

非定型的な業務のための情報システムの 開発におけるデータモデリングの役割

A specific role of data modeling in developing information systems for unstructured business functions

小 幡 孝一郎

Koichiro Obata

1. データモデリングの発展経緯	11
2. データモデリングのパラダイム	12
2. 1 パラダイムとは	12
2. 2 データモデリングのパラダイムの概要	14
2. 3 機能主義者のデータモデリングに関するパラダイム的な前提	16
2. 4 社会的な相対主義者のデータモデリングに関するパラダイム的な前提	18
3. データモデリングの実践のあり方	23
3. 1 伝達的な要求定義の考え方とその問題点	23
3. 2 これからの情報システムの要求定義におけるデータモデリングの定義	25
4. まとめ	28

1. データモデリングの発展経緯

一般にモデルとは本質的に複雑なものを理解するために、その特定の面だけに着目してそれを抽象化した表現である。抽象化では常にある目的にとって重要な要素だけが抽出され、それ以外の要素を捨てられる。

データモデルは情報システム開発の様々な段階でそれぞれ異なる目的のために作成され使用される。例えばシステム分析の段階で使用される概念データモデルの目的は、ユーザーの情報システムに対する要求の中のデータに関する部分、即ちデータ要求をユーザーと開発者の双方に対して簡潔に且つ意味のある方法で表現することである。(Teorey 1994)。

1960年代の中頃からデータベース技術の進展と相まって、情報システム開発におけるデータベースアプローチの重要性が主張されるようになった。これは企業活動の様々な分野を支援する複数の情報システムの開発を全体として効率的に推進し、かつ情報を出来る限り組織全体で共有出来るようなシステムを開発するためには、機能分野毎に別々にシステムを開発するという伝統的なアプローチではなくて、個別のアプリケーションから独立したデータの定義に基づいて組織の情報要求を統合的表現にし、それを基盤として個別の情報システムを開発する事が必要であるという考え方である。データモデリングという考えが関心を集めるようになったのはこのころからで

あると考えられる。

これをデータベースアーキテクチャに関する国際標準の確立という形で支援しようとする提案が2度にわたって行われた、その第1回は1971年にCODASYL DBTGレポートによって提案された標準データベースアーキテクチャ（2層スキーマ）である。これは、論理レベルにおいて全体のデータベーススキーマとユーザーのためのサブスキーマとを分離することによって、たとえデータの記述の一部に変更があっても、それがそれが自分のサブスキーマの範囲外であればユーザーは自分のアプリケーションを変更しないで済む、ということをも可能にしようというものであった。

しかしこれには次の2つの基本的な問題点があった。

- ・データベーススキーマを記述するモデル言語が異なる場合、それらの対応をとることが容易でない。
- ・モデル記述言語がレコード指向であるために、データの論理的な性質を十分に表現できない。

これらの問題を解決すべく、1975年に次の世代のデータベースアーキテクチャの標準としてANSI/X3/SPARCの3層スキーマ（ANSI/X3/SPARC, 1975）が正式に提案された。これはデータベースを内部スキーマ、概念スキーマ、及び外部スキーマの3つのレベルに分けて管理すべきであると規定している。内部スキーマではデータの物理的な記憶領域の編成方法が定義される。外部スキーマはユーザーに最も近いレベルであって、各々のユーザーが見る見方でデータが記述されている。概念スキーマはこの二つのレベルの間にある間接的な表現であって、データに対するコミュニティユーザーの見方（ビュー）であると定義された。この3層スキーマを提案した人たちは、概念スキーマはデータの意味を捉えなくてはならないと考えていた。

しかし残念なことにANSI/X3/SPARCの提案からは、概念スキーマが何を意味しているのかは完全には明らかになっていない。上記のようなコミュニティユーザーの見方、外部レベルと内部レベルの間にある間接層、データの意味の規定、等の説明によって何をいいたいのかという精神は理解できるが、尚多くのことが答えられないままになっている。例えば外部スキーマの要素は必ずそれに対応する概念スキーマの要素を持っているが、それにも関わらず概念スキーマとはある程度独立にデータを定義できることが期待されている。さらに概念スキーマが特定のデータモデル（記述方法）に依存すべきであるのか、それともそれとは独立なデータの見方であるべきなのかは全く明らかでない。

このような潜在的な問題点があるにも関わらず、概念スキーマという考え方はデータモデリングの分野に大変大きな刺激を与えた。これはデータ要求をもっと高いレベルで表現するための様々なモデル記述方法（例えばオブジェクト指向モデルのような）が提案されたことから明らかである。

一方これを契機として、これらのモデル記述方法の根底にある考え方に基本的な違いがあることも明らかになってきた。この興味ある問題について、Hirschheimらの「情報システムの開発とデータモデリング：その概念的及び哲学的な基盤（R.Hirschheim, H.K.Klein, K.Lyytinen, 1995）に包括的な分析がある。次章はその一部の紹介である。

2. データモデリングのパラダイム

2.1 パラダイムとは

ある専門家のコミュニティが採用していて、それに基づいてその人々が認識を共有することが

出来、また共同の実践活動に従事することが出来るようになるような、最も基本的な前提条件の集まりをパラダイムと呼ぶ。パラダイムを構成する代表的な要素は、知識及びその獲得方法に関する前提と、専門家の活動の対象となっている物理的及び社会的な世界に関する前提である。この種類の前提は科学的な研究に携わる人々だけでなく、全ての専門家集団について存在する。開発の専門家はシステム設計のために、(対象とする)社会的な世界に対する探索を行い、またシステム実現のために、同じ社会的な世界に対する介入を行うことが必要になる。そこで2つの関連する前提条件を区別するのが自然である。即ち、システムの開発者がシステムを設計するために必要となる知識を獲得する方法に関する前提(認識論的な前提)と、開発者が社会的な世界をどのようなものであると見るかに関する前提(存在論的な前提)である。

2. 1. 1 パラダイムの認識論的な前提と存在論的な前提

BurrellとMorganに従って、(社会的な世界に関する)パラダイムの前提は、主観主義対客観主義という次元と、秩序対コンフリクトという次元の、2つの次元に従って分類されることに着目したい。いうまでもなくこれらの標識(呼び方)は哲学的な立場を明らかに示すための単純化に基づくものである。

第一の次元での客観主義の立場の本質は“自然科学におけるモデルと方法を人文科学の研究に適用する”というところにある。つまり客観主義においては、社会的な世界をあたかもそれが自然的な世界であるかのように取り扱う(Burrell & Morgan 1979)。客観主義者は、自然的な世界と社会的な世界の両方に全く同じように適用される汎用の科学的な方法、という考え方に固執している。

これに対して主観主義の立場は、自然科学の方法を社会的な世界の研究に適用するのは適切でないとして、各人の主観的な経験の深みを探究することによって、人間の生活の基盤を理解しようとする。その基本的な関心は、各個人が自らの存在する世界を創り出し、変更し、そしてそれを解釈する方法を理解することにある(Burrell & Morgan 1979)。従ってHabermasが指摘するように、自然科学と人間文化の科学という二元論を認めることになる。

第二の次元での秩序ないしは統合主義者の視点は、秩序、安定性、統合、コンセンサス、機能的な調整などによって特徴づけられる社会的な世界を強調する。

これに対してコンフリクトないしは抑圧視点は、社会の変化、コンフリクト、分裂、制圧などの面を強調する。

この2つの次元を組み合わせると、次のような4つのパラダイムが出来上がる(図参照)。

即ちQ；

- ・機能主義(客観主義&秩序)
- ・社会的相対主義(主観主義&秩序)
- ・急進的な構造主義(客観主義&コンフリクト)
- ・ネオヒューマニズム(主観主義&コンフリクト)

この特定の枠組みを選択したのは、それを使うことによって、情報システムの開発に対する様々な方法について、単純でしかも哲学的な基盤に支えられたやり方で、それぞれの前提の違いを識別することが出来るからである。

2. 1. 2 4つのパラダイムの特徴

ここに紹介する4つのパラダイムは、はじめBurrellとMorganによって組織及び社会的な研究という文脈の中で見出されたものであるが、我々はそれが情報システム開発（及びデータモデリング）の領域においても同じように有効であると考え。しかし情報システムの世界では、これらのパラダイムの多くは常識的な信念や背景的な知識の蜘蛛の巣の中に深く根付いているので、それらが実際にどの様なかたちで影響しているかを示すことは容易ではない。

・機能主義（客観主義&秩序）

このパラダイムは、現状、社会的な秩序、社会的な統合、コンセンサス、要求の満足、合理的な選択、等を説明することに関心を持っている。つまり、社会システムを構成する個々の要素が全体として一つの機能を果たすために、どのようにして互いに作用し合うのかを説明しようとしてとめる。

・社会的相対主義（主観主義&秩序）

このパラダイムは、個人毎の意識と主観という領域の中で、また行為の観察者ではなく、社会的な行為者という枠組みに基づいて、社会的な社会をとらえようとする。この視点に依れば、“社会的な役割や制度は、人々が自分の世界に付与する意味の表現として存在する”。(Silverman 1970)

・急進的な構造主義（客観主義&コンフリクト）

このパラダイムは、社会や組織に現存する制約条件を打破する或いはそれを超越することの必要性を強調する。ここでは主に経済的な力関係の構造の分析に焦点が当てられている。

・ネオヒューマニズム（主観主義&コンフリクト）

このパラダイムは、急進的な変化、解放、潜在的な可能性を求め、変化を理解するためには、様々な社会的或いは組織的な勢力が演ずる役割が重要であると主張する。そこでは解放に対する、イデオロギー、権力、社会的な強制や制約など、あらゆるかたちの障害に焦点が当てられる。

2. 2 データモデリングのパラダイムの概要

データモデリングという探究のプロセスには、伝統的な科学の理論の構築との間に、本質的に類似の性質があるので、上記の各パラダイムはデータモデリングのツールや方法に対して、それぞれ大きな影響を与えてきた。

データモデリングの領域では、パラダイムを分類するのに使われる2つの次元の中で1つだけ、即ち客観主義と主観主義とを区分する次元だけが顕著である。あらゆるデータモデルは皆その対象とするシステムの秩序視点を前提とし、また殆どのデータモデルは現実を表現することに焦点を当てている。即ちデータモデリングの世界では機能主義の視点が圧倒的に多数を占めている。組織的なコンフリクトの中でデータモデルが何らかの役割を果たす、という考えは殆ど注目されていない。従って急進的な構造主義のアプローチは見出すことが出来ない。もしそのような視点が存在するならば、データモデルは敵対的な討論を通して真実を発見するのをどの様に支援できるか、という主張によってそれを認識できたはずである。データモデリングの文献の中には、様々な異なるユーザーの視点や異なる意味論があることに注目しているものはあるが、これらの中で、データモデリングを急進的な変革のために使うことが出来る、と述べているものは1つもなかった。従ってデータモデリングにおいては、急進的な構造主義ではなくて、機能主義と社会的相対主義を識別することは可能である。後者は現実をモデル化するのではなくて、異なるユーザー言

語のルールをモデル化することに焦点を当てている。モデル化の対象が現実であるかそれとも言語であるかという違いは、パラダイムを分類するときに利用した客観主義と主観主義の違いに対応するものである。後に見るように、データモデリングにおけるネオヒューマニスト視点を確立するための基盤的な作業は行われている。しかし現時点ではまだ、ネオヒューマニストパラダイムが存在すると考えるのは適切でない。

データモデリングに対する様々なアプローチが、知識の性質に関するパラダイミ的な前提に関してどの様に異なるかを明らかにするために、ここではデータモデルの構築をそのアプリケーションドメインに関する限定された理論の構築と対比してみる。この視点は決して新しいものではなくて (Kent 1978; Naur 1985)、既に多くの人々が認識、データ、実世界、知識及び表現の間には重要な関連性があることに着目している (Churchman 1971; Checkland 1981; Stamper 1987; Goldkuhl and Lyytinen 1982 a; Lyytinen 1987)。

従ってここではこれらの研究成果を踏まえて、現存するデータモデリングのアプローチが以下の諸項目に関してどの様な前提に立っているのかを調べることが自然な流れである。

- イ) 現実の性質 (存在論)、及びデータモデルによって表現される知識の性質、これはこの知識を収集するための方法や手段を含む (認識論) : 即ちパラダイムの "主観主義-客観主義" という次元。
- ロ) 社会的な世界の性質、社会的な文脈と呼ぶことができるものは何か : 即ちパラダイムの "秩序-コンフリクト" という次元。
- ハ) 言語的な要素の表現。

上記の3つの哲学的な論点をより明らかにするために、これらに関する次の4つの質問を用意する。

- (1) モデル化されるものは何か (存在論) ?

この質問は環境或いはモデリングドメイン即ち議論の対象となる世界 (universe of discourse) の性質に関するデータモデリングの最も基本的な前提を指している。現時点においてはデータモデリングの存在論 (ontology) に関して議論するための共通に使用される用語 (terminology) は存在しない。データモデリングの存在論には、ただ1つ或いは複数のユーザーシステムが存在するのか、それともそれは全く存在しないのか、更に各ユーザーシステムの基本的な構成要素はオペレーション、役割、意志決定、社会的な活動、発話行為、又はそれ以外の何かのどれであるか、と言うようなアプリケーションドメインの性質に関する基本的な前提も含まれる。

- (2) モデルが妥当であるのは何によるか (認識論) ?

この質問は我々がどのようにして議論の対象となる世界に関する妥当な解釈や知識を得ることが出来るのか、についての基本的な前提を指している。データモデリングの作業中に遭遇する (問題及び効果という) 異なる種類の不確実性にどのように対処すべきであるとしているかは、言うまでもなく異なるアプローチを比較するために考慮すべきただ1つの事例ではないが、異なるアプローチが知識やその探索方法に関して異なる前提に立っていることを示すよい例ではある。

- (3) 社会的な世界とデータモデリングの間にはどのような関係があるか (社会的な文脈) ? この質問はデータと行動との関連性に関する基本的な前提を指している。データと行動の間には2通りの関連性がある。

先ず第一にデータはある目的を達成するために使われるが、その目的はデータモデルがつくられたのと同じ社会的な文脈の中で形成されている、という繋がりがある。別の動機に基づいて行われた意思決定の合理性を説明するためにデータが使用された時に、初めて本来の目的が見えてくるということもある (Kling 1980)。

第二に行動に対するデータの関係は、データの使用に対して言語の使用に類似した解釈をすると明らかになる、というものである。言語が効率的に使用されるためには、発話は常にそれが使用される特定の場に結びつけられなくてはならない。この観点から言語は指標的 (indexical) である、即ち言語の重要性は特定の場、つまりその発話が行われた具体的な環境との結びつきに依存している (Suchman 1987)。これと同様の社会—言語学的な前提がデータの効率についても当てはまるが、そのことはデータモデリングに対する伝統的なアプローチの中では明らかにされていない。

- (4) モデルはどの程度うまく表現されているか (表現) ?

この質問は対象システムを表現するために最も適切、且つ有効な表現方法に関する基本的な前提を指している。

2. 3 機能主義者のデータモデリングに関するパラダイム的な前提

- (1) 存在論的な質問に対する答え：

世界は所与であってかつ、具体的な対象物 (オブジェクト) から構成されている。それらの対象物は各々固有の特性を持っていて、かつ互いに別の対象物と結びついている。所与であるとは、世界は人による理解や評価が行われる以前から存在しているのであって、社会的な相互作用の結果として創り出されるようなものではない、という意味である。このような世界は従って、ユーザーの考え方や社会的な実践を超越して存在する。(情報システムを開発する際の) 対象システムとは、データモデルが関連している現実世界の一部分の事である。データモデルは対象システムに関する適切な事実を全て反映しているので、それは対象システムを単純化した“絵”であると考えられる。次の定義はデータモデルに対する機能主義者の視点を端的に表している。

“データモデルは実世界を表現するための仕様言語である (Mylopoulos 1981) ”

データモデルは事実に関する仕様であると考える上記の視点と類似したものには、「偏見を含まない企業モデル」という考え (Chen 1977)、「概念スキーマと情報ベースのための概念と用語」というタイトルの報告書 (van Griethusyen 1982) 等がある。機能主義と実在論 (realism) は、Tarskiの真理の対応理論 (correspondence theory of truth) を介して結びついている、という点に着目する必要がある。このことは、上で引用した van Griethusyen において明確に認識されている。従って以下において実在論という言葉がことわり無く使われる場合には、それは今述べたような意味においてである。

- (2) 認識論的な質問に対する答え：

妥当なモデルを構築することが出来るのは、対象システム即ちアプリケーションドメインに対して、適正な観察及びデータ収集の方法を適用することによってである。データモデルは対象システムの絵のようなものであるから、それが現実の対象システムにどの程度うまく対応しているかをチェックすることによって、モデルの正確さを決めることが出来る。このような方法によって、時間をかければモデルと現実との対応度を改善していくことが出来る。

環境の変化に対しても同様な方法を適用することが出来る。この様にデータモデルに関する誤りの訂正或いは変化への対応を継続的に行うためにユーザーの参画が求められるが、そこでユーザーに期待される役割は次の2点に限定されていると考えられる：

イ) データモデルの作成者に対して、"生の"データやその定義を提供する。

ロ) 作成された公式のモデルが真実の状況に対応していることを確認するという意味で、それが妥当であることを保証する。

この様な意味でのユーザー参画の効率を上げるために、ダイアグラムによる表現やワークスルーなど様々な工夫が提案されてきた。しかしこれらの道具を使うことは決して「妥当性の検証とは真理の対応理論の線にある正確性のテストのことである」という基本的な前提を変更するものではない。

(3) 社会的な文脈に関する質問に対する答え：

秩序と安定性を達成するために規制が重要である。組織における様々なプロセスは基本的に組織の秩序と安定性の維持を指向しており、データモデルはこれらのプロセスが利用する情報を提供するように設計される。データモデルは組織の効率性と効果を改善するための様々な意図的な介入を支援することを通じて、組織がその目標を達成するのに貢献する。データモデルは決して、組織の基本的な目標や方針に関する議論を促進するために設計されるのではない。方針を策定する人は、組織にとって何が望ましいことであるか、また組織の方針が個別の決定に対して合理的な判断根拠となるためには、それをどのように組織に伝えたらよいか知っている。この様な前提の下では、もしもデータモデルが組織とその環境とを矛盾のない形で表現しているならば、それは組織をコントロールするために使用される情報の質を向上することが出来る、と信ずることは合理的である。この様にコントロールに重点が置かれていても、データモデルそのものは政治的に中立であると考えられる。組織内部にコンフリクトがあったとしても、それは組織内に存在するパワーによって解決されるのであって、アナリストは組織の方針の決定に対してなんら命令する立場にない。

(4) 表現に関する質問に対する答え：

データモデルの質は、厳密さと使い易さという2つの競合する視点から分析される。形式の厳密さという視点からは、データモデルは完全で、曖昧さや矛盾が無く、厳密な推論が可能ないように形式化(formalized)されていなくてはならない。この視点からの理想的なデータモデルは、対象システムに関するあらゆる質問に答えられるような形式的な計算方法(formal calculus)を提供するものである。質問への答えは公理と推論の規則から導かれなくてはならない(Bubenko 1986)。このことは限られた範囲においては、例えばリレーショナル完全性に関する研究によって達成されている(Codd 1972)。述語計算を利用する形式主義はRusselとWhiteheadの著名な研究成果に基づいているので、この目標に対してきわめて有望であると思われる。それ故に、機能主義のデータモデリングにおいて、対象システムのオブジェクトとその特性(properties)及びオブジェクト相互の関連性を直接に表す構成要素を提供する形式主義が数多く提案されたのは驚くには当たらない。広範囲に使用されている1つの表現方法は、対象システムをエンティティ、その属性及びエンティティ相互の関連性のインスタンスによって描こうとする(Chen 1976; Teichroew et al. 1980)ものであるが、リレーショナルモデルにも類似の考えはある。また従来のデータモデリング言語では、1つのインスタンスはただ1つのタイプに属するという単純なタイプ理論を採用しているが、

これに関する問題点が最近広く認識されるようになって、タイプ理論は“カテゴリー”を包含するように拡張されている (Elmasri and Wiederhold 1985)。しかしながら正統的なデータモデリングアプローチにおける意味論は依然として、各データ項目の意味はそれが表す対象物の集まりに対応するという外延論 (denotational theory) に固定されている (van Griethuysen 1982)。従ってデータモデルの中にもっと意味を含めようとする努力の焦点は、イベントの時間的な発生順序、存在依存性、オブジェクトの特定性など、実世界の構造を反映していると考えられる制約条件を定義することに当てられている (Codd 1979)。データモデルの品質に関するもう1つの基準、即ち使い易さは、例えばERモデルのような特定の種類の形式主義が何故広く使われているのかを説明するために利用されてきた。使い易さという問題は、経験的には幾つかの研究において取り上げられている (Batra and Davis 1989; Batra et al. 1988)。しかしこの基準は、異なるデータモデリング言語の中から適切なものを選ぶための有効な方法が見当たらないと言う点で、適用が困難であるという事が分かってきた (Tsichritzis and Lochovski 1982)。

要約すると、データモデリングに関する方法論とツールを詳細に調べた結果、機能主義の強い影響が明らかになった。これらの方法論が機能主義に基づいていることを示す根拠は次の2組の基本的な前提である：

イ) 実証主義的な認識論 (positivist epistemology) と結びついた実在主義的な存在論 (realist ontology)

ロ) 人間の組織に対する規制乃至は秩序重視的な視点

しかしながら、これらの前提は必ずしもデータモデリングの考えが提案された時点から、広く共有されてきたわけではない (kent 1978)。

2. 4 社会的な相対主義者のデータモデリングに関するパラダイム的な前提

(1) 存在論的な質問に対する答え：

社会を構成する各個人はコミュニケーション過程の中で自分の立場を定義する。そのコミュニケーション過程を通じて、対象システムが社会的に構築されていく (シンボリック相互作用論)。この基本的な前提は「事物が実際そこに存在しているかどうかに関わり無く、言語ゲームの創発的な組合せ (an emergent set of language games) を適用する解釈過程を通してのみ、我々はその事物にアクセスすることが可能になる」というものである (Truex 1993)。ユーザーもアナリストも組織的な現実が存在すると主観的に信ずることがあるかも知れないが、それは実は人間によって編集されたもの (human authorship) である、ということをおぼえているのである。この様な客観化が行われるプロセスは“具象化 (reification)”と呼ばれる (Berger and Luckman 1967)。習慣、言語、伝統、制度化などがこの中で重要な役割を果たす。

これらの考えに基づいて、Boland (1979) はシステム設計に対する意思決定モデルに基づくアプローチと行動に基づくアプローチの違いを指摘した。前者は客観的な表現とアルゴリズム的な操作に大きく依存しているのに対して、後者は；

情報システムの設計において重要なのは、前もってモデル化された組織的な現実に設計の結果が適合しているかどうかではなくて、組織の構成員がシンボリックな相互作用を通じて組織的な現実を構築していくのに、設計の結果を適切に使うことが出来るかどうか、という

ことである。要するに情報システムはシンボルの環境であって、状況を理解する (sense making) プロセスがその環境の中で遂行されるのである (Boland 1979)。

この様な基本的な考え方をデータモデルに適用すると、「データモデルはユーザーがアプリケーションドメインの中で互いにコミュニケーションするために使用している言語をモデル化しよう努力すべきである」ということになる。これは機能主義的なアプローチの“現実写像 (reality mapping) 視点”に対比して、“公式言語開発 (formal language development) 視点”と呼ぶことが出来よう。

つまり公式言語開発視点によれば、データモデルは実世界のモデルであると言うよりはユーザー言語のモデルである。ユーザー言語は；

業務の中で使われる言語であって、業務プロセスを支援又は変更する、業務体系の編成、知識や価値観の共有化、更にその状況を構成している社会的な関係を確認する、等の目的で使用される (Holmqvist and Andersen 1987)。

ユーザー言語の例としては、大組織の中で創り出された多くの頭字語 (acronyms)、普通の言語でありながら特別な意味を持たされたもの、などがある。

ユーザー言語は複雑なルールのシステムであって、その (ルールの) 多くは話者にとっても表面化されてはいない。従って、モデル化することに注意を注がなくてはならないのは、所与のオブジェクトやその特性ではなくて、その言語の統語論、意味論、及び語用論を定義しているルールである。ルールに基づくデータモデルを作成するための手法やツールに関しては幾つかの試みがなされているが、(ユーザー) 言語は完全な仕様化が難しい日常の発話行為の中から生まれるので、完全なモデルを作成する事が可能であると期待する事は現実的でない。情報システムはユーザー言語よりはもっと構造化され、形成化された言語の上に構築されている、という特徴を持っている。このことが情報システムの効用を制約する大きな原因となっている。このことを考慮に入れると、データモデルは、ユーザーが互いに理解し合ったり、環境を理解するために使用している言語を部分的に再構築している、ということが出来る。

言語をモデル化する際の基本的な問題点は、対象システム (即ちユーザー言語) を構成する基本的な構成要素 (building block) が何であるかを決定することである。ユーザー言語の基本的な構成要素としては明らかに、文 (sentence) と発話行為 (speech act) という2通りの選択が可能である。前者は人間の言語は述語論理を使ってモデル化することが出来ると提案したFregeの言語観と同じ線上にある。そのように言語の基本的な構成要素として文を選択する場合には、ユーザー言語をモデル化するためにエンティティー・属性記法を使うこともできるであろう。

一方、言語が文ではなくて発話行為から構成されていると見る場合には、データモデリングの焦点は広がる。発話行為はコミュニケーションの基本的な単位であって、それによって話者は、(質問という発話行為を利用して) 一片の情報を手に入れる、或いは (約束という発話行為を利用して) コミットメントを知らせる、等何らかの言語外の目的を達成しようとするものである。Searleは人間の基本的な意図を伝えるために、発話行為には5通りの基本的なタイプがあるという仮説を提案した (Searle 1979)。発話行為をモデル化するためには、その3種類の側面を表現出来ることが必要である。即ち：参照されているものは何か (叙述と参照)、意図 (発話内行為)、その発話行為の結果として予測される事態 (発話媒介

行為の効果)。ERモデルはこの中で高々叙述と参照をとらえることが出来るだけである。

発話行為という視点からのデータモデルは、発話行為に対応する基本的な“動き (move)”からなる言語的叙述 (linguistic discourse) のモデルである (Auramaki et al. 1988)。一連の簡単な発話行為の意味をとらえるためのデータモデリングに関するある種の形式論が提案されている (Lehtinen and Lyytinen 1986)。しかしこれを、方針決定に関する議論のような、もっと複雑な領域に拡張していくことは重要な研究課題である。コミットメントに関する発話行為をめぐる設計されたシステムの例が、(Winograd and Flores 1986) の最後の章に報告されている。

(2) 認識論的な質問に対する答え：

あるデータモデルが別のモデルよりも正確であるかないか、或いは適切であるかないかと言うのはどのような意味においてであるか、というこの質問に対する社会的な相対主義者のデータモデリングの答えは解釈学の視点に立つものである (Langefors 1977 ; Capurro 1986)。解釈学は、データモデルは次の3点において基本的な制約を受けていると指摘する。：

第1はどのようなデータモデルも、何らかの偏りを免れることが出来ない (たとえ、カッコに入れる (bracketing) 事によってその偏りに対する理解を改善することが出来るとしても) と言う点である。

第2に、データモデリングを通じて少なくとも二つの異なる意味の視界 (horizon of meaning) が交わる、つまり二通りの解釈を伴う、という点である。

第3は複雑な組織では、異なる意味の視界がどれだけあるかを予め決めることが出来ない、という点である。従ってデータモデリングのプロセスはオープンエンドでなくてはならない。又それ自身の限界に対して批判的であることが必要である。

上記の考え方から、社会的な相対主義者の視点に立ったときの、データモデリングの研究や実践の方法を導くための次のような原則が生まれる (Capurro 1986)：

イ) どのデータモデルにも基本的な偏りがある。その偏りが分かれば、そのデータモデルつくる基になっていた事前理解 (contingent preunderstanding) がどのようなものであったかを突き止めることが出来る。

ロ) ある種の自己批判的、内省的な対話であるカッコ入れを通して、ある程度まではこの偏りを透明にすることが可能である。

ハ) カッコ入れを、互いに基本的に相容れない先入観 (preconception) のいずれかを選ぶための手順であると考えてはならない。データモデルに対する解釈学的なアプローチでは、(上記のようなプロセスを通して) 偏りがいずれは殆ど問題にならなくなる、或いは完全に除去される、という考えに対してきわめて懐疑的である。

これらの原則を当てはめてみると、データモデリングの実務においては、例えばアナリストとユーザーと言うような、異なるコミュニティにおいてはそれに固有の意味の視界の間の根本的な相違があることを考慮に入れなくてはならない。異なるコミュニティがそれぞれ固有の意味の視界を持っているというのは、それらが出会ったとき互いに相手を自分の視界の中で解釈しなくてはならないという問題に直面し、従って相互理解はリスクにさらされる、という意味である。しかし二つの視界が融合し、生じがちな誤解が克服される可能性は存在する。その理由は「双方の視界がオープンである」、即ち「各コミュニティはシンボリックな相互作用 (互いに相手と一緒に、問題を共有しようとする) を通じて、各々の視

界を広げ、それをある程度まで融合させることが出来る」からである。意味の視界という考え方をデータモデリングに適用することによって、更に二つの結論を導くことが出来る。

第一の結論は「一つのデータモデルが作成されると言うことは、少なくとも二つの視界が融合されるということの意味する」ということである。仮にユーザーが分析のプロセスに加わってなくても、開発者は、例えばユーザーと開発者という、異なるコミュニティが持っているそれぞれの意味の視界（又は理解のコンテキスト）の橋渡しをしなくてはならない。どちらのグループもそれぞれ別々の言語ゲームを内部に持っている。ユーザーにとっても状況は対称的である。もしユーザーが自分で開発を行おうとするならば、開発のために使われる仕様化言語やモデリングツールに内在する（開発者の世界に固有の）意味の視界に対処しなくてはならない。

データモデリングにおける視界の融合は3ステップにわたって行われる。

第1のステップでは開発者が、ユーザーの間では共有化され、且つ少なくとも当初は開発者にとって未知の、意味を理解しようとする必要はない。

第2のステップでは、開発者が理解できたことを、ある適切な（論理的に整合性のある）形式に変換しなくてはならない。その形式は少なくともある部分に限っては、現存するユーザー社会の言語ゲームの代わりを果たすことが出来る、ということ暗黙の前提としている。

第3のステップでは、第2のステップで作成された形式に依る表現を再びユーザーの意味の視界（の言葉）に変換し、日常の行動の一部にしないといけない。（これはもちろん融合された意味の視界が妥当なものであるとして受け入れられた場合のことである。そうでないときは、新しい事前理解に基づいて、上記のサイクルを繰り返さなくてはならない。）

上記のどのステップにも危険は満ち満ちている。これらの危険に対しては綿密な注意が必要であるが、通常ウォークスルー、ドキュメンテーション、ユーザー教育、等のやり方は不十分である。

第2の結論は「システムティックな“カッコ入れ”を通じて視界の融合を促進することが出来る」ということである。この理由は、ユーザーと開発者の間の対話に基づく方が、それぞれのグループ内部の討議に基づくよりも有効なカッコ入れが出来る、ということによる。この点は特に「どのような代替的な概念モデルを作成しようとするときも、それに先立ってそのモデルに対応する“根底定義”を構築しなくてはならない」というCheckland (1981) の考えによって明らかにされた。ユーザーと開発者側の2重の解釈が必要であると言うことは、一面では硬直性と誤解を生み出す危険性を持っているが、同時にそれは対話を通じて相互理解を促進する機会でもある、と考えられる。

組織内には異なるユーザーグループの数と同じ数だけの意味の視界がある。ここで異なるというのは、ユーザーは各々異なる教育や職業的訓練を受けており、従ってそれぞれのグループ毎に特化した言語と実践とを伴った固有の熟練を共有している、という意味である。言い換えると、各々のグループ毎の日常業務での興味の中心は互いにかかなり離れている、と見なくてはならない。同じ病院の医師のグループと看護婦のグループはこの好例である。従って二重の解釈は更に多重の解釈に拡張されなくてはならない。誤解の危険はアナリストとユーザーの間だけでなく、異なるユーザーコミュニティの間にも存在する。

この問題に関する一つの結論は「組織全体にわたる完全に整合的なデータモデルは存在し得ない」ということである。いうまでもなく、ある組織において戦略データモデルが存在し、

そのモデルは組織の全体をカバーしている、ということはある。これはしかし、ハイレベルの企画スタッフというグループに固有の、もう一つ別の意味の視界を提供しているに過ぎない。そのような性格上、このモデルは別のユーザーグループとの対話の基盤となるから、その結果として異なる視界の融合、つまり組織学習をもたらすことが可能である。実務的には、企業モデルは常にこの様な目的で使うために作成されたはずである。しかしデータモデルについて述べた文献の中には、これとは別のことを述べているものもある。例えば組織全体にわたって整合性のあるモデル（スキーマ）を構築し、その約束を強制的に守らせるべきである、という提案があった。同様な考えの基で、企業全体で共通に使用される全く曖昧さを持たない語彙を定義すべきであるとか、ローカルなデータベースがグローバルにも妥当であることをテストすべきである、という提案も行われている。このどれもが、解釈学的な観点からは、適切ではないと考えられる。

(3) 社会的な文脈に関する質問に対する答え：

この質問の意味は、データモデルはどの様にして組織の行為者 (organizational actors) 達が互いに理解したり、学習したりするのを支援することができるのか、ということである。社会的相互主義の文脈ではデータモデリングの作業は、組織の中でそれぞれ別々の視点を持った複数のグループ（文化）が共存している、という前提に立って行われる。コンフリクトは問題ではなく、従ってある個人又はその個人が属するグループの利益のために意図的にデータをゆがめる、というようなことは特に注意すべき問題であるとはみなされない。また異なる視点の妥当性を支持する証拠が現れたときでも、それがどのくらい重要であるかは問題にならない。どの視点も皆偏見であって、解釈サイクルの繰り返しを通して時間とともに変化していくと考えるからである。

(4) 表現に関する質問に対する答え：

実在論の基ではデータモデルは現実を表す手段である。従って（同じ対象システムに関する）複数のデータモデルが存在する場合、（現実写像の）正確性や完全性という基準によって、それぞれが（どの程度適切であるかが）評価される。しかしデータモデルが、何を現実と見るかという事と、そのことの妥当な証拠は何であるかという事の両方を同時に決めるとなると、これは不可能になる (Boland 1979)。なぜならば、構築主義的な視点の基では、データモデルは社会的なコミュニティがその環境を理解するための一つ的手段であるからである。つまりそれは、意味のある重要なものを、それ以外のものから識別するための一つ的手段であるからである。つまりそれは、意味のある重要なものを、それ以外のものから識別するためのフィルターとして使われる。社会的に構築された現実を維持するのを支援することによって、データモデルは、人々の知覚を制限した上でそれを伝える、証拠の入手可能性に影響を与える、どの様な種類の証拠が望ましいかを示唆する、警告する、隠蔽する、等の手段を通して人々の行動に影響を与える。このことはデータモデルの妥当性を立証する行動にも適用することが出来るので、結局は循環する事になり、前述の状況が生まれる。それ故に、真実の状況との偏差を最小化するか、或いは所与の現実のあまりに多く、又はあまりにわずかを篩にかける表現を見つける、といったようなやり方でデータモデルの妥当性を表すことは出来ない。

データモデルを作成するという事、或いは適切なデータモデルを選択すると言うことに対して、上で述べたことがどのような意味を持っているかは、これまでの文献では余り取り

上げられていない。我々はデータモデルの役割は、論議を呼ぶ表現ではあるが、科学的なコミュニティにとっての理論の役割と同じような視点でとらえるべきである、と言う提案をしたいと思う。とりあえずデータモデルは、それが社会的なコミュニケーションを促進するのに役立つようなやり方で、構築すべきである、という提案をすることが出来るかもしれない。理想的には、よいデータモデルは、真摯で、適切で、明確かつ情報を十分に伝えるようなコミュニケーションを促進すべきであろう。しかしこの様な見方は理想的すぎて、既に述べた幾つかの重要な観点を見落としている。

3. データモデリングの実践のあり方

3. 1 伝統的な要求定義の考え方とその問題点

わが国の主な企業や病院、諸団体などの組織では、情報技術の進展に合わせて、ビジネスの基幹的な業務を対象とし、業務効率の向上を主目的とする情報システムの開発が盛んに行われてきた。この様な定型的な業務においては、その業務の内容は金融、流通など業種毎にほぼ似たようなものとなるので、主なコンピューターメーカーが皆顧客の業種別に組織を編成し、それに対応できる専門の技術者を育成してきたのは周知の事実である。

この様なやり方で情報システムの開発に対する大規模な投資が行われたのは、ユーザー企業においてその投資に見合う利益を明確に見積もることが出来ただけでなく、これを支援するメーカーの側においても、この様な社内組織に対する投資を、類似のプロジェクトに再利用することによって十分に回収できたからに他ならない。このことは、どのような情報システムを開発すべきかを定める要求定義の段階で、個別企業の個別部門に特有の状況を深く分析する必要は殆どなくて、むしろ同じ種類の情報システムの開発に関わった経験を持つ外部の専門家がユーザーの業務を調査した上で、自分たちの経験と方法論に基づいて要求仕様を決めることが可能であり、むしろその方がスムーズな開発をもたらした、という事を物語っている。

後の分析のために、この当時の（今も使われている）開発方法論がどのようなものであったかを見ておくことは意味があると考えられる。各メーカーはそれぞれ固有の方法論を開発し、適用してきたが、それらは皆共通の基本的な考え方を基盤として持っている事が分かる（情報処理学会 1991）。それは基本的に「情報システムの開発は次のような幾つかのステップからなる問題解決のプロセスである」とするシステム工学の考え方である。

- ステップ1 問題の確認と定義
- ステップ2 目標（望ましい姿）の定義
- ステップ3 現状から目標に至るための代替案の検討
- ステップ4 最適案の選択
- ステップ5 実施
- ステップ6 結果の確認と評価

このシステム工学の考え方とそれに基づく開発方法論の要求定義のプロセスには二つの暗黙の前提がある事に注目しなくてはならない。

その一つは情報システムの開発における現状の問題や到達すべき目標は、自然科学の対象世界と同じように、未だ発見されていないだけであって既にそこに存在しており、従ってそれを「客観的に」定義することは原則的に可能である、というものである。

もう一つはそれを発見するためのプロセスにおける関係者の役割に関連する。すなわち、そのような「客観的な事実」を発見するのは主として開発の専門家であるアナリストの役割であり、そのための材料を提供し、更にアナリストが発見した「事実」に誤りがないかどうかを（提供されたモデルの検証という作業によって）確認するのは当事者であるユーザーの役割であるというものである。

これは明らかに前章で紹介した機能主義者のデータモデリングに関するパラダイム的な前提に対応しているが、この様な基本的な考えに基づく開発方法論が、先に述べたような定型的な基幹業務を対象とする情報システムに対してきわめて有効であったことは容易に理解できる。

しかし、定型的な基幹業務のための情報システムの開発が一段落すると（主要企業で1980年代中にはほぼ行き渡ったと見られている）、開発対象の中心は自然に、非定型的な傾向が強いスタッフ業務に移ってきた。スタッフ業務は基幹業務に比べて一般に問題の構造が複雑であるだけでなく、意思決定の仕方がきわめて個別かつ状況依存的であるという特徴を持っている。

従ってこの様な性質のスタッフ業務を対象とする情報システムの開発に対して、外部の専門家が自分たちの経験を別のプロジェクトでそのまま活かすことは殆ど不可能であるだけでなく、スタッフであるユーザー自身も情報システムに対する自分の要求を定義する事がきわめて難しい。即ちシステム工学の考え方を基本に持つ従来の開発方法論が有効であるための基盤が失われたのである。それにも関わらず、この新しい状況に対応するための新しい方法論を抜本的に開発することなく、従来の考え方とやり方でこの複雑な情報システムを開発しようとした組織では、せっかくシステムが開発されてもそれは実際には使いものにならない、という事態が発生した。いわゆる第3次（勘定系）オンラインシステムの開発を完了した金融機関で、その次の対象になったスタッフ向けの情報系システムの開発プロジェクトの多くがこの様な状況を経験したことは広く知られている。

情報システムの要求仕様をどのような考え方で決めるべきかという視点からこの問題を考えてみると、これまでの開発方法論には少なくとも次の3つの基本的な問題点があることが分かる（小幡 1995a）。

第1は、問題状況（開発の文脈）を的確に把握するということに対する関心の低さである。定型業務では先に述べたように、個別のユーザーが自分たちの問題をどのように理解しているかを正確に把握するニーズは余りなかったので、方法論はこの問題を重視しなくて済みますことが出来た。従って多くの方法論では問題の把握については、基本的な考え方というよりは経験に基づくガイドライン的なものしか用意して居らず、現実にはプロジェクトを担当する開発専門の技術者（いわゆるSE）の裁量に委ねるといのが実態であった。このようなやり方で、開発対象の非定型業務の問題を、当事者であるスタッフがどのように見ているかを的確に把握出来る、と考えることには非常に無理がある。

第2は、情報システムの対象世界はどのような性質のものであるか、という基本的な考え方（存在論）に関するものである。それは先に述べたシステム工学の考え方、即ち「情報システムの対象世界は自然科学の対象世界と同じように、未だ発見されていないだけであって既にそこに存在しており、従ってそれを客観的に定義することは原理的に可能である」というものであった。状況が複雑であればあるほど、同じ様な環境にあっても、何が問題であるか、どのような情報に特に価値があるか等について、関係者が皆同じように考えるとは限らない。それはむしろバラバラである。そのような状況においては本来的に、情報システムの対象は一人一人の人間の経験や

価値観によって異なると考える方が自然である。これは今までの自然科学が対象とする客観的な世界ではない。このような基本的前提に基づいて、社会的な意味で複雑な世界を対象とする情報システムに対するユーザーの要求を定義できると考えることにはやはり無理がある。第1の問題点と同様に、定型的な業務においてはその単純性の故に、この問題点も顕在化しないで済んだと見ることが出来る。

第3は、開発プロジェクトにおけるユーザーの役割に関するものである。先に述べたようにそれは、「客観的な事実を発見するのは主として開発の専門家であるアナリストの役割であり、そのための材料を提供する事と、アナリストが発見した「事実」に誤りがないかどうかを（提供されたモデルの検証という作業によって）確認するのが当事者であるユーザーの役割である」というものであった。

今までの方法論においては、特に要求定義の段階でエンドユーザーの参画が大切であるとされている。これは勿論エンドユーザーが上に述べた役割を果たすことなしに「正しい」要求仕様を生み出すことが不可能だからであるが、それ以上に「エンドユーザーは、たとえモデルの妥当性を検証するだけの役割であったとしても、自分が要求定義に参画したシステムでなければ完成したシステムを受け入れない」というところにポイントがある。このことは逆に見ると今までの方法論においては、エンドユーザーは元々情報システムの開発に対するオーナーシップを持っていない、或いは持つことができない、という認識があることを示している。つまり今までの方法論ではエンドユーザーはオーナーシップを持っていないと考えるが故に、エンドユーザーが関知しない情報システムを開発することが可能であったのである。そのようにして開発された情報システムに対しては、仮にそれが結果として「正しい」要求仕様に基づくものであったとしても、そうでない場合はなおのこと、エンドユーザーがコミットしないのは当然である。そこでこのようなことが起こらないように、要求定義のプロセスにエンドユーザーを参画させるための工夫が必要になった。この目的のために多くの具体的な方法が提案されているのは周知の通りである。しかしここに述べたように、今までの方法論は本質的にエンドユーザーのための方法論ではなくてシステムの開発者である専門家のための方法論である。

非定型的な業務の場合は、先に述べたようにユーザー自身の内部においても、自分の業務を予め明確に定義できていない（それをやる意思もない）のであるから、そのままの状況で開発の専門家であるアナリストがどのような要求仕様を提示したとしても、それに対して正しいかどうかの判断を下すことは不可能であり、従ってそれにコミットすることはない。このようなやり方は、結果としてどのようなシステムが開発されたとしても、ユーザーはそれを受け入れない、ということになる。

3. 2 これからの情報システムの要求定義におけるデータモデリングの定義

データモデリングは情報システムの開発において重要な役割を持っている。特に要求定義の段階でのデータモデル（概念データモデル）は情報システムを利用する組織体のそのシステムに対するデータ要求、即ち情報システムの対象となっている世界のデータの意味と構造とを簡潔に示している。第2章で紹介したように、この作業をどのような基本的な前提に基づいて行うかによって、結果として得られたモデルは、一見同じように見えてもその意味は全く異なるものになってしまう。情報システムの開発は基本的に、それを利用する組織のデータ要求に基づいて行われるのであるから、開発方法論とデータモデリングは、その基本的な前提において一致しなくてはならない。

前節での開発方法論に関する分析から、少なくとも非定型的な業務を対象とする限り、機能主義に基づくパラダイム的な前提では、情報システムの開発は成功しないことが明らかになった。これに対して、社会的な相対主義に基づくパラダイム的な前提の基本的な特徴は次の2点であった。

イ) 存在論

「対象システムは客観的に所与のものとして存在しているのではなくて、社会を構成する各個人のコミュニケーション過程を通じて、社会的に構築されていく」

ロ) 認識論

「客観的に正しいデータモデルは存在しない。データモデリングの過程を通して、様々な異なる意味の視界が交わり合う」

この考え方には、前節において従来の開発方法論の基本的な問題点を特定した時に、我々がもっていた問題認識と基本的に相容れるものがある。特にこの認識論(解釈学)に基づいて、Capurro (Capurro 1986) が提案しているデータモデリングの方法に関する原則は、具体的な方法論を考える上で示唆的である。ここに再びそれを引用する。

イ) どのデータモデルにも基本的な偏りがある。その偏りが分かれば、そのデータモデルつくる基になっていた事前理解がどのようなものであったかを突き止めることが出来る。

ロ) ある種の自己批判的、内省的な対話であるカッコ入れを通して、ある程度まではこの偏りを透明にすることが可能である。

ハ) カッコ入れを、互いに基本的に相入れない先入観のいずれかを選ぶための手順であると考えてはならない。データモデルに対する解釈学的なアプローチでは、(上記のようなプロセスを通して) 偏りがいずれは殆ど問題にならなくなる、或いは完全に除去される、という考えに対してきわめて懐疑的である。

第2章で紹介したように、Hirschheimらはこの原則と同時に「意味の視界の交わり」を利用してデータモデリングを進めるための具体的なステップを紹介している。即ち；

データモデリングにおける視界の融合は3ステップにわたって行われる。

第1のステップでは開発者が「ユーザーの間で共有化されているが、開発者にとっては(少なくとも当初は) 未知の意味」を理解しようとつとめなくてはならない。

第2のステップでは、開発者が理解できたことを、ある適切な(論理的に整合性のある)形式に変換しなくてはならない。その形式は少なくともある部分に限っては、現存するユーザー社会の言語ゲームの代わりを果たすことが出来る、ということ暗黙の前提としている。

第3のステップでは、第2のステップで作成された形式に依る表現を再びユーザーの意味の視界(の言葉)に変換し、日常の行動の一部にしなくてはならない。(これは勿論融合された意味の視界が妥当なものであるとして受け入れられた場合のことである。そうでないときは、新しい事前理解に基づいて、上記のサイクルを繰り返さなくてはならない。)

我々は以前に、前述の基本的な問題認識に基づいて今までの開発方法論の要求定義に代わる新しい要求定義のあり方を探索したとき、P. Checklandが提唱するソフトシステム方法論(S S M)が、手法レベルと認識レベルという二つの異なるレベルで、前節で述べた今までの開発方法論の3つの問題点の解決に貢献できる可能性を持っている事を指摘した(小幡 1995b)。

ここでの議論を容易にするために、S S Mの基本的な考え方を簡単に紹介する。

S S Mにおいては、問題状況に関連する人々が、基本的に次の3種類の活動要素からなる一種の

学習サイクルを限りなく繰り返すことによって状況の改善を目指す。すなわち；

イ) 問題状況の表現

現実の問題状況をありのまま表現する。そこでは

- ・どのような関係者がこの問題状況に登場するか
 - ・それぞれの関係者は状況をどのように認識しているか
 - ・それぞれの関係者はその相手と互いにどのような関係にあるか
- などが生々しく絵で表現されている（これを豊かな絵と呼ぶ）。

ロ) 関連システムの作成

現実の問題状況をより深く把握するために、それと対比すべき「目的を持った人間活動のシステム」という概念上のモデルを作成する。これを「関連システム」と呼ぶ。それは現実世界そのものでも、又客観的に理想的であると考えられる世界でもなくて、問題状況に関係する人々がその状況の本質を理解する上で有効であると考えて自ら作り上げる概念上の人間活動のモデルである。関連システムは一つの問題状況に対して、それに関連する人々が持っている異なる認識の数だけ存在する。関連システムは、それが何のために何をするシステムであるかを記述する根底定義と、根底定義が満たされるためにはどのような種類の要素活動が必要であって、それらがどのようなつながりの基で展開されなくてはならないかを図的に示す概念モデルによって明確に表現される。関連システムは問題状況に対する関係者の理解を深めるために作成されるので、全く異なる視点に基づく複数のシステムが作成されることが望ましい。

ハ) 議論を通じての問題状況の理解と改善行動の選択

関係者は上記イ) で表現された豊かな絵と、ロ) で作成された（複数の）関連システムを対比し、議論を重ねることによって、現実の問題状況に対する認識を深める。この議論に基づいて現実にとるべき行動を選択し、実行したあと再びイ) へ戻る。

このプロセスは一見して、前述のデータモデリングにおける視界の融合の3ステップに類似しているように見える。これはSSMもまた解釈主義のパラダイムに基づく方法論であることから、ある意味では当然のことである。しかしここではとりあえず、この二つのアプローチの類似性を更に深く分析する代りに、視界の融合のために行われる3つのステップが、1. データモデルの作成に先立って行われる状況理解、2. 状況理解に基づくデータモデルの作成、3. 作成したモデルの妥当性の検証、に対応していると考え、この各々をSSMを構成する活動要素、特にイ) 問題状況の表現と、ロ) 関連システムの作成、の各々の中に埋め込むことによって、SSMの手法的なレベルの意義、即ち要求定義段階での「問題」のとらえ方を一層強化できることを指摘したい。

すなわち、問題状況の表現において、豊かな絵は人間の直観的な状況把握力に訴えることが出来るという点で確かに強力であるが、それだけに全く同じ比喩的な表現が、人生経験や価値観が異なる人に対して、全く逆の印象を与えてしまう危険がある事が指摘されている(P.Lewis 1994)。そこで、1. 先ず関係者の問題状況の理解の仕方を的確に把握した上で、2. 関係者の間で交わされる典型的な会話とその背後にある状況を説明するデータモデルを作成し、3. そのモデルの妥当性が関係者によって確認されたならば、そのデータモデルは豊かな絵が正しく理解されるのを支援するという意味で、その強い表現力を補足しているという事が出来る。

次に関連システムは、問題状況の本質に対する関係者の洞察を促進する上で、豊かな絵と同じ、

或いはそれ以上に大切な役割を持っている。豊かな絵との大きな相違点は、概念上ではあるが、人間活動のシステムの公式の（理論的な基盤に基づく）表現であるという点である。従ってその意味が全ての関係者によって、同じように理解されることの重要性は、豊かな絵の場合に劣らない。P. Lewisはこの点に着目して、問題状況が余り明確でない段階で利用されるための「解釈レベルでのデータモデル」という考え方を提案している（P. Lewis 1994）が、それはまさにSSMにおける関連システムのためのデータモデルである。彼によると、「従来のデータモデルの形式は問題領域が十分に定義されている場合にはきわめて有効であるが、組織が今後どのようにあるべきかという種類の問題が議論される段階では余り適切ではない。そこで、文化的な要素にもっと敏感で、現実が社会的に生成される可能性を認識できるような、従来のデータモデルとは異なる形式を持つ解釈レベル(interpretative level)のデータモデルを開発する事が必要である」、又「SSMの関連システムにおけるデータモデル、即ちそこで認識されるカテゴリー(cognitive categories)、それらの間の結合関係、及びその意味、等は前述の解釈レベルのデータモデルの考え方に適合すると判断することが出来る。この解釈レベルのデータモデルは、SSMの関係者が関連システムに関する議論を通じて問題状況に対する理解を深めるのに役立つだけでなく、従来のデータモデリングに対する意味のある出発点となることが期待される」

Lewisの考え方は基本的に今まで述べてきた我々の考え方と同じである。但し我々は彼のように「従来のデータモデルの形式、即ちモデルの構成要素とその表現方法は、解釈レベルのデータモデルの表現には適していない」とは考えない。例えばERモデルにおいて「実体は全て客観的に存在する具体的な物又は事ではなくてはならない」と考えれば、それは機能主義の立場に立ったモデルになるし、そうではなくて「実体は全て、個人のコミュニケーション過程を通じて社会的に生成される何かである」と考えれば、それは社会的な相対主義の立場から見たデータモデルになるであろう。我々はそのことよりもむしろ、社会的な相対主義乃至は解釈主義の立場から見た対象世界の意味を、出来る限り正確かつ自然に反映できるモデルを選ぶことが大切であると考え。その意味ではオブジェクト指向モデルが、解釈レベルのデータモデルの適切な表現形式として適切である可能性が高いと思われる。

4. まとめ

2. において、Hirschheimらによるデータモデリングのパラダイミ的な前提条件に関する包括的な分析を紹介した。

3. 1において、定型的な業務を対象とするシステム開発プロジェクトで成功を収めてきた従来の開発方法論は、伝統的なシステム工学の考えを基本としているので、

- イ) 問題の文脈（当事者の問題認識）を的確に把握するという点に関する関心が低い
- ロ) 情報システムの対象世界は客観的な存在であるという基本前提に立っている
- ハ) 基本的に開発者のための方法論である

という基本的な問題点を持っていること、そしてその結果、複雑で非定型的な業務を対象とする情報システムの開発には適合できなくなっていることを指摘した。

3. 2において、機能的主義のパラダイムに基づくデータモデリング方法論がこれと同様の立場に立っていることから、同様の環境に対しては同じように不適切であることを指摘した。さらに、これに代わる社会的な相対主義（特に解釈学）のパラダイムに基づくデータモデリング方法

論を、同じパラダイムに基づくSSMと組み合わせることによって、SSMを要求定義段階で使用する場合に、その問題把握力を一層強化できる可能性があることを指摘した。

しかしそれを具体的に確認するためには、残された課題は多い。

例えばSSMを手法レベルで適用するというのは、通常要求定義に先立ってSSMによって関係者の問題認識を深めることを意味するが、そこにおいては更に次の諸点を明らかにする必要がある。

- ・豊かな絵の表現力を補足するデータモデルは具体的にどのような物になるか
- ・関連システムの意味を明らかにするためには、具体的にどのようなモデリング方法を採用すべきか
- ・豊かな絵と関連システムに基づく議論を通して、問題状況に関する関係者の洞察が深められるが、それをどのようにして評価するか
- ・以上のステップを踏んだ後、情報システムの要求定義に入るが、これまでの議論の成果を具体的にどのように要求定義に反映するか

さらに問題の抜本的な解決のためには、SSMを認識レベルで適用する、即ち要求定義だけでなく、情報システム開発の全てをSSMの考え方で統合することが必要である。これに関する具体的な研究は今後待たなくてはならない。

参考文献 (第3章でHirschheimらが引用しているものを含む)

Teorey, T.J. 1994 Database Modeling & Design: The Fundamental Principles. 2nd ed. Morgan Kaufmann.

R.Hirschheim, H.K.Klein, K.Lyytinen, 1995, Information systems development and datamodeling: Conceptual and philosophical foundations. Cambridge.

ANSI/X3/SPARC, 1975, Study Group on Data Base Management Systems, Interim Report. FDT-Bulletin7(2).

Burrell, G. and Morgan, G., 1979, Sociological Paradigms and Organisational Analysis. Heinemann.

Silverman, D., 1970, The Theory of Organisations. Heinemann.

Kent, W., 1978, Data and Reality. North-Holland.

Naur, P., 1985, Programming as Theory Building. Microprocessing and Microprogramming, 15 254-261.

Churchman, C.W., 1971, The Design of Inquiring Systems. Basic Books.

Checkland, P., 1981, Systems Thinking, Systems Practice. Wiley.

Stamper, R., 1987, Semantics. In Boland, R.J. and Hirschheim, R.A. (eds.), Critical Issues in Information Systems Research. Wiley.

Goldkuhl, G. and Lyytinen, K., A Language Action View of Information Systems, In Gintzberg, M. and Ross, C. (eds.), Proceedings of the Third International Conference on Information Systems, Ann Arbor, MI. 13-30.

Lyytinen, K., 1987, Two Views of Information Modeling. Information and Management, 12(1), 9-19.

Suchman, L., 1987, Plans and Situated Actions-The Problem of Human-Machine

Communications. Cambridge University Press.

Mylopoulos, J., 1981, A perspective on Conceptual Modlings. In Brodie, M.L. and Zilles, S. (eds.), SIGPLAN Notices, 16(1), 98.

Griethuysen, J. van, (ed.), 1982, Concepts and Terminology for The Conceptual Schema and the Information Base. ISO Report, ISO/TC97/SC5/N695, March.

Bubenko, J.A., 1986, Information Systems Methodologies—A Research View. SYSLAB Report No.40, The Systems Development and Artificial Intelligence Labolatory, University of Stockholm.

Chen, P., 1976, The Entity—Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems, 1(1), 9—36.

Teichroew, D., Macasovic, P., Hershley, E. and Yamamoto, Y., 1980, Application of Entity—Relationship Approach to Information Processing Systems Modeling. In Chen, P.P. (ed.), Entity—Relationship Approach to Systems Analysis and Design. North—Holland, 15—38.

EL Masri, R. and Wiederhold, G., 1985, The Entity Category Relationship Model. Data and Knowledge Engineering, 1(1), 75—116.

Batra, D. and Davis, J., 1989, A Study of Conceptual Data Modeling in Data Base Design: Similarities and Differences Between Expert and Novice Designers. In Proceedings of the 10th International Conference on Information Systems. Mineapolis, MN.

Tsichrizis, D. and Lochoovsky, F., 1982, Data Models. Prentice—Hall.

Truex, D., 1993, Information Systems Development in the Emergent Organization. PhD Dissertation, Thomas J. Watson School of Engineering, SUNY.

Berger, P. and Luckmann, T., 1967, The Social Construction of Reality: A Treatise in the Sociology of Knowledge. Doubleday.

Boland, R.J., 1979, Control, Causality and Information Systems Requirements. Accounting, Organizations and Society, 4(5), 259—272.

Holmqvist, B. and Anderson, P.B., 1987, Work Language and Information Technology. Journal of Pragmatics, 11, 327—357.

Auramaki, E., Lehtinen, E. and Lyytinen, K., 1988, A Speech—act Based Office Modeling Approach. ACM Transactions on Office Information Systems, 6(2), 126—152.

Lehtineu, E. and Lyytinen, K. 1988, Action Based Model of Information Systems. Information Systems, 13(4), 299—317.

Winograd, T. and Flores, F. 1986, Understanding Computers and Cognition. Ablex Publishers.

Langfeors, B., 1977, Hermeneutics, Inforlogy and Information Systems. TRITA—IBADB NO.1052, University of Stockholm.

Capurro.R., 1992, Informatics and Hermeneutics. In Floyd, C. Zullighoven, H., Budde, R. and Keil—Slavik, R. (eds.), Software Development and Reality Construction. Springer—Velag, 363—375.

Lewis, P., 1994, Information Systems Development. Pitman.

情報処理学会編、情報システムの計画と設計、1991、培風館。

小幡孝一郎、1995a、新しい要求分析のあり方、「情報処理」第36巻、第10号。

小幡孝一郎、1995b、要求定義とソフトシステム思考法、「情報研究」第16号、文教大学情報学部。

(情報学部教授)