

# 要求定義とソフトシステム思考法

小 幡 孝 一 郎

## Issues on defining requirements and SSM as an alternative approach to addressing them

Kouichiro Obata

### 概要

情報システムの開発において、システムに対する要求を分析し仕様を決定するプロセスは、開発の上流工程においてプロジェクトの成功の鍵を握る重要なポイントの一つとして位置づけられ、このプロセスを適切に遂行するためのガイドとして、多くの著名な方法論が提案され、実施されてきた。

しかし最近の情報技術の発展に伴う様々な変化、特に対象業務の質と開発・利用環境の変化によって、構造化システム分析・設計方法論に代表される従来の方法論を要求定義のプロセスにおいて適用しようとしても、うまくいかないことが多く見受けられるようになった。

ここでは、この不具合が従来の方法論の出発点にある基本的な考え方から発生しているので、その解決は容易でないことと、ソフトシステム思考法（SSM）がまさにこの様な問題に対処するための基本的な思考と行動の枠組みを提供する可能性を持っていることを述べる。又この問題の今後の方向と課題について簡単に触れる。

尚本報告は筆者が先に“大岩元他編、情報システムにおける要求工学に関する調査研究、ソフトウェア工学研究財団、平成7年3月”に投稿した同名の論文を元にして、これに加筆、訂正を施したものである。

In an attempt to ensure that only those systems are developed which do just what users really require, a great emphasis has been placed on the early stages of information systems development where the requirements of users are analysed and determined. It is well known that most development methodologies have proposed respective guidelines for how to approach to this critical process.

However, it is not uncommon recently for us to observe systems development projects in which such typical methodologies as structured analysis and design do not work as well in determining requirements of users as they used to do before. This tends to be more evident when they are applied to a more complex situation where people involved have basically different views of the situation.

We try to make it clear in this paper that it is not easy to find a complete solution to the inability of these methodologies to define users requirements in a complex situation, since it stems from the fundamental philosophy about the nature of information systems which underpins these methodologies, and, on the other hand, that soft systems methodology provides a basic framework to deal with problems of this kind. We will, then, briefly review subject areas which seem to be worthy of further research.

## 1. システム開発における要求定義と問題点

### 1. 1. システム開発における要求定義の位置づけ

ソフトウェア製品の特性として、それを実際に使ってみる前に、それがどのようなものであるか（どのような働きをするか、それを使うことによって自分の仕事かどのように変わるか等）をユーザーが事前に、しかも正確に、理解することはきわめて難しい。特にそれによって支援される業務がビジネスの中心にあるような場合は、完成した後でそれがユーザーに受け入れられないものであることが判明することのリスクは致命的である。このためシステム開発のなるべく早い時点で、システムに対するユーザーの要求を明確に定め（要求定義）、それに従ってその後の開発工程を進めることが重要視されてきた。幾通りもの開発方法論がこの考えの基に提案され、それに基づいて数多くのシステムが開発されてきたことは周知の通りである。

### 1. 2 従来の要求定義の方法の特徴

要求定義は前述のようにシステム開発の成功・失敗を決定するきわめて重要な工程であるにも関わらず、従来の方法論では必ずしも開発現場の要請に十分に答えてきたとはいえない。

代表的な方法論の一つである構造化システム分析・設計方法における要求定義（システム分析）のプロセスの概略は以下の通りである<sup>(1)</sup>。

まず現状分析の段階で、アナリストは自分がとらえることができたユーザーをとりまく現実世界の状況を、方法論が規定する（DFDなどの）モデルを使って表現する。これに対してユーザーはそのモデルが現実を正しく反映しているかどうかを確認するよう求められる。この確認が得られるといよいよ本題の要求定義にはいる。これは先ず物理的な制約を取り除いた概念レベルの世界で行われるが、その結果つまり開発すべきシステムの本質的な姿もまたモデルの形で表現され（要求論理モデル）、再びユーザーの確認が求められる。このプロセスで（アナリストにとって）最も重視されていることの一つはモデル化のスキルである。

この様な要求定義のプロセスが生まれた背景には、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がある<sup>(2)</sup>。

- 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

従来の開発方法論の考え方、即ちシステム工学とそれに基づく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。

### 1. 3 情報システムをめぐる状況の変化とそのインパクト

数年前まではビジネスの基幹的な（オペレーショナルな）業務の効率の向上を主目的とする情報システムを、メインフレームアーキテクチャの基で、開発するプロジェクトが主流であった。この場合要求仕様、従ってその背後にある要求定義のプロセスが問題になることがあまりなかったのは、対象が定型的な業務に限られていたために、解決すべき問題の構造が多くの場合誰の目にも明らかであったこと、及び対象ユーザーにとっては自分の業務の殆どがこのシステムによってカバーされるので、要求仕様を定義する公式のプロセスに前もって参画し、例えばユーザーインターフェイスのあり方について意見を表明しておくことが実際に担当業務を遂行する上で不可欠の条件であったこと、等による。

しかしメインフレームによる集中処理を中心とする基幹情報システムの開発が峠を越し、開発プロジェクトのテーマがいわゆる情報系に移行し始めたのと時をあわせて、マイクロコンピューティングと統合ネットワークによる分散処理を中心とする情報技術が実用の域に達し、このことに伴って情報システムの開発環境、利用目的、利用環境などが大きく変化してきた。特に開発対象業務の質と、システムの開発・利用環境が変化したことが、前述の要求定義のプロセスに対して深刻なインパクトを与えることになった。

#### イ) 開発対象業務の質の変化

非定型的なスタッフ業務はオペレーショナルな業務に比べて問題の構造が複雑であるだけでなく、スタッフの意志決定の仕方はきわめて個別的且つ状況依存的である。従ってその意志決定を支援するために必要な情報を「客観的に」定義することは、アナリストにとっては勿論、ユーザー自身にとってもオペレーショナルな業務の場合とは比較にならないほど難しい。

#### ロ) 開発・利用環境の変化

メインフレームによる集中的な処理だけではなく、一人一人のエンドユーザーによる分散的な処理が可能になってくると、ユーザーは予め全ての要求仕様を明確に定義する（公式の）プロセスに参画しないと自分が使うシステムを作ってもらえない、ということはない。ある程度情報技術の素養のあるユーザーは、自分で出来る範囲で、自分がやりたいように自分のシステムを構築したり、改善することができるようになった。これは前述の自分の担当業務の殆どを新しく開発される（全社的な）情報システムに依存する定型的な業務のユーザーの利用環境とは全く異なるものである。

このような開発・利用環境は上記のスタッフ業務の特性にマッチしているので、いわゆるエンドユーザーコンピューティング環境は急速に普及した。この結果、アナリストを中心とする開発チームがユーザーの全面的な参画の基で情報システムに対する要求仕様を事前に、全体的に、し

かも明確に定義する、というこれまでのやり方でユーザーに協力を求めるやり方は現実的なものではなくてきた。この点を無視して従来の要求定義のプロセスを強行しても意味のある結果を期待できないのは当然である。

## 2. ソフトシステム思考法の概要とシステム開発におけるその意義

P.Checklandによって創始された代表的なソフトシステム思考法であるソフトシステム方法論(SSM)<sup>(3)</sup>を取り上げる。ここではその概要とシステム開発における意義についてだけ述べる。

### 2. 1 SSMの概要

次の記述<sup>(4)</sup>はSSMの特徴を簡潔に紹介している。

“SSMは社会的な意味で問題のある分野において、その分野に関連する人々の中に「理想的には終りのない」学習過程を巻き起こすことによって改善をもたらすための方法論である。

学習過程の中で関係者はシステム概念を利用して現実世界をどう認識するかを表現し、その表現が適切であるかどうかについて繰り返し議論する。次に議論の結果に基づいて現実世界に於て何等かの行動を起す。そこで発生した変化に関して、再びシステム概念を利用して現実世界の認識を表現し、それに対してまた繰り返し議論する、というプロセスを繰り返す。

この認識表現とそれに関する議論はその都度構築されるシステムのモデルを介して行われるが、これらのモデルは問題領域を説明するものではなくて、理想的に考えるならば状況はこのように一つのまとまり(システム)として認識されるであろうという一つの見方である。問題領域に関してはいかなる客観的或は完全な説明も与えられないということが前提となっている。”

もう少し具体的に述べるとSSM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サイクルを限りなく繰り返す。すなわち；

イ) 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(豊かな絵と呼ぶ)。

ロ) 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。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) 上記イ) で表現された現実の問題状況と、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、関係者の現実の問題状況に対する認識が深められる。この議論に基づいて現実にとるべき行動を選択し、実行する。

### 2. 2 システム開発(要求定義)におけるSSMの意義

1. で述べたように、従来の構造化手法に代表される開発方法論の問題点が顕在化し、それに代わる明確な方法論が未だ見いだされていない現状において、SSMの考え方が持つ意義はきわ

めて大きいが、それは二つのレベルに分けてとらえることができる。

一つはシステム開発の基本的な枠組みとして現在の構造化手法をそのまま使用する中で、要求定義のプロセスの中にSSMのやり方を部分的に取り入れることによってその弱点を補強する、というものである。これを「手法レベルの意義」と呼ぶことにする。

もう一つは構造化手法の明確には述べられていない前提（システム工学の基本哲学）に関連する問題点に着目し、そこにSSMの考え方を取り入れることによって、情報システムの開発に対する人々の基本的な認識を変えていく、というものである。これを「認識レベルの意義」と呼ぶことにする。これは今の構造化手法の基本的な枠組みの中で、それらの問題解決をはかることができるという事を前提にはしていない。

## 2. 2. 1 手法レベルの意義

### イ) 「問題」のとらえ方

従来の方法論では関係者によって識別された多くの問題を因果関係のチェーンでつなぐことによって問題領域全体の構造を明らかにし、とるべき行動の必要性や合理性を説明する、というやり方が代表的であった。問題領域の範囲と問題の構造が明確である場合はこのやり方で十分であったが、1. で述べたように情報システムの対象領域が拡大し、また対象業務が複雑になると、何が問題か、その原因は何かをめぐって関係者が完全な合意に達する、即ち現実の問題状況は誰がそれを見ているかに関わらず客観的に表現出来る、という前提が余り現実的でなくなる。

SSMでは「問題」が客観的に存在するのではなくて、「関係者によって認識された問題状況」だけが存在する、と考える。しかもこの「認識された問題状況」は言葉ではなくて、「豊かな絵」によってビジュアルに表現される。問題が社会的な意味で複雑になればなるほど、このような方法の方が関係者の認識を自然に表現することが出来ることは明らかである。

### ロ) 「あるべき姿」の導き方

構造化手法は要求論理モデルがきわめて重要であるとしながら、それを導く概念と方法については基本的にアナリストの裁量に委ねている。SSMでは関係者によって把握された問題状況に対して直ちに解決方策の検討に入るのではなくて、先ずこの問題状況に対する関係者の理解を深めるために、それと対比すべき関連システムを作成する。それは現実世界そのものではなくて、そこに関連する人々が問題状況に関して持っている認識に基づく人間活動のモデルである点がポイントである。従ってそれは一つの問題状況に対して、その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、その定義が満たされるためにはどのような人間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、それ自体が要求モデルにならないとしても、意味のある要求モデルを導き出すための重要な出発点になる可能性を持っている。

## 2. 2. 2 認識レベルの意義

### イ) 世界観の顕在化

SSMでは上記のように、関連システムという形で目的を持った人間活動を表現することがきわめて重要であるが、これを導く基本的な考え方、特に背景となっている関係者の世界観を明示することがこれと並んで重視される。状況が複雑であればあるほど、関係者の世界観は皆同じと

は限らない。世界観が異なれば、それに基づく関連システムも変わってくる。各々の世界観を表に出しながら、最初に認識された「問題状況」と対比してどの関連システムが適切であるかを徹底的に議論する。これを何度か繰り返すことによって状況に対する関係者の理解を深めていく、というのがSSMの基本的な考え方である。これは、システム工学の基礎にある実証主義とは対照的に、SSMが解釈主義的な立場に立っていることを表している。

従来の構造化手法では多くの場合、世界観は暗黙の了解事項であって表に出ることはなかった。少なくとも複数の世界観の存在は認められなかった。従って何のためにどのような情報システムを開発する必要があるのかという最も基本的なところで、関係者の考え方が一致しないまま開発が進められる危険性は潜在的には十分あったのである。状況が複雑になるとともにこの問題が顕在化してきたのは、その意味では当然のことであった。

情報システムは必ず何らかの人間の活動を支援するために構築されるのであるから、有用な情報システムが開発されるためには、それが支援すべき人間の活動に対する深い理解が必要であることは論をまたない。SSMはまさにそのための方法論である。

#### ロ) オーナーシップの確立

情報システムの開発において、それが完成した後に初めてユーザーの要求を満たしていないことが判明し、このためこのシステムがユーザーに受け入れられないものになる、というのは考えてみると奇妙な話である。今までの方法論において、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るのは、勿論それなしに「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が要求定義に参画したシステムでなければ彼らに受け入れら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「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。しかし本質的に今までの方法論は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、ということができる。

要求仕様が適切に定義されることは望ましいことではあるが、もしも情報システムの開発に関するオーナーシップをエンドユーザーが保持しているならば、仮に要求仕様を予め適切に定義できなくて結果的に不具合のあるシステムが開発されたとしても、それは自分が決めた要求の結果なのだから、それに対してエンドユーザーがコミットしないというとはあり得ない。このことは、今までの方法論においては要求仕様の決定が最も重要であるとされてきたけれども、実はそれよりも重要なことがあり得ること、ないしは「情報システムの開発は単発的な開発プロジェクトで行う他にない」という従来の常識に対する疑問の可能性、をも示している。

### 3. ソフトシステム思考法とシステム開発との結びつきに関する幾つかの考え方

英国はSSMの発祥の地であるだけに、かねてから情報システムに携わる人々からもSSMに対して大きな関心が寄せられていた。特に英国システム協会(UKSS)に所属する人々の著作の中に、情報システムの開発にSSMのシステム概念を利用することにふれたものが多いので、協会のメンバーが一堂に集まってこのテーマ、即ち情報システムとシステム概念の利用、について討議をしようという企画が生まれた。討議に参加するメンバーには予めポジションペーパーを提出することが求められたが、それが(多少の校正を加えられた後に)まとめられ、雑誌SYSTEMISTの特集号<sup>(5)</sup>として発行された。少し古いきらいはあるが、この特集には歴史的な価値があると考えられるので、ここでごく簡単に紹介する。

#### 3. 1 概要

このセミナーの主題は「SSMと情報システム」であったが、参加者の殆どはこれをもっと具体的に「方法論としてのSSMをシステム開発の方法論(SDM)、例えば構造化方法論、と結びつけることに関する問題」という意味でとらえていたようである。この背景として、従来のSDMは何らかの形でSSMのような考え方を取り入れることが必要である、という認識が全員に共通にあったと見ることができる。たとえばPrior<sup>(6)</sup>は次のように述べている。

“この様な(今までの)アプローチの特徴は問題状況の探索に対する関心が欠如していることである。その結果何故、どのような情報システムが必要になるか、という点については仮説に基づくことにならざるを得ない。(中略)情報システムは単にデータを処理するためのハードウェアとソフトウェアの組合せではない。それは特定の人が特定の状況で使うものである。誰が何に関する情報を受けるべきかを決めるのはこの様なシステムの使用に関する文脈である。SSMの利用価値はそれによってより広範囲の問題を探索することが可能になり、どのような文脈の下でどのような情報が必要になるかを的確に理解できるようになることである。”

したがってこれらの論文は大まかに次の質問に対する答えによって分類することができる<sup>(7)</sup>。

- イ) この二つの方法論を直接結びつけることができる(賛成)かできない(反対)か
  - ロ) 上記の問いに賛成の場合、次の2通りの結びつけ方が考えられる。いずれがよいか
- 接ぎ木型:

SSMをSDMのフロントエンドに置く

埋め込み型:

いずれか一方を基本的な枠組みとして、もう一方をその中で利用する技法として使う

接ぎ木型の基本的な考え方は、先ずはじめにSSMを使用して問題状況を理解し、目指す姿を一つの意図を持った人間活動のシステムとして認識する。この認識の表現の一つである概念モデルを出発点として、構造化手法で使われる論理DFDを導きだそうとするものである。つまりこれは第2章で述べたSSMの「手法レベルの意義」に着目したやり方である。様々なつなぎ方が提案されているが、これに対する主な反論は技術的なものではなくて、このやり方では論理DFD以降は従来の開発手法に従って展開されるので、(それを従来の考え方しか身につけていない専門家に任せると)はじめにSSMで得られた状況に対する深い理解が全部失われてしまう、

というものである。ここに従来の開発手法、ないしはそれに閉じこもっている人々に対する拭いがたい不信感を読みとることができる。

埋め込み型には当然2通りの全く違ったやり方がある。構造化手法の枠組みの中にSSMの考え方を取り入れるというやり方の一つの具体例は、従来実証主義的な立場で（技術的に）取り扱われてきたデータモデリングをSSMの解釈主義的な立場で実行しようというものである<sup>(8)</sup>。

もう一方は「SSMを情報システム開発プロジェクトを管理するための基本的な枠組みとして使用し、その中で必要に応じて構造化手法の技法（DFDなど）を使う」という考えである。これは第2章の「認識レベルの意義」に対応するやり方である。その基本には、「情報システムの開発の始めから終わりまでをSSMの実行主体（＝エンドユーザー）が管理する」という前提があるはずであるが、このことにはっきりと言及したのは後述するStowellとWestの論文だけであった。

### 3. 2 個別紹介

この特集号に寄せられた論文はいずれも興味深いものであるが、その中でも特に注目されるものを簡単に紹介する。

#### J. Mingers：SSMと情報システム；オーバービュー<sup>(7)</sup>

彼はこのセミナーの世話役の一人であった関係上、彼の論文にはこのテーマについてそれ以前に発表されたいろいろな考え方を分析、評価した記述がある。それとは別に彼自身は、SSMには重要な点で従来の開発手法にない優位性（特に学習を重視する点）があるので、SSMを開発プロジェクト全体を管理するための枠組みとして使う埋め込み型を支持するといっている。

#### F. Stowell&D. West：情報システムのクライアント主導型設計のための手段としてのSSM<sup>(9)</sup>

これもSSMを開発プロジェクト全体を管理するための枠組みとして使う埋め込み型を支持しているが、それらの中で「情報システムの開発の始めから終わりまでをSSMの実行主体（＝エンドユーザー）が管理する」という前提にはっきりと言及した唯一の論文である。彼らのクライアント主導型設計という発想は、情報システムの導入が屢々ユーザーに強い不安感をもたらし、これに基づくユーザーの抵抗のためにせっかく開発された情報システムが役に立たなくなることが多い、という問題に対する解決策として生まれたものであるが、SSMの哲学に対する理解の仕方は既に述べた我々のものにきわめて近い。ただこの論文は考え方を述べているだけで、具体的な方法論には触れていない。（これについてはその後の展開<sup>(13)</sup>が発表されている）

#### R. Prior：SSMと情報システムの開発を結びつける<sup>(6)</sup>

この論文の冒頭で、彼は自分の考えを次のように要約できると述べている。

第1に、情報システムを設計し開発するためには、SSMによってもたらされる洞察が必要である。第2に、論理DFDの利用は、SSM分析によって生成された活動モデルを情報システムの設計や開発に既に使われている技法やツールに変換する可能性を持つ場を提供する。第3に、この変換は決してSSM分析がこの時点で完了したのである、あるいはそれが従来の構造化分析、設計技法のためのフロントエンドとしての役割を提供したに過ぎないと言う意味ではない。この変換を行うのはSSM分析者の役割である。そして設計と開発のプロセスにおいて、SSM分析の成果物（特に根底定義と概念モデル及びその背景にある世界観）は継続的に参照され、様々な決定の拠り所とならなくてはならない。

この様に彼の考えは形の上では従来の開発手法の枠の中にSSMの手法を取り込む埋め込み型であるが、一貫してSSM活動の成果を基準にすべきであると主張しているところに特徴がある。Mingersはそのまとめの報告の中でPriorの考えは接ぎ木型であるとしているが、この論文ではそのように理解することには無理がある。

F.Gregory：SSMから情報システムへ；論理的な進め方<sup>(10)</sup>

彼もこのセミナーの世話役の一人である。彼の考え方は概要で述べた大まかな分類によれば接ぎ木型に属するが、接ぎ木を行うために「論理言語学的モデル」というきわめてユニークなモデルを導入している。ただこれが今までの開発手法にどのようにつながるのかは余り明確でない。

Y.Merali：分析的なDFD；物質主義への代案<sup>(11)</sup>

彼の主張は、DFDには（余り認識されていないが）思考のツールとして有効な役割があること、また論理DFDは概念をモデル化し精緻化するための、そして情報システムの概念的なモデルを構築するための強力なツールを構成する、というものである。このために彼は上記のGregoryの「論理言語学的モデル」を出発点にして、彼の「分析的なDFD」を導く方法を提案し、更にこれによって概念モデルが持っている「豊かさ」を論理的なシステム設計の段階まで保持できることを示している。

J.R.G.Wood：SSMを情報システムと結びつける<sup>(12)</sup>

彼の考えは論文の結論に要約されている。

機能主義から解釈主義のパラダイムへの移行の意義は「情報システムに対するユーザーの要求をもっと正確に記述する」という今までの取り組みを単にもう少し洗練されたものにする、というところにあるのではない。それはむしろ情報そのものが経験される実世界を客観的に記述したものであるとする今までの考えから、それが本質的に主観的な性格を持つものであるという認識への移行を意味するのである。

つまりSSMの哲学を土台にして、情報システム開発方法論を新しく構築し直せ、というのが彼の主張である。

#### 4. まとめ

要求定義に関して今までの情報システム開発方法論が持っていた基本的な問題点の中で、最近の情報システムの環境変化、特に情報システムを必要とする組織の状況が複雑になってきたことに伴って顕在化してきたものは次の3点に集約できる。

1. 有用な情報システムを開発するためには、問題状況とその背景をなす文脈に対する深い理解が不可欠であるにもかかわらず、そのような認識が不足していること。
2. 現状の問題や到達すべき目標は方法論を適切に使用することによって必ず発見することが出来、従ってそれを客観的に定義することが可能であるという客観主義、乃至は実証主義の立場に立っていること
3. あくまでも開発者のための方法論であり、そこでのユーザーの役割はアナリストが必要とする材料を提供し、またアナリストが発見した事実に誤りがないことを確認することである、という考え方。

これに対してSSMは1と2に関しては全く逆の主観主義、乃至は解釈主義の立場に立って、関係者が問題状況に対する理解を深めていくための思考と行動の枠組みを提供している。従って上記の3に対応して、SSMを基盤とする新しい開発方法論を具体的に構築することが出来れば、それは従来の開発方法論が直面している諸問題を根本的に解決するための有効な手だてとなり得るであろう。その方向としては基本的に次の二つが考えられる。

- イ) 開発プロジェクト全体をSSMの枠組みの中で管理し、その中で具体的な開発を行っていくための方法論。先に紹介した埋め込み型の2番目のタイプがこの考え方である。これを更に具体化していくことは今後の大きな研究課題である。
- ロ) 情報システムの開発を従来のように一つのプロジェクトのライフサイクルの中で捉えるのではなくて、組織活動の中で永続的に行われるSSMに基づく取り組みの一つであるという考え方。これはきわめて魅力的な考えであるが、概念的にも手法的にも一層の研究が必要である。

#### 参考文献

- (1) Cutts, G., Structured Systems Analysis and Design Methodology, Blackwell, 1991
- (2) Whitten, J.L., Bentley, L.D. & Barlow, V.M., Systems Analysis and Design Methods, 2nd ed., IRWIN, 1989
- (3) Checkland, P. & Scholes, J., Soft Systems Methodology in Action, Wiley, 1992
- (4) Bulow, I.von, The bounding of a problem situation and the concept of system's boundary in soft systems methodology, Journal of Systems Analysis, 16,1989
- (5) SYSTEMIST, United Kingdom Systems Society, vol.14, no.3, 1992
- (6) Prior, R., Linking SSM and IS development, SYSTEMIST, vol.14, no.3, 1992
- (7) Mingers, J., SSM and information systems: An overview, SYSTEMIST, vol.14, no.3, 1992
- (8) Lewis, P.J., The feasibility and desirability of a closer linking of SSM with data-focused information systems development, SYSTEMIST, vol.14, no.3, 1992
- (9) Stowell, F. & West, D., SSM as vehicle for client-led design of information systems, SYSTEMIST, vol.14, no.3, 1992
- (10) Gregory, F. SSM to information systems: A logical account, SYSTEMIST, vol.14, no.3, 1992
- (11) Merali, Y., Analytic data flow diagrams: An alternative to physicalism, SYSTEMIST, vol.14, no.3, 1992
- (12) Wood, J.R.G., Linking Soft Systems Methodology and information systems, SYSTEMIST, vol.14, no.3, 1992
- (13) Stowell, F. & West, D., Client-Led Design: A Systemic Approach to Information Systems Definition, McGraw-Hill, 1994