



Title	問題状況の構造化過程へのコンピュータの採用：創造的な問題提起のために
Author(s)	増山, 和夫
Citation	デザイン理論. 1985, 24, p. 2-24
Version Type	VoR
URL	https://doi.org/10.18910/52570
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

問題状況の構造化過程へのコンピュータの援用

——創造的な問題提起のために——

増 山 和 夫

1. はじめに

本来、CAD (Computer Aided Design) は、デザイン活動へのコンピュータの広義な援用を意味する概念であった。然るに昨今では、その実用的活用という点から、作図作業の機械化、生産性の向上に重点を置いて、Computer Aided Drafting として、狭義に扱っていることが多い。また、CAE (Computer Aided Engineering) は、技術計算など設計過程における解析的な側面へのコンピュータの援用を志向するものとして、人々の関心をひいている。両者は、コンピュータ・グラフィックスの技術的発達とともに、形状モデルを介して互に接近しつつある。前者が、作図作業のみに留らず、データ・ベース技術を基盤にして、形状処理の過程に解析のためのソフト・ウェアを取り込もうとしているのに対して、後者は、解析結果を視覚化して理解し易くするために、図形処理機能を積極的に導入しようとしている。

形状データを中心とした、図形処理および解析過程へのコンピュータの援用とは別に、デザインにかかわる調査データの分析、技術知識の蓄積・検索などへの援用も少なくはない。なかでも、デザイン過程の初期段階における問題認識、問題定義のためのコンピュータの援用は、CAD の概念の提唱とほぼ同時に試みられている。この方面の研究は、1964年に発表された Christopher Ale-

xander の“Notes On The Synthesis Of Form”^①に始まり、デザイン問題の構造同定 (Problem-Structuring)、広義には、問題解決 (Problem-Solving) の方法論の研究として今日に至っている。^②

いわゆる CAE、あるいは、CAD/CAM が、コンピュータ・グラフィックスの技術を細部設計に適用するという、図形処理的な作業の自動化を中心としているのに対して、調査データの分析、知識データ・ベースの構築などへのコンピュータの援用は、問題の構造同定を通して、問題定義、情報の構成・構造化、分析、評価など多様な可能性を秘めている。この方面の研究は、何ごとかを為すための手順、方法として「いかに」を究明することよりも、「何を」為すべきかを明らかにすることを志向している点で重要である。因みに、デザイン活動の対象とすべき「問題」は、予め「問題」として明らかになっているのではない。われわれの前にあるのは「問題のある状況 (問題状況)」である。そのような状況を前にして、何を問題とすべきかを考えることがなければ、具体的なデザイン活動は始まらない。

2. デザインをどう捉えるか

広義の CAD は、文字通りコンピュータをデザイン活動の道具として利用することである。従って、デザイン過程のどの部分に、どのような形でそれを援用できるか、あるいは、援用すべきであるかを考えることが重要である。そのために、一方ではコンピュータの可能性を追究しなければならないが、他方で、デザイン活動はいかにあるべきかを明らかにすることが必要である。

コンピュータの基本的な能力は、記憶と演算である。いわゆる CAD システムが一般的にそなえている性能は、情報検索、情報表現、特性評価、製造情報の作成などである。いまさら繰り返すまでもなく、コンピュータは、それを使うおうとする人間が考えている以上のことは何もできない。デザイン過程におけるコンピュータの可能性は、それを使うデザイナーに依存している。そこで重

要なことは、デザイナーがそのデザイン活動をどうあるべきだと考えるかである。即ち、CAD について考えることは「デザインとは何か」を考えることであり、具体的な過程と作業の中で、どのようにコンピュータの能力を有効に活用できるかを考えることである。

現状の CAD システムは、設計の補助的な作業である手作業を自動化しているにすぎず、設計の主作業としての知的、創造作業を行うものでないとする意見^③もあるが、この場合でも、設計とは何かという問いが残る。

「デザインとは何か」については、多様な捉え方があるが、デザイン概念の歴史的変遷、デザインと他の専門領域との関係（位置づけ）、あるいは、デザインの実践活動について考察すると、昨今のデザイン活動は、明らかに人間志向になって来ていることがわかる。

デザイン概念の史的変遷をみると、手工芸中心のもの作りの時代には、芸術の問題であったデザインの概念が、工業生産体制のもとでは技術的問題として捉えられる。公害、資源、エネルギーの問題の顕在化とともに、デザインは、生産側の問題であるばかりでなく、使用者（消費者）および、その生活と深くかかわるものとして捉えなければならないとされるようになった。デザインは、社会的な問題であり、デザイナーの働きかける対象は、作られるべき製品そのものではなく、それをとりまく（使用する）人間であるということになる。人が生きて行く上で起る一切の社会的問題が、デザイン活動のテーマであり、究極的には「人間とは何か」という哲学的な問いに至ることにもなる。

デザインと他の専門領域との関係に注目すると、デザインは、多様な領域の研究成果を統合して、具体的な生活空間の形成に参与すべき位置にあることがわかる。このことに関しては、Jay Doblin のモデル（DESMOD）が興味深い^④（Fig.1）。このモデルは、人間の精神作用（行動）を2つの普遍的な軸の上に位置づけ、これらを直交させることによって得られる。図のように、デザインは4つの主要な領域と重なる部分を持っている。このことは、デザインが、

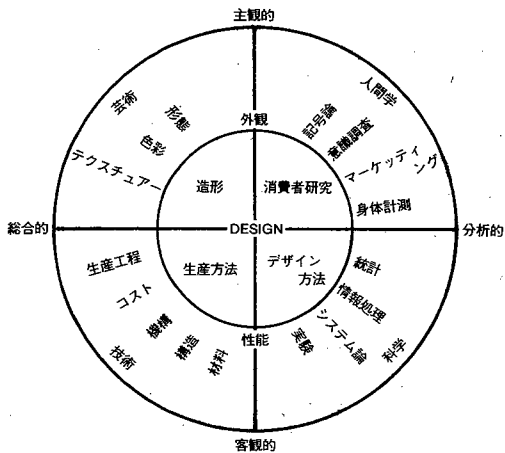


図 1. DESMOD (Design Model)

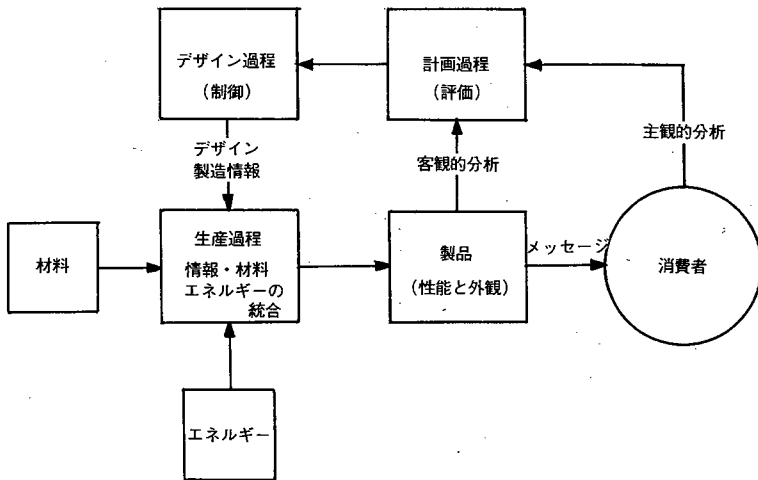


図 2. ものが作り出される過程のモデル

各々純粹であるとされている他の専門領域を、全体として調和的に統合しようとするものであることを示している。デザイン活動は、芸術作品を作るのではないし、技術（Technology）がするように、具体的に製品を作ることもない。最終的な結果に至る手順、方法を明らかにすることがその使命である。その内の「手順」「方法」を重視する立場に立てば、デザインは、ものの造形、材料、加工法など、芸術や技術の領域とより深くかわるものとして扱えられる。他方、「最終的な結果」が何であるかに注目するとき、デザイン活動は、望ましい効果、あるべき状態、ものごとの意味を明確にすることから始められなければならないから、デザインは、人間学（Humanics）の領域とより強い関係を持つものとして扱えられる。

デザイン方法の研究の目的は、この望ましい最終的な結果が何であることを明確にするために、複雑多様な情報を処理する有効な方法を究明することにある。

デザインの実践活動は、製品が作り出される過程と切り離しては考えられない（図2）。すべての製品は、情報と材料とエネルギーを総合することによって成り立つ。図中の矢印は情報の流れを表わし、各構成要素は情報の変換過程を表わしている。製品が作り出される過程は循環的であるが、原理的には、製造情報を内包している「デザイン過程」から始まるのが普通である。ところが、これからのもので作りは、「いかに作るか」よりも「何を作るべきか」を重視すべきだとされる。その「何を」は、問題状況を前にして、どのような問題を提起するかにかかっている。問題提起は、「計画過程（Planning）」の所産である。

問題状況は、人々の生活の中にある。何かがなければならないことは分かっているものの、何があればよいのか分らない状況。今のままで満足しているわけではないが、望ましい状態がどのようなものか説明できない状況。このような状況は、それに対して直接解答が得られるような問題そのものではない。問題は、立場、視点を定めた時にはじめて現われる。その意味で、問題は作り出されるものであると言える。問題状況を素材にして、創造的な問題を提起するこ

とがデザイン活動の始まりである。

3. デザインと問題解決

製品が作り出される過程を参照して、デザイン過程へのコンピュータ援用の現況を見ると図3のように表わせる。コンピュータ・グラフィックスの技術を利用した、図形处理的な作業を中心とするCAD/CAM、あるいは、CAEは、デザイン過程と生産過程との間の関係、つまり、製造情報の作成にかかわる領域へのコンピュータの援用である。デザイン過程以前にあって、デザイン目標の創成にかかわる領域にコンピュータを援用しようとするのは、広義の「問題解決」の概念である。

デザイン過程を「概念集合から実体集合への変換過程」と定義するなら、デザイン過程とCAD/CAMおよび問題解決過程との関係は図4のように表わすことができる。

問題解決過程の一般的概念は「ある目標に向ってとられる意図的な行動である」と定義できる。現状に対して何らかの働きかけをして、意図した通りの望ましい状態を作りだそうとすることである。とるべき行動が予め決っていない

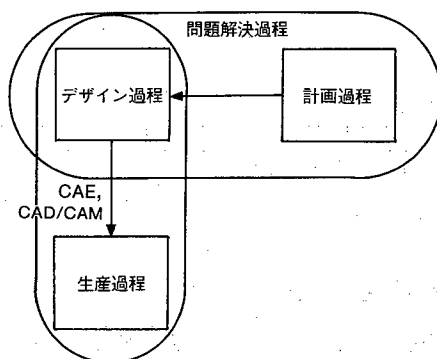


図3. デザイン過程へのコンピュータ援用の現況

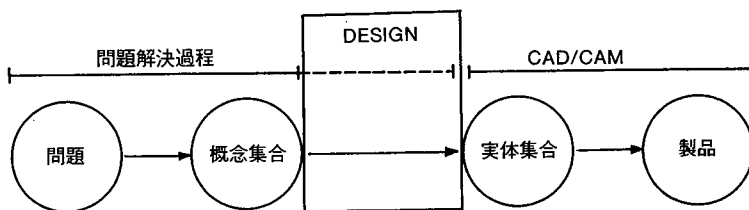


図 4.

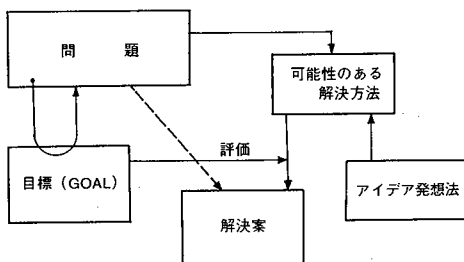


図 5. 問題解決過程の概念

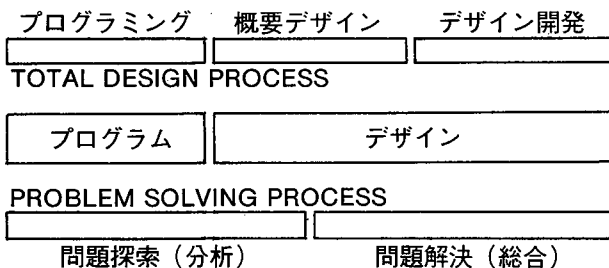


図 6. デザイン過程と問題解決過程

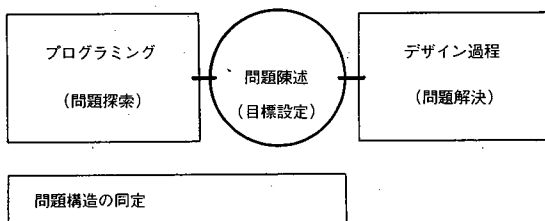


図 7. 問題構造の同定とデザイン目標の設定

ところが動物の本能的行動とは異なる。^⑤

この場合の「問題」は、ともに明らかになっている現状と望ましい状態との差をどのように埋めるかということである。この問題は、要求、制約条件、限界などを明確にし、それを目標 (Goal) として設定することによって定義される。問題解決過程の概念を図示すると図 5 のようになる。定義された問題から直接解答を得るのは容易なことではない。そこで通常は、多様な発想法を使って可能な限りの解決方法を提案する。多くのアイデア、発想から、予め設定しておいた目標に照らして、最善の解決案を選定する。

この概念図を詳細に検討すると、広義の問題解決過程は、問題探索過程 (Problem Seeking Process) と狭義の問題解決過程に分けられることがわかる。これを全般的なデザイン過程と対比すると図 6 のように表わせる。^⑥

問題探索過程と狭義の問題解決過程の間でデザイン目標が設定される (図 7)。整理された与条件とデザイナーの主張とをまとめて、何が心要か、何をなすべきか、望ましいことは何か、何が最も重要かを明確にしなければならない。この問題構造の把握からデザイン目標の創成に至るまでの過程は、問題構造の同定過程 (Problem Structuring Process) である。

問題構造同定の手順は、問題の構成要素をすべて洗い出すことから始まる。それら構成要素とは、前記のように、そのプロジェクトが最終的に満たすべき

要求、制約条件、性能仕様などである。これらの構成要素を列挙したままでは、それら相互の間に脈絡がなく、渾沌としている。情報が不足しているのではなく、手に負えないほど多くの情報、知識、データが無秩序な状態であることが、問題定義を困難にしている。そこで、為すべきことは、これらの情報、データを相互の間の関係の有無に基づいて構造化し、的確かつ首尾一貫した陳述として組織立てることである。そして、それを幾つかの扱い易い下位問題に分割する^⑦。

問題を幾つかの構成要素に分節し、それらの間の相互関係に基づいて、合理的な構造を発見する。ここで、個々の構成要素とは、要求、願望、あるいは、解決案などであるから、この方法は、問題をその解決案に照して構造化しようとする。このことが、その特徴であると同時に限界でもある。

4. 問題の構造

従来のシステムティックな問題構造の同定法では、コンピュータを援用するための準備として、要素リストの作成、要素間の相関関係の定義、相関行列の作成などを手作業で行わなければならない。これは、大変面倒な作業であるし、相関行列の成分値の信憑性は保証されにくい。また、この方法は、デザイン過程の極く初期の段階で、明確かつ完全に評価基準が設定できることを前提にしている。つまり、問題定義をするに際して、それを既存の枠組に嵌め込もうとする姿勢が見られる。これは、この方法がシステム工学、あるいは、システム分析の考え方を基盤にして成り立っていることを示している。要求されているものがはっきりしていて、それを最も安価で効率よく実現するための最善の方法を選択しようとする。工学的設計の多くは、このように要求と目標とが始めから与えられている問題を対象としている^⑧。従って、ここでの問題は「何を」ということではなく「いかに」ということである。

従来の問題構造同定法の難点は、このようにハードなシステム工学的考え方

を、人間とその生活というソフトなデザイン問題に適用しようとするところにあった。

デザイン活動の対象となるものは、本来受動的な経験としてしかなく、明確に説明しにくい形でしかない。従って、予め目標を設定することは不可能である。現実の生活空間に現われた問題状況を、既存の評価基準や枠組にとらわれないで同定できる方法がなければ、新たに生活空間の形成に継がる問題提起はできない。

経験の中で受動的に発見された問題状況を、その構成要素と要素相互の関係によって構造化する。構造化された問題状況を1つのシステム・パターンとして同定する。このパターンから問題状況の性質を読みとり、それを望ましい状況へ向けて創造性に転換することを考える。創造的飛躍によるパターンの変換、変形、組み換えの結果得られる新しいシステム・パターンは、何が為されるべきかを表わす概念モデルである。

われわれが、混沌、茫漠とした状況を秩序立った形で認識できるのは、その状況に分節的な記号を割りつけて「切り分け」を施しているからである。従って、問題状況の構造を同定するためにも、その状況の記号による分節が有効であろうと考えられる。しかし、上記のように、問題状況は受動的に認知されるものであり、説明し難いものであるということは、それが尋常な方法では分節、構造化し難いということである。そこで、デザイン活動の対象となる問題状況の特性に注目すると、それは、人々の日常生活の中での行為、行動にかかわるものであることがわかる。人間の行為、行動、あるいは、事象の生起現象は、一連の時間、空間の中で分節し、相対的に位置づけ、パターン化することができる。

記号を用いて分節された世界、あるいは、人間活動は、その境界に記号を割りつけ、それを新たな分節とする時、新しい境界が生れる。既に分節されているものを新しく分節し直すことができる。分節された行為、行動あるいは、現

象の境界に直観、隠喩、感情的要素などによって飛躍をもたらし、新たなパターンを構成することができる。^⑨分節とパターン化によって構造化された問題状況に新たな切り分け、パターン変換を施すことによって望ましい状況としての概念モデルを作る。この概念モデルは、与えられた情報から論理的に正解が導かれるというように、連続的、収束的な思考によって得られるものではない。それは、問題自体が明確に定義されていない場合に、与えられた情報から様々な新しい情報を作り出そうとする、主として不連続的で、発散的思考、創造的飛躍によって得られるものである。従って、この概念モデルは、問題状況に対してとられた特定の立場、考え方を反映している。それは、創造的に提起された「問題」である。ここに提起される問題は、何が、どんな順序で起らなければならないかを表現している。これは「何を」のレベルで、具現化すべき特定のシステムを定義したものであるが、所与の問題状況に対しては「いかに」を表現するものである。この概念モデルを「何を」として把え、それを「いかに」達成するかを考える時、この「いかに」をあらためて「何を」のレベルで把え、それに関する「いかに」を考えなければならないこともある。従って、どのレベルの「何を」に対する具体的な手段として媒体（道具）を位置づけるかは、デザイナーにとって大変重要である。

5. 問題状況の構造化のために

問題状況の調査とその構造化のために応用できる方法の1つに、従来発想法として知られているKJ法がある。^⑩この方法によれば、野外調査（探検）に際して、調査目的に関連がありそうな事柄をすべてノートに記録する。その記録を整理して、ひとまとまりの内容ごとにカードを作る。それらを要素として、意味内容の親近性によるチーム編成、グループ編成を行い、最終的には、意味関係によるチーム相互の論理的な配列を得る。それを図解したり、文章化して、現状の構造化、現状についての陳述を行う。カード化された要素間の相関関係が数値化できるならば、それをデータとして数量化4類を適用し、親近性によ

るチーム編成の過程にコンピュータを援用することもできる^⑪。しかし、この場合、その結果だけから構成要素の論理的配列を自動化に引き出せるものとはならない。また、要素間の相関関係を数値化する作業は、従来のシステマティックな問題構造の同定法における相関行列の作成作業と同様、大変面倒で、信憑性が保証されにくい。

要素間の関係づけは、類似性だけに依らなくても善いはずである。時間的前後関係、空間的前後関係、上下関係、あるいは、上位概念、下位概念という関係もあり得る。場合によっては、このような関係づけの方が客観的であることも多い。特に、デザイン活動の対象となる問題状況を、人々の生活の中にある行為、行動、あるいは、事象の生起現象の集合として捉えようとする時には、時間的、空間的に関係づける方が容易である。

必要があれば、時間的、空間的な近接関係を意味論的な同義関係に、そして、時間的前後関係、空間的前後関係、上下関係を同様に階級関係に対応づけて考えることもできる。

ここに提案する問題状況の構造化のための方法は、問題状況の構成要素への分節とそれら要素間の時間的前後関係を記録する VTR 分析装置と、要素間の関係をデータとして、問題状況の構造同定ができ、対話的に構造変換、概念モデルの構築ができるプログラム (DCOTH : Decomposition of Directed Graph and Composition of Thesaurus) とから成っている。

VTR 分析装置を使えば、ビデオ録画された問題状況を、マイクロ・コンピュータで制御可能なビデオ再生機によって再生しながら、構成要素の生起過程を時間データとともに記録することができる。具体的には、録画された状況の中に現われる行為、行動、および、その主体、あるいは、生起する事象を予め抽出する。それらをその状況の構成要素として、個々に、コンピュータのキーボード上の任意のキーに割りつけ、登録する。再生映像の始まる時刻を初期値として、任意に設定して分析を開始する。映像を観察しながら、登録した

要素が現われた時点で、それに対応するキーを押すと、その時の相対的な時刻と出現した要素とが、1組のデータとして記録される。これを繰り返して、記録された状況の中に出現するすべての要素とその生起経過とを記録することができる。コンピュータとビデオ再生機とは、ビデオテープ上のCTL信号を介して、同期するようになっているため、巻き戻し、再生、早送りなどを繰り返しても、初期設定された開始時刻からの相対的な時間は常に保持されている。従って、どの時点からでも、記録を始めることができるし、過去のデータに新しいデータを挿入することも出来る。一度の再生では、1つの要素にだけ注目して観察し、記録することが許される。要素ごとにこの観察を繰り返すことによって、多様な要素が出現する複雑な状況でも、比較的容易にデータとして記録することができる（図8）。

このデータから、任意の要素の出現頻度、あるいは、特定の要素の前後に出現する別の要素などを知るのには、極めて容易である。

DCOTHプログラムでは、問題状況をその構成要素に対応する頂点と、それらの間の関係を表わす方向性のある稜から成る有向グラフとして表現できると

```
THE NUMBER OF SEGMENTS IS.... 156
*** DISPLAY DATA ***
```

TIME	SEGMENT	RH	LH	W	LG
@@ START @@					
0: 0:21 ->	1		MONEY		
0: 0:29 ->	2		MONEY		
0: 0:31 ->	3		MONEY		
0: 0:35 ->	4		BUTTON		
0: 0:45 ->	5			BENDING	
0: 0:47 ->	6			BENDING	
0: 0:47 ->	7	DOOR			
0: 0:50 ->	8			BENDING	
0: 0:56 ->	9			BENDING	
0: 0:56 ->	10	DOOR			
0: 0:57 ->	11		DOOR		
0: 0:57 ->	12	GOODS			
0: 0:58 ->	13		DOOR		
0: 0:59 ->	14			BENDING	
0: 1: 3 ->	15				BENDING
0: 1: 8 ->	16	CHANGE			

図8. VTR 分析装置による行為、行動分析（部分）

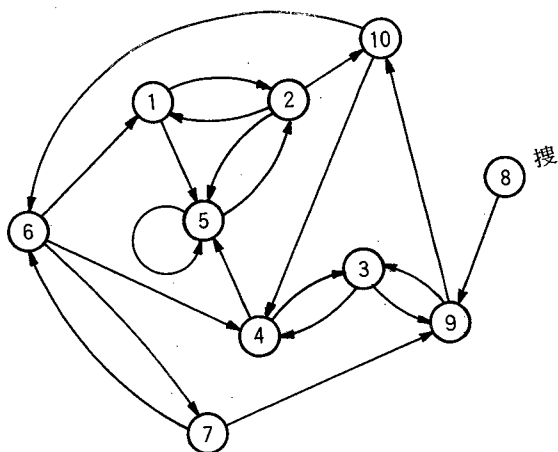


図9. 問題状況の有向グラフによる表現

RELATION MATRIX *****
FACTORS : 10 STRING : 21

	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	1	0	0	0	0	0
2	1	0	0	0	1	0	0	0	0	1
3	0	0	0	1	0	0	0	0	1	0
4	0	0	1	0	1	0	0	0	0	0
5	0	1	0	0	1	0	0	0	0	0
6	1	0	0	1	0	0	1	0	0	1
7	0	0	0	0	0	1	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	1	0	0	0	0	0	0	1
10	0	0	0	1	0	0	0	0	0	0

図10. 有向グラフの行列表現

REACHABILITY MATRIX *****
FACTORS : 10 STRING : 75

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	0	0	0	1	1
2	1	1	1	1	1	0	0	0	1	1
3	1	1	1	1	1	0	0	0	1	1
4	1	1	1	1	1	0	0	0	1	1
5	1	1	1	1	1	0	0	0	1	1
6	1	1	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	1	0	1	1
8	1	1	1	1	1	0	0	1	1	1
9	1	1	1	1	1	0	0	0	1	1
10	1	1	1	1	1	0	0	0	1	1

図11. 到達可能行列

考える (図9)。そして、この有向グラフによって表現 (パターン化されたもの) の構造をより明確にするためにコンピュータを援用する。(なお、DCOTH プログラムと VTR 分析装置とは、互に独立しているので、DCOTH プログラムのデータは、必ずしも、時間的關係、空間的關係によって関係づけられたものに限らない)

グラフの構造を明確にするために、このグラフを頂点間の結合性を利用して、いくつかのサブ・グラフに分割する。有向グラフをサブ・グラフへ分割しようとする場合、2つの方法がある。^⑩1つは、サイクル結合をしている頂点を取り出す方法である。サイクル結合した頂点の小集合をグラフのリーフ (LEAF) といい、同じリーフに属する頂点に対応する要素は、何らかの意味で近接 (同義) 関係にあるとする。もう1つの方法は、リーフを更に分割してローブ (LOBE) を取り出す方法である。まずリーフの中でサーキット結合をしている稜を取り出し、それらの稜の始点、終点となっている頂点の集合を取り出す。ローブを構成している頂点に対応する要素は、リーフの場合と同様の関係にある。頂点間にある稜の数、即ち、道の長さが短いほど近い関係にある。

問題状況を表わす有向グラフを、頂点の数に等しい行と列を持った行列で表わし、行列の性質を利用して、リーフおよびローブに分割するためのプログラムを開発した。

任意の頂点Aから他の頂点Bへ向う稜があれば、成分ABを1とし、なければ0とする(図10)。この行列から得られる到達可能行列(Reachability Matrix)を求める(図11)。この到達可能行列において、任意の頂点A、Bについて、成分ABが1で、成分BAも1であるような頂点を挿す。このような頂点の集合が、サイクルに結合をしている頂点の集合であり、リーフを構成している(図12)。このようにして得られたいくつかのリーフについて、1つのリーフに属する頂点から他のリーフに属する頂点に向う稜があるならば、前者は後者よりも前(上位)に位置する要素群であると言える。このことは、到達可能行列において、各行の成分の内、1の値を持つものの数を数えればただちに判明する(図13)。DCOTHプログラムでは、この性質を利用して、自動的にリーフ間の階級関係を判別できる。

リーフをローブに分割するには、リーフを構成している頂点の数に等しい行と列を持つ正方行列を作る。そして、この行列について、初めの有向グラフにおいて、任意の頂点Cから頂点Dに向う稜があれば、成分CDに稜の名前に相当する任意の記号を割り当てる。求めるサーキットとは、ある頂点が始点であり、かつ終点であるような径路である。このような径路を作る稜は、行列の対

* < LEAF >										
NO.	T/FACTR.	FACTORS								
1	2	6	7							
2	7	1	2	3	4	5	9	10		

図12a

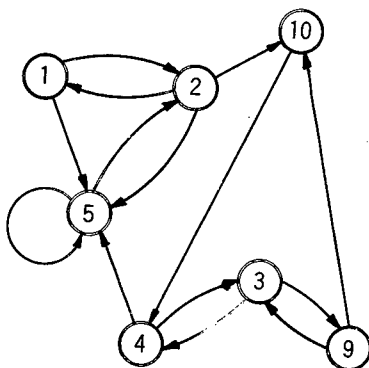


図12b

*< ADVANTAGE >

FACTOR	ADVANTAGE SCORE
1	7 *****
2	7 *****
3	7 *****
4	7 *****
5	7 *****
6	9 *****
7	9 *****
8	8 *****
9	7 *****
10	7 *****

図13. 要素の優位性の判別

*< LOBE >

LEAF	STEP	ODR.	T/FACTR.	FACTORS
1	2	1	2	6 7
2	1	1	1	5
2	2	1	2	1 2
2	2	2	3	1 2 5
2	2	3	3	3 4 9
2	2	4	2	3 4
2	2	5	2	2 5
2	2	6	2	3 9
2	3	1	3	1 5 2

図14. ローブとその構成要素

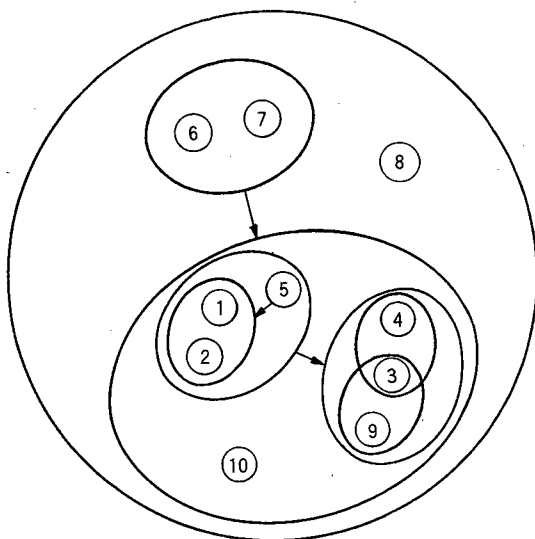


図15. 問題状況の構造

プロジェクトの名前	プロジェクトの説明	構成要素のリスト	稜線の数	関係行列で値が1の成分の位置リスト
PROJECT 0	GENERAL PROJECT	1, 2, 3, 4,, m	23	2, 5, 13,, n
PROJECT 1	AAA.....	1, 3, 5, 10,, l	21	2, 5, 11,, n
PROJECT 2	****.....	2, 6, 7,, j	25	1, 3, 8,, l
PROJECT 3	BBB.....	5, 6, 9,, l	10	4, 5, 7,, p
PROJECT n	###.....	1, 4, 7,, p	50	1, 3, 4, 6,, s

プロジェクトのリスト

要素の名前	要素の説明
FACTOR 1	PPP.....
FACTOR 2	*****.....
FACTOR 3	###.....
FACTOR m	%%%.

要素のリスト

要素の名前	連想要素
FACTOR 1	2, 5
FACTOR 2	1, 5
FACTOR 3	4, 9
FACTOR 4	3, 9
FACTOR m	

要素と連想要素

図16. DCOTH プログラムのデータ構造

```

*< ASSOCIATED FACTORS ***** LOBE >

FACTOR      T/FACTORS      ASSOCIATED FACTORS
    1          2          2  5
    2          2          1  5
    3          2          4  9
    4          2          3  9
    5          2          1  2
    6          1          7
    7          1          6
    8          0
    9          2          3  4
   10          0
  
```

図17. 要素とその連想要素

角成分となる。最初に作った行列について、0でない対角成分があれば、その成分に相当する稜の始点と終点を成している頂点は、道の長さが1つのロープを作っている。次に、その対角成分を0に置きかえた行列（K）を作り、それを2乗して得られる行列の0でない対角成分からは、道の長さが2のロープが得られる。再びそれを0にかえた行列に行列（K）を掛けた結果の行列の0でない対角成分からは、道の長さが3のロープが得られる。このような操作は、対角成分以外の成分がすべての0になった時に終る。以上の手順をすべてのリーフについて行えば、あらゆるロープを抽出できる。ロープ間の階級関係は、そのロープが属するリーフの階級および道の長さによって判定する（図14）。

有向グラフで表現された問題状況を、リーフやロープを構成する要素群に分割し、それら群同士の間の階級関係に基づいて、その構造を明らかにすることができる（図15）。

DCOTH プログラムのデータ構造は図16の通りである。1つの問題状況の分析のために入力された、一纏まりの要素とそれらの間の関係とを1つのプロジェクトとして、プロジェクトごとにファイルに保存する。また、すべてのプロジェクトに登録された要素とそれらの間の関係を整理して、一纏めにしたもの（を一般プロジェクトとして保存する。一般プロジェクトとは、過去に経験した1つ1つのプロジェクトに現われた要素の内、重複するものは1つにし、要素間の関係もそれに応じて整理した形で、2つ以上の問題状況にわたる、より大きなプロジェクトとして自動的に登録、保存され、新しいプロジェクトが登録された時には、それに応じて更新される。これは、過去の経験の蓄積されたものである。また、既成概念にとらわれない問題状況の分節、関係づけ、更には、問題提起ができるかも知れないという期待がかけられている。

ロープに分割された個々の要素については、それぞれに対して、それが属しているロープの他の構成要素を連想要素（近接関係、同義関係にある要素）として保存するようになっている（図17）。これは、その要素が後日何らかのプ

プロジェクトの構成要素になった時、その連想要素も、関係深いものとして自動的にそのプロジェクトに登録されるようにするためである。これは、過去のデータや経験を可能な限り活用しようとするのと、このプログラムによって働くシステムに、ある程度の学習能力を持せるための機能である。

自動的に登録された要素とプロジェクトの他の要素との関係は、上記の一般プロジェクトを参照して決定される。既存のプロジェクトに、全く新しい要素を追加しようとする時には、要素の名前とそれについての簡単な説明文を登録した後、そのプロジェクトに属するすべての要素間の関係を示す行列をターミナルに表示し、成分の値を必要に応じて変更する。

DCOTH プログラムでは、要素の追加、削除、関係の変更がすべて対話的に行える。従って、このプログラムは、問題状況の構造化のために使えるばかりでなく、構造化された状況について、パターン変換を行い、その結果を改めて構造化して検討することもできる。

おわりに

ものそのものを成り立たせるための科学技術と、ものを具体的に作るための生産技術の発達が目覚ましい。しかし、人間のために達成すべき効果が何であるか、そのためにどのような媒体の実現が期待されるかを究明する技術が充分発達しているとは言い難い。いわゆるハードな技術に対するソフトな技術の立ち遅れ、あるいは、それらの乖離についての指摘が多い。客観的な分析と総合によるハード・ウェアの実現へ向けて応用されて来たシステム分析、システム工学が、物理的的人工物を作る上で多くの貢献をしたことは疑う余地がない。しかし、あらかじめ設定された要求、目標を効率的に達成しようとするハードなシステム思考を、人間にかかわる問題にまで適用しようとする姿勢が、科学技術に対する反駁、反技術の主張を生むことになったのも事実である。

何を問題とすべきであるかが、未だ不明確な問題状況に対しては、問題とそ

の解決目標とが所与となっていなければ適用できない方法論は無力である。また、受動的で漠としている問題状況を直観的に把握することも甚々難しい。渾沌、茫漠とした世界を認識するために、われわれが知っている方法は、そうした世界を記号（言葉）によって分節し、分類、整理して秩序立てることである。ここで提案した問題状況の分節と構造化の方法の基盤はそこにある。また、それは認識のための方法であるばかりでなく、表現のための方法でもある。その意味で、DOCTH プログラムは、現状の構造化のために有効であるばかりでなく、構造変換、新しい構造の創成による概念モデルの構築のためにも使える。

問題状況の構造化は、問題状況の表現であって、それが、一元的に問題を定義しているのではない。表現された状況をどのように把えて、何を問題とするかは、われわれに任されている。明らかにされた構造の内に、歪や偏り、不均衡を発見したり、測定したりするために、公式化された方法を適用すべきかどうかについては、慎重にならざるを得ない。もちろん、過去の経験や集積された知識を参照して、対象とする構造を検討することは重要である。しかし、「いかに」よりも「何を」を重視しようとするならば、問題提起こそ創造的であればならない。非凡な感性と創造性が期待されるところに、既存の枠組を適用しようすることは慎まなければならない。構造変換、あるいは、新しい構造の創成によって得られた概念モデルこそ、望ましい状況の表現であるから、そこには創造的な飛躍がなければならない。構造化された問題状況を前にして、デザイナーとしての立場と主張を明らかにし、デザイン目標となるべき原型を発想することこそ、創造的な概念モデルの構築であり、問題提起である。

DCOTH プログラムは、筆者が、文部省派遣在外研究員として、IIT（シカゴ）のデザイン科に滞在している間に開発したものである。その契機になったのは、Charles L. Owen 教授が開発した RELATN プログラムと VTCON プログラムである。^⑩

滞在中、Owen 教授、佐藤啓一助教授、そのほか多くの方々から受けたご厚意と貴重な助言に対して、心からの謝意を表したい。

参考文献

- ① Christopher Alexander "NOTES ON THE SYNTHESIS OF FORM" Cambridge, Massachusetts and London 1964, Harvard Univ.
- ② Gray T. Moore, ed. "EMERGING METHODS IN ENVIRONMENTAL DESIGN AND PLANNING 1970, MIT
- ③ 吉川弘之「一般設計学」『PIXEL 別冊 No.3』1983, 図形処理情報センター
- ④ Jay Doblin による Design Theory の受講ノートより。1984, ID, IIT
- ⑤ W.J. Mitchell "COMPUTER-AIDED ARCHITECTURAL DESIGN" NEW YORK, CINCINNATI, TORONTO, LONDON, MELBOURNE 1977, Van Nostrand Reinhold
- ⑥ W. Peña "Problem Seeking" Houston 1977, CBI
- ⑦ W.J. Mitchell, ibid.
- ⑧ Peter Checkland 著, 高原康彦, 中野文平監訳「新しいシステム・アプローチ — システム思考とシステム実践 —」1985, オーム社
- ⑨ 高辻正基「記号とはなにか」1985, 講談社
- ⑩ 川喜田二郎「発想法」1967, 中央公論社
- ⑪ 柳井晴夫, 岩坪秀一「複雑さに挑む科学」1976, 講談社
- ⑫ 西之園晴夫 (京都教育大学, 教育実践研究指導センター) 他によって開発されたものである。
- ⑬ 渡辺 茂編者「コンピュータ時代の数学」1968, ダイヤモンド社
グラフの分割方法, 行列の計算方法については, 分担執筆者三浦宏文「知識をひきだす法 — IR のための数学 —」を参考にして, FORTRAN でプログラミングした。
- ⑭ Charles L. Owen "DCMPOS: An Algorithm for the Decomposition of Nondirected Graphs .. Gray T. Moore, ibid.