ソフトウェア工学 86－3
（１９９２．７．２０）

非構造データ析出のモデル

米川　清 *

* 三井情報開発株式会社

データモデルについて、筆者の現状認識を述べる。
次に、非構造データは、構造化分析によるデータモデルと別体系のモデルとし、２つのデータモデルを提唱する。
上記の概念整理は、言語理論やポスト構造主義から強い影響を受けている。
以上から、大規模情報システムを巨大な概念装置として再利用し、ＤＯＡの立場から、２つのデータモデル分類のメソッドを報告する。

Data Modelling for unstructured data

Kiyoshi Yonekawa *

* Mitsui Knowledge Industry Co.,Ltd

On my thesis, I introduce a new paradigm of Software Engineering, which use linguistics and post-structuralism applied to social science.
The paradigm is composed of two Data Model, that is, structured Data Model and unstructured Data Model.

1.はじめに

大規模ビジネス情報システム構築に従事した実務経験を通じて、大規模情報システム全体を概念装置として再利用し、データ中心のアプローチ(DOA)によるシステム再構築を想定したデータ分析の私論を報告する。既存の大規模情報システムは、構造化分析により設計されたソフトウェア資産とする。

筆者は、ＯＯＤＢの理論的研究が主に計算機科学からの支援により進展する事を期待しつつ、直観的には実世界そのものを内部構造化する事に懐疑的である。[1)]

上記の筆者の矛盾した思惟規範を予め明記しておく。

筆者は、実世界の動的側面への対応について、当面は従来の関係データモデルを改良しつつ、ＤＯＡ設計との調整を図る折衷主義が実務的であると思料する。

2.現状認識

現在のソフトウェア資産は、構造化分析、構造化設計、構造化プログラミングから構築され、ソフトウェアプロセス成熟度の改善が維持される。また、各フェーズを支援するCaseツール群も豊富である。プロセス中心のシステム設計は、今後も評価される。

その一方、ＯＯＤＢ宣言をトリガーとして、先進的技術（マルチメディア、ＣＡＤ etc）への応用面での制約も指摘される。

筆者は、ＤＯＡの立場から、以下の問題提起を行う。

(1) 情報構造を分析する話題から、プロセス中心設計のソフトウェア資産にリバース・エンジニアリングを試みても、シンタックスは抽出されるが、ＤＯＡ設計に於ける有益なデータは救済できない。意味データモデル構築では、データをコード化する過程で捨象されるコードの剰余価値が大切である。従って、数学的形式化と平仄をあわせ、言語学や記号論の理論を援用した概念整理が重要である。

(2) ＯＯＤＢはデータベース設計者がオブジェクト自体の構造を予め決定するので、ＯＯＤＢにより実世界のビヘイビアに動的応答が可能になるとは、にわかに考え難い。即ち、オブジェクトを内部構造化する事自体が、構造化のパラドックスとなり、むしろポスト構造化分析の手法を模索すべきではないか。

以上から、筆者は実世界の個々に差異を有する運動を抽象化し、構造化分析を行う事に悲観的観測を持つ。意味データモデルでの意味内容とは、ある時点での解釈や現象を示すのであり、解釈には幻影が宿る。時間の経過により意味内容や情報価値は瞬時に変容する。

3.ポスト構造化分析と２つのデータモデル

構造とは「データ集合上のｎ項の関係のネットワーク」として理解するなら、記号論でいう超越的中心である「ゼロ記号」の存在をメタファーとする。本論文では、is a kind

of でのスーパークラスが「ゼロ記号」である。

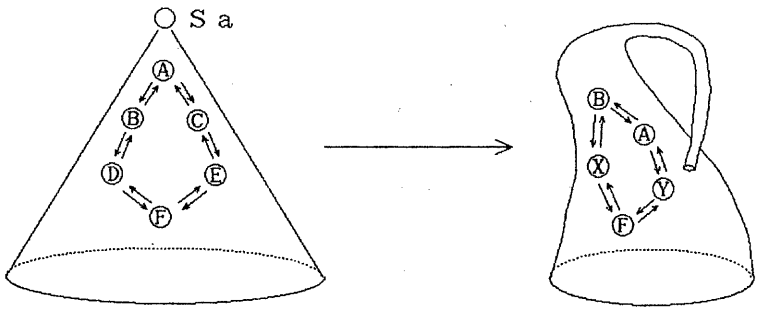

図-1　クラインの管の援用

図-1左のＳＤＭ（Semantic Database Model）[2] では、Ａを「ゼロ記号」とする。

実世界の動的なビヘイビアとの類推ではＡを「貨幣」とすれば、Ｂ→Ｄのプロセスを経て、Ｆの「商品」に化体し、「商品」が売却され入金すれば、Ａの「貨幣」へまた、ジャンプする。上記は、キャッシュ・フローのサーキットとして有益である。[3]

然し、営業循環を反復する過程では外部要素が内部へ侵攻したり、内部要素が外部世界へ駆逐されたり、データ自体が自己組織化もする。その結果、構造自体に「揺らぎ」が生じ、Ａが「ゼロ記号」として存在しえなくなり、サブクラスのＢ（例えば資産）がゼロ記号に移動したりする。結論的には構造化モデルとしての歪曲は否定出来ない。即ち、図-1の右にシフトする。上記モデルは、ドイツ数学者クラインのトポロジーの援用である。[3]

以上から、オブジェクトを内部構造化する事を筆者は否定する。筆者は関係モデルの宣言的なデータ操作やＯＯＤＢのユーザーインターフェスの利点を重視しつつ、構造化モデルを「データを一定の状況に閉じこめる事」と定義する。

構造化モデルとして認識不可能なデータは、全て非構造データモデルとして取り扱う。非構造データモデルでは、内部構造や集合関係の概念を全て排除する。

従って、２つのデータモデルが共存する事になるが、[4] 両者の関係を図-2に示す。

図-2では、構造化モデルをメタレベルとし、非構造データモデルをオブジェクトレベルに位置付ける論理階型（logical type）とし、相互参照を遮断する。

上記の分類手法は、ソフトウエア工学に近似にして、遠く離散している社会科学に於けるポスト構造主義の学説に依拠する。レヴィ＝ストロースらの構造主義では「純粋数学」の集合論を理論基盤として、中心概念は「構造とは関係のネットワーク」として分析した。しかし、このモデルでは静的世界は表現可能であるが、動的世界への対応の視点が欠落していると指摘される。

以上の論理を図-3に示す。

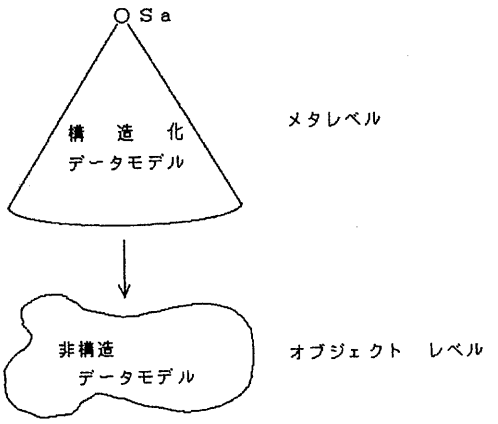

図-2　　データモデルの論理階型

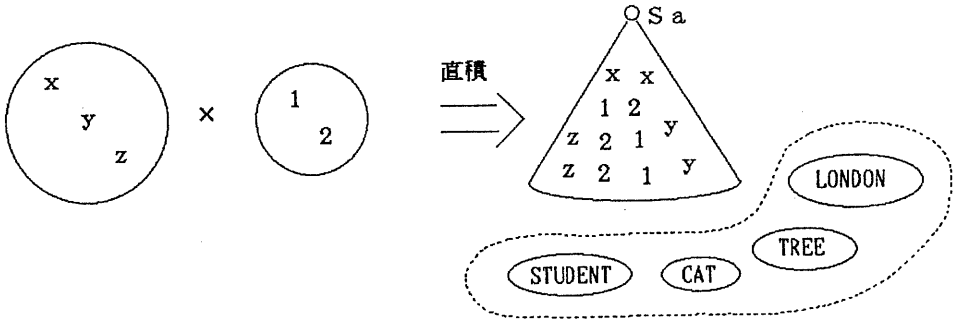

図-3　　構造モデルとリゾーム

図-3で破線部で示す実世界に散在する相互関連性を欠くカテゴリーをどの様に認識すべきか。

例えば、「学生Aは猫を飼っている」

「学生Aの家には、大きな木がある」

そして、「学生Aはロンドンに在住している」かもしれない。

然し、上記の論理の筋道は論理学上は、非論理的である。

クラス概念からは、「学生」「動物」「植物」「地名」のメンバーである。

従って、ＳＴＵＤＥＮＴとＣＡＴの関係は、φである。この破線部は、ポスト構造主義ではリゾームとして定義され、自然発生的で脈絡を欠く空集合と解釈する。

筆者は実世界を構造化モデルとリゾームの無限集合としてとらえ、リゾームを非構造データモデルと定義する。

構造化モデルでは、「記号表現・記号階型・配列の規則」を定義し、非構造モデルは、「記号内容・記号の剰余価値・記号値で表現不可能なもの」を含む。

4.2つのデータモデルと再利用技術との係り

本節で、大規模情報システムを再利用した、2つのデータモデル導出の為の方法を述べる。筆者は、既存の巨大ＤＢに蓄積されたデータ資源をあえてジャンク（junk）とみなし、大規模情報システムでのデータ処理過程の中からトランザクション・データが最大レングス（length）になった時点でのデータを採取し、各レコードから構造的項目を析出し、構造の外にとり残された非構造項目を明瞭にする手法をとる。この非構造データは、利用者が情報構造分析の上で、関心を有するドメインを特定する過程で有益である。更に、実体の特性や振舞いを事前列挙する事も可能とする。[5)]

図-4（下図）は、プロセス中心設計による大規模情報システムのマクロ概念図の１例である。この様なシステムは、利用者からは巨大なブラックホールと化しており、利用者の仕様変更にスムースに対応出来にくく、利用者の情報生産上のネックとなる。

筆者は、大規模情報システム全体を概念装置ととらえ、現在蓄積されているデータ資源よりトランザクション・データに着目し、データが計算結果やプログラム・エディット等を通じて最も冗長となったポイントに「引込線の仕組み」を仮設し（図-4の破線部）、一定期間データ・リポジトリにプールする。

プロセス中心設計で構築された巨大ＤＢは、「死んだデータ部品の集積」として捨象する。

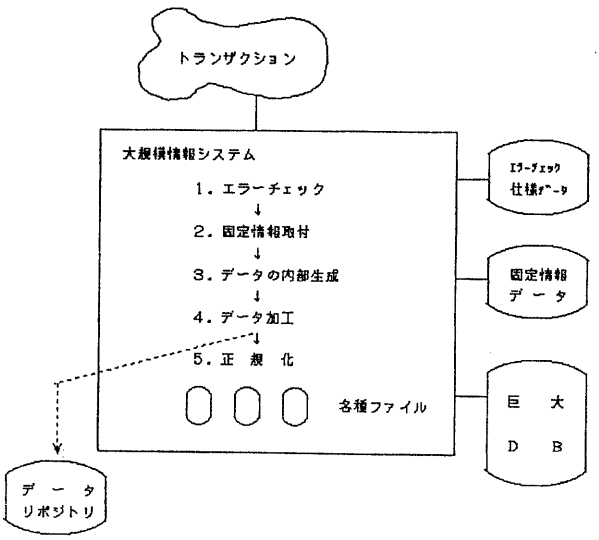

図-4　　大規模システムの概念図

即ち、関数従属に基き正規化され、参照整合性などの設計者の意図が深く介入した巨大ＤＢから、複合オブジェクトを生成するなど、物理学のエントロピーでの熱力学第２法則に背反する事を想起させる。筆者は、データ・リポジトリを全てのデータの集合体とし、この集合体から構造化データを抽出してゆき、構造の外にとり残された残余部分を非構造データとして認識する。（図-5）

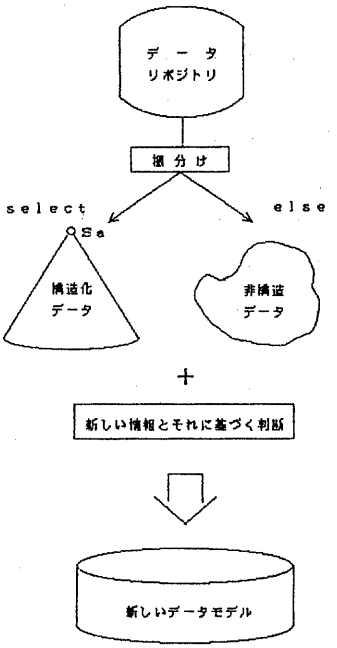

図-5　　データモデルの分類手法

以下に、非構造データ析出の手順を述べる。

(1)　データ・リポジトリとエラーチェック仕様データの突合せ

エラーチェック仕様データには、項目間チェック、コードチェック、レコード間チェック等の整合性チェックの全てがメタ情報としてデータ化されている。[6)]　データ・リポジトリにプールされたデータは整合性チェックのフィルターを通過している以上、エラーチェック仕様モデルに登録のある項目は、構造化データである。

(2)　データ・リポジトリと固定情報ファイルの突合せ

固定情報ファイルには、標準化された利用者のコード及びその属性、処理パターンが登録されている。固定情報ファイルに存在するデータは、全て構造化データである。

(3)　データ・リポジトリと共有データの突合せ

トランザクションの最終目的地（Data･destination）は、各種ファイルであり、ファイルは各種ビューと対応づけられ目的別に設計される。筆者は各種ファイル間の共

有資源に着目する。（図-6）

ファイルA∩ファイルB　　ファイルC⊂ファイルD

図-6　　共有資源の抽出

図-6の斜線部は構造化データである。

(4)　データ・リポジトリとプログラム生成項目の突合せ

情報システムでは、処理目的を明示させる為に、処理区分、種別、マーク等が内部生成される事も、ままある。この場合、リバースにより、当該ロジックの抽出は可能である。諸条件及び内部生成コードも構造化データとみなす。

上記手順から、残余項目の非構造データが析出される。導出された非構造データから、共通部分を排除し、より純粋化する。また、公理主義の立場から、推論規則により、関係を演繹してナビゲーションの支援がえられるなら、構造化部分を析出する。

更に、論理学上の判断の蓋然性を追求する「確率の論理」の手法を適用すれば下記の通りである。

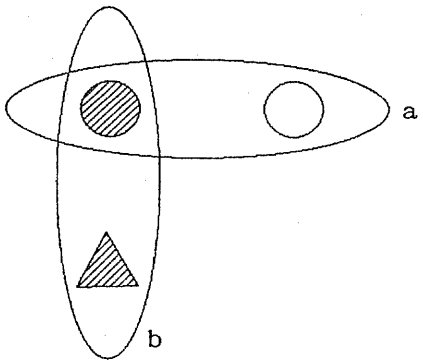

図-7　　確率の論理

図-7のモデルでは、「円のグループ」としてａの立場をとるか、「斜線のグループ」としてｂの立場をとるか、非構造項目群とするかは、設計者の恣意性に委ねる事となる。[7)]

5.おわりに

本論文で、筆者は非構造データモデル（リゾーム）を提唱した。ここでの論旨は、データの部品化や継承性を論ずるものではなく、既存システムを再利用した実体データ分析上のメソッドを述べた事を確認しておく。

概念定義は行ったものの、非構造データモデルの実現可能性については、その手がかりさえつかめない状況である。

実世界を全般的に俯瞰すると、言語学で云う形を持たない言葉が散見される。「美」や「善・悪」という言葉が、それである。

ソフトウェア工学は、これらを論ずる研究ではないが、実世界の中からどの範囲迄を主体（entity）として取り出すかは議論する必要がある。[8)]

本論文では、実世界の中から利用者の関心領域を鮮明にする方法として、ソフトウェア資産そのものを概念装置として再利用する動機から、未消化でまとまりを欠く私見を発表させて頂いた。

参考文献

1) 二木，大堀，柴山，安村，竹内，上田，村井，萩谷：理論は実践を導けるか，実践は理論を生かせるか？，情報処理，Vol.33, No.3, pp272-289, (1992).

2) Hammer, M. and Mcleod, D.: Database Description with SDM: A Semantic Database Model, ACM Trans. on Database Syst., Vol 6, No.3, pp.75-92 (Sep. 1981).

3) 浅田　彰：逃走論，筑摩書房(1984).

4) 有澤　博：意味データモデルの最近の動向，情報処理，Vol.32, No.9, pp1023-1031 (1991).

5) 堀内　一：システム開発パラダイムと高水準データモデル，情報処理，Vol.32, No.9 pp1014-1022(1991).

6) 米川・会田：エラーチェック仕様のデータモデル，情報処理学会ソフトウェア工学研究会，1992年5月(1992).

7) 沢田允茂：考え方の論理，講談社(1989).

8) Brodie, M.L. and Ridjanovic, D: On the Design and Specification of Database Transactions in On Conceptual Modelling (Brodie, M.L., Mylopoulos, J. and Schmidt, J.W.Ed.), pp272-312, Springer-Verlag (1984).