DSS設計論に関する一考察

飯島　淳一

1. 序　論

情報処理システムについての研究，特にその設計論は，管理科学における重要な分野の1つである．従来の情報処理システムが対象としている問題は，比較的定型的で構造化された問題である．一方，ＤＳＳが対象とする問題は，定型的な問題よりも非定型的な問題であり，構造の明確な問題よりも不明確な問題であるといわれ，いわゆる非構造あるいは悪構造問題と総称されるものがその特徴であるとされている．したがって，ＤＳＳの設計論は，ＤＳＳが対象とする問題のこのような特徴を踏まえて，展開する必要がある．

さて，従来の情報システム設計論は，対象とする問題が定型的で構造化されており，人間の意思決定の関与する余地が比較的少ないということから，入力から出力への変換を所与とした，オートマトンをモデルとして展開されていた．ＤＳＳは悪構造問題を対象とするので，ユーザーは対象とするシステムに対して積極的にコミットすることが必要であり，また，ユーザーはその目標を変えたり，さまざまな価値判断を行なうことが考えられる．したがってＤＳＳの設計論は，入力から出力への変換を所与としたオートマトンではなく，意思決定要素を陽に含んだ目標追求システムを用いるべきである．このような観点から，ここでは目標追求システムにもとづいた設計論の概念的考察を行なう．また，コンサルティングを行なうＤＳＳを目標として，われわれが従来から提案しているＤＳＳの構築枠組 act DSS と，ここで展開したＤＳＳ設計論との関係についても述べる．

2. 従来の情報システム設計論

ＤＳＳの設計論について考察する前に，従来の情報処理システムの設計論について簡単に述べる．従来の情報処理システム設計論は，大きく分けてウォーターフォール型モデルにしたがったアプローチとプロトタイピングアプローチに分類できる．ウォーターフォール型モデルにしたがったアプローチは，従来の工学において通常行なわれているアプローチと類似のトップダウンアプローチで，**図1**のようなものである．このようなアプローチの下で，従来からさまざまなシステム設計論が展開されてきた．ここで注意しておきたいことは，これらが対象としている情報処理システムは**図2**のように，ある仕様にしたがって入力を出力に決定論的に変換するシステムを考えているということである．このような情報処理システムの設計は，入力と出力の間の関係として与えられ

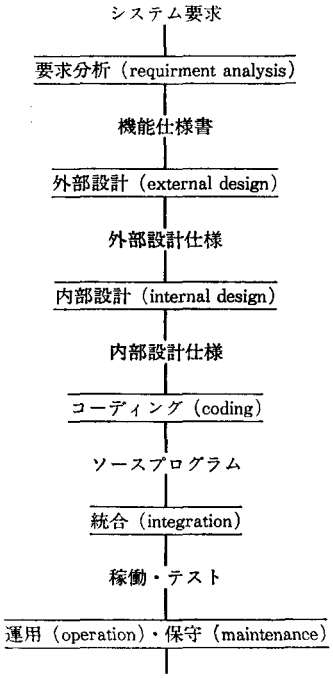

図 1　システム設計におけるウォーターフォール型モデル

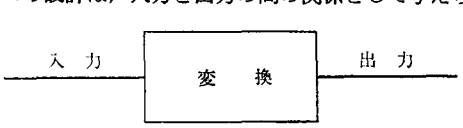

図 2　入力-出力システム

いいじま　じゅんいち　東京工業大学　経営工学科
〒152 目黒区大岡山2—12—1

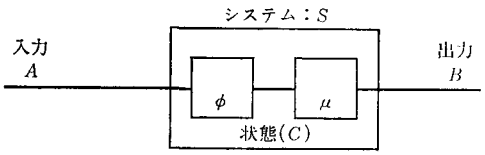

図 3　システム・モデル

る機能仕様を，図3のようなオートマトンによって記述されるモジュールの結合によって実現し，さらにその有限オートマトンをコード化するプロセスとして考えることができる[1].

3. ＤＳＳ設計論の概念的基礎

システム理論ではオートマトンのほかに，目標追求システムと呼ばれるものが代表的なモデルとして考えられている[2]. この節では，目標追求システムをモデルとして用いたＤＳＳ設計論を展開する.

まずはじめに，意思決定の対象となる“問題”を次のような3つ組み $\langle C ; c_0, F\rangle$ で表現する.

定義 1：問題記述

問題記述 (problem description) とは

$PD=\langle C ; c_0, F\rangle$

である．ここで，

C：状態空間と呼ばれる集合；

$c_0 \in C$：初期状態；

$F \subset C$：目標状態の集合，$c_0 \notin F$；

である.

ＤＳＳのユーザーははじめに漠然とした問題状況を持っているが，それを構造化するために，まず上記のような問題記述に変換する必要がある．これはウォーターフォール型モデルにおける要求分析に相当するフェーズで，ここでは問題分析フェーズと呼ぶことにする．問題記述における状態空間を規定するものは，問題にかかわる変数とその変数の間の関係(制約)であるから，問題分析のフェーズでは，どのような変数がどのように関係し合っているかを明確にしなければならない．また現在の状態 c_0 がどのようなものであるか，目標とする状態 F がどのようなものであるかをも明確にする作業が必要になる.

問題記述 $PD=\langle C ; c_0, F\rangle$ に対して，問題解決が行なわれる．ここでは問題解決を，目標追求システムをモデルとして用いることにより考えてみよう．目標追求システムは，決定問題と決定基準の組によって定義される．まずはじめに，決定問題を定義する.

定義 2：決定問題

決定問題とは，六つ組 $\langle M, X, Y, V, P, G\rangle$ である．ここで，

M：代替案集合；

X：外部入力集合；

Y：出力集合；

V：評価集合（弱順序集合）

$P : M\times X \to Y$：プロセス；

$G : M\times X\times Y \to V$：目的関数

である.

問題記述 $PD=\langle C ; c_0, F\rangle$ における状態集合 C と決定問題とは次のような関係にあると考えることができる：

1)　$C \subset M\times X\times Y$；

2)　$(m, x, y)\in C \leftrightarrow P(m, x)=y$

すなわち，プロセスは問題記述における状態集合を規定しているのである．従来のオートマトン的モデルとプロセスが異なる点は，入力を2つのカテゴリに分割したことである．これは，このシステムを利用するもの（以下，ユーザーと呼ぶ）の存在を陽に仮定し，ユーザーが操作できるものとできないものに分類したという点が重要である．前者を外部変数，後者を代替案と呼ぶ．ここでは，話を簡単にするために P を関数としている．プロセスにおける M, X, Y を適当な集合上のベクトルの集合と考えれば，プロセスをあるオートマトンにもとづいて規定できるようにすることができる．このオートマトンに対応する（構文的）言語表現は，ＤＳＳで用いるシミュレーションプログラムに相当する.

以下，記法として，$\underline{DP}$ によって決定問題の集合を表わすことにし，$\underline{R}=\{R \mid (\exists DP\in \underline{DP})$ (M は DP の代替案集合 & $R \subset M\times M$ & R は M 上の弱順序集合$\}$ とする.

定義 3：決定基準

関数 $\delta : \underline{DP} \to \underline{R}$ が，任意の $DP=\langle M, X, Y, P, G,\rangle \in \underline{DP}$ に対して，$\delta(DP) \subset M\times M$ という弱順序関係を与えるとき，δ を**決定基準** (decision principle) と呼ぶ．以降，$\delta(DP)$ を $\leq_{\delta(DP)}$ と表記する.

決定問題と決定基準の組を目標追求システム表現と呼ぶ.

定義 4：目標追求システム表現

決定問題 DP と決定基準 δ の組み $\langle DP, \delta\rangle$ を，目標追求システム表現 (goal-seeking system representation) と呼ぶ.

定義 5：解

$DP=\langle M, X, Y, P, G\rangle$ を決定問題とし，δ を決定基準とする．このとき，解を次のように定義する：

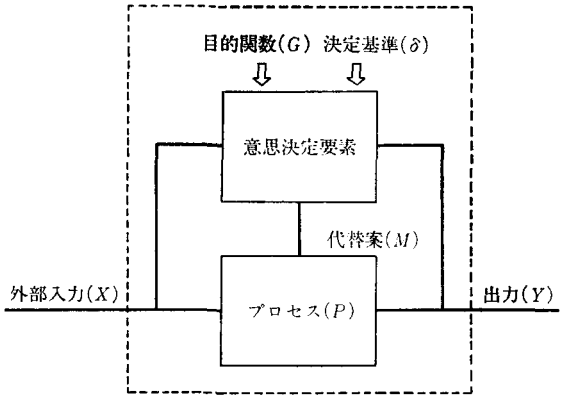

図 4　目標追求システム

$$Sol(DP,\delta)=\{m,x,y)\,|\,m\leq_{\delta(DP)}m'\rightarrow m'\leq_{\delta(DP)}m)$$

問題記述に適合する目標追求システム表現は，その解が，与えられた問題記述における目標状態に含まれるようなものである．目標追求システムを図示すると図4のようになるが，これからわかるように，目標追求システムは従来の入力-出力システムを，オートマトンとは異なる形で構造化したものであり，プロセス，目的関数，決定基準を分離した点が特徴である．

問題記述は複数の部分問題記述に分解されるが，各々の部分問題記述に対応して，複数の目標追求システム表現を構成することができる．この分解は，これらの目標追求システム表現を，さらに上位の層によって統合した二階層のシステム表現が，与えられた問題記述に適合するような分解が望ましい．問題記述を複数の部分問題記述に分解するフェーズを部分問題分割フェーズと呼ぶ．これは，情報処理システム設計の外部設計に相当する．各部分問題記述に対して，目標追求システム表現を与えるフェーズを目標追求システム設計フェーズと呼ぶことにする．これは情報処理システム設計の内部設計に相当する．このようにして，問題記述と適合するような目標追求システム表現を実現する二階層システム表現を構築する作業として，ウォーターフォール型モデルにしたがったＤＳＳの設計論を展開することができる．

以上をまとめると，図5のようなＤＳＳ設計のウォーターフォール型モデルになる．

4.　ＤＳＳ設計論の実施[3], [4]

3.で考察した設計論は，特定ＤＳＳに対する設計論である．この節では，コンサルティングを行なうＤＳＳ構築ジェネレータとして提案している actDSS について述べ，3.で展開したＤＳＳ設計論との関係について考察する．

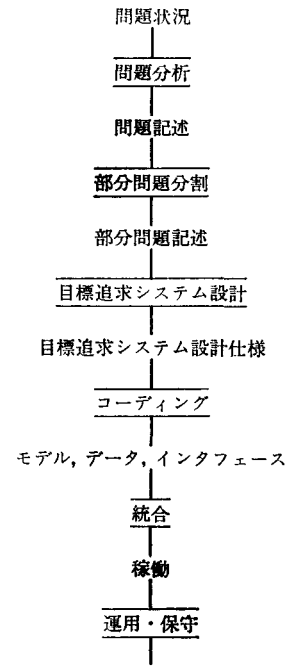

図 5　ＤＳＳ設計のウォーターフォール型モデル

actDSS を構成する“ハードウェア”は，インタフェースの役割を果たすスプレッド・シート，数値処理やグラフ表示などを行なう従来型のＤＳＳジェネレータ，記号処理および全体的な情報処理の制御を行なうプログ・インタープリタから構成されている．このようなコマンドをここではタスク要素と呼ぶことにする．また，これらのタスク要素や数値情報，記号情報が登録されたワークシートをタスクユニットと呼ぶことにする．*actDSS* は，上記のタスク要素の発火やその構造体であるタスクユニットのロードとセーブをアドホックに行ないながら，問題に対する理解を深めることができる枠組みである．

actDSS における情報処理は，図6のように，問題クラス同定（PCI-Problem Class Identifier）問題領域特定(PDS-Problem Domain Specifier)，問題構造特定(PSS-Problem Structure Specifier)，解領域の構成(SDC-Solution Domain Constructor) の4つの部分からなっている．

ＰＣＩは，知識ベースとして，キーワードとそれに対応する典型的な問題名のリストを用意しており，ユーザーとの制限された自然言語を用いた対話により，ユーザーの抱える問題状況に対応する問題名の書かれたワーク

シートを表示する．

ＰＤＳは与えられた問題名に対応して，現在意思決定者が問題としているものに対して，外部入力変数名の集合 V_X，出力変数名の集合 V_Y，評価の基準となる目標変数名の集合 V_G を決定する．これらの３つの変数群による問題の特徴づけを問題領域（problem domain）と呼んでいる．

選択された問題領域にもとづいて，ＰＳＳは問題領域を具体的な意思決定問題として定式化する．すなわち，V_X に対するデータの集合 X を適当なデータベースから探し，操作変数名の集合 V_M を同定し，そのデータ集合 M の領域を確保し，V_G にもとづいて目的関数を構成し，問題に適したプロセス P（対象問題の入出力関係）を作成することにより，意思決定問題 $\langle M, X, Y, P, G\rangle$ を構成する．現在はあらかじめ P と G のクラスを用意し，それらの要素を組み合わせることでこれを行なっている．この段階で選択された論理モデルはあくまでも１つの可能性の表現であり，ユーザーとシステムが論議を行なうためのたたき台である．

ＳＤＣでは，ＰＳＳの作成した論理モデルによる解を提示し，ユーザーの意向にしたがってさまざまな分析を行なう．ここでは，論理モデルとしてシステムが構成した意思決定問題の提示とユーザーの与えた意思決定基準にもとづく代替案の導出，what-if 分析，多目標問題に対するコンフリクト解決，不確実性に対するリスク分析，問題構造や問題領域の修正，あるいは意思決定問題それ自身の変更などが行なえる．

このように，actDSS では，問題状況を持ったユーザーが，問題クラス同定，問題領域特定，問題構造特定，解領域構成の４段階を通じて問題解決を行なうことを考えている．また，記述言語は基本的には論理型言語であるプロローグを用いており，プロローグのプログラムによって，指定されたタスク要素の実行を制御することができる．すなわち，プロローグのプログラムはすぐに実行可能となるようなものであり，その意味で，プログラムがオペレーショナルな仕様に相当している．

actDSS ではまずはじめに，問題状況を持ったユーザーとの対話により，問題のクラスや問題領域を同定するわけであるが，この段階は，3. の問題分析フェーズに相当する．この段階で，問題記述が定まる．問題構造の同定の段階は，3. の部分問題分割と目標追求システム設計の段階に相当する．actDSS では設計者は3. で展開したＤＳＳ設計論にもとづいてＤＳＳを設計するが，ユーザーは，プロセスや目的関数，決定基準などをリアルタイムに変更して，結果をみることができる．この意味で，ユーザーの視点からはユーザーの求めるＤＳＳに対するプロトタイピング・アプローチを行なっていることになる．

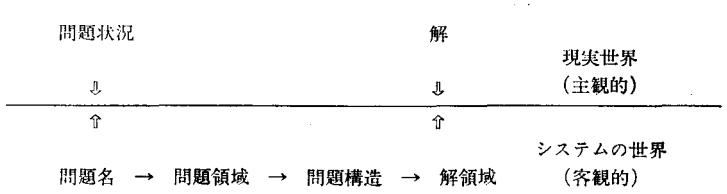

図 6　actDSS における問題解決

5. 結　論

ＤＳＳは従来の情報システムとは異なり，構造化されていない問題を取り扱うという特徴がある．従来の情報システムの設計論の対象となっているシステムのモデルはオートマトンであった．このようなモデルにしたがった設計論では，“active”なＤＳＳを設計することができない．そこで，システム理論におけるもう１つのモデルである目標追求システムをシステムモデルとして，ライフサイクルモデルにしたがって，ＤＳＳ設計論について考察した．

情報処理システムの設計は，オートマトンをシステムモデルとして，数理論理的言語を用いて定式化し，厳密な理論展開ができる[１]．同様に，ＤＳＳの設計を目標追求システムをシステムモデルとして定式化することにより，ＤＳＳ設計について厳密に議論することができる．これについては今後の課題である．

参考文献

[１] 飯島淳一：“ウォーターフォール型モデルによるシステム設計の定式化”，「システム設計の理論と実際」（出版予定）に所収，近代科学社，1990

[２] 高原康彦：システム工学の理論，日刊工業新聞社，1973

[３] 高原康彦，飯島淳一，佐藤　亮，劉　学平，ミゲル・バレラ・ムリジョ：“新しいＤＳＳ概念とその実施構造”日本経営工学会，投稿中

[４] 高原康彦，飯島淳一，李　東，W.パークプーム：“新しいＤＳＳの概念枠組”，日本オペレーションズ・リサーチ学会，投稿中