

DSSのためのコンピュータ・システム —その効果的なフレームワークについて—

広 内 哲 夫

A Computer System for DSS —On its Effective Framework—

Tetsuo HIROUCHI

A Decision Support System (DSS) is by nature general-purpose system because it supports a variety of managers who have different decision styles and different problems. However, it seems that no effective general-purpose DSS has yet come into existence, although the components of DSS such as database technology, modeling techniques, inexpensive graphic display etc., have progressed enough to build DSS.

I think that the shortcoming has resulted from the fact that an effective framework to integrate those components into DSS is still not established.

In this paper, I propose an effective framework that is based on the study of unstructured-problem solving and considerations for use by managers as non-computer specialists. We illustrate a system realized on the framework.

1. はしがき

この数年、対話型システムをベースにした問題解決を支援するコンピュータ・システムの構築が、各企業により試みられている。このようなシステムは一般に「意思決定支援システム (Decision Support System, 略DSS)」と呼ばれている。このDSSの進展に伴い、管理者が自分の仕事の進み具合に合わせてこのシステムを利用して、自分の問題解決 (意思決定) を効果的に進めることができる環境が、企業組織体に出現しつつあるといえる。⁽¹⁾

ところで、このDSSのアプローチは、効果的な意思決定プロセスにおけるコンピュータの役割に関して、次のような仮定がその基礎になっている。⁽¹⁾

(1) コンピュータは管理者を支援するのであって、管理者に取って替わることがあってはならない。

(2) 問題解決に当っては、管理者の洞察力あるいは判断力が必要とされる非構造的なものがその対象となる。

(3) 効果的な問題解決プロセスは、本質的にインタラクティブなものであるので、人間

とコンピュータの即時的な対話が必須である。

筆者が調査したところによると、DSSの目的を持つと考えられるシステムの中で、システムの機能面から上記の仮定(2)、(3)を充分満しているものは稀であった。

(2)に関しては、ほとんどのDSSが機能分化を押し進めて構築されてはいなかった。そのため、管理者は、非構造的な問題をコンピュータ・システム上に充分に表現できず、既存のシステム制約下でそれを固定的に解かざるを得ないようであった。

(3)に関しては、大部分のDSSが対話型システムとして構築されてはいた。しかし、それらは特定のレベルの利用者、例えば、初心者あるいは上級者のどちらかを想定しているために、中間レベルの利用者にとっては、利用しづらいシステムであった。

DSSを構成する基本的な個々のシステム技術、例えば、汎用データベース技術、経営科学手法、問合せ言語、グラフィック表示技術などは、現在、それぞれ充分満足のいく高い水準に達している。しかしながら、未だ幅広く管理者に受け入れられるDSSが存在していないと思われる理由は、それらの個々のシステム技術を総合化するところの技術が確立されていないからであるといえよう。

意思決定のために必須の要素であるデータとモデルを管理者に広く利用させるには、図1に示すように、それら2つの要素と人間の間にインターフェースとして介在するもう1つの要素を確立しなければならない。筆者は、このインターフェースをDSSフレームワークと呼ぶ。このフレームワークが上記の個々の基本的なシステム技術を総合化するところのものであり、それが管理者のコンピュータ使用を強力に支援しなければならないと考える。

従来、DSSの領域で成功したといわれるシステムは、コンピュータの専門家を対象としており、それは特定の問題を扱うために、データとモデルのみにシステム化の努力が払

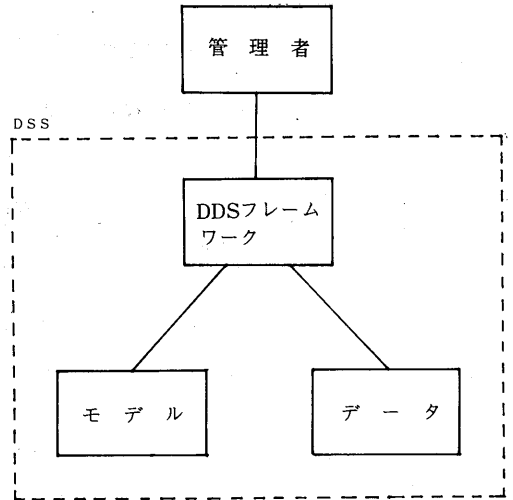


図1 DSSの構成要素

われていたようである。

本論文では、DSSがより広範な管理者に対して、①非構造的な問題を解決するための柔軟性、および②彼等自身で容易に使用できる使い易さを、如何にしたら提供できるか、またそのためのコンピュータ・システムは如何にあるべきかという筆者の考え方を示す。⁽⁶⁾そして、その考え方に沿って、筆者らが開発したDSSフレームワークであるところのDSS1100を紹介する。⁽⁶⁾

2. 柔軟性の考え方

H.A.Simonによれば、人間が非構造的な問題を解く場合、彼はその非構造的な問題を細分化し、自分の知っている慣れ親しんだ方法で解けるところの小さな構造的な副問題までに還元して解くという。⁽²⁾これは一般的、共通的な手法などの組み合わせによって、その非構造的な問題を解決できることを意味する。

一般に、管理者が非構造的な問題を解く場合、頼りにする手法は、経営科学手法、統計手法、四則演算、データベース検索、グラフ表示などの比較的単純で易しいものである。

そこで、筆者は、従来モデルと呼ばれてい

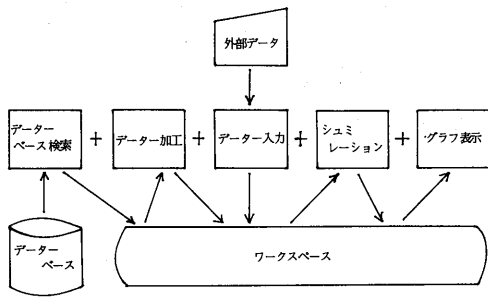


図2 小モデルの連結

たところのものを拡大解釈して捉えようとするものである。すなわち、データベース検索やグラフ表示などの単一の機能もモデルと考えるのである。このモデルは明確に単一機能に限定されるので、ここでは小モデルと呼ぶ。

このようにすれば、小モデルを上記の副問題に対応させることができるので、逆に、小モデルの柔軟な組み合わせにより、非構造的な問題を解くことが可能となるといえよう（図2参照）。

管理者が小モデルを用いて非構造的な問題を解くためには、DSSに次に述べるようなシステム環境が用意されていなければならない。

小モデルは単一機能のみから構成されている。そして、それは単独で稼動することができる。システム内には小モデル間のデータの引き渡しを行うためのワークスペースが設けられている（図2参照）。このワークスペースは管理者にとって、メモエリアの働きをし、必要があれば、外部から種々のデータをこのワークスペースに投入することもできる。小モデルは、他の小モデルとの間でワークスペース内で、データの授受を正しく行うことができるように、ある一定の基準（プロトコル）のもとで作成されている。

また、管理者が、小モデルの存在を鮮明に意識できるようにするため、それらの小モデル群は、システム内の見通しの効くモデル保

存エリアに保存されている。そして、管理者が必要に応じて、当該エリアの中の目的とする小モデルを取り出して、即座に実行できるような機構が用意されている。管理者はこの機能を利用して、次々と小モデルを選び出して実行していく。そして、それに従って、小モデルは連結されて、より大きな意思決定モデルと成っていく。

このようなシステム環境を実現することが可能であれば、管理者は彼の意思決定プロセスに従って、容易に彼の望む意思決定モデルを創造することができるといえる。

3. 使い易さの考え方

システムの使い易さは、問題を解くという本質からは重要ではないにしても、管理者を効果的に支援するという観点からは非常に重要である。⁽⁵⁾

システムの使い易さに関するオー一番目は、システムの操作性に関する問題である。管理者は一般にコンピュータの専門家ではない。そのため、使用したい小モデルの存在を承知していたとしても、それを充分操作できるとは限らない。大部分の管理者は小モデルの基本的な概念を知っているにすぎないのである。しかしながら、一方、人間は一般に教えられ、使い込むと、それに習熟していくものである。管理者も使い込めば、最終的には自己の力量に応じた利用スタイルを確立するであろう。

DSSはこのように、初心者から習熟者までを幅広く受け入れなければならない特質を持っている。そこで、DSSには、次の2つの対話形式を設ける必要がある。

一つは初心者向けのシステム主導型の対話形式である。これは小モデルが何をどんな様式で入力を要求するかを、システムが使用者に教え、誘導していく形式である。これは試行錯誤的な対話が可能な形式でもある。

もう一つは習熟者向けの利用者主導型の対話形式である。これは使用者が、事前に小モ

デルの要求する入力内容を知っている場合、その使用者が一時的にシステムに命令を与えていく形式である。

使用者の習熟過程にシステムが対応していくためには、この二つの対話形式が混在して利用できるようにしておくことが、非常に重要である。

システムの使い易さに関する二番目のものは、問題の構造化についてである。⁽³⁾当初、非構造的な問題も、その出現回数が増すに従って、徐々に問題の解決方法も定まっていき、構造的な問題に転化していく。もし、この構造化への過程に、システムが対応できないと、頻繁に使用者がシステムに介入する必要が生じ、使いづらいものとなってしまう。従って、問題の構造化の進展とともに、徐々に使用者の操作をシステム側に移し替えることができる、次のような機構を設ける必要がある。

問題の解決方法の固定化は、最初、個々の小モデルの入力過程において現われ、次第に小モデル群の組み合わせ過程にも出現するようになる。そこで、小モデルの会話に関する入力イメージおよび小モデル群の組み合わせに関する入力イメージを、システム側に登録できるようにする。使用者が類似の問題に対応する場合、登録された入力イメージを再び弾力的に呼び出すだけで良いようにしておく。

このような機構が完備していれば、使用者はより簡単な操作によって、構造化の進んだ問題を解決することができるといえる。

4. 柔軟性の実現

筆者らが採用したDSSフレームワークのインプリメンテーション・アプローチは、モデルを単一の問題解決機能のみから構成し(以後、このモデルをユニット・モデルと呼び、これは第2章で小モデルと呼んだものに対応する)、ユニット・モデルの持つ共通の基本的な情報処理機能をすべて抽出して、それをDSSフレームワークとして独立させて実現

する方法である。この情報処理機能とは、会話制御機能、ファイル操作機能、出力制御機能、データ管理機能など、従来のモデルの持つ必須の機能である。

このアプローチは次の利点をシステムに提供する。

(1) ユニット・モデルは、その固有の問題解決アルゴリズムだけから構成することができる。従って、他の情報処理機能を含ませる必要はない。

(2) ユニット・モデルに対して、共通のデータの基盤を与えることができる。

(3) 使用者とシステムとの対話形式を、ユニット・モデルの間で共通的に定めることができる。

筆者らは、DSSフレームワークとしてDSS1100を実現するために、図3に示すシステム構造を採用した。そのDSSフレームワークは、図3の中の5つの機能別モジュール、モデルベース、中継データ・ファイルおよびシステム・ファイルから構成されている。ここで、モデルベースと中継データ・ファイルは、第2章で述べたモデル保存エリアとワークスペースにそれぞれ対応するものである。

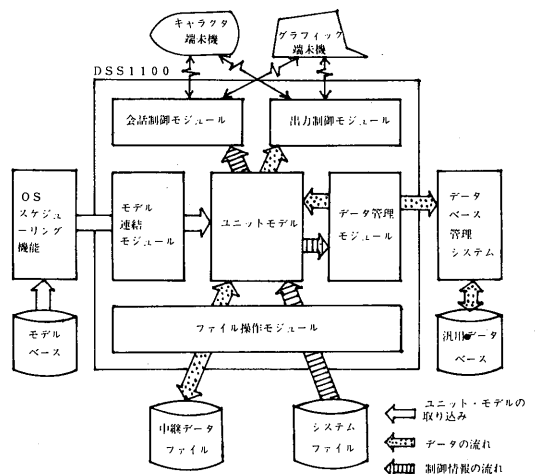


図3 DSS1100のシステム構成

表1 システム・ファイルの種類

ファイルの名称	ファイルの内容
データ定義ファイル	データベースの検索の経路、およびレコード・レイアウトを定義する。
機密保護ファイル	データの機密保護を行うための情報を登録する。
会話情報ファイル	ユニット・モデルごとの会話の順序、形態、および会話メッセージを登録する。
処理手順ファイル	事前に既知である会話入力イメージを登録する。
デコード情報ファイル	コードをデコードした情報を登録する。

機能別モジュールは次の観点から作成されたソフトウェア・モジュールである。

(1) モデル連結モジュール：ユニット・モデルを連結して作動させる。

(2) ファイル操作モジュール：中継データ・ファイルへの入出力操作をユニット・モデルから解放する。また、このモジュールの存在により、各ユニット・モデル間の整合性を保証する。

(3) 会話制御モジュール：会話入力に関する一切をユニット・モデルから解放する。

(4) 出力制御モジュール：出力に関する一切をユニット・モデルから解放する。

(5) データ管理モジュール：ユニット・モデルがらデータを独立させる。また、データベース管理システムに容易にインターフェースを取らせる。

システム・ファイルは、①モデルの開発を容易にするための情報、②システムを稼働させるのに必要な情報、③更新の可能性の高い情報、などをユニット・モデルから独立させて登録させておくファイルであり、表1に示した5つのファイルから構成されている。

モデルベースはユニット・モデル群を保存するためのファイルであるが、データベースのデータのようにユニット・モデルの存在を強く意識させ、ユニット・モデルの取り出し、追加、更新を容易に行えるように設定されたものでもある。このファイルの効用は後述する。

中継データ・ファイルは、ユニット・モデ

ル間での情報の引き渡しを行うために設定されたものである。ユニット・モデルはデータの入出力の対象をすべて中継データ・ファイルとしている。あるユニット・モデルで処理された結果（データ）は、中継データ・ファイルを介して、次に連結されるユニット・モデルの入力データとなる。

中継データ・ファイルは、使用者に当該ファイルのデータを鮮明に意識させるために、一種の関係データベースとなっている。中継データ・ファイルの操作はファイル操作モジュールによって、表（マトリックス）形式の一部であるベクトル形式で行われる。出力データ群には、当該モジュールにより自動的に一定のデータ名が割り付けられる。使用者はこのデータ名を指定するだけでよく、データの物理的な格納構造を意識しなくてもよいようになっている。

ここで説明したモデルベース、中継データ・ファイルおよびモデル連結モジュールなどのDSSフレームワークが、従来の意思決定モデルを創り出す働きをする。これらのシステムの働きは次のように描写できる。

使用者によりDSS1100が開始されると、モデル連結モジュールは会話制御モジュールを介して、使用者に何のユニット・モデルを使用するかを問う。使用者が特定のユニット・モデルの名前を与えると、当該ユニット・モデルはモデル連結モジュールによってモデルベースから呼び出され、図3に示すユニット・モデルの位置に取り込まれる。そして、そのユニット・モデルの実行が開始される。

当該ユニット・モデルは中継データ・ファイルを入力源とし、そのユニット・モデルの持つ問題解決アルゴリズムに従って問題解決に当る。この際、基本的な情報処理に関しては、会話制御、出力制御、データ管理、ファイル操作の各機能別モジュールに委ねる。

機能別モジュールは、システム・ファイル

表2 意思決定モデルの作成例

モデル名	ユニット・モデルの連結順序
データ検索モデル	データベース検索UM→帳表作成UM
重回帰分析モデル	データベース検索UM→重回帰分析UM →グラフ作成UM
時系列予測モデル	データベース検索UM→データ入力UM →データ加工UM→時系列予測UM →グラフ作成UM

(注) UMとはユニット・モデルの略

に登録されている情報を参照しながら、ユニット・モデルに替わって基本的な情報処理を行う。そして、ユニット・モデルは処理結果を中継データ・ファイルに出力する。

当該ユニット・モデルの実行が終了したら、モデル連結モジュールは会話制御モジュールを介して、再度、使用者に次は何のユニットモデルを使用するかを問う。このパターンは使用者がDSS1100に終了を指示するまで繰り返される。

以上がシステムの動作の概略である。上記のパターンの繰り返しの過程を通じて、データは中継データ・ファイルを仲立ちとして、各種のユニット・モデル間に引き次がれ、その間に有用な情報に転化していく。

このフレームワークにより、管理者は種々のユニット・モデルを連結して、一連の意思決定モデルを創り上げることができる。例えば、モデルベースにデータベース検索、重回帰分析、データ入力、データ加工、時系列予測、グラフ作成、帳表作成などの各種のユニット・モデルが用意されていると、表2に示すような意思決定モデルを創り上げることができる。

この種の組み合わせは、ユニット・モデルの数が増せば、それに従って多種多様な組み合わせが生ずる。その結果、システムはより柔軟性を増すことになる。

ユニット・モデルは、すでに述べたように、単一の問題解決アルゴリズムのみを含むプログラムである。このため、管理者は、彼の抱

いている問題を解決するユニット・モデルを容易に作成することが可能である。これをモデルベースに格納すれば、彼を含め誰もが、当該ユニット・モデルを既存のユニット・モデルと組み合わせて利用することができる。それ故、モデルベースに保存されているユニット・モデルは、管理者達の共通の財産とみなすことができる。

5. 使い易さの実現

コンピュータとの対話を実現する一般的な方法は、コマンド方式あるいはメニュー方式である。コマンド方式は利用者主導型でかつ柔軟な対話の様式を実現できる。しかし、それは一般にコンピュータ専門家向けの対話方式といえる。

一方、メニュー方式はシステム主導型の対話を実現できる。この方式は初心者向けの対話方式であり、使用者は対話の流れを教えられながら、システムに指示を与えることができる。しかし、それは若干柔軟性に欠けるきらいがある。

そこで、筆者らはDSSフレームワークに、この両方式の利点のみを取り入れ、習熟過程の追求可能な対話方式を実現した。

表3 逐次入力によるデータの検索

(1)	プログラム NOヲタイプインシテクワサイ
TYPE IN :	50
クワシユンケンクワ	
(2)	レコードメイヲタイプインシテクワサイ
TYPE IN :	1
クワアケシヨクホク	
(3)	デンNOヲタイプインシテクワサイ
TYPE IN :	ALL
ケンズクハ	17クワデス
(4)	シヨクビンNOヲタイプインシテクワサイ
TYPE IN :	1010
セキ	
(5)	データノシユルイヲタイプインシテクワサイ
TYPE IN :	PLAN,RESULT
ケイカクチ	
シツセキチ	
(6)	データノヒツケクヲタイプインシテクワサイ
TYPE IN :	7905.S
79ネン5カツイコウノデータ	
(7)	データメイヲタイプインシテクワサイ
TYPE IN :	A,B
ワリフシテデータメイハA001カラA017デス	
ワリフシテデータメイハB001カラB017デス	

表 4 一括入力によるデータの検索

```
(1)      プログラム NO ラ タイプ イン シテ クタ サイ
TYPE IN : 50:1:ALL:1010:PLAN.RESULT:7904,S:A,B
```

通常の会話は、表3に示すようなメニュー方式の逐次入力を基本とした。これはシステムが使用者に何を入力すべきかを会話メッセージにより問い、使用者が逐次、それに答える会話形式である。この会話形式は習熟度の未だ低いレベルにある使用者を対象とするものである。表3は、データベース検索ユニット・モデルが選択され、前章で述べた会話制御モジュールが、当該ユニット・モデルに替って、使用者に順次必要な入力を求めている様子を示している。

ユニット・モデルを使い込んで、全ての入力すべき項目とその順序を知った最も習熟度の高いレベルにある使用者に対しては、システム側からの会話メッセージ表示をブロックして、一括した入力が行える会話形式を用意した。これは使用者が逐次表示される会話メッセージに回答しているある段階から、その逐次に行われる会話を止め、以後の会話のすべてあるいは一部を一度に入力する方法である。表4に示される一括入力は、表3の逐次入力と等価な入力である。

これらの逐次入力と一括入力は全てのユニッ

表 6 HELPコマンドによるコードの表示

```
(3)      デン NO ラ タイプ イン シテクダ サイ
TYPE IN : *HELP
          1XX   トウキヨウ チク
          2XX   オオサカ チク
          3XX   ナゴヤ チク
TYPE IN : *HELP 1
          11X   チョウワ
          12X   ミナト
          13X   シナガワ
TYPE IN : *HELP 12
          121   アカサカ シデン
          122   アヤマ シデン
          123   ロウホクンキ シデン
```

ト・モデルに対して混在して組み合わせて使用できるようにした。これらの入力の間の切り換えは使用者が行う必要はなく、前述の会話制御モジュールによって自動的に行われる。

一方、試行錯誤的な会話ステップの制御、入力パラメータの教唆、入力の取り消し、訂正などの会話制御の操作はコマンド方式で行い、会話操作の柔軟性を向上させた。表5はHELPコマンドでモデルベースに登録されているユニット・モデルの内容を見ている例、表6は同じくHELPコマンドで店コード体系を段階的に調べている例である。

構造化過程を追跡する機能としては、次の2つの機能を設けた。問題の構造化に対しては、単体マクロ機能と呼ばれる機能を設けた。問題の構造化はユニット・モデル単位の会話

表 5 HELPコマンドによるユニット・モデルの一覧

```
(1)      プログラム NO ラ タイプ イン シテ クタ サイ
          タイプ イン ホウホウ カ"ワカラナゲレバ" *HELP ラ タイプ イン シテ クタ サイ
TYPE IN : *HELP
          <<データベース.ケンサク>>   <<ヨソク/ファンセキ/カコウ>>   <<グラフ(グラフイック)>>
          50 タンシ ユン.ケンサク       5   データ.カコウ               21 (25) シンケイレツ.グラフ
          51 ヒカク.ケンサク           6   シンユウカイキ.ファンセキ   22   ソウカン.グラフ
          52 トウケイ.ケンサク         11  センキ ノビリツ             23 (26) ホウ.グラフ
          <<チヨウヒヨウ>>              12  センネントウキ ノビリツ     24 (27) レータ.チャート
          2   リポート.ライター         13  トレント
          <<イチラン.シュツリョク>>      14  EPA ホウ                     <<データ サクセイ/コウシン>>
          3   ソクセイ.リスト           15  シズウヘイガツ ホウ        61  MSヘンズウ.サクセイ
          4   データ.リスト             <<ズケイ シツリョク>>        62  MSヘンズウ.コウシン
          TYPE IN : 50                 29  スケイ シツリョク         63  MSファイル.ロード
          <<ズケイ シツリョク>>        64  MSファイル.セーブ
```

表7 単体マクロ機能によるデータの検索

```
(1)      プログラム NO ライフイン シテ クワサイ
TYPE IN : $URIMAGE1
(3)      デンメイ ライフイン シテ クワサイ
TYPE IN : 121,122,123
アカサカ シデン
アヤマ シデン
ロツボニンキ シデン
```

の固定化から始まる。これは事前に既知の一連の会話入力に名前を付け、それを前に述べたシステム・ファイルに登録するものである。使用者は当該ユニット・モデルを利用する場合、その単体マクロ名を会話により与えるだけでよい。システムの会話制御モジュールがあたかも、全ての入力がなされたのと同様な処理を行う。表7は、大部分の会話が固定化されたデータベース検索の例である。店コードに関しては、未だ固定化されていないので、システム側から問い合わせが行われている。`$URIMAGE1`という単体マクロ名で登録されている会話入力イメージは、店コードを除いて表3と同一である。

ユニット・モデルの構造化が押し進められると、次はユニット・モデルの組み合わせによる構造化に進んでいく。そこで、システムには単体マクロの組み合わせ情報をシステム・ファイルに登録し、それを処理する連動マクロ機能を設けた。使用者は、唯一の連動マクロ名を入力するだけで、複数のユニット・モデルを連動して実行することができる。

筆者らの開発したDSS1100では、上記のメニュー方式の会話とコマンド方式の会話を混在して利用することができる。しかも、それらの会話の中で、単体マクロ機能と連動マクロ機能を自由に用いることが可能である。そのため、DSS1100の会話方式は、初心者から習熟者まで、幅広くかつ習熟レベルに応じて支援することが可能になっている。

6. あとがき

従来、DSSの名で呼ばれていたシステムは、どちらかと言えばコンピュータの専門家向きであり、また、個々のモデル群が十分な相互関連を持たず、統合化されていると言い難かった。筆者らの開発したDSSフレームワークであるところのDSS1100は、システム統合化の観点から、システムの柔軟性と使い易さの実現を目指したものである。使い易さに関して言えば、DSS1100は、コンピュータに不慣れな管理者をも使用可能にしているところに、その特徴がある。

しかし、最大の特徴は、非構造的な問題を解決するために、管理者自らが各種の単一機能を持つユニット・モデルを連結して、一つの意思決定モデルを組み立てて行くことができることにある。そして、この組み立ての過程において、管理者が随時、試行錯誤的にシステムに介入することができるので、彼は創造されたモデルの意味およびそのモデルから出力される情報の特質を的確に把握することができるのである。

DSS1100はUNIVAC1100シリーズのコンピュータで稼動するものであり、主に銀行、証券会社において資金運用計画などの計画業務に適用されている。

DSSにおける小モデルの種類およびデータの在り方について、紙面の都合上記載できなかったが、それらについては、参考文献一覧の文献(6)、(7)を参照されたい。

なお、本論文は筆者と日本ユニバック(株)の小坂武氏との共同著作である文献(6)を母体として著わされたものである。筆者の単独執筆の形をとってはいるが、その陰に氏からの絶大なる御協力があつたことを明記しておく。ここに小坂氏に深く感謝の意を表する。

参考文献

- (1) P.G. Keen, "‘Interactive’ computer systems for managers: A modest proposal", Sloan management review, spring, 1977.
- (2) H.A. Simon, "The new science of management decision", Revised edition, Prentice-hall, New Jersey, 1977.
- (3) G.A. Gorry and M.S. Scott Morton, "A framework for management information systems", Sloan management review, fall 1971.
- (4) J.J. Donovan and S.E. Madnick, "Institutional and ad-hoc DSS and their effective use", Proceedings of conference on decision support systems, A quarterly news letter of SIGDBP, Vol. 8, No. 3, winter 1977.
- (5) Eric D. Carlson, "A approach for designing decision support systems", SIG DATA BASE, winter 1979.
- (6) 広内哲夫・小坂武者 "経営情報システムの新しいアプローチ"
Computer Report 1979年2月号～1979年9月号(8回連載).
- (7) 原野秀永・広内哲夫著 "M-DSS (管理者のための意思決定支援システム)"
オペレーション・リサーチ, Vol. 25, No. 11, 1980.