

実体/事象モデリングによる要求定義の一手法

堀内 一 木戸 恭彦
 (日立製作所コンピュータ事業本部) (同 左)

1. はじめに

近年、データ中心(data oriented)アプローチによるシステム設計、ソフトウェア設計手法の提案は多くなっている。(1)(2)(3) 特に、データベースを利用する適用業務システムの設計はデータベース中の標準データに従いながら、その制約下で行われねばならない。このようにデータ中心アプローチが可能となるか、否かは、才1に、データ自体が個々の適用業務システムによる情報生産のための共通部品としての要件を、どの程度満たすかに依存している。才2に、共通部品である標準データの完全性や安定性を維持する仕組みや組織的統制に依存し、才3に、共通部品によって、個々の情報生産と設計、定義するための方法論に依存する。

本稿では、かゝる観点から、既存データ群を形式化し、そのデータを基に、データ更新系の形式化を図ることを述べ、さらに、形式化されたデータとその更新処理を前提に、個々の情報要求と分析、定義する手法と述べる。

2. システム形式化の概念と手順

システム形式化とは、複数の既存システム群への体系的に従がめしめらることをいう。個々の適用業務システムが従うべき体系を明示する手段の一つとして概念データモデルや企業データモデルを考へる方法が、文献(3),(4)に示されている。図1にそのようは概念、図2に手順を示す。図2の点線で示した過程はデータ管理者による実施される。実線で示した過程は適用業務システムごとに実施される過程である。また、システムモデルの構成を図3に示す。

モデリング		形式化		変換
調査	システムモデリング	データ体系化	データシステム体系化	個別システム構築
〃	〃	〃	〃	〃
〃	〃	〃	〃	〃
〃	〃	〃	〃	〃

図1. システム形式化の手順

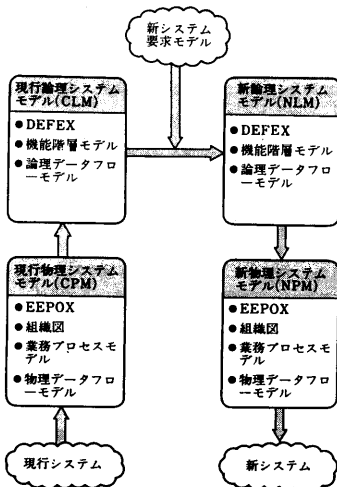


図3. システムモデルの構成

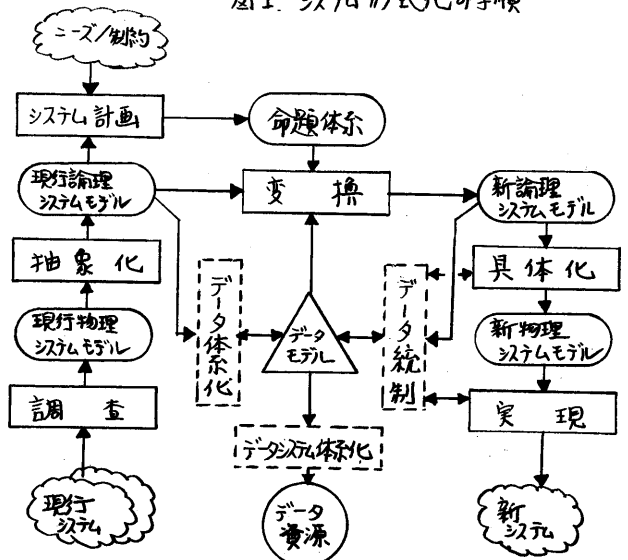


図2. システム形式化手順

3. 現行システム群の形式化

統一的記述手段で表現されたシステムモデルを頼りに現行システム群の形式化が行われる。ここで、形式化とは断片的なシステム群を再構築するための、統合体系をデザインすることを言う。この過程は、大きくデータ体系化 (data architecture design) とデータ体系に基づくシステム体系化 (data system architecture design) に分けられる。形式化の意図は、図3に示すように、これまで、適用業務システムごとに、固有に確保されてきたデータ更新処理と業務処理を分離し、データとその更新処理をデータ資源と見なし、一元管理することにある。

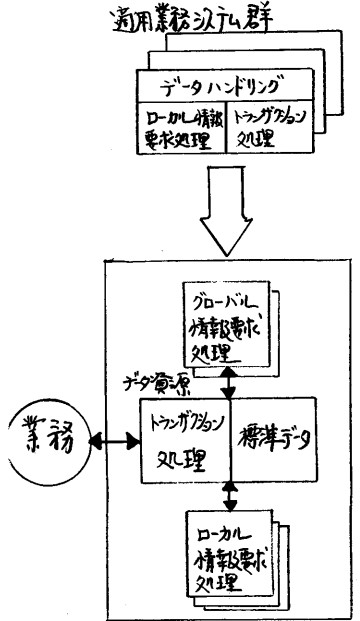


図3. 形式化の目標

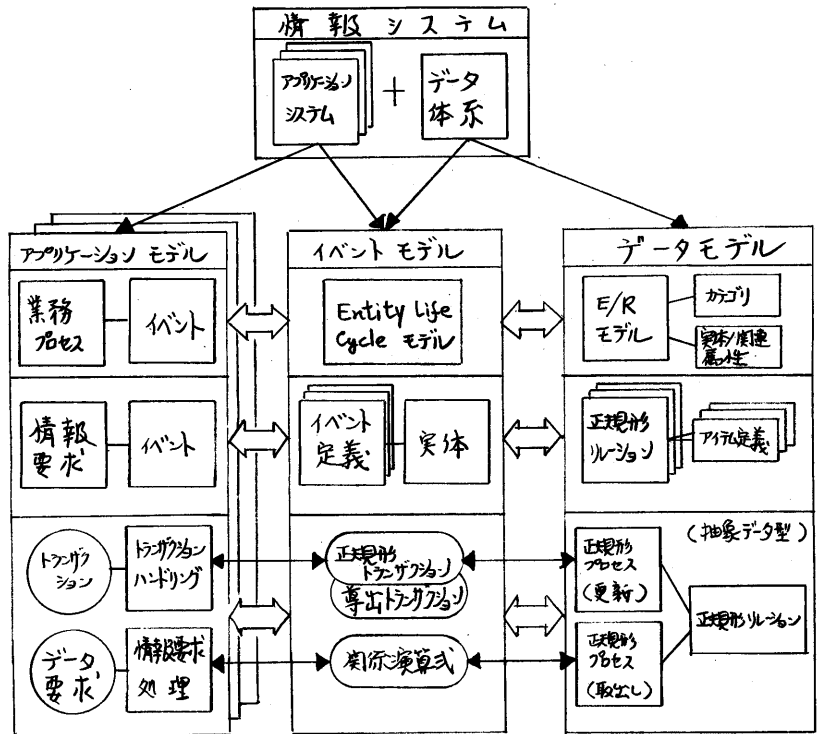


図4. システム体系

図4は、形式化されたシステムのモデルを示すものである。まず、情報システムはデータ体系と適用業務別のアプリケーションシステム群に大きく分けられる。データ体系はデータモデルとしてとらえることができ、アプリケーションシステムはアプリケーションモデルとしてとらえられる。データモデルとアプリケーションモデルを仲介する役割としてイベントモデルが存在する。図4の各モデルは、上層が概念レベルであることを示し、下方向に具体化過程をとる。最下層はインプリメンテーションレベルを示す。

3.1 データモデル

データは、実体/関連 (E/R) モデルの段階で、カテゴリの設定を受け、体系化され、各実体、関連に対応する正規形リレーションを割り付けられる。正規形リレーションはアイテム定義情報と結びつけられる。これらの結びつきがデータディクショナリによって維持される (図5参照)。

概念データモデルは現
行システムモデルから、
帰納的に作成する。その
手順を図6.に示す。

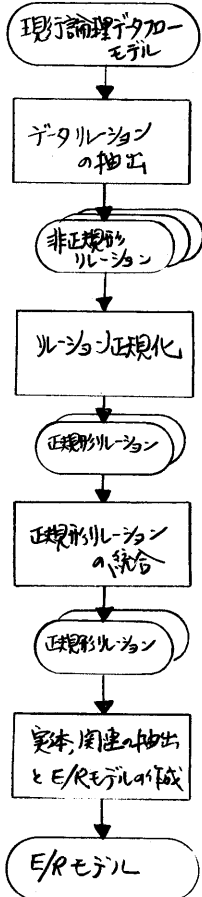


図6 帰納的プロセスによるE/Rモデルの作成

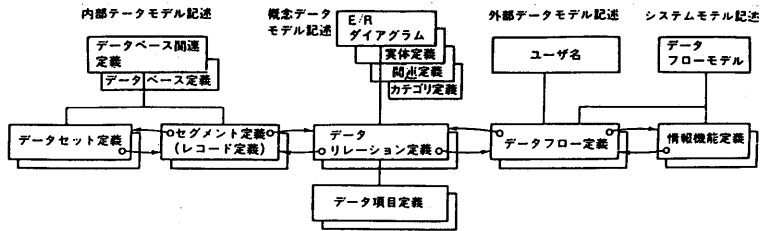


図5. データの役割

図7.は、このような手順に従って作成されたE/Rモデルである。◇型と○型との関連には全て正規形リレーションが対応する。(◇型に対応するリレーションは関連を表現するものではなく、実体を表現するものがある)。

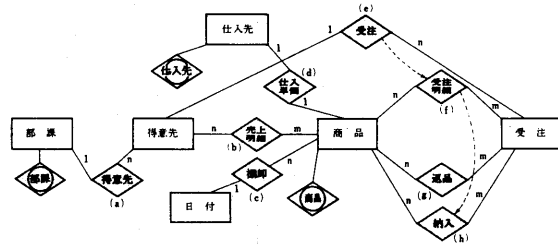


図7. E/Rモデル

図7のE/Rモデルは、図8に示す論理データモデルから作成されたものである。

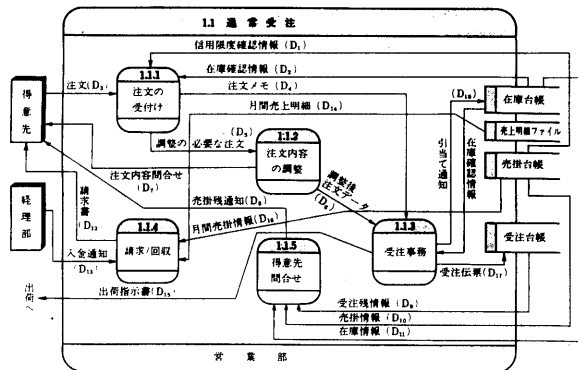


図8. 論理データモデル

図8の論理データモデルは、図9に示す正規化の対象となるデータフローは、D1, D2, D9, D10, D11, D14, D16, D17, D18, である。

3.2 イベントモデル

システムが反応すべき有意な出来事をイベント(event)と呼ぶ。イベントに着目したシステム設計手法は多数、提案されている。(5)(6)(7)(8)(9) k.A. Robinson⁽¹⁰⁾は実体ライフサイクルをイベントごとの之、M.A. Jackson法⁽¹¹⁾と結びつけてプログラム構造を設計する手法を述べている。また、C.J. RosenquistはRobinsonの提案を受け、Lindgreen記法、有限状態グラフを用いたシステム記述手法を示している。⁽¹²⁾また、V. De AntonellisとB. Zontaはイベントの定義記法とPetri Netsによるイベント関連の記述を述べている。⁽¹³⁾

本稿では、正規形データリレーションの実体との対応が一貫している性格をもつことに着目し、正規形リレーションとイベントの対応を、実体ライフサイクルとし

と定義しはばり、そのサイクルから正規形リレーションの基本的な更新プロセスを導く方法を考える。そして、そこから個々のアプリケーションに依存しない情報システムの基本的構成(データシステム体系)を決定するシヒを考ふる。

(1) イベントとトランザクション

イベントは実体と同様に非定義概念である。しかし、実体と同様、識別可能である。イベントは実体の状態変化や、実体と実体の間の動的な関連によって発生する。つまり、イベントは必ず対象実体をもつ。

トランザクションとは、イベントの発生を情報システムに伝達するための情報であり、それ自体、一つのデータリレーションである。トランザクションによる正規形リレーションは更新(追加, 削除を含む)される。(図9参照)

(2) イベントと実体

一つのイベントは一つの、実体と対象とすることもあるが、同時に複数の対象実体をもつこともある。その対応は1:nの対応となる。しかし、実体が存在する限り、その実体の発生から消滅までの一連の過程(life cycle)を支配するイベントが存在する筈である。例えば、この、実体側からは、その実体のライフサイクル推移を引起すイベントを一気に決定でき、イベントの連鎖として表現される。(図10参照)

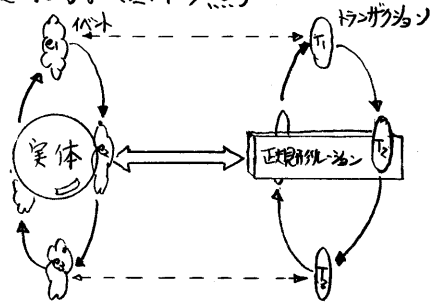


図9 イベントとトランザクション

(3) トランザクション正規化

一つのイベントによる、発生するトランザクションは、一つの正規形リレーションだけを対象とするとは限りはばり。多くの場合、一つのイベントは複数の対象実体をもつ。例えば、この、複数の実体に関する情報をもつ複雑なトランザクションが発生する(図11参照)。一つのトランザクションから正規形データに対応したトランザクションを生成することをトランザクション正規化と呼ぶ。正規形リレーションは、唯一の更新対象をもつ(図12参照)。

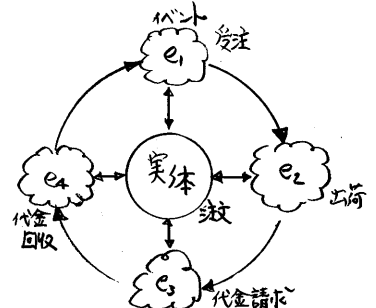


図10 実体ライフサイクル

(4) リレーション更新プロセス

データの更新プロセスはトランザクションが、その起動要因となる。非正規形トランザクションと対象とする更新プロセスは、イベントごとに固有なものとなり、業務ごと、イベントごとのプログラムに重複して散在するものとなる。(図13参照)

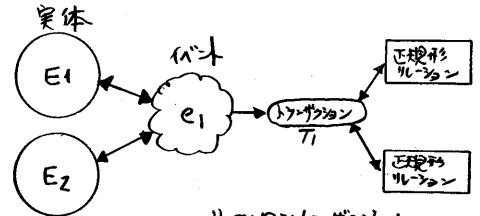


図11 非正規形トランザクション

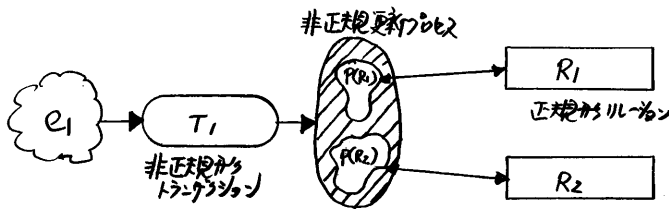


図13 非正規形更新プロセス

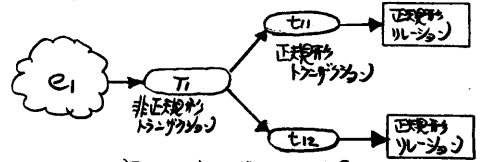


図12 トランザクション正規化

これに対し、正規形トランザクションは更新対象となる正規形リレーションが唯一である。したがって、その更新プロセスは正規形リレーションごとに用意でき、正規形データとこの更新プロセスを組み合せることが可能となる。(図14参照)

(5) 更新プロセスの定義

正規形データとトランザクションが定義されたらば、更新プロセス(P_i)は次のように定義できる。

$$P_i = (t_i, R_i, f(I_n))$$

- t_i : 正規形トランザクション名
- R_i : 対象リレーション名
- f : 更新オペレータ(U:更新, I:挿入, D:削除)および生成式
- I_n : 正規形対象アイテム名

更新プロセス定義は正規形トランザクション定義と同一の性格をもつ。(図15参照)

(6) イベントの分析と定義

イベントの抽出と分析の手順は図16に示すようなものとなる。図17に業務プロセスフローの例を示す。この記法はDELTA記法(14)に従う。図18に業務プロセスフローから抽出したイベント定義票の例を示す。

これらの情報から、イベントと実体、さらに業務プロセスと組織のクロスリファレンス(EEPOX: Entity-Events-Process-organization Cross Reference)を作成する。(図19参照)

実体ライフサイクルが定義されたら、データフローモデルを把握してこのデータフローからトランザクションを抽出し、

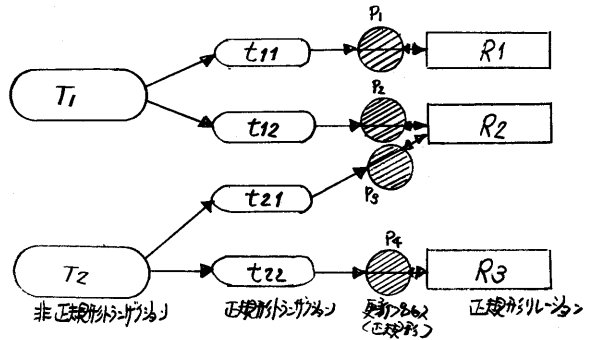


図14. 正規形更新プロセス

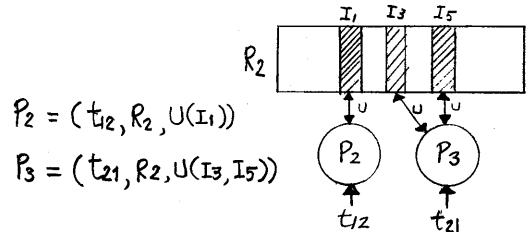


図15. 更新プロセス定義

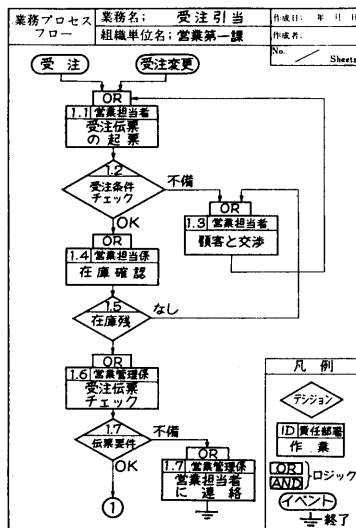


図17. 業務プロセスフロー

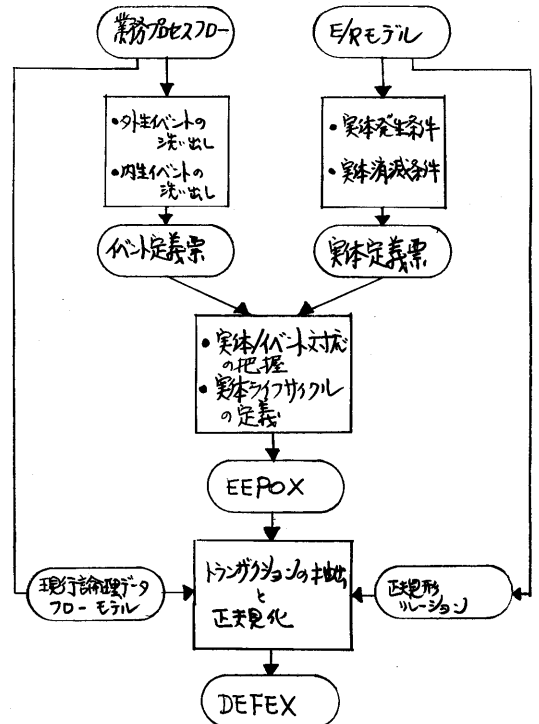


図16. イベントモデル化手順

(2) 汎用トランザクションプロセスの設計

トランザクションが唯一のレコードだけを更新するような制約を加えながらデータ更新。汎用トランザクションプロセス(UTP)を簡潔に報告は、M. Tamir 等による、2行なわれしている。(18) データを正規化し、その更新トランザクションも正規化するならば、トランザクションマッピングと更新プロセスに分けて汎用プログラムの設計が可能となる(図23参照)。

■ トランザクションマッピング; イベントごとに発生するトランザクションを正規化し正規形トランザクションを生成する。この時、図24に示すトランザクションマトリクスを参照する。

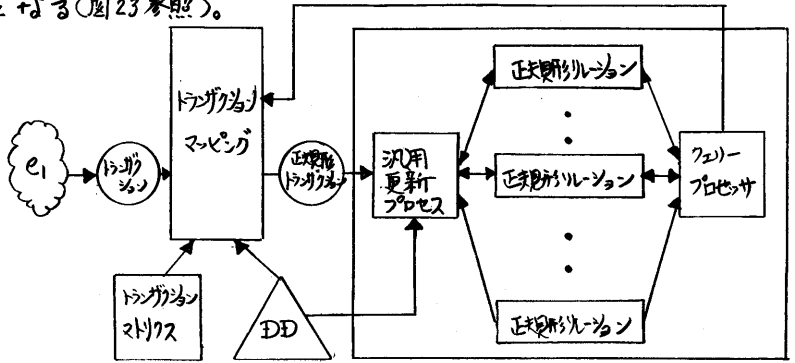


図23. トランザクションプロセス

■ 汎用更新プロセス;

正規形トランザクションによる該当する正規形リレーションに対し、指定された更新処理を実行する。トランザクションマッピングから引渡される正規形トランザクションのフォーマットは図25に示すものとなる。

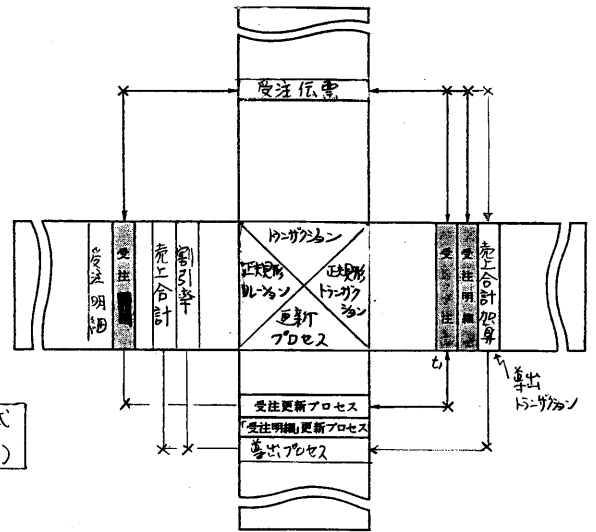
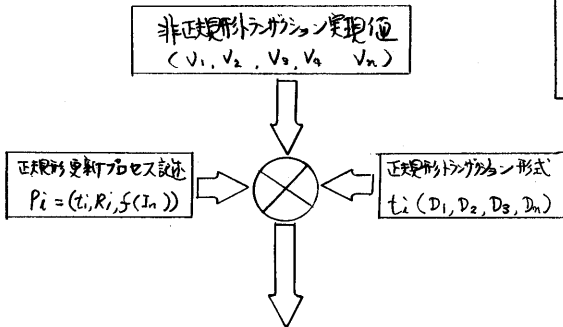


図24. トランザクションマトリクス



プログラムID	サブプログラムID	シリアル番号	対象レコード名	アクセスキー値	アイテム更新情報 1		2
					対象アイテム名	更新レコード指定	
						トランザクションデータ値	

図25. 正規形トランザクションフォーマット

■ 導出トランザクション

正規形トランザクションは更新すべき正規形リレーションが唯一つがある。しかし、正規形リレーション相互には、データ項目間に依存関係をもちあっている。例えば、次のようなリレーションを考へてみる。売上合計(---, 当月売上合計) 受注(---, 顧客ID, 受注金額, ...), 割引率(顧客ID, 割引率), 2つのリレーション間の

2. 当月売上合計 = $\sum(\text{発注金額} \times \text{割引率})$ と言う依存関係があるとき、売上合計リレーションを更新するためのトランザクションは、発注トランザクションを正規化するだけでは得られない。このような時、トランザクションマッピングルーチンは、割引率リレーションを読み込み、新しいトランザクションを生成しなければならぬ。このようなトランザクションを導出トランザクションと呼び、トランザクションマトリクスに定義しなくてはならない(図24参照)。

4. 抽象データ型に基づく新情報要求の分析と定義

正規化された、体系化されたデータと正規形データの更新プロセスが決定されれば、新システムの要求のほとんどは正規形データから非正規形の情報を作り出すリレーショナル操作が主体となる。新情報要求を標準化したデータの下で処理するために重要となるのは、情報要求分析過程と、標準データと情報要求の差異に対する処置を確実に行うことである。

図26にそのような手順を示す。また図27に情報要求に基づくデータ要求定義票の記述例を示す。

データ要求定義票	データフロー名またはユーザ名	データ要求タイプ	プロセス名	ユーザ名	作成者
正規形リレーション名	正規形リレーションのデータ構成	リレーション操作			構文記号
1 挿入	G	挿入先コード+挿入先日付+商品コード+商品数量+発注日付+受注金額	①-②-③-④		月別売上明細 +商品コード +1日-発行額 合計 +商品数量
2 削除	G	挿入先コード+商品コード+受注日付+受注金額+受注数量+納期	①-②-③-④		月別売上明細 +商品コード +1日-発行額 合計 +商品数量
操作ID	操作タイプ	操作条件			
S1	SELECT	挿入先コード			
P1	PROJECTION	挿入先コード+月別売上額			
S2	SELECT	S1と同じ挿入先コード			
S3	SELECT	商品コード			
P2	PROJECTION	挿入先コード+商品コード+受注日付+受注金額+受注数量-1			
C1(R)	CALL	受注金額/受注数量			
J1	JOIN	R1@R2			

図27. データ要求定義票

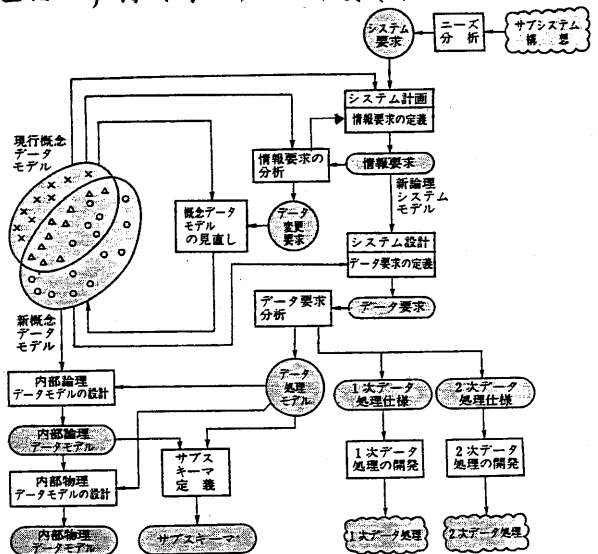


図26. 形式化されたデータによる新システムの設計過程

5. 所望に

本稿に述べた手順によるシステム開発事例は数件あり、現在、その適用途上にある。しかし、更新プロセスの標準化と一歩化に必要なのは、従来からのプログラム設計スキルに欠乏したインパクトを孕み、データ管理上の組織的統制の確立が技術以外の要因から、難しいこともあって、実務での適用は難しい。データ標準化と形式化は、データが共有資源であることへの認識の一般化と、データ資源管理組織の確立が大前提となる。今後、管理方法論との体系化を進めたい。

参考文献

- [1] J.D.R., 精緻化設計, 日経情報(1976) [7] C.Rolland et al, "Tools for...", Proc. 5th VLDB(1979) [13] V. De Antonellis et al, "Modelling Event...", Proc. 7th VLDB (1981)
- [2] K.T. Orr, Structured Systems, London (1977) [8] K.Grindley, Systematics, Comp. J., 9 (1966) [14] J.N. Warfield, "A Unified Systems...", Battelle Memorial, (1972)
- [3] C. Finkelstein, "Information Engineering", ComputerWld, 3/11-4/5 (1981) [9] M.W. Alford, "A Requirement...", Proc. 2nd ICSE (1976) [15] J.V. Gottag, "Abstract Data...", CACM, 7 (1977)
- [4] IBM, BSPマニアル, GE20-0527-3, (1981) [10] K.A. Robinson, "Entity/event...", Comp. J. 8 (1977) [16] K.S. Shanker, "Data structure...", COMPUTER, 9 (1980)
- [5] W.E. Riddle et al, "Behavior...", IEEE, Trans. on Soft. Eng. SE-4 (1978) [11] M.A. マン, 精緻化設計, 日経情報(1976) [17] B. Liskov, "Specification tech...", IEEE, Trans. Soft. Eng. 3 (1975)
- [6] P. Lindgreen, "Event directed...", pp.109-124, Formal Models and practical Tools for Information System Design, IFIP TC-8, (1979) [12] C.T. Rosengquist, "Entity Life...", Comp. T. (1982) [18] M. Tamir, "A Data-Based Application...", Proc. of 6th VLDB (1980)