D-004

時変動実体を扱う情報システムの構築

Developing Information Systems Which Handle Temporal Entities

並松鏡友(ヴァリューシステム)　田村浩一郎（中京大学）

Toshinari Namimatu　Koichiro Tamura

1．はじめに — なぜ「時変動」？

単純なグループウェアを例にとる．グループ G のメンバーが変動するとき，G 宛の記事 A は，閲覧時点で G に所属するメンバーに対してのみ表示されるべきであろう．しかし，A が D 日予定のカレンダー記事だとすれば，カレンダーを閲覧したとき，その閲覧時点に関係なく， D 日における G の所属メンバーに対してのみ，その日の予定として表示されるべきであろう．さて，どうすれば両者を統一的にあつかえるか．

グループに限らず，実世界の事物のほとんどは変動するから，時変動性は，本来どの情報システムも考慮するべき課題である．しかし，多くのシステムは，単にデータを更新するか，あるいはアドホックなプログラミングで対処しているため，バグを生みやすく，システムの開発，運用，発展を困難にし，セキュリティを危うくし，モービリティを下げ，史料価値を貶める．体系的かつ簡潔な方策が強く望まれるが，包括的に時変動データベースについて論じた Jensen[1]においても，例えば上記問題に答える統一的方策は見いだせない．

2．時変動を考慮しない実体の状態とその言明

現在の標準的な情報システムの構成と対象世界の表現法の要点は次のようになろう．

情報システムを**操作層**（コントロールとビュー）と**表現層**（モデル）に分ける．操作層は表現層を利用して，ユーザの要求に応える．表現層は，**実体**（いわゆる「事物」）を要素とする対象世界 Wの状態，即ち実体間の相互関係を次のようにして表現する．W の部分集合を対象，その間の写像を射とする集合圏 Ωにおいて，射 $A: X\to Y$について，$Y=\Pi_i Y_i$（直積）と表せるとき，Aを**属性**といい，$A_i: X\to Y_i$をその**成分**という．属性は実体の特徴を他の実体で与える．実体 $e\in X$に対し，$A(e)\in Y$を eの**状態**，$<e, A(e)>$を eの状態の**言明**という．言明の集合$\{<e, A(e)> \mid e\in X\}$を**類** Aといい，対象 Xについての言明を形成する．属性 Aの成分 A_Fのコドメインが属性 Bのドメインとなるとき，A_Fを類 Bへの**外部参照子**と呼ぶ．このとき，両類の状態言明を介して属性 Aで A_Fを $B\circ A_F$に置換して作られる属性の類を類 A,Bの**結合**という．表現層をさらに**機能層**と**記録層**に分けたとき．集合である類を，機能層は内包記法として定義し，記録層は外延記法として表記し記録する．実体の表記は，リテラルはそのまま，非リテラル実体は識別子（ユニーク数）で代用する．

表現層に対し，操作層は実体の状態言明の**取得**（検索）ないし**変更**（生成，更新，削除）を要求する．ここで，状態は常に「今」の状態であると措定され，従って，対象世界の状態変動即ち状態言明の変更となる（McTaggart の A series[5]）．

ここに時間と時変動の明示的表現はない．改めて基本的な定義から見直す必要がある．

3．時変動実体のコマによる状態言明

時変動を表現するため，時間の部分集合（**期間**）を対象，その間の推移を射とする圏を Θとし，関手圏 Ω^Θを用いる．属性 A（とその成分）は Ω^Θの射，即ち自然変換であり，圏 Θに依存する． $\tau\in$

Θ, $A(\tau)$: $X(\tau)\rightarrow Y(\tau)$, $e\in X(\tau)$としたとき，$<\tau, e, A(\tau)(e)>$をeの**コマ**と呼び，τを**状態期間**という．コマは，τにおける実体eの状態が$A(\tau)(e)$であることの言明である．コマの集合$\{<\tau, e, A(\tau)(e)> | \tau\in\Theta, e\in X(\tau)\}$を類$A$とする．類の結合は，時点を共有する（互いの状態期間の共通部分が空でない）コマを介して行われる．

記録層において，記録の生成から削除までの期間を**記録期間**ρとし，$<\rho, \tau, e, A(\tau)(e)>$をコマの**記録**とする（削除は削除時間を記録するのみとし，更新は削除と生成で行う）．かくて，コマの記録の総体が対象世界Wの「過去，現在，未来」の状態の言明を形成する(McTaggartのC series[5])．

実世界を表現するためには，空間的順序など，さらに種々の言明要因を設ける必要があるが，詳細は他に譲る[2].

4．設定時点

システムの全域変数として3種の設定時点（時間）を設ける．第1は，記録を変更する時点（**変更時点**）で，現在時を量子化して設定する．第2は，システムの記録を見る時点（**表示時点**）で，現在時をデフォールトとし，ユーザが随意に設定する．この時点が記録期間に入る記録がその時点で（論理的に）存在する記録である．第3は，（システムが記録する）対象世界の状態を見る時点（**言及時点**）で，この時点が状態期間に入るコマ記録が，（与えられた表示時点がその記録期間に入っていれば）その時点における実体の状態を示す．表示時点をデフォールトとする．

5．簡単な例

頭書の例に戻る．表現層における類として，「記事」，「宛先グループ」，「グループ」，「グループメンバー」，「ユーザ」を設ける．「グループメンバー」類では，「グループ」類と「ユーザ」類への外部参照子によりグループとそのメンバーの関係にある両実体が示され，記録期間と状態期間がその関係の成立期間を定める．他の類も同様である．記事の閲覧では，「記事」類から「ユーザ」類まで，言及時点のコマの状態を介して結合して得られた類に対し，閲覧するユーザ宛の記事を検索し，表示する．一般記事としての閲覧の場合は，与えられた表示時点を言及時点に設定する．一方，カレンダー閲覧では，「記事」類の属性成分の一つ「予定日」の値を言及時点とする．

6．実装と有効性の検証

対象世界の時変動性を統一して扱うために，Ruby on Rails[3]のActive Record（機能層に該当）として，コマの共通属性と機能（取得や変更など）を定義するクラスを作成した．応用では，対象世界のコマの類をこのサブクラスとする．この実装方式により，時変動性の組み込みによる所要時間のオーダへの影響はない．中京大学理系学部イントラサイト[4]を題材とし，この機構の有効性を検証した[2]．所要時間の実験データ例を**表1**に示す．

メッセージ投稿	0.15
メッセージ一覧	0.45
スケジュール閲覧	0.58
シラバス一覧	0.81

表 1　所要時間の例(単位：秒)

(DELL Inspiron 6400, WEBrick, MySQL5.0 使用)

参考文献

[1] C.S.Jensen: Temporal Database Management, http://www.cs.aau.dk/~csj/Thesis/

[2] 並松鏡友「時変動オブジェクトを実体とした情報システムの構築」，中京大学情報科学研究科情報科学専攻 2006年度修士論文

[3] http://www.rubyonrails.com/

[4] https://intrasite.sist.chukyo-u.ac.jp/

[5] http://www.ditext.com/mctaggart/time.html