

## データモデルを導入した仕様化技法

大 槻 繁

(株)日立製作所 システム開発研究所

データベースを中心としたソフトウェア・システムの仕様化技法を提案する。仕様記述モデルは、データベースの構造に対しては実体・関連モデルを、システムを構成する各プログラムの機能に対しては入出力データ構造の対応関係を基にしている。

仕様化の手順は、システム導入前の分析をワークシート類を基に行う実世界分析ステップ、情報の蓄積要求を抽出しそれを実体・関連図式で記述するデータベース仕様構築ステップ、及び、そのデータベースを主対象とする入出力を定義する機能仕様構築ステップから構成される。本技法で得られる最終成果は、形式的仕様であり、実体・関連モデルに従ったデータベースに対するデータ操作指定として直接実行可能である。

本論文では、技法の手順、仕様記述モデル、仕様の表記法について、事例を通して検討し、要求仕様の早期確認、安定性、プロトタイピング等の観点からこれ等の有効性について述べる。

### A SPECIFICATION METHOD BASED ON THE ENTITY-RELATIONSHIP DATA MODEL

SHIGERU OTSUKI

Systems Development Laboratory, Hitachi, Ltd.  
1099 Ohzenji Asao-ku, Kawasaki, 215 Japan

We propose a specification method for database oriented software systems. The representation models are based on the entity-relationship model for describing the database schema, and Jackson's input-output data structure correspondence for defining the program functions.

The design process consists three stages. The first stage is an analysis of the real world by focusing on worksheets. The second stage is a database structure construction abstracting requirements of the information management. The final stage is a construction of the function by defining input or output for the database. The final products of this method are formal specifications. Specifications are therefore directly executable in the environments, which support data manipulator towards the entity-relationship model.

This paper describes main phases of this method, description models, and these representation diagrams. The main phases of this method are illustrated by small example, and also advantages are discussed from the view point of the requirement validation, stability, and prototyping.

## 1. はじめに

ソフトウェアに対する要求の抽出不能性や不確定性の問題に対処するために、1980年代初頭のライフサイクル論争を契機として、オペレーショナル・アプローチ、自動プログラミングのパラダイム等のソフトウェア生産形態を抜本的に見直す提案が為されてきた。ここでは、生産形態自身を統一的な視点から整理し、抽象度の高い形式的な仕様記述モデル、並びに、その仕様化の手順<sup>4)</sup>について論ずる。

ここで述べるものは、データベース中心のシステムを対象とした仕様化技法である。データベースに対しては、この分野において概念設計モデルとして定評のあるP. P. ChenのER(実体・関連)モデル<sup>1)</sup>を、機能についてはM. A. Jackson(JSP)による入出力データ構造対応関係<sup>2)</sup>を仕様記述の基盤としている。

## 2. ソフトウェア生産形態

種々の生産形態を整理体系化するために、我々は、野木の《山モデル》<sup>3)</sup>を拡張し、RPC生産形態モデル(Real world / Problem specification / Computer Instantiation cubic model)を技法のフレームワークとして設定した。これは図1に示すように、抽象性、厳密性、最適性という3つの軸からなる座標を考え、この上に《実世界》R、《問題仕様》P、《計算機世界》Cを置き、これ等の間の軌跡を生産形態として表現するものである。

例えば、R. Balzerの自動プログラミングのパラダイム<sup>6)</sup>に基づく生産形態は、図中破線で示すよ

うに、《実世界》という曖昧で、かつ、具体レベルの問題を抽象化して形式的な《問題仕様》としてまとめ、これを具体レベルの《計算機世界》に機械的変換によって実現しソフトウェア・システムを得るという図式である。一般に、R→Pの過程を「仕様化」と呼び、P→Cの過程を「変換」と呼んでいる。

従来のウォータフォール型の生産形態は、図中点線で示すように、問題仕様は自然語或は絵等による非形式的で、かつ、計算機による実現を含んだ形でまとめられ、これを基に、モジュールの分割、構造化が進められ、次第に詳細化され最終的ソフトウェア・システムが得られる。前者の抽象化の過程を「要求定義(分析)」と呼び、後者の具体化の過程を「設計」と呼ぶことが出来る。

ここで論ずる仕様化技法で想定している生産形態は、図中太い実線で示されており、基本的には、Balzerのパラダイムに従ったものである。本技法は、仕様化の過程に関わるものであり、その最終成果である《問題仕様》は、十分な形式性を備え、実行(歩行)可能なものである。理想的な充分高い抽象レベルの問題仕様記述モデル(図中破線の生産形態)では、計算とデータとの間の差異もなくなり、より《変換》時の柔軟性

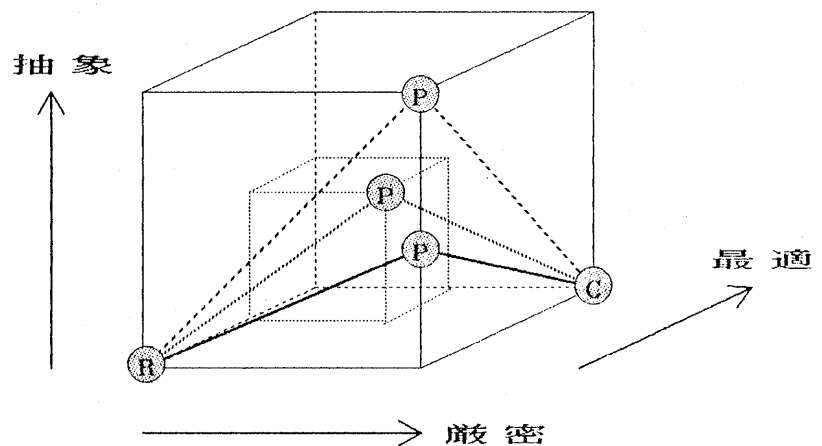


図 1 R P C 生産形態モデル

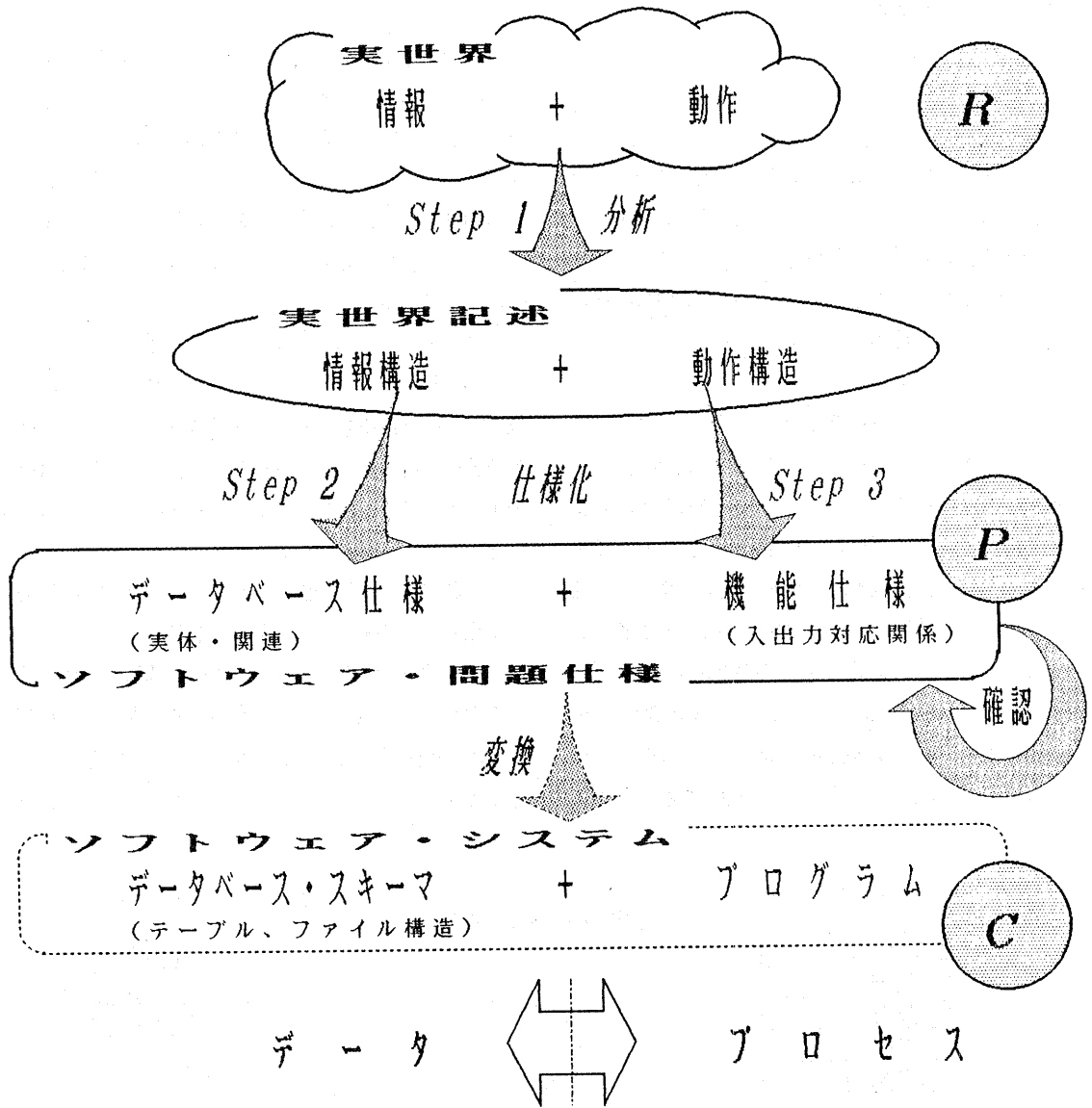


図 2 データモデルに基づく仕様化技法の概要

を確保することが出来るが、実用に耐え得るモデルを構築することは一朝一夕には成し得ない。

我々は、情報蓄積型の事務処理プログラムに代表されるデータベースを中心に据えたソフトウェアを対象分野を絞り、そのための仕様記述モデルを開発した。

このモデルの抽象度は、計算とデータとの融合化が起こらない限りで比較的高いレベルに設定されている。また、この分野のソフトウェアに対する要求は、データベースに蓄積すべき情報要求が比較的独立しており、その記述をモデル中で独立させて置くことは、実用上

の視点からも有効である。

### 3. 技法(手順)の概要

本技法の手順は、図2に示すように大きく分けて3段階から構成される。ステップ1は、ソフトウェア・システム導入前の実世界業務分析であり、ステップ2と3とは、それぞれ対象システムに対する情報蓄積要求と機能要求との抽出・仕様化である。

#### (1) ステップ1 実世界分析

先ずソフトウェア・システム自身の仕様化の前に、補助的な分析を行う。ここでは、システム導入前の業務分析を、既存の伝票やカード等のワークシート類を中心にして行う。実世界の情報を出発点とすることは、ユーザの要求を抽出するための第一歩として価値がある。なぜなら、ユーザは、現状業務についてならば、明確に述べる事が出来るし、また、この段階で言葉の誤解を解き、ユーザと開発者間のコミュニケーション・ギャップを埋めておくことは有効であるからである。

る。

ここでは、ソフトウェア・システムの機能の多くが、既存の手作業業務の効率化として位置づけられることを前提としている。貴重な業務上のノウハウは、この段階で抽出される。特に、手作業業務は、伝票、カード等のドキュメントによって運用されており、手順もマニュアル化されているか、いないにせよ明確に語り得るものである。

本ステップの詳細は、ワークシート類の収集や業務の洗い出しを行う《概要記述》、ワークシート類の抽象的なデータ構造を分析する《情報分析》、業務手順の概要を分析する《行動分析》、更に、この行動分析結果をデータ構造の対応関係によって詳細化する《機能分析》とから構成される。

この最初の《概要分析》において、現状業務の概要と手順並びにドキュメント形式が収集される。

実世界で行われている個々の単位業務は、その業務の入力となる情報と、出力となる情報、及び、それ等の間の対応関係によって表現することができる。この考えはJSPに基づいている。《情報分析》、《行動

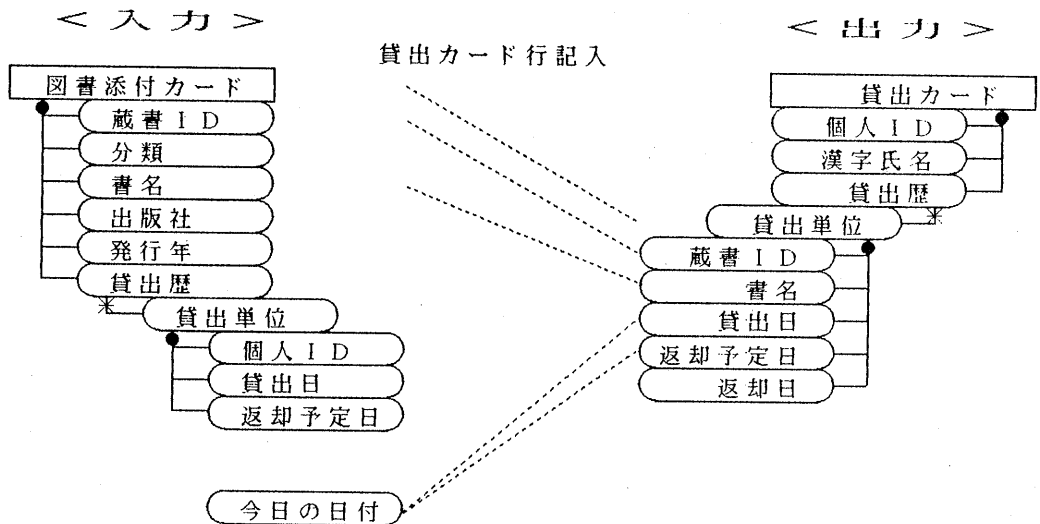


図3 実世界業務の機能分析例

分析》、《機能分析》は、この単位業務の記述を得るための一つの手順を与えるものである。情報分析段階では、概要分析段階で収集されたワークシート類の抽象的な構造を、ジャクソン・モデルに従って、直積、直和、列の組合せによって記述する。こうすることにより、ワークシートの本質的な構造が抽出され、レイアウトや色づけといった付随的な特性が捨てられる。行動分析段階では、業務手順の概略を、接続、選択、繰り返しの組合せによって記述する。これは、概要分析段階で得られた業務手順マニュアルや、実務あるいは管理者へのインタビューによって得られた情報を整理し、記述するものである。機能分析段階では、情報分析段階で作成した情報構造と、行動分析段階で作成した動作構造とを基に、曖昧性のない現状業務記述を作成する。ここでは、各単位業務を入出力情報と、これ等の間の対応関係によって規定する。

データ構造、或は、手順構造を記述する際にはJacksonモデルに従い、直積（連接）、直和（選択）、列（繰り返し）の基本構造の組合せによって記述している。また、本技法全体を通じてこれ等の基本構造の表記法は、●、⊕、\*のマークで統一されている。

図3は、カード業務を主体とした図書館の貸出業務に関わる実世界分析結果の例である。

## (2) ステップ2 データベース仕様構築

ソフトウェア・システムは、複数のプログラムの総体である。データベースは、このシステムに含まれるプログラムが共通に使う基盤となる情報を蓄積する所である。従って、変更の起こりにくい安定したもの

を構築しなくてはならない。本ステップが機能仕様構築より前にある主要な理由は、この要求抽出の優先度と安定性とによっている。

データベース仕様の記述モデルは、ERモデルを基にしている。本モデルは、人間の状態に対する認識過程との適合性が高いものである。安定したデータベース構造を得るためには、その場限りの実体や関連ではなく、実世界を反映した構造を導入しなくてはならない。

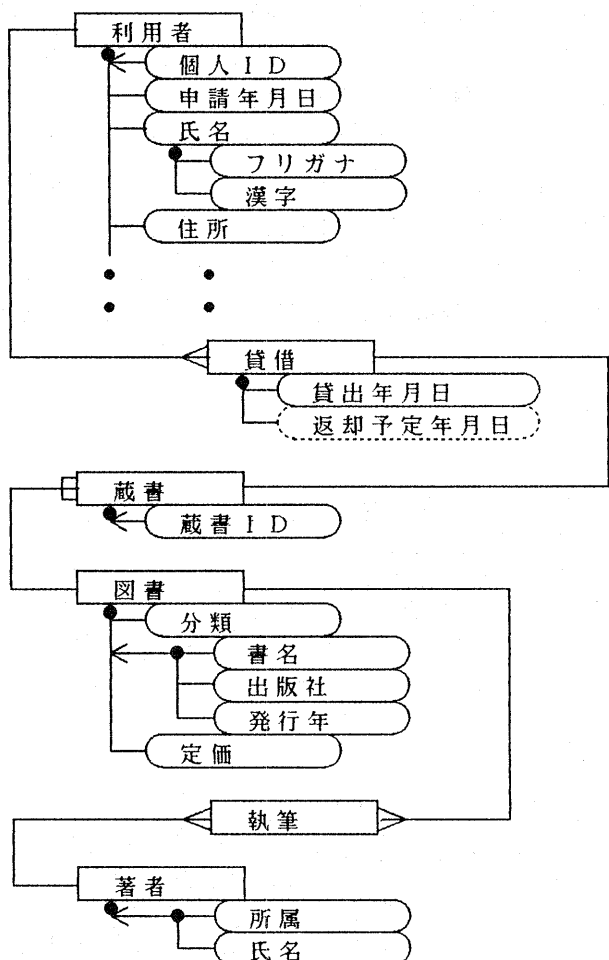


図 4 データベース仕様例

本ステップは、システムで蓄積すべき実体型と関連型とを抽出する《実体・関連記述》、これに属性を割り当てる《属性割当て》、種々の制約条件を付加する《制約記述》から構成される。

実体・関連記述段階では、まず実体の抽出を先に行う。実世界の構造を反映させるということは、実世界の《もの》に対応した実体をデータベースの中に設けるということである。このデータベースの中の実体のことを「ソフトウェア実体」と呼ぶ。ソフトウェア実体によって構成される世界は、これから構築して行くシステムの目からみた限定化された世界である。主な抽出の視点は、例えば、実世界の『個人』を、ソフトウェアの世界で『利用者』としてシミュレートすることに見られるように、当該システムで扱う情報への絞り込み（部分集合化）がある。

関連については、『貸出』業務は、『図書館』が『蔵書』を『利用者』に『貸与』することであるといった、業務の関与範囲からこれを抽出することが出来

る。これは、実世界での『貸与』という契約関係をシミュレートするものである。

《属性割当て》では、ステップ1で得られた実世界の分析結果を基に、ソフトウェア・システムへの範囲限定を行いながら実体関連構造に属性情報を割り付けることによってほぼ漏れなく記述することが出来る。すなわち、実世界の反映という視点から、属性情報のソフトウェア・システムで必要なものの取捨と実体の識別情報の同定が行われる。

ここで重要なことは、データモデル上で属性が構造（直積、直和、列）を持つことを許していることである。データモデルとして、関係形式モデルではなく、ERモデルを採用した主な理由は、実体概念が表現できる点と、この構造属性が導入できる点にある。これによって、実世界の情報構造との対応関係が明確になり、さらに、データモデルの抽象化のレベルを高く保つことが可能になる。

《制約記述》では、さらに、この実体・関連記述の

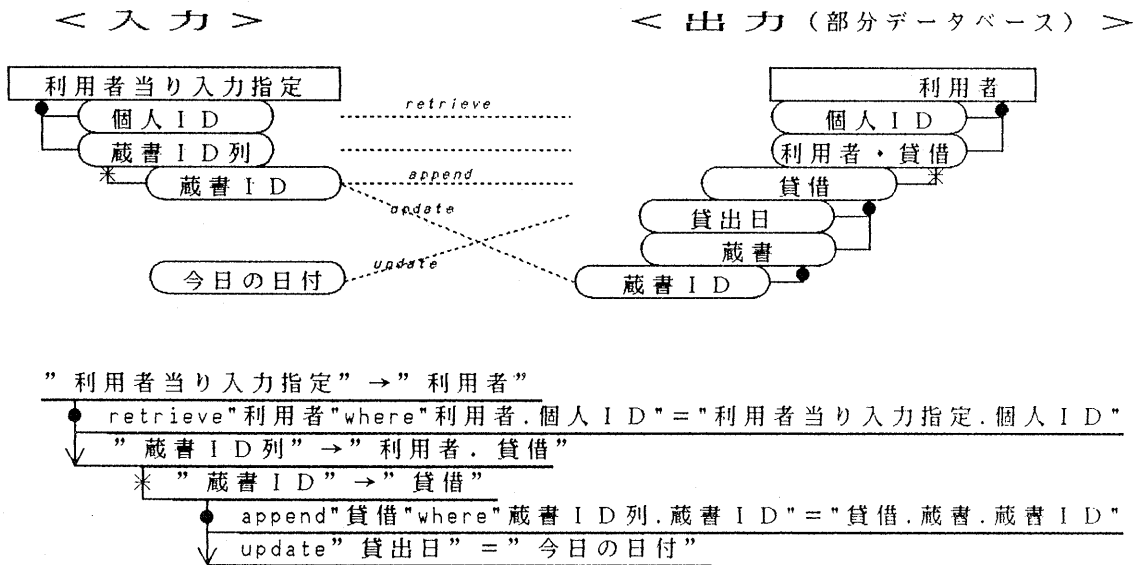


図 5 機能記述例

精密化を行う。ここでは、主に、関連のマッピングレ  
ーション、属性値間の従属性等が記述される。

図4は、データベース仕様の記述例である。

### (3) ステップ3 機能仕様構築

最後にソフトウェアの個々の機能をジャクソン・モ  
デルに従って記述する。即ち、機能は、入力データ、  
出力データ、並びに、これ等の間の対応関係によって  
規定される。

これを得るために、本ステップの詳細は、ソフトウ  
ェア・システムの個々のプログラムの機能の概略を記  
述する《概略機能記述》、データベースから当該機能  
を行うために必要な部分を抽出する《部分データベ  
ース記述》、プログラムの機能を部分データベースと入  
出力データとの対応関係によって記述する《機能記述》  
から構成される。

この時、入出力データ的一方或は両者がデータベ  
ースになっている場合が多い。データベースの構造は、  
ERモデルに従ったネットワーク構造であるが、一方、  
ジャクソン・モデルでは、データ構造は木構造である  
ことが要請される。このため、当技法では、データベ  
ースの一つのビューとして木構造を定義できるように  
している。これは「部分データベース」と呼ばれ、図  
5に示すように、データベースの一部を成すと同時に、  
これに対する木構造型の問い合わせ言語と見なすこと  
が出来る。

## 4. 考察

### (1) 手順の妥当性：

要求の抽出不能性に対処するために、ステップ1に  
て、実世界分析を行うが、これは定型化された手作業  
業務を前提にしている。実際には、非定型業務や、ソ  
フトウェアの実現、導入以前には、存在しない機能を

も抽出できなくてはならない。また、エンハンスのよ  
うに、既存のソフトウェアさらにはデータベース・シ  
ステムに対して拡張を行う場合は扱っていない。現段  
階では、ステップ1は、技法全体からみて補助的な位  
置づけになっている。

仕様化の最も重要なステップはデータベース仕様構  
築ステップであり、これが、機能仕様構築ステップよ  
り先行していることは、仕様の安定性からみて妥当で  
ある。より汎用的な技法であるJSDにおいてもモデ  
リング工程を先行させ、システムの機能は後から付加  
される手順を採用している。

### (2) 仕様モデルの妥当性：

直積、直和、列という構造を入出力データ構造、属  
性データ構造、手順構造に対し統一的に導入し、体系  
の簡便化を図っており、表記法もこれに従っている。  
無論、仕様には、タイミングやコントロールにかかわ  
る本仕様モデルだけでは記述し得ない事柄も多いが、  
本稿で述べた様な事務データベース・システムでは、  
かなりの部分に適用可能と考えられる。

データベース・モデルとして、属性に構造を導入し  
たERモデルを採用したが、これは、前節で述べたよ  
うに、実世界との対応と仕様の抽象度の点から望まし  
い性質を持っている。また、このデータモデルの上  
には、完備なデータ操作体系<sup>5)</sup>を構築することが出来る  
ことも利点の一つである。

機能の表現にジャクソン・モデルを採用しているが、  
この方法では、いわゆる構造不一致の問題が起り得  
る。構造不一致が起こった場合には、Jacksonの提案ど  
うりに中間ファイルを設定して回避すればよい。しか  
し、ここで提案した技法では、データベース・モデル  
と融合化を図っているためにこの問題は起りにくい。  
境界不一致は、一般に、抽象化のレベルの異なったも  
のが混在するときに起こるが、これはステップ2で概  
ね回避される。順序不一致は、もともとデータベース  
を仕様記述モデルに含んでいるために起らない。具

体的には、データベースと部分データベースとの間で自動的に解消される。また、脈絡不一致は、順序不一致の特殊な場合と見なすことが出来る。

### (3) プロトタイピングの視点:

本モデルに従った仕様編集や、最終仕様に対する解釈実行機能を支援系として提供すると、プロトタイピング支援システムを構築可能である。この歩行可能な仕様を得ることが可能である裏付けは、ジャクソン・モデルにおいて入出力のデータ構造とそれ等の間の対応関係からプログラム構造が導出出来ること、部分データベースがERモデルに従ったデータベースへの問い合わせ言語（基本演算の体系）になっていることである。

開発過程の後半に当たる変換・設計（実現）では、データベースはChenの方法に従って、概念、論理、物理へと変換し、機能は、JSPに従って、中間ファイルの解消、最適化を進めて行けば最終的なソフトウェア・システムを原理的に得ることが出来る。

## 6. おわりに

本稿では、データベースを中心とした事務処理システムを対象とした仕様化技法として、ジャクソン・モデルとERモデルとを融合化した仕様記述モデルとその上の技法について論じた。安定したデータベース構造の構築を主目的としており、更に、最終的な仕様は、実行(歩行)可能なレベルまで詳細化することが出来る。また、JSD技法との適合性もよいし、適用範囲を限定した実用的な視点から見ても有効な方法と考えられる。

## 【参考文献】

- [1] Jackson, M. A. : "Principles of Program Design", Academic Press, 1979
- [2] Chen, P. P. : "The Entity-Relationship Model toward a Unified View of Data", Trans. Database Syst. 1, 1, Mar. 1976
- [3] 野木 : 「要求定義技術の最近の動向」, 情報処理, Vol. 27, No. 1, 1986
- [4] 大槻 : 「データ中心型ソフトウェア仕様化技法の一考察」, 情報処理第35回全国大会, 1987 9月
- [5] Parent, C. and Spaccapietra, S. : "An Algebra for a General Entity-Relationship Model", IEEE Trans. on Soft. Eng., Vol. SE-11, No. 7, July 1985
- [6] Balzer, R., Goldman, N. M., While, D. S. : "Operational Specification as The Basis for Rapid Prototyping", SEN7-5, 1982