

時区間データモデルと TSQL2 のデータ選択能力の比較

天笠 俊之[†] 田頭 利規[†] 金森 吉成[†] 増永 良文^{††}

[†] 群馬大学工学部情報工学科 ^{††} 図書館情報大学

著者等は実世界の時間的側面を記述するために、実世界の履歴情報を時区間で表現する時区間データモデルを提案した。このモデルでは時区間に対して色々な操作を施すことによってさまざまな問い合わせを記述することができる。時間的データベースに対する問い合わせ言語はユーザーにとって親和性が高く、かつ時間に関して豊富な問い合わせ表現が可能でなくてはならない。本稿では時区間データモデルについてこの評価を行うために、本モデルと TSQL2 のデータ選択能力の9項目について比較を行った。この結果、時区間データモデルによって TSQL2 と同等の表現が可能であることを示すことができた。

A Comparison of the Data Selection Power between Time Interval Data Model and TSQL2

Toshiyuki Amagasa[†] Toshimi Tagashira[†] Yoshinari Kanamori[†]
Yoshifumi Masunaga^{††}

[†] Department of Computer Science, Gunma University

^{††} University of Library and Information Science

We have proposed the time interval data model which represents the real world in terms of time intervals. This model allows us to express many temporal queries by applying operations to the intervals. It is essential that a temporal query language should be user-friendly and provide rich expression power. In this paper, we compare the time interval data model with TSQL2 on nine properties of data selection power. The comparison discloses that the data selection power in the time interval data model is nearly equal to that of TSQL2.

1 はじめに

一般に実世界の時間的側面は複雑である。これをモデル化するために我々は時区間データモデルを提案した [1]。このモデルにおいて事象の履歴は時区間の集合 (収集) として捕らえられ、この収集に操作を施すことによって色々な問い合わせを記述することができる。

時間的データベースの問い合わせ言語はユーザにとって親和性が高く、かつ時間に関する豊富な問い合わせ表現が可能であることが重要である。しかし、さまざまな時間のモデルがあり表現能力を理論的に規定するのは難しい。例えば、関係データベースでは関係完備なる概念によってデータモデルの表現能力を規定したが、時間のモデルに関してはこのような理論的な枠組は存在しない。そこで、本稿では TSQL2 [2] との比較によって時区間データモデルの問い合わせ表現の能力を測る。

2 時区間データモデルの概要

2.1 時区間と収集

時区間データモデルでは実世界において、ある事象が存在していた期間を時区間 (time interval) で記述する。時間軸にはさまざまなモデルが存在するが、ここでは離散モデル [3] を採用する。離散モデルにおいて時間は整数と同型である。時間軸は chronon と呼ばれるそれ以上分割することのできない最小の区間のリストとして表現される。この時、時区間上の任意の時刻は chronon によって表される。時区間は時間軸上の閉区間であり、連続した chronon の集合である。

本研究では時区間を事象の存在を表す実時区間と空時区間とに区分して扱う。前者はある事象が存在していた期間を表し、後者はそれが休止していた期間を表す。

[定義 1] 実時区間 (real time interval) と空時区間 (null time interval) [1] t_s, t_e ($t_s \leq t_e$) を時間軸上の任意の時刻とする。 a_i ($1 \leq i \leq n$) を属性名とする。属性名の集合 $A = \{a_1, a_2, a_3, \dots, a_n\}$ を属性リストと定義する。この時、 $(t_s, t_e; A)$ を実時区間、属性リストを持たない $(t_s, t_e; \emptyset)$ なる時区間を空時区間と定義する。空時区間は (t_s, t_e) と略記する。

任意の 2 つの時区間の間には 13 種の時間的關係が存在することが示されている [4]。例えば、本論文では時間軸のモデルとして離散モデルを採用しているので 2 つの時区間 $t = (t_s, t_e)$ と $u = (u_s, u_e)$ が *meets* の関係にあるとは次の条件を満たすときをいう。

$$t \text{ meets } u \text{ iff } t_e + 1 = u_s$$

この時間的關係は 2 つの時区間の関係を述べているだけでなく、2 つの時区間の間の演算と解釈することもできる。例えば $t \text{ meets } u$ は、時区間 t, u が *meets* なる関係を持つときに真を返す述語であると同時に、 t と u を接合した新たな時区間を返す演算であるとも言える。

時区間を集合として取り扱うために、時区間 (duration) や属性リストなどに時間的順序を持たせたリストを定義する。

[定義 2] 収集 (collection) [1] 収集は時間的順序を構造として持った時間値 (time value) の集合である。ここで時間値とは時刻、時区間、収集、期間、属性リストのいずれかである。

実世界のある事象を考えると、その履歴は一般に存在と休止を繰り返す。これは実時区間と空時区間を組み合わせて表現することができる。これを複合時区間と呼ぶ。

[定義 3] 複合時区間 (composite time interval) [1] $t_i = (t_{s_i}, t_{e_i}; A)$ を時区間とする ($1 \leq i \leq n$)。 $A = \emptyset$ の場合、 t_i は空時区間である。このとき、

$$(\forall i)(1 \leq i \leq n-1)(t_i \text{ meets } t_{i+1})$$

を満たすような収集 $C = \{t_1, t_2, t_3, \dots, t_n\}$ を複合時区間と呼ぶ。

複合時区間はその要素が *meets* なる時間的關係で連なっているという制約を持った収集であると考えることができる。時区間データモデルでは、複合時区間や収集に対して何らかの操作を加えることによって、時間に関する問い合わせ処理を記述・処理する。

2.2 時区間の演算

Leban 等 [5] による共通 (intersection) 演算と被覆 (cover) 演算は時区間の演算の中でも最も基本的なものである。本研究で定義する時区間には属性リストを持っているので、同様の演算を行う際にこれを考慮しなければならない。ここでは紙面の都合により定義の詳細 [1] は省いて、演算の例を示す (例 1, 図 1)。

[例 1] t_s, t_e, u_s, u_e ($t_s < u_s < t_e < u_e$) を時間軸上の任意の時刻とし、 $t = (t_s, t_e; A_1)$ と $u = (u_s, u_e; A_2)$ を時区間であるとする。

$$t \cap u = (u_s, t_e; A_1 \cup A_2)$$

$$t \cup u = \{(t_s, u_s - 1; A_1), (u_s, t_e; A_1 \cup A_2), (t_e + 1, u_e; A_2)\}$$

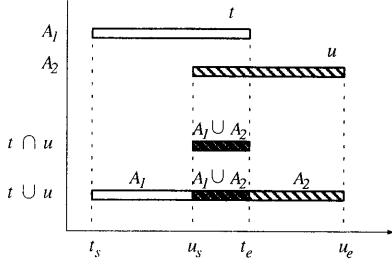


図 1: 共通演算と被覆演算

繰り返し (foreach) 演算は収集に対して Op なる演算を繰り返し適用する演算である。 Op は Allen による 13 種の時間的演算子や共通演算, 被覆演算などである。

$$\{C : R : t\} \equiv \{c \cap t \mid c \in C, c R t\} \setminus \{\epsilon\}$$

$$\{C . R . t\} \equiv \{c \mid c \in C, c R t\} \setminus \{\epsilon\}$$

ここで ϵ は $(-\infty, \infty)$ なる時区間である。

[例 2] 繰り返し演算 $C_A = \{(1, 10; A), (11, 30), (31, 40; A)\}$ を複合時区間とする。

$$C_A : \cup : (5, 35)$$

$$= \{(5, 10; A), (11, 30), (31, 35; A)\}$$

$$C_A . \cup : (5, 35)$$

$$= \{(1, 10; A), (11, 30), (31, 40; A)\}$$

繰り返し演算は第 2 引数に収集を取ることもできる。この場合第 2 引数の要素それぞれについて演算を適用する。

繰り返し演算と共通演算, 被覆演算を組み合わせることによって, 収集の間の共通演算と被覆演算を定義することができる。これらをそれぞれ拡張共通演算, 拡張被覆演算と呼ぶ。前者は 2 つの収集の論理積 (and), 後者は論理和 (or) を取った収集である。これらは問い合わせ処理を行う際に有効である。 C_1 と C_2 を収集とする。これらに拡張共通演算と拡張被覆演算を施した例を示す (例 3, 図 2)。

[例 3] 拡張共通演算と拡張被覆演算

$$C_1 = \{(t_0, t_2; A_1), (t_2, t_3), (t_3, t_4; A_1), (t_4, t_5), (t_5, t_6; A_1)\}$$

$$C_2 = \{(t_1, t_3; A_2), (t_3, t_5), (t_5, t_6; A_2)\}$$

とする。

$$C_1 \cap C_2 = \{(t_1, t_2; A_1 \cup A_2), (t_2, t_5), (t_5, t_6; A_1 \cup A_2)\}$$

$$C_1 \cup C_2 = \{(t_0, t_1; A_1), (t_1, t_2; A_1 \cup A_2), (t_2, t_3; A_2), (t_3, t_4; A_1), (t_4, t_5), (t_5, t_6; A_1 \cup A_2)\}$$

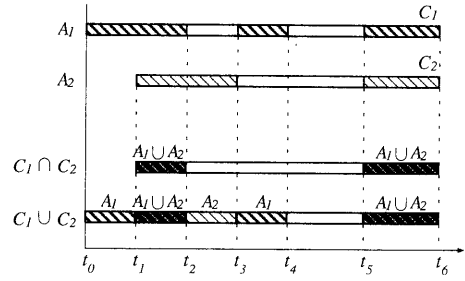


図 2: 拡張共通演算と拡張被覆演算

収集から任意の順番を指定してその要素を取り出す演算が選択 (selection) 演算である [6]。引数が負の場合, 後ろから数えて該当した時間値が選択される。

[定義 4] 選択演算 $t_i = (ts_i, te_i; A_i) (1 \leq i \leq n)$ を時区間とし, $f (1 \leq f \leq n)$ を整数とする。

$$f / \{t_1, t_2, t_3, \dots, t_f, \dots, t_n\} = t_f$$

また, f が $-n \leq f \leq -1$ の場合,

$$f / \{t_1, t_2, t_3, \dots, t_f, \dots, t_n\} = t_{f+n+1}$$

$t = (t_s, t_e)$ を時区間, C を収集とすると単項演算は以下のように定義される。

- $\text{real}(t)$ …… t が実時区間の場合, t を返す。
- $\text{null}(t)$ …… t が空時区間の場合, t を返す。
- $\text{startTime}(t)$ …… t_s を返す。
- $\text{endTime}(t)$ …… t_e を返す。
- $\text{duration}(t)$ …… $t_e - t_s + 1$ を返す。
- $\text{idcnt}(t)$ …… A を返す。
- $\text{count}(C)$ …… C に含まれる要素の数を返す。
- $\text{sum}(C)$ …… C の要素の和を返す。

以上の演算を用いて問い合わせを記述した例を以下に示す。

[例 4] 「従業員 A の在籍期間を求めよ」
 C_A を従業員 A の在籍履歴を表す複合時区間とする。

$$\text{sum}(\text{duration}(\text{real}(C_A)))$$

[例 5] 「1996 年 1 月に在籍していた従業員の人数を数えよ」

複合時区間 EMP を従業員全体の在籍履歴とする。

$$\text{count}(\{\text{EMP} . \cup : (1/1/1996, 1/31/1996)\})$$

Name	Dept	V
Al	Ship	{10, ..., 15, 20, ..., 25}
Al	Load	{21, ..., 24}
Bill	Load	{35, ..., 45}

表 1: 従業員リレーション

3 データ選択能力の比較

3.1 データ選択能力

時間的データベースでは、データを検索するためにユーザーとの親和性が高く、時間に関係した豊富な記述が可能であることが要求される。データの有効時間による検索を有効時間選択 (valid-time selection) と呼ぶ [2]。データベースには有効時間とはそのデータが実世界で有効だった時間のことである。

TSQL2[2] では既存の時間的問い合わせ言語を調査し、実践的視点から以下に述べる項目の記述が可能ないように言語を設計した。その項目とは、timestamp referencing, event extraction, interval constructor, event and interval comparison predicates, temporal predicates, valid timeslice, temporal ordering である。

本稿ではこれらの項目に関して TSQL2 と時区間データモデルの比較を行い、有効時間選択の記述能力を比較することにした。

3.2 TSQL2 のデータモデル

TSQL2 におけるリレーションの例を表 1 に示す。TSQL2 で定義されるリレーションは通常の (時間を考慮しない) 関係データベースの属性に相当する明示属性 (explicit attribute) とこれの時間情報を保持するタイムスタンプ属性 (timestamp attribute) の 2 種類の属性を持つ。表では {Name Dept} が明示属性、{V} がタイムスタンプにあたる。

タイムスタンプのドメインには 4 つの型がある。これを表 2 に示す。また、TSQL2 で用いている用語と本研究の時区間データモデルでの用語の対応を表 3 に示す。

問い合わせ言語の設計において型の定義の問題は重要であって、ユーザーに型の違いをなるべく意識させないような設計が望ましい。TSQL2 のデータ型もこのように設計されている。すなわち要素は任意の chronon の集合、期間は連続した chronon の集合、イベントは単一の chronon である。従ってイベントと期間は要素の特殊な場合であると考えられる。

イベント (event)	時間軸上の任意の chronon. (例)1996/2/16
期間 (period)	端点の定まった時間軸上の閉区間。 (例)1996/2/16 ~ 1996/2/23
区間 (interval)	端点の定まっていない時間軸上の区間。 (例)2年間
要素 (element)	期間の集合

表 2: TSQL2 の時間に関するデータ型

時区間データモデル	TSQL2
時刻 (time point)	イベント (event)
時区間 (time interval)	期間 (period)
期間 (duration)	区間 (interval)
収集 (collection)	要素 (element)

表 3: 概念・用語の対応関係

3.3 比較項目

本論文では以下の項目についてモデルの表現能力を比較する。

- ・タイムスタンプ参照 (Timestamp referencing)
データベース上のタプルについてそのタイムスタンプを抽出する方法。

- ・抽出 (Extraction)
抽出にはイベント抽出と期間抽出の 2 つがある。任意の期間は開始イベント (starting event) と終了イベント (ending event) から成り立っており、イベント抽出では、期間からこれらのイベントを取り出す。これは問い合わせ表現に有効である。期間抽出とは要素 (期間の集合) から期間を取り出すことである。

- ・構成子 (Constructor)
開始イベントと終了イベントを組み合わせることによって区間を構成することができる。またいくつかの期間を組み合わせることによって要素を作り上げることができる。これらをそれぞれ期間構成 (period construction), 要素構成 (element construction) と呼ぶ。

- ・比較演算 (Comparison)
期間や時刻などの大小比較をするための述語である。

- ・タイムスライス (timeslice)
一般に、タプルの有効時間を表す期間や時刻は時間軸上に散在している。タイムスライスとは、ある時刻で有効なタプルを選択する操作である。例えば、「1995 年 1 月 30 日に在籍していた社員」のような問い合わせがこれにあたる。

・時間的整列 (temporal ordering)

あるタプル群に対してそれぞれの時間順にソートを行う。この操作によって時間を基準にタプルを取り出すという操作ができる。例えば、「全ての社員について、在籍期間の短い順に 10 人を選べ」という問い合わせが考えられる。

3.4 データ選択能力の比較

以下では、上に示した比較項目に関して TSQL2 と時区間データモデルのデータ選択能力の比較を行う。

タイムスタンプ参照

・ TSQL2

TSQL2 では任意のタプルに対して VALID 演算を適用することによってタイムスタンプを参照することができる。また、タプル変数を用いて明示的に EMPLOYEE.PERIOD のような記述も可能である。

・本モデル

時区間データモデルではタイムスタンプそのものを取り扱っているため、TSQL2 におけるタイムスタンプ参照のような概念は存在しない。時間を参照するには $t = (t_s, t_e; A)$ のような区間変数 (interval variable) を用いるか、 $(t_s, t_e; A)$ のように、値を直接記述する。

抽出

・ TSQL2

抽出にはイベント抽出子 (event extractor)、期間抽出子 (period extractor) の 2 種類がある。イベント抽出子は BEGIN と END である。これらは与えられた引数からそれぞれ最初と最後のイベントを取り出す操作である。引数としてはイベント、期間、要素の全てを取ることができる。期間抽出子は FIRST と LAST であり、それぞれ最初と最後の期間を抽出する。これは期間と要素に対して適用可能な操作である。

TSQL2 ではユーザーにタイムスタンプの型を意識させないように、全ての型について共通の関数を用意している。例えば、期間や要素から先頭 (最後) のイベントを取り出す *BEGIN(END)* という関数はイベントにも適用することができる¹。これは期間の特別な場合がイベントであって、イベントも期間と考えられるからである。

・本モデル

¹この場合 *BEGIN* と *END* は同じ値を返す。

一方、本研究では時区間から時刻を取り出すために *startTime*, *endTime* の演算を定義した。これらは *foreach* 演算と併用することによって収集に適用することもできる。時区間と収集を同様の手続きで扱うことができるという点は TSQL2 と同様である。また、収集から時区間を抽出するために選択演算子を提供している。

TSQL2 では与えられた引数から最初と最後のイベントまたは期間しか抽出することができないのに対し、本モデルでは選択演算子を用いることによって任意の順番でそれらを抽出することができる。よって本モデルの方がより詳細な記述が可能である。

構成子

・ TSQL2

TSQL2 ではイベント構成子として *FIRST*, *LAST* を、期間構成子として *PERIOD* を、要素構成子として *INTERSECT*, *+*, *-* を提供する。

FIRST は引数として与えられたタイムスタンプの中で最も早いイベントを返し、*LAST* は最も遅いイベントを返し、期間構成子 *PERIOD* は引数として 2 つのイベントを取り、これらを開始事象、終了事象とした新しい期間を返す。要素構成子の 3 つの演算は集合演算 (*INTERSECT*, *+*, *-*) であり、要素はこれらの演算で閉じている。前述した通り TSQL2 においてイベントと期間は共に要素の特化した形であるから、これらは全て要素構成子の引数として取ることが可能である。

・本モデル

時区間データモデルではデータ操作を定義していないために、TSQL2 の期間構成子にあたる関数は定義されていない。要素構成子は収集の間で共通演算を取ることによって同等の操作が可能である。

比較演算

・ TSQL2

事象、区間、要素の時間的な順序を比較するために、*PRECEDES*, *=*, *OVERLAPS*, *CONTAINS*, *MEETS* の 5 つの比較演算子を定義している。*MEETS* 以外の比較演算子は要素、区間、事象に関して全て適用可能である。

この比較演算子は Allen の 13 種の時間的關係を組み合わせたサブセットになっている。これは問い合わせを表現するのに必要最低限と考えられる比較演算子を選択した結果である。

・本モデル

本モデルでは Allen の時間的關係に加えて、問い合わせ表現に有効であると考えられる幾つかの關係を追加した。

タイムスライス

・TSQL2

TSQL2 ではタイムスライス文を定義していない。これは、SQL-92 からの拡張を最小限にとどめるという TSQL2 の設計方針に基づいている。タイムスライスと同等の操作は SQL の入れ子表現によって可能である。

・本モデル

本研究でも直接タイムスライスを表現することはできないが、{C.U.t} によって同等の操作を実現することができる。

時間的整列

・TSQL2

TSQL2 のリレーションにおいてタプルは独立な存在である。従って有効時間の順番でタプルをソートしようとする場合、集約関数を用いてこれを行う。

・本モデル

時区間データモデルでは、時間的順序を持つ時間値の集合を収集と定義した。従って、このような時間的順序付けは必要ない。

3.5 まとめ

以上の比較の結果をまとめたのが表 4 である。本モデルでは時区間の追加などのデータ操作を考えていないので、TSQL2 における期間構築 (period construction) の操作を行うことができない。

TSQL2 における期間抽出は本モデルの選択演算に対応する。期間抽出では先頭の要素 (FIRST) または最後の要素 (LAST) としか指定することができないのに対し、選択演算は任意の順番を指定して時区間を抽出することができる。

タイムスライス文は TSQL2 においても時区間データモデルにおいても定義されていないが、タイムスライス文そのものを用いなくても同等の表現が可能である。

この表から、データ選択能力に関して時区間データモデルは TSQL2 と同等かそれ以上の記述能力を持つことが分かる。

	TSQL2	時区間 DM
event extraction	○	○
period extraction	○	◎
event constructors	○	○
period constructors	○	×
element constructors	○	○
time comparison	○	○
mixed comparison	○	○
time slice clause	△	△
temporal ordering	use aggregate functions	○

表 4: 表現能力の比較

4 おわりに

本稿では時区間データモデルと TSQL2 を比較することによって、時区間データモデルのデータ選択能力を評価した。結果として、時区間データモデルでは TSQL2 と同等かそれ以上の記述が可能であることを示すことができた。

今後の課題として、時区間データモデルの問い合わせ言語への組み込みなどが挙げられる。

参考文献

- [1] 福田紀彦, 天笠俊之, 金森吉成, 増永良文. “時区間に基づく拡張時間データモデル”. In *Proc. of Advanced Database System Symposium '94 (Tokyo, Japan)*, pp. 175-184, Dec. 1994.
- [2] The TSQL2 Language Design Committee. “*THE TSQL2 TEMPORAL QUERY LANGUAGE*”. Kluwer Academic Publishers, 1995.
- [3] Clifford, J and A. U. Tansel. “On an Algebra for Historical Relational Databases: Two Views”. In *Proc. of ACM SIGMOD International Conference on Management of Data.*, pp. 247-265, May 1985.
- [4] J. F. Allen. “Maintaining Knowledge about Temporal Intervals”. *Communications of the ACM*, Vol. 26, pp. 832-843, Nov. 1983.
- [5] B. Leban, D. McDonald, and D. Forster. “A Representation for Collections of Temporal Intervals”. In *Proc. of the AAAI-1986 5th Int. Conf. on Artificial Intelligence*, pp. 367-371, 1986.
- [6] R. Chandra, A. Segev, and M. Stonebraker. “Implementing Calendars and Temporal Rules in Next Generation Databases”. In *Proc., Tenth International Conference on Data Engineering*, pp. 264-273, Feb. 1994.
- [7] 天笠俊之, 田頭利規, 金森吉成, 増永良文. “制約を導入した時区間代数”. 情報処理学会データベースシステム研究会資料, No. 104-24, July 1995.