データモデリングにおける時間依存性の取扱いについて

4M-4

宮原 広行, 橋本 恵二

富士通株式会社 システム本部

1. はじめに

データベースを主流とするビジネスアプリケーションシステムの開発において，概念データモデリングの重要性が認識されてきている[1]．

筆者らは，データモデリングを含むシステム要求分析技法C－NAPⅡ[2]を開発しており，これまでその適用を何例か行っている．

その中で，特に，データの時間軸上の意味と静的なデータモデルとのギャップをどのように埋めるか，という問題に何度か出会った．

本稿では，イベントシーケンス分析という方法を用いて，データモデリングにおける時間依存性の取扱いについて述べる．

2. データモデリングの困難

2.1 関数従属関係の時間依存性

データモデリングでは，業務の仕組みやユーザの要求から，データの包括的な意味を表現する概念データモデルを作成する．概念データモデルは，エンティティ内あるいはエンティティ間の論理的なデータの構造を記述する．エンティティとは，その業務で管理したい対象である．

システム内で扱うエンティティにデータ項目を集めてくる基準は，関数従属関係である．つまり，エンティティの識別キー項目の値を一つ指定すると，そのデータ項目の値は一つに決定されなければならない．識別キー項目とは，エンティティの個々のレコードを一つ一つ識別するためのデータ項目である．

しかし，関数従属関係を考える時に難解なのは，従属関係が時間に依存する場合である．例えば，「受注No.を決めると希望納期が決まる」は常識的に成り立ちそうである．しかし，「受注No.を決めると出荷日が決まる」かどうかは即答できないだろう．出荷日は受注を受けた時点では決まらず，その後に顧客先へ出荷が行われて初めて決定されるからである．

このように，識別キー項目が確定される時点と，データ項目が決定される時点との間に時間的な隔たりがある場合，関数従属関係を正確にとらえるのが難しくなる．

2.2 従来の理論

データの時間依存性を取扱うために，エンティティライフサイクル分析[3]やＪＳＤの応用[4]などが考えられる．これらの理論は，業務の動きをエンティティの状態遷移で表現する．

しかし，このような方法を現実の業務に応用するには次のような問題がある．

① データ更新イベントをすべて扱うため，モデルが複雑になりすぎる．

② ユーザとの共同作業や確認作業を考えたとき，難解すぎる．

分析法の規則性は少し落としてでも，誰にでもルールのわかりやすい方法が必要である．

3. イベントシーケンス分析

3.1 イベントの発生順序とエンティティの状態概念

業務システムにおいて，業務活動を行うことにより色々な出来事（イベント）が発生する．これらの出来事の間に，業務の都合により又活動する側の都合により，順序関係が生じる場合がある．

例えば図1に示すように，「予約受付」というイベントの発生とそれに対応する「予約確認」というイベントには時間的な順序関係がある．つまり，ある「予約確認」が発生するには，その前に必ず対応する「予約受付」が起こっていなければならないという関係である．

図1 予約エンティティの状態遷移

On Time Dependency in Conceptual Data Modeling

Hiroyuki MIYAHARA, Keiji HASHIMOTO

FUJITSU, Ltd.

このようなイベントの時間的順序関係を考慮すると，エンティティに状態概念が自然に生まれる．つまり，予約エンティティの一つのレコードに注目したとき，そのレコードは予約があってまだ確認が済んでいない状態なのか，あるいは確認が済んだ状態なのか，既にチェックインしてしまった後なのか，といった見方が生まれてくる．

3.2 強い状態遷移と弱い状態遷移

データモデリングにおいて，エンティティに起こるすべてのイベントに対して，エンティティの状態遷移を調べると，モデルが複雑になるという問題がある．そこで，強い状態遷移と弱い状態遷移という概念を持ち込む．

今，予約エンティティの構成が

予約＝予約No.＋予約者名＋ﾁｪｯｸｲﾝ予定日＋ルームNo.

となっているとする．ここで下線は識別キー項目である．この業務システムでは予約確認があった時点でルームNo.をアサインするとする．

もし「ﾁｪｯｸｲﾝ予定日」が変更になったとき，普通は「ﾁｪｯｸｲﾝ予定日」の値を変更する．すると，予約エンティティは元の「ﾁｪｯｸｲﾝ予定日」を忘れてしまう．この場合，予約エンティティが「ﾁｪｯｸｲﾝ予定日」の変更を受ける前の状態か後の状態かは区別しようがない．一方，予約エンティティが仮予約の状態か確認済の状態かは「ルームNo.」で区別できる．つまり，このデータ項目に意味のある値が入っているかどうかで状態を区別できる．

前者の場合は，エンティティに変化があっても陽に状態の区別ができないので，弱い状態遷移と呼び，後者のように，識別キー項目に関数従属するデータ項目が新たに加わった場合を，強い状態遷移と呼ぶ．

エンティティの状態遷移を調べるのに，すべてのイベントを相手にする必要はない．強い状態遷移を引き起こすイベントだけでエンティティの状態遷移は識別できる．この作業は比較的簡単で，データモデリングは十分行える．

3.3 イベントシーケンス図

イベントシーケンス分析という手法に用いる表記法として，イベントシーケンス図を用いる．例えば，図1に示す状態遷移を，図2のようなイベントシーケンス図に表現する．

エンティティに対して，その1インスタンスのレコードが影響を受けるイベントを時間順に並べ，その状態遷移を明らかにする．その上に，「そのエンティティがどの時点で何を知ることができるか」という観点でデータ項目をアサインする．

また，状態遷移とデータ項目の関係を一緒に記述することで，視覚的にわかりやすくなるように工夫した．

但し，2つ以上のイベントが繰返して起こるようなイベントシーケンスの形式性は犠牲にしている．

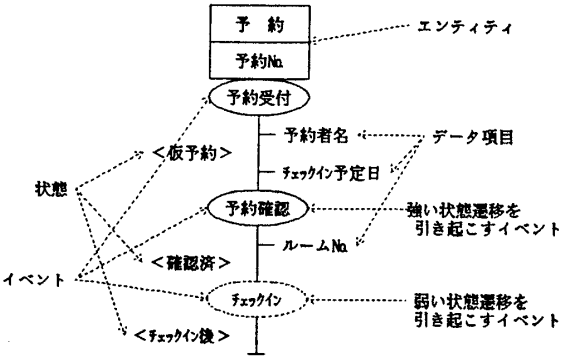

図2 予約エンティティのイベントシーケンス図

4. 評価

筆者らは，イベントシーケンス分析の有効性を調べるために，ある生産管理業務に適用した．

この業務の特徴的な点は，生産ライン中の製品の単位が工程毎に変化するため，状態遷移を起こすエンティティが多数あること，見込み生産と受注生産が混在するため，受注と生産ラインのエンティティとの関連や在庫の概念が動的に絡み合うことである．

イベントシーケンス分析を適用してみて，その有効性を高く評価した．実際，この業務ではエンティティ間の関連，データ項目の従属関係の時間依存性が著しいため，通り一遍の正規化では歯が立たなかったであろう．またこの分析を行ったことによって，最終的な概念データモデルにおいて，各々の関連が何故生じているかが明確に説明できるというメリットもあった．

参考文献

1) 味村，山田，堀内「データベースシステムの設計と開発」オーム社 (1982)
2) 橋本，永田「データ中心アプローチに基づく上流工程支援」情報システム研究会 (1988)
3) C.J.Rosenquist:Entity Life Cycle Models and their Applicability to Information Systems Development Life Cycles, THE COMPUTER JOURNAL, Vol.25, No.3, 307-315 (1982)
4) Cameron,J.R.:An Overview of JSD, IEEE Trans. Softw. Eng. Vol.SE-12, No.2, 140-158 (1986)