sharp Corporation

制データベースの研究^{5),6),7)}があるが, TVDMは時制データベース管理を対象としたものではなく, 版管理における版の導出関係と時間情報の管理を主目的としている。

我々は, TVDMでの各種データ操作を行うためのデータベース言語として, TV-Quel (Temporal Version Quel) を提案した²⁾. TV-QuelはPOSTQUEL^{8),9)}をベースとした言語である。「履歴型」に付随する各種オペレータをTV-Quel中で用いることで, 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可能である。

しかし, TV-Quelで版の導出関係や時間情報に関する条件記述を行うためには, 「履歴型」のオペレータの利用規則を正しく理解する必要がある。特に, オペレータの引数の意味や型を理解していることが必要である。このため, オペレータの利用規則に十分に精通しているとは限らない末端利用者向けには, より直観的理解を促すような利用者インタフェースの提供が望ましい。このひとつのアプローチとして, 導出関係・時間情報を組み合わせた問い合わせ条件を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考えられる。

本稿では, このような背景のもとに, 導出関係・時間情報に関する問い合わせ条件の視覚的記述を特徴とする言語, TVQE (Temporal Version Query by Example) を提案する。また, その処理系の設計と実装について説明する。さらにまた, 図式を用いた導出関係・時間情報に関する条件記述の評価を主な目的とした, 利用者の使用感に関するTVQEとTV-Quelの比較評価実験を行い, その結果について報告する。

TVQEはリレーショナルデータベース言語であるQBE (Query by Example)^{10),11)}をベースに, 導出関係・時間情報といった履歴情報に関する図式的な条件記述を行う機構を付加した言語である。これまでに, データベースに対する視覚的問い合わせ言語については多くの研究が行われてきた^{12),13)}。しかし, 本研究の対象である導出関係や時間情報の管理に対応した研究は少ない。QBEをベースにした時制データベースに対する問い合わせ言語の研究としては, Ozsoyogluら¹⁴⁾やTanselら¹⁵⁾によるものがあるが, いずれもQBEの表形式の枠組の中での時間情報記述を対象としており, 本研究におけるような2次元図式を条件記述に用いたものではない。また, これらは時制データベースを対象としたものであり, 本研究の対象である版管理を対象とはしていない。また, 時制データベースのひとつであるタプル列データモデルを対象とした

視覚的問い合わせ言語の提案¹⁶⁾もあるが, QBEの基本概念をベースとしたものではなく, また対象とするデータモデルや問い合わせも本研究とは異なる。

本稿の第2章では, 本研究の背景であるTVDMとTV-Quelについて説明する。第3章では, TVQEの概要について述べる。第4章では, TVQEの実装システムであるTVQEインタプリタの設計と実装について述べる。第5章では, TVQEの利用者から見た使用感の評価のためのTV-Quelとの比較実験について述べる。これまで, 問い合わせ言語の比較評価実験としては, 筆記試験による実験を行った例がよく知られている^{17),18)}。今回の比較評価実験では, 筆記試験による実験に加え, 実装システムを用いた実験についても試みた。第6章では, TVQEの記述力と拡張について考察する。第7章では, 本稿のまとめを述べる。

2. TVDM

図1は設計作業の進捗とそれにもなう版の導出の様子を表した例である。図中の円は版を表し, 上向きの矢印は導出関係を表す。この例では, ENGINE-1という設計対象について, たとえばその第1版に改良を加え, 第2版を時刻 t_2 の時点で導出したことを示す。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。一般に, 版集合中の版の導出関係は枝分かれを持った木構造やDAG構造をとる。版管理においては, 「昨年6月以前に設計された版を元に改訂した版は?」等の導出関係と時間情報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には, 版の導出関係や時間情報といった履歴情報を一体として管理することが必要となる。

TVDM (Temporal Version Data Model) は, このような版管理を考慮に入れた拡張リレーショナルモデルとして提案された²⁾。TVDMでのデータ表現では, 版とその履歴に関する情報は版リレーション (version

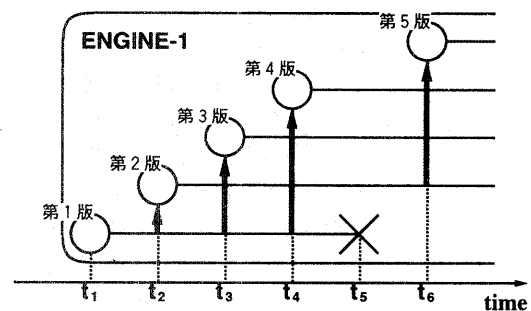


図1 設計の進展と版の生成

Fig. 1 Design evolution and version derivation.

版リレーション
ENGINE

| _vsetid | _vno | power | weight | factory |
|---------|------|-------|--------|---------|
| 1 | 1 | 140 | 250 | Tsukuba |
| 1 | 2 | 155 | 240 | Tsukuba |
| 1 | 3 | 155 | 245 | Tsukuba |
| 1 | 4 | 150 | 240 | Tsukuba |
| 1 | 5 | 160 | 245 | Moriya |
| 2 | 1 | 150 | 270 | Kashima |
| 2 | 2 | 160 | 270 | Kashima |
| 2 | 3 | 155 | 270 | Kashima |
| 2 | 4 | 170 | 280 | Kashima |
| : | : | : | : | : |

履歴リレーション
ENGINE

| _vsetid | _history |
|---------|----------|
| 1 | |
| 2 | |
| : | : |

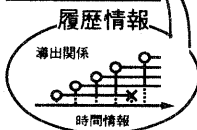


図2 版リレーションと履歴リレーション

Fig. 2 Version relation and history relation.

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を1タプルとして表現する。版リレーションは各版の属性記述のためのユーザ定義の属性を持つ。さらに版リレーションには、各版の属する版集合を表す「版集合識別子型」の属性 (_vsetid) と版番号を表す「版番号型」の属性 (_vno) が自動的に追加され、両者のペアが主キーとなる。履歴リレーションは各版集合を実体としてとらえ1タプルとして表現したものである。履歴リレーションは版集合識別子を表す版集合識別子型の属性 (_vsetid) と履歴情報を格納するための「履歴型」の属性 (_history) からなり、_vsetid が主キーである。

版リレーションおよび履歴リレーションにおける属性 _vsetid, _vno の属性値は、TVDM における版の生成操作に基づきタプルが追加された時点でシステムにより自動的に割り当てられる²⁾。また、利用者からはその参照のみが可能でその値を更新することはできない。版集合識別子型および版番号型の値同士は、二項オペレータ = により等価性の判定が可能である。

履歴型は1つの版集合に関する履歴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である。履歴型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。

我々は、TVDM の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 を提案した²⁾。TV-Quel は POSTQUEL^{8),9)} をベースに設計されている。TV-Quel の検索用コマンドの retrieve では、ターゲットリストや where 節中に上述の履歴型の参照用オペレータを記述することで、導出関係と時間情報を組み合わせた問い合わせを可能にする。

以下に図2のリレーションを対象とした retrieve コマンドの例を示す。なお、コマンド中の “<-” は意味的には “ε” を表す。以下の例では、ターゲットリストに履歴型に付随するオペレータ CREATED_TIME を、where 節に CHILD, PARENT, CREATED_TIME, DELETED_TIME を用いることで履歴情報に関する問い合わせ条件の記述を行っている。

Q. 版集合識別子1のエンジンの第4版の兄弟の版のうち、第4版の生成時刻において存在し、かつ筑波工場 (“Tsukuba”) で開発された版の版番号と生成時刻を求めよ。

```
retrieve (E._vno, ctime =
    CREATED_TIME(EH._history, E._vno))
from E in ENGINE, EH in _ENGINE
where E._vsetid = 1
and E._vno <- CHILD(EH._history,
    PARENT(EH._history, 4),
    CREATED_TIME(EH._history, 4))
and E.factory = "Tsukuba"
and EH._vsetid = E._vsetid
and CREATED_TIME(EH._history, 4)
    < DELETED_TIME(EH._history, E._vno)
```

3. TVQE (Temporal Version Query by Example)

TV-Quel では、履歴型のオペレータを用いることにより履歴情報に関する問い合わせ条件を記述することが可能であるが、その際、履歴型のオペレータの利用規則、特に、引数の意味や型を理解していることが必要である。したがって、オペレータの利用規則に精通しているとは限らない末端利用者向けには、より直観的理解を促すような利用者インタフェースが望ましい。このためのひとつのアプローチとして、履歴情報に関する問い合わせ条件を図式表現を用いて視覚的に記述する方法が考えられる。我々はこのアプローチに基づいた問い合わせ言語 TVQE (Temporal Version Query by Example) を提案する。履歴条件に関する条件記述を図式表現を用いて行ううえでは、全体の言語仕様も TV-Quel のような1次元的なものではなく2次元的であることが望ましい。そこで、2次元なりレシヨナルデータベース言語として最も広く知られている QBE^{10),11)} をベースに、履歴情報に関する視覚的条件記述を行う機能を追加する方針で設計を行った。

図3に TVQE の画面構成要素を示す。この画面構成要素中で、例示要素 (example element)、定数要素

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

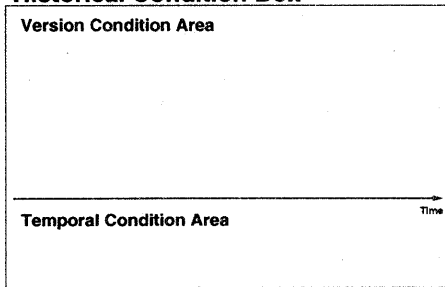


図3 TVQEの画面構成要素
Fig. 3 Visual components of TVQE.

(constant element), 比較演算子, 論理演算子および履歴条件のための記号を記入することにより問い合わせを記述する. QBEと同様, 例示要素は変数に相当し, 下線を引いて表す. 同じ例示要素を複数記述することで, 結合演算や参照関係の記述等を行う. 定数要素は定数に相当する.

画面構成要素のそれぞれの役割は以下のとおりである.

- **Output Skeleton**
問い合わせ結果のリレーションのスキーマを指定する. 上段に利用者が任意に決定する属性名, 下段に例示要素を記述する.
- **Relation Skeleton**
版リレーション名とその属性名が示されており, 属性名の下に, 定数要素, 例示要素, あるいはそれらを含んだ式を記述することで, 版リレーション中のタプルに関する各種条件を指定する.
- **Condition Box**
論理演算子 \wedge, \vee 等を用いたやや複雑な検索条件の指定に用いる.
- **Historical Condition Box**
履歴情報に関する検索条件の図式的記述に用いる. 表示されている時間軸の上を Version Condition Area, 下を Temporal Condition Area と呼ぶ. Version Condition Area には, 版を表す記号と, 導出関係等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

表1 履歴情報に関する条件記述用の記号

Table 1 Symbols to describe historical conditions.

| | | |
|---|----------------------|---|
| 1 | 版を表す記号 | ENGINE (1, <u>a</u>) |
| 2 | 親子関係を表す矢印 | ENGINE (1,3) → ENGINE (1, <u>a</u>) |
| 3 | 時間的直前直後関係を表す矢印 | ENGINE (1,3) - - - - -> ENGINE (1, <u>a</u>) |
| 4 | 親子関係の推移的閉包を表す矢印 | ENGINE (1,3) - * -> ENGINE (1, <u>a</u>) |
| 5 | 時間的直前直後関係の推移的閉包を表す矢印 | ENGINE (1,3) - - * - -> ENGINE (1, <u>a</u>) |
| 6 | 版の削除を表す記号 | ENGINE (1, <u>a</u>) → X |
| 7 | 版が最新でなくなった時点を表す記号 | ENGINE (1, <u>a</u>) → Δ |
| 8 | 時刻を表す記号 | □ |
| 9 | 時間的前後関係を表す矢印 | □ - - - - -> □ |

す記号, 例示要素, 定数要素等を用いて, 版操作に関するイベントの時刻, 特定の時刻, およびそれらの時間的前後関係に関する条件を記述する.

上に述べた4種類の画面構成要素のうち最初の3つの役割は, QBEにおけるものと基本的には同じである. TVQEに特徴的なのは, Historical Condition Box である. Historical Condition Box で履歴情報に関する条件記述に用いる記号の分類を表1に示す. なお, Historical Condition Box 中に書かれた個々の条件はすべて and 条件で連結されたものとして解釈され, or, not, グルーピング, 集約関数等を用いた条件記述は行うことができない.

- (1) 版を表す記号
Relation Skeleton で記述した版リレーション中のタプルに対応する版を表すため, リレーション名 (版集合識別子, 版番号) として表す. 表1の例は, リレーション ENGINE の版集合識別子が1, 版番号が a の版を表す.
- (2) 親子関係を表す矢印
版の間の親子関係は対象となる2つの版を表す記号の間にこの矢印を引くことで表す. 表1の例では, リレーション ENGINE の版集合識別子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) 時間的直前直後関係を表す矢印
親子関係と同様に, 対象となる2つの版を表す記号の間にこの矢印を引くことで表す. 表1の例では, 第 a 版が第3版の時間的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矢印
親子関係の推移的閉包は矢印2と“*”を組み合わせることで表す. 表1の例では, 第 a 版は第3版

を元にして幾度かの導出を行ってできた版であることを示す。

- (5) 時間的直前直後関係の推移的閉包を表す矢印
記号4と同様、矢印3と“*”を組み合わせることで表す。表1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) 版の削除を表す記号
版の削除を表す。記号8と組み合わせることにより版が削除された時刻を表す。図4に、削除時刻の表現例を示す。この例では、記号8がリレーションENGINEの版集合識別子1、版番号aの版の削除時刻を示す。
- (7) 版が最新でなくなった時点を表す記号
版集合中でその版が最新の版でなくなった時点を表す。記号8と組み合わせることにより版が最新でなくなった時刻を表す。図5に、最新でなくなった時刻の表現例を示す。この例では、記号8がリレーションENGINEの版集合識別子1、版番号aの版が最新でなくなった時刻を示す。
- (8) 時刻を表す記号
版操作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の下に記したときは版の生成時刻を意味する(図6の左側の記号8)。記号6の下に記したときは版の削除時刻を意味する(図4)。記号7の下に記したときは版が最新でなくなった時刻を意味する(図5)。特定のある時刻、たとえば1995年3月14日などを表すには、単に矩形内にその時刻を表す定数

要素を記述する(図6の右側の記号8)。記号8はHistorical Condition Box中のTemporal Condition Areaに記述する。

- (9) 時間的前後関係を表す矢印
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6に、記号1, 8, 9を組み合わせる例を示す。この例では、リレーションENGINEの版集合識別子1、版番号aの版の生成時刻が1995年3月14日より早いことを表す。記号9はTemporal Condition Areaに記述する。

図7に、第2章で示したTV-Quelの問い合わせ例をTVQEで記述したものを示す。

TV-QuelとTVQEによる問い合わせ記述を比較して、TVQEは時間情報と導出関係を組み合わせる複雑な検索条件をより直観的な理解を促す形で表現できるように思われる。より系統的な比較実験の結果については、第5章に述べる。

なお、TVQEのベースとなったQBEは記述力に限界のあることが知られている^{19),20)}。このことから、TVQEの記述力はTV-Quelの記述力と等価ではない。この点に関する考察は第6章で行う。

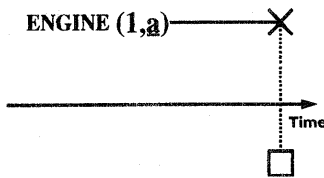


図4 削除時刻の表現例

Fig. 4 Representation of version deletion time.

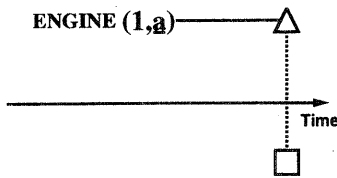


図5 最新でなくなった時刻の表現例

Fig. 5 Representation of time when the version becomes not current.

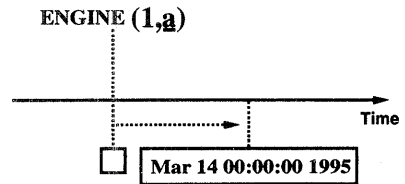


図6 時間的前後関係の表現例

Fig. 6 Representation of chronological order of two time points.

| Output | vno | ctime |
|--------|-----|-------|
| | a | b |

| ENGINE | _vsetid | _vno | power | weight | factory |
|--------|---------|------|-------|--------|---------|
| | 1 | a | | | Tsukuba |

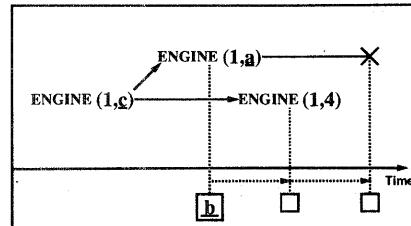


図7 TVQEによる問い合わせ例

Fig. 7 TVQE query example.

4. TVQE インタプリタの設計と実装

本研究では、TVQE の処理系として、TVQE インタプリタの設計と実装を行った。本実装は、TVQE の特徴である Historical Condition Box の実現にポイントを置いた基本的なものであり、QBE がサポートし、TVQE でも本来 Historical Condition Box 以外の部分では記述可能な or, not, グルーピング、集約関数等については現在は実装していない。実装環境は Sun OS 4.1.2, および X ウィンドウシステム V11R5 である。図 8 に TVQE インタプリタの構成を示す。TVQE インタプリタは画面表示部と問い合わせ実行部から構成される。画面表示部は 4 つの画面構成要素と検索結果を表示する。問い合わせ実行部は画面構成要素に記述された TVQE の問い合わせを等価な TV-Quel の問い合わせに変換し、これを実行する。

本システムの実装では、TVDM に基づくデータ管理を行うため、拡張可能関係 DBMS POSTGRES^(8),9) を用いた。POSTGRES では、組み込みのデータ型に加え、新たに利用者が定義したデータ型を属性の定義域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した。また POSTGRES には、関数の定義の機能があり、これを用いて履歴型のオペレータを作成した。また、POSTGRES の C 言語用インタフェースである関数ライブラリ LIBPQ をベースとして、TVDM のインタフェー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。LIB-TVQ 関数は TV-Quel の基本構文に基づいた命令を実行可能である。

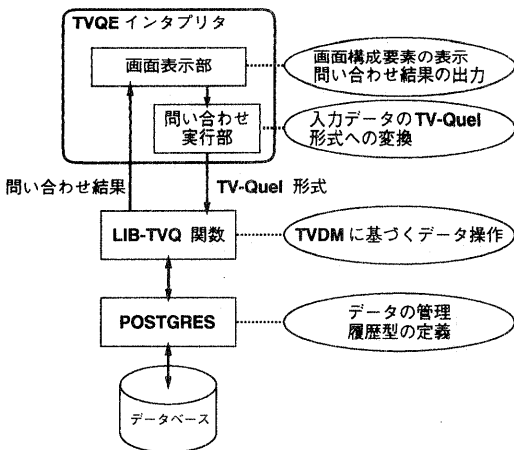


図 8 TVQE インタプリタの構成
Fig. 8 Architecture of TVQE interpreter.

4.1 画面表示部

画面表示部の主な機能は、TVQE の 4 つの画面構成要素を表示し問い合わせ条件を入力することと検索結果を画面に表示することである。画面表示部の実装には Tcl/Tk²¹⁾ を用いた。Tcl はアプリケーション独自のコマンドを追加できるインタプリタ言語であり、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る。本実装では、LIB-TVQ 関数を Tcl/Tk のコマンドとして実行できるように、Tcl を解釈実行するインタプリタの拡張を行っている。画面構成要素は Tk が提供する GUI オブジェクトである“ウィジェット”を用いて実装した。ウィジェットを使用して実装した画面構成要素の例として、Historical Condition Box の構成を図 9 に示す。

4.2 問い合わせ実行部

問い合わせ実行部は画面構成要素に記述された問い合わせを LIB-TVQ 関数の仕様に沿うように TV-Quel の構文に準ずる形に変換し、LIB-TVQ 関数を呼び出す。問い合わせの変換の手順を図 10 に示す。図 11 に TVQE による簡単な問い合わせ例を示す。図 11 の問い合わせについて、変換処理 (8) まで終了した変換結果を図 12 に示す。図 12 の中間結果について、変換処理 (9) を施した最終的な変換結果を図 13 に示す。

図 11 の問い合わせを例として、図 10 の変換手順について説明する。

変換処理 (1) Output Skeleton 中の各列ごとに、TV-Quel のターゲットリスト中の“属性名 = 例示要素”の形式に変換する。図 11 の Output Skeleton は検索結果として 2 つの属性を必要とするので、図 12 のターゲットリストの 2 つの式に変換される。

変換処理 (2) Historical Condition Box の記号 1 ごと

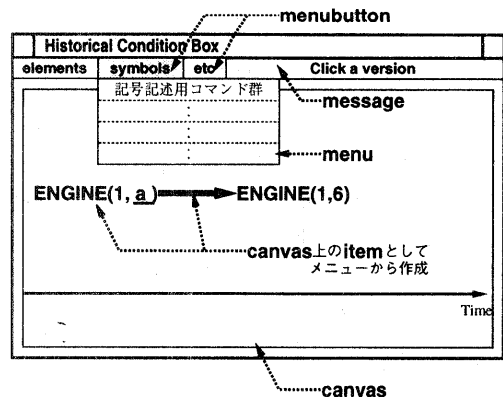


図 9 Historical Condition Box の構成
Fig. 9 Historical Condition Box.

```

for (Output Skeletonの各列について) {
    ターゲットリストに"属性名 = 例示要素"を追加 ... (1)
}
for (Historical Condition Box での各記号1について) {
    版リレーション用タプル変数の割当て ... (2)
    履歴リレーション用タプル変数の割当て ... (3)
    "(2)のタプル変数._vsetid=(3)のタプル変数._vsetid"
    の条件式を where 節に追加 ... (4)
    "(2)のタプル変数._vsetid=記号1の版集合識別子"
    および"(2)のタプル変数._vno=記号1の版番号" の
    条件式を where 節に追加 ... (5)
}
for (各Relation Skeletonについて) {
    for (Relation Skeletonの各行について) {
        版リレーション用タプル変数の割当て ... (6)
        for (Relation Skeletonの各列について) {
            if (定数要素/例示要素の記述がある) {
                "(6)のタプル変数. 属性名=要素"
                の条件式を where 節に追加 ... (7)
            }
        }
    }
}
for (Historical Condition Box での各図式表現について) {
    履歴型オペレータによる条件に変換し,
    where 節に加える ... (8)
}
ターゲットリストと where 節の例示要素の除去 ... (9)
    
```

図10 問い合わせ実行部における変換手順

Fig. 10 Translation procedure in the query execution part.

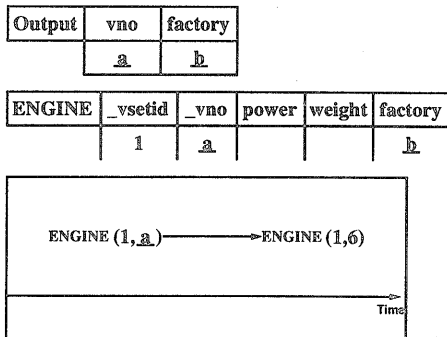


図11 TVQEによる問い合わせ例
Fig. 11 TVQE query example.

とに, from 節に記述されるべき版リレーションのタプル変数を割り当てる. 図11の Historical Condition Box の左側の記号1に対してはタプル変数 V1 が, 右側の記号1に対してはタプル変数 V2 が版リレーション ENGINE のタプルとして割り当てられる.

変換処理(3) Historical Condition Box の記号1ごとに, 対応する履歴リレーションのタプル変数を割り当てる. 図11の Historical Condition Box の左側の記号1に対してはタプル変数 H1 が, 右

```

retrieve(
    vno = a, (1)
    factory = b (1)
)
from V1 in ENGINE, (2)
    H1 in _ENGINE (3)
    V2 in ENGINE, (2)
    H2 in _ENGINE (3)
    V3 in ENGINE, (6)
where V1._vsetid = H1._vsetid (4)
    and V2._vsetid = H2._vsetid (4)
    and V1._vsetid = 1 (5)
    and V1._vno = a (5)
    and V2._vsetid = 1 (5)
    and V2._vno = 6 (5)
    and V3._vsetid = 1 (7)
    and V3._vno = a (7)
    and V3.factory = b (7)
    and a = PARENT(H2._history,6) (8)
    
```

図12 変換処理(8)まで終了した問い合わせ

Fig. 12 Intermediate translation result after the step (8).

```

retrieve(vno = V1._vno, factory = V3.factory)
from V1 in ENGINE, H1 in _ENGINE,
    V2 in ENGINE, H2 in _ENGINE,
    V3 in ENGINE
where V1._vsetid = H1._vsetid
    and V2._vsetid = H2._vsetid
    and V1._vsetid = 1
    and V2._vsetid = 1
    and V2._vno = 6
    and V3._vsetid = 1
    and V3._vno = V1._vno
    and V1._vno = PARENT(H2._history,6)
    
```

図13 変換の完了した問い合わせ

Fig. 13 Completed translation result.

側の記号1に対してはタプル変数 H2 が履歴リレーション _ENGINE のタプルとして割り当てられる.

変換処理(4) (2)と(3)で割り当てたタプル変数を使用して, 版リレーションと履歴リレーションの結合条件として, "(2)のタプル変数 ._vsetid = (3)のタプル変数 ._vsetid" を where 節に追加する. 変換処理(5) Historical Condition Box の記号1ごとに, where 節に "変換処理(2)のタプル変数 ._vsetid = 記号1の版集合識別子" および "変換処理(2)のタプル変数 ._vno = 記号1の版番号" の形式の条件が追加される.

変換処理(6) Relation Skeleton 中の各行ごとに, 変換処理(2)と同様に, from 節に記述されるべき版リレーションのタプル変数を割り当てる. 図11の Relation Skeleton のタプルに対してタプル変

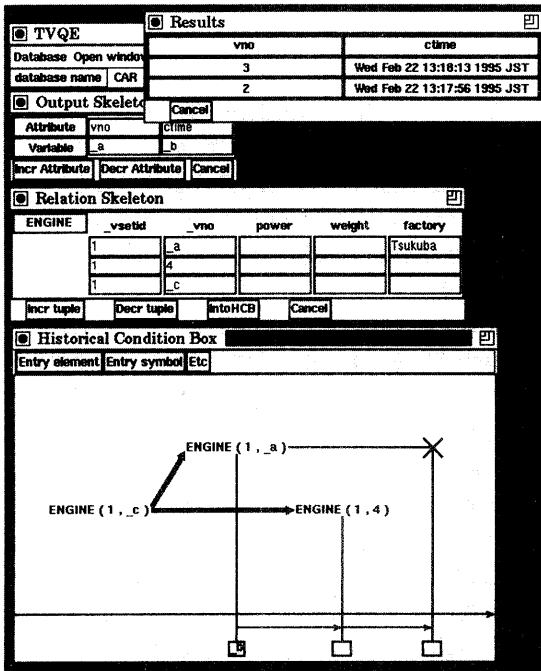


図 14 TVQE インタプリタの実行例

Fig. 14 Session example of TVQE interpreter.

数 V3 が、版リレーション ENGINE のタプルとして割り当てられる。

変換処理 (7) Relation Skeleton 中における定数要素/例示要素の記述された列ごとに、where 節に“変換処理 (6) のタプル変数、属性名 = 定数要素/例示要素”の形式の条件が追加される。図 11 の Relation Skeleton のタプルについて 3 つの条件式が生成される。

変換処理 (8) Historical Condition Box の履歴条件に関する条件を履歴型のオペレータを用いた条件に変換する。図 11 の親子関係を表す記号は、履歴型オペレータ PARENT を用いた条件式に変換される。

変換処理 (9) ターゲットリストと where 節には、まだ例示要素が残っているためこれを消去する。図 12 の 2 つの条件式 “V1. _vno = a” と “V3. factory = b” を使用して、a と b を代入により消去する。

図 14 に実装を行った TVQE インタプリタによる第 3 章で使用した問い合わせの実行例を示す。

5. 比較評価実験

本章では、TV-Quel を比較対象とした TVQE の使用感に関する評価実験について述べる。比較評価実験

表 2 解答時間 (6 問)・採点時間 (54 件) および正解・不正解の割合

Table 2 Summary of experiment.

| | TVQE | TV-Quel |
|------|--------------|---------|
| 解答時間 | 19 分 | 41 分 |
| 採点時間 | 25 分 | 90 分 |
| 正解率 | 完全な正解 | 61.4% |
| | 単純な間違いのみの不正解 | 6.7% |
| | 両者の合計 | 68.1% |
| | | 48.3% |

としては、筆記試験による実験と実装システムを利用した実験の 2 種類を行った。

5.1 筆記試験による比較評価実験

5.1.1 実験方法

筆記試験による比較評価実験では、日本語で与えられた版管理に関する問い合わせ内容を TVQE と TV-Quel で記述する試験を行った。被験者としては、データベースの基礎的知識を有する 20 名の学生を対象とした。筆記試験に際しては、事前に TVQE と TV-Quel、およびその関連事項に関する簡単な解説資料を配布し、各自でそれぞれを事前学習してもらった。学習に要した時間は平均 2 時間 30 分であった。TVQE 6 問、TV-Quel 6 問、合計 12 問の問題を用意した。また、テスト終了後に使用感についてのアンケート調査を行い、正解率、解答に要した時間、採点に要した時間と合わせて、TVQE と TV-Quel の比較評価の尺度とした。TVQE と TV-Quel のそれぞれの問題を同一の 6 問とした場合、先に解いた問題が他方を解くヒントになる恐れがあると考え、対象となるリレーションや必要となるオペレータを変えた、難易度のほぼ等しい 6 問ずつを用意した。また、TVQE 6 問を先に配布・解答する組と TV-Quel 6 問を先に配布・解答する組の 2 組に分けて試験を行った。実験に用いたリレーションを付録 A.1 に示す。また、問題については紙面の都合上 TVQE に関する問題のみを付録 A.2 に示す。

5.1.2 実験結果

● 解答時間

解答時間は「書きやすさ」を表すひとつの尺度であると考えられる。今回の実験では、1 被験者の 6 問に対する解答時間は、TVQE が最短 13 分、最長 27 分、TV-Quel が最短 31 分、最長 52 分であった。いずれの被験者についても、TVQE の解答時間は TV-Quel の解答時間より短く、TVQE の解答時間を 1 としたとき、TV-Quel の解答時間は最小 1.6、最大 2.9 であった。表 2 に被験者 1 人あたりの平均解答時間を示す。平均して、TVQE は TV-Quel の約半分の時間で解答が記述された

ことが分かる。

● 採点時間

採点時間は「読みやすさ」を表すひとつの尺度であると考えられる。表2に9人の被験者の解答(54件)の採点に要した時間を示す。TVQEの採点時間はTV-Quelの3分の1以下の時間であった。

● 正解率

正解率は「理解しやすさ」を表すひとつの尺度であると考えられる。一般に、TV-Quelによる記述量はTVQEによる記述量と比べてかなり多く、単純な書き間違いなどのミスが入り込む可能性が高いと予想される。そこで正解率に関しては、単純な記述ミスのみによる不正解は正解とした場合の正解率についても求めてみた。今回の実験では、すべての被験者について、TVQEの正解率がTV-Quelの正解率を上回った。また、単純な記述ミスのみによる不正解は正解とした場合の正解率についても、TV-Quelの正解率の方が高かった被験者が3名であったのに対し、TVQEの正解率の方が高かった被験者は13名であった。表2に全被験者に対する正解率、単純な記述ミスによる不正解の割合、およびこれらの合計を示す。正解率、単純な記述ミスを無視した正解の両者で、TVQEが優位性を示した。

● 不正解

明らかな不正解の原因となったエラーをその種類に基づき分類した。表3に分類結果を示す。

履歴条件記述に関するエラー 履歴情報に関する条件記述はTV-Quelでは履歴型のオペレータを、TVQEではHistorical Condition Boxを使用して記述を行う。表3は、履歴情報に関する条件記述においてTVQEの方がエラーが入り込む余地が少ないことを示している。

履歴リレーションに関するエラー TV-Quelでは、履歴型のオペレータを使用する問い合わせに対し、履歴リレーションを明示的に参照することが必要になる。これに対し、TVQE

は履歴リレーションを意識する必要がなく、この種のエラーが起り得ない。このことが正解率の格差の大きな要因となった。

リレーション間の参照関係に関するエラー 表3から、リレーション間の参照関係に関する条件記述に関してはTVQEが優位であることが分かる。これはTVQEのベースであるQBEの特徴によるものと考えられる。

● アンケート結果

テスト終了後のアンケートとして、それぞれの言語の使用感を1から5までの5段階で表す質問をした。数字が大きいほど使いやすいことを示す。また、それぞれの言語について正解している自信の程度を同様に5段階で表す質問をした。数字が大きいほど自信があることを示す。表4にこれらのアンケート結果の平均を示す。使用感に関してはTVQEがより好評であったが、問い合わせの自信という点では大差がないという結果となった。

5.2 実装システムを利用した比較評価実験

同様の比較評価実験を前章で述べたTVQEインタプリタ、およびすでに実装済みのTV-Quelインタプリタ^{5),6)}を使用して行った。筆記試験の被験者のうち5名による実験を行った。問題の数は、TV-Quel 3問、TVQE 3問、合計6問を用意した。筆記試験による実験と同様に難易度のほぼ等しい問題3問ずつを用意し、TVQE 3問を先に解答する組とTV-Quel 3問を先に解答する組の2組に分けて試験を行った。実験に使用したリレーションは筆記試験のものと同じである。問題については紙面の都合上TVQEに関する問題のみを付録A.3に示す。本実験では、正しい問い合わせが成功するまで何度も問い合わせを発行してもらい、その回数を求めた。さらに、解答に要した時間とアンケートを評価の対象とした。

本実験では、TV-Quelに多い文字の入力は手書きよりもキーボード入力の方が速いことが予想できること、TVQEでは画面構成要素に関する実際の画面表示操作のため、筆記試験の場合と比べて若干余分な時間が必要なことなどが、筆記試験の場合と異なる。

表5に本実験の結果を示す。問い合わせ発行数は各被験者が1問あたりに発行した問い合わせ回数平均値である。また、解答時間は、各被験者が3問の解答

表3 不正解の原因の分類
Table 3 Classification of causes of errors.

| | TVQE | TV-Quel |
|------------------|------|---------|
| 履歴条件記述のエラー | 31 | 42 |
| 履歴リレーションのエラー | 0 | 28 |
| リレーション間の参照関係のエラー | 10 | 25 |
| その他 | 1 | 8 |
| 合計 | 42 | 103 |

表4 使用感と正解の自信の程度のアンケート結果
Table 4 Summary of questionnaires.

| | TVQE | TV-Quel |
|-----|------|---------|
| 使用感 | 4.0 | 2.8 |
| 自信度 | 2.9 | 2.7 |

表5 実装システムを利用した比較実験の結果

Table 5 Summary of experiment using the prototype system.

| | TVQE | TV-Quel |
|----------|--------|---------|
| 問い合わせ発行数 | 1.13 回 | 1.40 回 |
| 解答時間 | 12分40秒 | 24分00秒 |
| 使用感 | 4.0 | 2.4 |

を終了するまでの平均時間である。5段階評価は筆記試験の場合と同様にその値が大きいほど扱いやすいことを表す。本実験結果でも筆記試験の場合と同様、いずれの点においても TV-Quel に対する TVQE の優位性が示された。

5.3 実験のまとめ

TVQE と TV-Quel との比較評価実験の結果は、書きやすさ、読みやすさ、理解しやすさ、使用感に関して、おおむね TVQE が有利といえるものであった。今回の実験の条件の下での TVQE の優位性は TVQE の持つ以下の特徴によるものと考えられる。

- (1) Historical Condition Box での図式表現による履歴条件記述はその理解の促進や記述の容易化に有効である。
- (2) オペレータを図式表現することで、オペレータの引数の意味、型を意識せず記述できる。
- (3) 履歴リレーションを直接意識する必要がない。
- (4) QBE をベースとした表形式によるリレーションの表示により、リレーション間の参照関係の理解・記述が容易である。

今回の実験は、データベースの知識を有するものの設計業務に関する知識を持たない被験者によるものであった。したがって、実際の設計業務に関する知識の豊富な利用者が利用した場合には上記の評価結果がそのまま当てはまらない可能性も考えられる。このような利用者は、設計データの意味やその相互関係を熟知し、それらの知識を検索を行ううえで活用することができる。しかし、利用者が本来の設計業務に集中するためには、その業務知識やデータに関する知識をできるだけストレートに表現することが可能なデータベース・インタフェースが与えられることが望ましいことはいうまでもない。上記の TVQE の特徴は、利用者がデータベース言語やその構文にとらわれずに本来のデータや問い合わせの意味に集中することができるという点で、設計業務に関する知識を有する利用者に対してもプラスに作用するものと期待される。

6. TVQE の拡張に関する検討

TVQE の記述力は TV-Quel の記述力と等価ではな

```

retrieve(I.iname)
from I in ITEM
where {(S.sname)
      from S in SUPPLY
      where S.iname = I.iname}
⊆ {(S.sname)
   from S in SUPPLY
   where {(D.dname)
         from D in DEPT}
      ⊆ {(D.dname)
        from D in DEPT,
        S2 in SUPPLY
        where S.sname = S2.sname
        and S2.dname = D.dname}}

```

図15 QBE では記述不可能な POSTQUEL 問い合わせ
Fig. 15 POSTQUEL query example that cannot be expressed in QBE.

い。その最も基本的な理由としては、ベースとなった QBE の記述力の限界があげられる^{19),20)}。図15は、リレーション ITEM (iname, ...), SUPPLY (sname, iname, dname), DEPT (dname, ...) からなるデータベースにおいて、「すべての部門にある商品を提供している供給者だけが扱っている商品を求めよ」という問い合わせを TV-Quel のベースとなった POSTQUEL で記述したものである⁸⁾。この問い合わせは QBE による1つの問い合わせでは記述不可能である¹⁹⁾。Ozsoyoglu らは QBE の拡張を提案している^{14),19)}。彼らによる拡張版 QBE の特徴は、問い合わせ中に入れ子状の副問い合わせと集合値に関する比較条件記述を許す点である。本論文で提案した TVQE の特徴は、図式による導出関係や時間情報に関する条件記述であり、Ozsoyoglu らによる拡張とは直交するものである。したがって、彼らの提案を組み入れることにより TVQE の記述力を強化することは可能と考えられる。

記述力に関する第二の問題点としては、Historical Condition Box の記述力がある。現仕様では、Historical Condition Box 中の個々の条件は and 結合をしたものとして解釈される。QBE の Condition Box と同様に、and 条件のみでなく or 条件も記述可能とするための仕様拡張の検討が今後必要である。

4.2 節で示した TVQE から TV-Quel への変換を用いた実装は、TVQE の特徴である Historical Condition Box による条件記述が、ベースとなった QBE の基本的枠組みと融合して処理可能である点を示すことを主目的としたため、QBE ではサポートされ本来 TVQE でも Historical Condition Box 以外では記述可能な or, not, グループング、集約関数等については考慮していない。しかし、QBE から SEQUEL への変換²²⁾など、QBE 自身の変換についてはより一般化

された形での提案も行われており、もともとの QBE では許されるこれらの記述を組み入れた形での変換処理の拡張は可能と考えられる。ただし、詳細の検討は今後の課題としたい。

7. おわりに

本稿では、履歴条件の図式表現による記述を特徴とする視覚的問い合わせ言語 TVQE を提案した。また、TVQE インタプリタの設計および実装を行い、その実現可能性を示した。さらに、すでに提案済みの TV-Quel をひとつの比較対象として、図式を用いた履歴条件記述の有効性を筆記試験および実装システムを利用した実験を通して検討した。TVQE では版管理を目的とした履歴情報に関する図式表現のみを対象としたが、同様のアプローチはより一般的に、抽象データ型に関するオペレータの引数の意味や型を利用者から隠蔽するうえでも適用可能なものと考えられる。この意味では、同様のアプローチを一般のオブジェクトデータベースシステムのコンテキストに適用することを今後検討してゆきたい。

TVQE に関するこのほかの今後の研究課題としては以下の点があげられる。

- 記述力の拡張

第 6 章で述べたように、TVQE の記述力やその拡張についてより詳細な検討を行う必要がある。

- より多様な履歴条件記述の検討

時間に関する条件記述として 2 つの時刻の比較のみを検討してきた。しかし、月や年といった幅のある「時間」に関する条件記述が必要になる状況は容易に想像できる。このような時間条件の記述について直観的理解を損なわない記述方法の検討が必要である。

- データ定義・操作機能の追加

TV-Quel は検索だけでなく、版の導出・削除等のデータ更新に関する機能を有する。TVQE についても、図式表現を用いてこれらの各種データ定義・操作機能に対応するための検討を行う必要がある。

- 構成を含めた版管理の検討

設計データの版管理に密接に関係するものとして構成管理がある。TVDM では版および版集合を共にリレーションの 1 タプルとして扱っているため、版または版集合を対象とした構成の記述が可能であると考えられる。また、構成情報に関しても視覚化は非常に有効であると思われる。今後、構成管理を含めた版管理について検討を進めたい。

- 実装システムの機能強化

本来 QBE では記述可能であるが現システムでは未実装の or, not, グルーピング, 集約関数等の処理について、その詳細の検討と実装を行う必要がある。また、TVQE の利点のひとつはキーボードからの入力が必要でできる点である。しかし、現実装では、データベース名, リレーション名, 時刻の定数要素等の記述のためにキーボードからの入力が必要である。これらの記述もボタン等を利用することで、さらに操作性の向上が期待できる。さらにまた、エラーチェックの機能についても強化する必要がある。

本研究の比較評価実験では TVQE の比較対象として TV-Quel を用いたが、TV-Quel の改良によるその使い易さの向上も考えられる。たとえば、ビューを用いる等により TVQE と同様に履歴リレーションを明示的には意識しない条件記述を可能とすることによる改良が考えられる。これらの点の検討も今後必要である。

謝辞

本研究を進めるにあたり御協力いただいた筑波大学電子・情報工学系データベース研究グループの諸氏に感謝いたします。また、貴重なコメントを下さった査読者の方々にお礼申し上げます。なお、本研究の一部は文部省科学研究費補助金の助成による。

参考文献

- 1) Katz, R.H.: Toward a Unified Framework for Version Modeling in Engineering Databases, *ACM Comput. Surv.*, Vol.22, No.4, pp.375-408 (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, Vol.34, No.5, pp.1031-1044 (1993).
- 3) 井伊克益, 北川博之: 履歴型を用いた版管理システムの設計と実装, 第 47 回情報処理学会全国大会講演論文集, No.4, pp.127-128 (1993).
- 4) 井伊克益, 北川博之: 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, 情報処理学会データベースシステム研究会研究報告, Vol.94, No.62, pp.97-104 (1994).
- 5) Tansel, A.U., Clifford, J., Gadia, S., Jajodia, S., Segev, A. and Snodgrass, R.: *Temporal Databases*, The Benjamin/Cummings Publishing Company (1993).
- 6) McKenzie, L.E. and Snodgrass, R.T.: Evaluation of Relational Algebras Incorporating the Time Dimension in Databases, *ACM Comput.*

- Surv.*, Vol.23, No.4, pp.501-543 (1991).
- 7) Snodgrass, R.: The Temporal Query Language TQuel, *ACM Trans. Database Syst.*, Vol.12, No.2, pp.247-298 (1987).
 - 8) Rowe, L.A. and Stonebraker, M.R.: The POSTGRES Data Model, *Proc. 13th VLDB*, Brighton, pp.83-96 (1987).
 - 9) Rhein, J. and Kemnitz, G.: The POSTGRES User Manual, EECS Dept., University of California, Berkeley.
 - 10) Zloof, M.M.: Query by Example, *Proc. AFIPS National Computer Conf.*, pp.431-438 (1975).
 - 11) Zloof, M.M.: Query-by-Example: A Database Language, *IBM Syst.J.*, Vol.16, No.4, pp.324-343 (1977).
 - 12) Kunii, T.L. (Ed.): *Visual Database Systems*, North-Holland, Amsterdam (1989).
 - 13) Knuth, E. and Wegner, L.M. (Eds.): *Visual Database Systems*, II, North-Holland, Amsterdam (1992).
 - 14) Ozsoyoglu, G., Matos, V. and Ozsoyoglu, Z.M.: Query Processing Techniques in the Summary-Table-by-Example Database Query Language, *ACM Trans. Database Syst.*, Vol.14, No.4, pp.526-573 (1989).
 - 15) Tansel, A.U., Arkun, M.E. and Ozsoyoglu, G.: Time-by-Example Query Language for Historical Databases, *IEEE Trans. Softw. Eng.*, Vol.15, No.4, pp.464-478 (1989).
 - 16) Oomoto, E. and Tanaka, K.: Design and Implementation of a Visual Query Language for Historical Databases, *Database Systems for Next Generation Applications — Principles and Practice*, Kim, W. and Kambayashi, Y., Eds., World Scientific, Singapore, pp.184-191 (1992).
 - 17) Reisner, P.: Human Factors Studies of Database Query Languages: A Survey and Assessment, *ACM Comput. Surv.*, Vol.13, No.1, pp.13-31 (1981).
 - 18) Greenblatt, D. and Waxman, J.: A Study of Three Database Query Languages, *Databases — Improving Usability and Responsiveness*, Shneiderman, B., Ed., Academic Press, New York, pp.77-97 (1978).
 - 19) Ozsoyoglu, G. and Wang, H.: A Relational Calculus with Set Operators, Its Safety, and Equivalent Graphical Languages, *IEEE Trans. Softw. Eng.*, Vol.15, No.9, pp.1038-1052 (1989).
 - 20) Miura, T.: Nesting Quantification in a Visual Data Manipulation Language, *Data and Knowledge Eng.*, Vol.12, pp.167-196 (1994).
 - 21) Ousterhout, J.K.: An Introduction to Tcl and Tk, EECS Dept., University of California,

Berkeley.

- 22) Mcleod, D.: The Translation and Compatibility of SEQUEL and Query by Example, *Proc. 2nd ICSE*, San Francisco, pp.520-526 (1976).

付 録

A.1 比較実験で使用したリレーション

● CAR

| 属性名 | |
|---------|-------------|
| _vsetid | 版集合識別子 |
| _vno | 版番号 |
| eid | エンジンの版集合識別子 |
| evno | エンジンの版番号 |
| cid | シャーシの版集合識別子 |
| cvno | シャーシの版番号 |
| factory | 開発工場 |

● ENGINE

| 属性名 | |
|----------|--------|
| _vsetid | 版集合識別子 |
| _vno | 版番号 |
| displace | 排気量 |
| power | パワー |
| factory | 開発工場 |

● CHASSIS

| 属性名 | |
|---------|--------|
| _vsetid | 版集合識別子 |
| _vno | 版番号 |
| eval | テスト結果 |
| factory | 開発工場 |

● _CAR

| 属性名 | |
|----------|--------|
| _vsetid | 版集合識別子 |
| _history | 履歴型データ |

● _ENGINE

| 属性名 | |
|----------|--------|
| _vsetid | 版集合識別子 |
| _history | 履歴型データ |

● _CHASSIS

| 属性名 | |
|----------|--------|
| _vsetid | 版集合識別子 |
| _history | 履歴型データ |

A.2 筆記試験による比較実験で使用した問題 (TVQE用)

- 版集合識別子1番のエンジンの第3版の時間的直後の版の版番号を求めよ。
- 版集合識別子1番の自動車に組み込まれるエンジンの版集合識別子を求め、さらに、その版集合識別子をもつ版集合の1993年1月 (Jan 1993) に

における最新の版の版番号を求めよ。

- 版集合識別子 4 番のエンジンの第 2 版の兄弟の版の版番号とパワーを求めよ。
- 版集合識別子 4 番のエンジンの第 7 版から生成された版の中で、1992 年 10 月 (Oct 1992) から 1994 年 1 月 (Jan 1994) までに開発された版の版番号と開発工場を求めよ。
- 1994 年 12 月 (Dec 1994) 以前に開発された版集合識別子 2 番の自動車において、その自動車に組み込まれるシャーシの版集合識別子と、その版の時間的直前の版の版番号を求めよ。
- 版集合識別子 1 番の自動車の第 4 版に組み込まれるシャーシのテスト結果と生成時刻を求めよ。

A.3 実装システムを用いた比較実験で使用した問題 (TVQE 用)

- 版集合識別子 1 番のシャーシの 1995 年 1 月 30 日 6 時 (Jan 30 06:00:00 1995) における最新の版の版番号とテスト結果を求めよ。
- 版集合識別子 2 番のエンジンの第 4 版の兄弟の版の版番号とパワーを求めよ。
- 版集合識別子 1 番の自動車の第 4 版に組み込まれるシャーシのテスト結果と生成時刻を求めよ。

(平成 7 年 3 月 31 日受付)

(平成 7 年 10 月 5 日採録)



井伊 克益 (正会員)

1971 年生。1993 年筑波大学第三学群情報学類卒業。1995 年同大学大学院修士課程理工学研究科修了。同年シャープ (株) 入社。現在、同社福山 IC 事業本部第一事業部勤務。



北川 博之 (正会員)

1955 年生。1978 年東京大学理学部物理学科卒業。1980 年同大学大学院理学系研究科修士課程修了。日本電気 (株) 勤務を経て、1988 年筑波大学電子・情報工学系講師。現在、同大学同学系助教授。理学博士 (東京大学)。時制データベース、問合せ処理、ドキュメントデータベース、異種分散情報資源管理等に興味をもつ。著書「The Unnormalized Relational Data Model」(共著, Springer-Verlag) 等。電子情報通信学会, 日本ソフトウェア科学会, ACM, IEEE-CS 各会員。