

意思決定過程のコンピュータ化 に関する一考察

安 田 晶 彦

1. はじめに

現代経営管理を解明する上では、経営意思決定過程 (management decision-making process) の概念の検討が、非常に重要になって来ている。そして、特に経営管理のコンピュータ化について論じる際には、意思決定過程の理解は、決定的な鍵を握っている。勿論、経営管理に関する総ての要因を、単純に意思決定論に還元する訳にはいかず、経済的社会的背景のもとで、より高次の、組織的活動に固有の問題を捉えて、初めて体系的な理解が可能になるのだということは言うまでもない。ただ、差し当って管理は精神的活動の一種であり、経営管理論において、この精神的活動を考察する上での基本的なアプローチが意思決定論であるところから、経営意思決定の問題を、先ず取り上げてみる必要は十分にある。

さて、本稿では経営意思決定論研究の一環として、意思決定過程のコンピュータ化をテーマとして取り上げることとする。コンピュータ化について、今回特に明らかにしておきたいのは、ソフトウェアとプログラムの区別と関連をどう理解するかという点で、これまで十分に論じて来なかった、コンピュータ技術の、この重要な側面について、基本的な位置づけを試みたいと思っている。特にソフトウェアという用語を、社会科学的な分析のため、どの

ように捉えるべきかについては、今後の経営意思決定論研究にとっても、重要な基礎視角となるだけに、十分に検討したい。

本稿の構成としては、先ずコンピュータ・ソフトウェアについて検討し、従来ソフトウェア≒プログラムとして曖昧に捉えられて来たものを、もう少し厳密に検討するつもりである。

次に、ソフトウェアとの関連で、コンピュータ・プログラムをどう位置づけるかについて考察していく。また、プログラミング労働が、基本的にかくなる性質を持つのかについても、明らかにしていくつもりである。

そして、以上の分析を踏まえて、意思決定過程のコンピュータ化について、基本的なフレームワークを提示してみたいと考えている。

2. コンピュータ・ソフトウェア

コンピュータというものは、本質的には人間のコミュニケーション過程 (communication process) を、部分的に労働過程におけるコントロール (control) に持ち込んだ労働手段であり、そのためには、本来の動力用エネルギーとは別の、制御¹⁾用微弱エネルギーの処理 (processing) 構造と、制御用エネルギーの変動パターンを、動力用エネルギーのそれに変換する、増幅作用が不可欠となっている。

この点から、人間のコミュニケーション過程を人工物に投影した、情報²⁾過程 (information process) を確立するため、先ずあたかも「労働手段に話させる」ごとくに、微弱エネルギーの変動パターンを読み取る技術を、開発する必要がある。この技術を、本稿ではソフトウェア技術 (software technology) と呼ぶことにする。

ソフトウェアという用語は、元来コンピュータ技術の発達に伴って現われ

たものであり、あくまでもコンピュータに即して考えることが妥当であると言えよう。しかし今日では、日常語として比喩的に、ソフトないしソフト化という用語が頻繁に使われており、改めてコンピュータ・ソフトウェアとは何かを問い直すことも、必要な状況になって来ている。

コンピュータ化という状況を根底に置きつつ、ソフト化という観点から社会現象全般を論じたものとしては、代表的には林雄二郎氏の著書『情報化社会』³⁾をあげることが出来る。そこに見られるような思潮を基礎に置いて、今日ソフト化と言うときには、物としての実現自体ではなくして、構想 (idea) や設計 (design) の直接的具体化を指すようになって来ている。このこと自体は、先に述べたようなソフトウェア技術の目的にも当てはまる面があり、ソフト化を広義に捉える立場としては、首尾一貫しているように見える。つまり、セーターそれ自体の、着ると暖かいという直接の使用価値ではなく、そのブランドが問題なのだと言うときには、生産力の一定の発達のもとで、社会的なコミュニケーション過程が、その影響力を急速に拡大しつつあるのだという感触を、得ることが出来るのである。

このようなソフト化の理解について、総てがコンピュータ化の直接の産物であると言い切る訳には当然いかず、コンピュータはソフト化の工学・技術面での現われだという考えも、妥当なように見える。だとすれば、コンピュータ・ソフトウェアについて論じる際には、工学・技術上の問題に、限定する旨の但し書が必要となろう。

ところがコンピュータ技術を支えるコンピュータ科学 (computer science) の方も、人間のコミュニケーション過程一般の理解に迫るべく、それなりに研究領域の拡大を目指しているので、注意が必要である。AI (Artificial Intelligence; 人工知能) 研究として、エキスパート・システムを始め、自然言語理解など、様々な面で先端的な研究とその実用化が積み重ねられている⁴⁾。ソフト化一般を問題にする際にも、構想や設計などの精神的諸活動が問題になる以上、理論的に深化を図る上では、コンピュータ科学からの様々な影響を

受けざるを得ないのである。

社会科学の分野に、無批判的にコンピュータ科学の諸理論を取り込むことや、人間の精神的活動を、コンピュータのアナロジーとして見ることの是非はともかく、ソフト化を広義に論じる際にも、コンピュータの位置づけは不可避のものとなって来ており、コンピュータに即したソフトウェアの理解は工学・技術面に限るといふ、但し書の効力以上のものがあると考えて良いのではないだろうか。

さて、人間が外界の諸事象を言語や記号を用いて表現するということが、即ちデータ表現は、人間の精神的活動にとっては、ごく普通の事柄である。ソフトウェアは差し当りデータ表現と深い関わりを持つのであるが、データ表現そのものがソフトウェア技術であるということでは決してない。そこで先ずデータ表現が、ソフトウェア技術とどのような関わりを持つかについて詳しく検討してみたい。

ソフトウェアの用語としての起源は、ライブラリー・プログラム (library program) から来ているが⁵⁾、ライブラリー・プログラムそのものが問題なのではなく、むしろ従来の労働手段以外の、即ちハードウェア以外の手段で、プログラムを保存し、実行するということが、やはり重要な問題である。勿論、ただ頭の中でプログラムを考え付いただけでは、何も実現しえないのであるから、何らかの労働手段によってプログラムが担われることなくしては、プログラムの実行はありえない。結局プログラムは何らかの物理的な媒体 (media) によって担われることになる。この媒体は、ペンで字を書いた紙のごとき、独立した手段ではなく、労働過程のコントロールを担うように、労働手段に組み込まれたものでなければならない。

このようにして、ソフトウェア技術は何らかの媒体上にデータ表現を行なう技術であると考えることが出来る。ソフトウェアに即して考えれば、データ表現は必ずしもプログラムの体裁をとっている必要はない。また、ソフト

ウェアは、差し当り紙や磁気テープのような柔らかい物だというのが当初の直感的な考えとしてあり⁶⁾、それは今日でも十分に当てはまるのであるが、問題は紙や磁気テープのような媒体を、不可欠の要素として組み込んだ、労働手段の構造の如何ということになるであろう。

技術論的に見て機械一般は、このようなデータ表現をまとまて行なうための手段ではなく、データ表現はただ機械の設計や機械操作のために、人間の頭の中や紙の上で必要とされるに過ぎない。しかし、一部の機械は、このようなデータ表現の手段であり、計測・通信機器がその典型である。それに対して、まとまったデータ表現が構造的に組み込まれた機械も、コンピュータ技術の出現前のソフトウェア技術の原型を示すものとして存在した。ジャカル織機がその典型である。後者の機械のように、媒体としての部分を持つものについては、例えば織物の模様のようなパターンを記号化し、それをまとめて媒体上の物理状態として保存することが、既に可能であった。

今日、コンピュータ以外で、こうしたソフトウェアのための媒体を持つものとしては、いわゆる AV（音響・映像）機器を、代表例としてあげることが出来る。従って、ソフトウェアをコンピュータに限る必要は必ずしもないとも言えるが、AV 機器は基本的な部分を電子技術に依存しており、そこでのソフトウェア技術は、コンピュータ技術と密接な関連を持って発展して来たものであるので、やはりソフトウェア技術をコンピュータ関連技術として見ることは、決して間違いではない。

そこで次には、コンピュータに即したソフトウェアの問題について考察していくことにしよう。コンピュータ・ソフトウェアの主体がプログラムであるところから、今日では、ソフトウェア≡プログラムであるという理解の仕方が普及している。前述のように、コンピュータ・ソフトウェア自体がプログラムの保存手段として生成したものであり、こうした理解も、概ね的を射たものと言えよう。

そこから、ソフトウェアは労働手段に対応した技術ではなく、プログラム設計者の構想そのものであるという考え方が出て来るのも、無理からぬところである。しかし、厳密に考えていくと、プログラムに相当する何らかの動作手順を実行する機能は、自動機械一般に存在し、プログラムの実現が必ずしもソフトウェア技術によらねばならないという、絶対的な条件はないのである。

例えば、マイクロプログラム (microprogram) という設計思想に見られるように、論理回路が直接担当する筈の種々の機能を、ソフトウェアの側に持たせ、回路構造を簡素化する方法が存在する⁷⁾。また、ファジィ (fuzzy) 理論⁸⁾の実用化に際して、既存のハードウェア上でソフトウェアを介して、ファジィ・プログラムを実現するか、専用のファジィ論理回路をハードウェアとして実現するかという選択も可能である⁹⁾。

このように見てみると、あるプログラムを実現するのに、ハードウェアなしにはそれが不可能というだけでなく、ハードウェアとソフトウェアの相対的な選択性もまた十分に存在するのだということが分かる。ハードウェアとソフトウェアの選択は、専用回路を用いて実行速度を上げるか、その点が致命的でない限りは、ハードウェアを意識せずに互換性のあるプログラムを作成する方を優先するかという問題にかかって来る。

従って、本稿で問題とするコンピュータ・ソフトウェア技術は、主記憶装置を中心にして、様々な補助記憶媒体間で、同一のプログラム、ないしはデータ群を変換する技術だと言うことが出来る。主記憶装置の存在は、何と言ってもコンピュータ・ソフトウェア技術が必要とされた、直接の要因に他ならず、様々な記憶媒体の開発が、ソフトウェア技術の発達ハードウェア的基盤となっているのである。

また、記憶媒体間のソフトウェアの互換性という点から、当然派生して来るのが、ソフトウェアの複写可能性である¹⁰⁾。これは、差し当り、ソフトウェアが商品として有償か否か、あるいは別の機種にプログラムを移植する

際の煩雑な手順といった点は別として、様々なソフトウェアを、多数のハードウェア上で有効に実現するための基盤となっている。この点は特に、パーソナル・コンピュータのソフトウェアについて言えることである。

以上のように、ソフトウェア技術は、特定の素材を媒体として利用し、まとまったデータ表現をそれに物理状態として設定し、労働過程で用いるための技術であることが明らかになった。このように考えれば、例えばビデオ・テープにもソフトウェア技術は必要であるので、ビデオ・ソフトといった日常語が、この定義にそぐわないということはない。議論が出て来るとすれば、ソフトウェア技術によって媒体上に実現されるデータ表現自体は、どういう位置づけになるのか、特にプログラムのような構想そのものは、いったい如何なる意味を持つのかについてであろう。次節においては、その点について検討したいと思う。

3. コンピュータ・プログラム

前節において検討したように、コンピュータ・プログラムは、差し当りソフトウェア上で実現されるものの、プログラムが表わしている、労働手段の一連の動作手順は、ハードウェアによっても、ハードウェア+ソフトウェアによっても実現可能であり、選択性を持っている。さて、この労働手段の一連の動作手順については、既に機械一般にその原型を見ることが出来る。即ち機械が、人間による道具を介した労働過程の制御を、たとえ一部であれ機構 (mechanism) の運動に置き換えたものであるのだから、プログラムとして構想されたものは、機構として実現されている訳である。

ところが機械の場合は、1つのプログラム=1つの機構であり、構想の具体化としてのプログラム自体が独立に問題とされることはありえず、むしろ機構そのものが課題とされれば十分なのであった。前節で例示した、まとも

ったデータ表現機構からなる機械についても、やはり、1つのプログラム＝1つの機構という原則は成り立つ。これはジャコール織機のさん孔紙などが、プログラム＝機構のもとでのデータ群のみを表わすに過ぎないからであり、結局このさん孔紙はプログラム＝機構のもとで、全くシーケンシャルに読み込まれるに過ぎないものである。

こうして、たとえ同じようにソフトウェア上のものであっても、プログラムとそれ以外のデータ群とは区別されるべきだということになるのであるが、従来の議論ではジャコール織機もまたプログラムを持つと考えられて来た面もあり、プログラムをどう理解するかが、ここでの重要な鍵となる。先ずプログラムというのは、一組の命令(instruction)セットを持ち¹¹⁾、命令の組合せによって、労働手段の動作手順を決定するものである。従って命令の数だけ動作の種類があり、命令があるまでは、何の動作も実現しないことが原則である。次に、命令を組み合わせたプログラムは、広い意味でのアルゴリズム(algorithm)あるいは問題解決(problem solving)手順¹²⁾を持つものであるという点が重要である。この点はソフトウェアを持つとは言え、ただシーケンシャルにデータだけを扱う労働手段とは根本的に異なる¹³⁾。石沢篤朗氏がワイゼンバウムを引用されながら、条件つき分岐(conditional branch)命令の意義について強調されるのも¹⁴⁾この点と共通するところがある。

1つのプログラム＝1つの機構という原則を突破するには、要するに機構が直接にプログラムを表わさないようにする必要があり、機構がプログラムに従って動作するための、機構からは自立したプログラムと、その保存機能が必要とされる。これが結局、プログラム配線式を経て、プログラム内蔵式により、コンピュータとして一応の完成を見たのである¹⁵⁾。

さて、コンピュータ・プログラムによって何らかの働きを担わせる場合、ハードウェア的に実現してしまう部分については、特にプログラムとは言わず、また通常のプログラミングにおいても意識されない部分でもある。従っ

て狭義のプログラムは、ソフトウェア上のそれと考えると、通常は差し支えないことになる。

この狭義のプログラムについて、もう少し詳しく検討してみると、先ずソフトウェア技術によって、何らかの媒体上に実現される前のプログラムは、やはり一種の設計書に他ならないと言うことが出来る¹⁶⁾。この設計書が他の一般の設計図と異なるのは、前者がコンピュータに入力され、自動的に実現されることによって、他の労働手段ではその設計図自体に直接現われない、一連の動作手順というものを、直接に表現することが可能になるという点である。

前者の点、即ちプログラムとして書かれたものが、直接に実行可能であるということ自体は、むしろソフトウェア技術にその基盤がある訳で、決してコンピュータ・プログラムそのものの特徴に依存している訳ではない。ソフトウェアを持つ機械について、たとえそのソフトウェアが、全くシーケンシャルに変換されるデータ群に過ぎないとしても、直接に実行されるということ、つまり加工データをもとに制作されたさん孔紙が、自動的に読み込まれる構造に、機械がなっているという点に変わりはない。

他方、後者の一連の動作手順を従来の設計図法に託することなく、直接に表現するという点では、ソフトウェア上の狭義のプログラムは際だった特徴を持っている。机上で立案されたプログラムは、従来の労働手段では、設計図へと読み換えられ、素材的制約を絶えず考慮しつつ、労働手段そのものの製造という形でのみ実現しえた訳である。たとえ機械による自動的な生産によったとしても、それがハードウェアであるという素材的制約を、ブラック・ボックス視することは不可能であると言って良い。

ここで、プログラムは一組の命令セットとアルゴリズムという端緒的な特徴を越えて、独自の発達を遂げることとなる。即ち、プログラムというものが、ハードウェアを意識せずにいろいろな作業目的を達成するものであると

いう点、そして、プログラムが人間のコミュニケーション過程を、労働過程に人工的に持ち込むことになったという点がこれである。前者については、差し当りソフトウェア技術の発達がこれを可能にしたと言え、さらにハードウェアの効率的な利用を目的とした、種々のコントロール・プログラムが、OS (Operating System) として体系化されることによって本格化した。そして後者については、特に言語プロセッサ (language processor) の発達によって、人工言語表現によるプログラムの作成が可能になったことが大きく貢献している。

先ず、前者の OS について考察していくと、次のようなことが言える。即ち、OS は種々のハードウェアを自動制御するための体系的なプログラムであり、ハードウェアを効率的に運用すると共に、プログラム作成時にハードウェアを意識しなければならない煩わしさを軽減することが、OS の課題となっていたのである。マン・マシン・インタフェースについては、OS は多くの場合一連のコマンドで動作し、ハードウェアの特徴、例えば CPU (Central Processing Unit) の性能や、入出力の方式といった点を意識する必要がないように考慮されている。このようにコンピュータの操作が、そのハードウェアを意識せずに可能になること、そしてハードウェアの自動制御ということが、OS では果たされているのである¹⁷⁾。

次に言語プロセッサについて見ると、最終的にソフトウェアとして信号化される前の状態として、アセンブリ語という記号表現、また高水準言語のような比較的的自然言語に近い表現で、プログラムを設計することが出来、それを自動的にソフトウェア化出来るようなプログラムが存在するという事は非常に重要である¹⁸⁾。この言語プロセッサの発達によって、プログラム技法はハードウェアから相対的に自立して、問題解決技法そのものへと近づいていくことになったのである¹⁹⁾。

このように特に OS と言語プロセッサという 2 大プログラム技法の発達は、従来の労働手段が普通に有している様々な素材的な特徴を、労働過程に

おける制約条件とはせずに、労働過程の制御を効率的に行なうことを可能にしたのである。そしてまた、それらのプログラム技法は、コンピュータ技術におけるソフトウェア技術の重要性を高める要因となったとも言つて良い。前に述べたように、コンピュータ技術において、ソフトウェアそのものは絶対的存在とは言えないまでも、コンピュータの効率的運用を図る上で、ソフトウェアに依存する部分が非常に大きくなっていることは事実なのである。

元来、プログラミングという行為は、計画策定（planning）過程という意思決定過程の内の1つから派生したものと考えられる²⁰⁾。広い意味での計画策定からは、先ず協働（cooperative）システムの発展に伴って設計過程が分化し、近代技術と結び付いて、特別な位置を占めるようになったと考えることが出来る。労働過程にとっては、計画策定にせよ設計にせよ、予め用意された精神的労働であるという観が強いものの、協働的行為のもとでそうした活動が不可欠のものとなつていったことは言うまでもない。

プログラムは最も広義には計画（plan）そのものであるとも言えなくもないが、やはりそれが労働手段そのものの機能として実現するという点にメルクマールがある。コンピュータ以外の従来の労働手段では、労働過程の制御を、労働手段そのものの構造に置き換える必要があった。従つて、机上のプログラムは先ず設計図に変換され、労働手段の構造として具体化するという手順を踏んでいた。

このような点を踏まえた上で、さらに考察しておかねばならないのは、狭義のプログラミング労働と狭義のソフトウェア作成労働についてであろう。プログラム配線式のコンピュータに見られるごとく、プログラムが必ずしもソフトウェア上で実現される訳ではない点に注目すると、机上のプログラミングと、それを實現する配線作業とは、一応別種の労働として捉えざるを得なくなる。

配線作業は、要するに機械操作労働ということになるが²¹⁾、ソフトウェ

アを普通に持つ今日のコンピュータに関して、プログラムをソフトウェア化する作業もまた存在する訳で、こうした労働を設計から派生したプログラミング労働と区別する上で、狭義のソフトウェア作成労働と考える必要が出て来る。この定義は、ソフトウェア労働という用語が、一般にプログラミング労働を指して用いられていることとは区別して、特に本稿での考え方を強調するためになされたものである。

狭義のソフトウェア作成労働は、かつてはコンピュータ端末としてのPCSのキーパンチであり、今日ではVDT労働に相当するが、これらはいずれも機械操作労働の一種に他ならない。キーパンチャーは通常、データのみの大量入力という作業、例えば会計伝票入力といった作業を担うが、本稿でのソフトウェアの位置づけが、プログラム入力もデータのみの入力も含みうるものであるため、結局、プログラム入力に関わるソフトウェア作成労働も、大量データ処理に関わるそれも、性質は変わらないものとして考えることが出来る。

勿論、プログラミングあるいはデバッグング (debugging) といった労働と、ソフトウェア作成労働とを同一人物が兼ねることは十分にありうる訳で、これは分業化が進む前の手工業の職人が、何でも兼ねていたのと同じことである。今日の分業化は、SE (System Engineer)、プログラマー、オペレータといった職種を生み出したが、技術論的に見て、これらの労働を本質的に分類しうるものは何かということを問題としたのである。

また、今日のソフトウェア作成労働は、機械操作労働の中では、極めて自動化の進んだ状況で行なわれているため、もはやタイプライターを打つ事務労働と、何ら区別出来なくなっている点も指摘出来る。これらは、組織のコミュニケーションも、工場の自動化のためのソフトウェア作成も、外見としては区別出来ず、結局コンピュータによる情報過程に、組み込まれてしまっていることを表わしている。ただし、情報過程の拡大が、総てのソフトウェア作成労働を、事務労働と見なして良いとする契機となるか否かに

については、疑問の残るところではある²²⁾。

また、プログラムの性質によっては、起動・終了の単なるボタン押し労働ではなく、コマンド入力などの操作を非常に多く要するものもあり、このようなプログラムのもとでの操作は、大量データ入力ともまた区別される傾向がある。また本格的なプログラミングではなく、プログラム開発支援ツールを用いた簡易のプログラミングも普及している。こうした、プログラミングとソフトウェア作成との中間的な労働の位置づけが、問題とならざるを得ない。こうなると、どこまでがプログラミングなのか、区別がつきにくくなる訳だが、現場労働から一切の計画的、設計的要素が取り去られていて当然だという、労働に関する固定観念の方が、むしろ問題なので、ここはプログラミングに専念する者と、プログラミングに一切関わらない者の2者のみの存在を、想定することのないように注意したい。勿論、組織的な問題として、前者の権限を持たないような職務編成は十分にありうるが、これは職務編成そのものの問題であって、ここで論じられる技術論的契機とは、とりあえず別と考えるべきであろう。

さて、これまでに述べたように、プログラミングが、労働過程に関わる人間の精神的活動としての、意思決定過程と深い関連を有する以上、意思決定過程におけるコンピュータ化の問題をどのように理解するかについて、これまでの考察を踏まえつつ、論じない訳にはいかないであろう。次節では、この問題の解明の糸口を探りたいと思う。

4. 意思決定過程とコンピュータ

コンピュータ・プログラムが、差し当り精神的活動の所産であることは疑いの余地がなく、ソフトウェア化される直前のプログラムは、主として机上の人工言語表現に他ならない。これは要するに、設計過程の一種であるが、

設計そのものは計画策定の技術的な分化形態であり、設計図は主として協働システムにおける、製品の製造をめぐる設計者と作業者間の、コミュニケーションの必要性により発達したものと考えることが出来る。計画策定自体についても、その重要性が高まったのは、協働システムの発達によっていることは言うまでもなく、例えば、個人的な工芸品生産においては、計画策定と設計とは、職人の頭の中に一体となっているに過ぎないのである。

既に明らかにしたように、ソフトウェア技術によって、プログラミング自体は、従来の設計技法とは、かなり異なる内容を持っている。プログラムは、差し当り純然たる数字の列によって表わすことが出来、これはいわゆる機械語と呼ばれている。そして、言語プロセッサの利用によって、プログラミング言語で表わすことも可能である。これはハードウェアに特定の機能を担わせることなくして、実際の機能はソフトウェア上のプログラムとして表現することの発達した形態であると言って良い。

意思決定過程が、差し当り人間に固有のもののだとしても、特にコントロール過程については、これを労働手段によって代替し、自動化を図ることは、機械の普及以降一般に可能であるから、コンピュータがコントロール過程を担うことも、別に不思議でも何でもない訳だが、やはり人間の精神的活動に固有の特徴があるとすれば、コンピュータによっては容易に代替しえない部分が当然あろうし、コンピュータの限界も、所詮はコンピュータは社会における手段に過ぎないのだという点から、明らかになっていくと思われる。

即ち、精神的活動を労働手段が代替すると言っても、労働手段自体が精神的活動を行なう訳ではありえないのであって、実際は人間の精神的活動に相当する部分を、人工的な物質の運動に投影しているに過ぎないのである。またここで精神的活動と言う場合には、やはり精神的活動の社会的な性格規定を抜きにしては語れず、意思決定もまた、そのような範囲でのみ問題とされることは言うまでもない。

例えばコンピュータの場合でも、作成したプログラムを、コンピュータの補助記憶媒体にセットし、それが主記憶装置へと読み込まれ、最初の命令がレジスタに送られたが最後、後は人間がそのプログラムを理解していようがいまいが、客観的物理的運動として、正確に作動するだけの代物となってしまう。ここでは設計書としてのプログラムは、電気信号のパターンとしてソフトウェア化されているが、これは人間の観念がそうさせたという類のものでは決してなく、物理的にある目的に応じた結果になることが、客観的に保証されているというだけなのである。この点では、機械が人間の意志通りに動くといっても、それが思念の結果でも何でもなく、ただ機構が目的を果たすように組み立てられているからだというのと同じことである²³⁾。機械の場合には、製品の出来具合を人間が直接確認することが接点となるだけで、コミュニケーション云々は、通常、問題とされていないのであるが、コンピュータの場合には、コミュニケーションの真似事らしきものが、人間との間にあるという点が重要である。

この限りにおいて、労働におけるコントロール過程が、労働手段によって代替され、またソフトウェア化されたプログラムの急速な発達によって、労働手段自体の生産過程のコントロール機能が、非常に高度化されるということが出来るのである。そこで重要な点は、同じく精神的活動の物質的運動への投影といっても、既に述べたように、機械の場合には機構そのものの運動として実現されるに過ぎないのに対して、コンピュータの場合には、人工言語表現のプログラムを、直接に投影することが出来るという点であり、つまりコンピュータが、人間の対自然関係と人間同士の社会関係との、中間項的なところを担っていることを示している。元々、プログラミング言語自体が、機械語と自然言語との間の過渡的な存在であり、それ故に、実に様々なプログラミング言語が消長して来た訳である。従って、意思決定過程の労働手段による代替といっても、コンピュータによるそれは、機械とは質的に異

なるものであると言うことが出来る。

このような、コンピュータによる意思決定過程の代替は、労働過程の直接的なコントロールの範囲に留まらず、精神的活動に直接及んで来る。元々、計算機や計測機器の発達で、人間のコミュニケーション過程の延長線上に位置づけられ、コンピュータに至っては、とりわけ人工言語表現により、コミュニケーション過程を本格的に技術上の課題としうる状況になっていることから、人間の精神的活動の一部を担う技術上の素地は十分に形成されていると言って良い。しかしながら、それがどのような形で進んでいくのかについては、これまで余り納得のいく説明がなされていなかったように思われる。

生産過程の分析では、コンピュータは、より高性能の自動制御装置として把握されることが多い。生産過程の自動化が進行すれば、当然計画策定と設計の役割が重要になる。例えば、石沢篤郎氏が特にCAD (Computer Aided Design) システムに注目されているように²⁴⁾、生産過程の直接的コントロールだけでなく、計画策定や設計の手段として利用すること自体は、既に周知の事実になりつつある。DSS (Decision Support Systems)、即ち意思決定支援システム²⁵⁾や、CADシステムは、今日では別に珍しい存在ではない。従ってここで問題としたいのは、計画策定や設計の自動化の事実そのものよりは、むしろその内容だということになる。

先ず、DSSにせよCADにせよ、それらは何れも支援 (supportあるいはaid) システムとして特徴づけられている。この名称からすれば、計画策定や設計は、コンピュータのみによって自動的に行なわれうるものではないと言えよう。確かに煩わしい計算や、厳格な製図法といった点についてはコンピュータが解決するが、計画策定や設計にとって、最も重要な構想については、コンピュータの演算結果として、自動的に出て来るものではない。

このように、計画策定や設計におけるコンピュータの利用は、あくまでも補助手段としてのそれに留まっていると言えよう。その場合、コンピュータ

の利用に先立って、企業における計画策定や設計の役割が重要になり、そうした過程自体が大規模に進められるようになったため、分業化、専門化が進んだことが、DSSやCADの発達の基盤となっているのは言うまでもない。ただ、コンピュータ化が進んでも、全く新たな構想の善し悪しの評価をする機能は、コンピュータの側にはないと言って良い。従ってDSSやCADは、やはりその名称通り、支援システムに他ならず、計画策定や設計の過程は、コントロール過程のように自動化しえないと言うことが出来る。

しかし、計画と称していても、それが単に事前的な意思決定であるというだけで、実際には新たな構想が何ら含まれないケースも多い。こうした事前的な意思決定の多くは、コンピュータの演算結果が代替しても、差し支えない側面を持っている。

このことに関連して、例えばバーナードは、意思決定を機会主義的側面と道徳主義的側面とに分けて考え²⁶⁾、また、サイモンは事実的要素と価値的要素とに分類したが²⁷⁾、やはり意思決定のコンピュータ化について考える場合にも、この視点は重要であると思われる。

問題を経営意思決定の領域に限定して考察していくと、先ず近代株式会社制度の普及以前の経営意思決定は、通常、企業主個人によって担われ、その過程に当事者以外が関わることは殆どなかった。もし現代においても、計画策定が経営者の直感とでも言えるような、全く漠然とした枠組みのもとで行なわれる場合には、経営意思決定のコンピュータ化はおろか、経営意思決定自体にさえ分析のメスを入れられないことになる。しかし、このような状況は、個人企業²⁸⁾では往々にしてありうるものの、実際の経営意思決定は経済的社会的な制約条件のもとで行なわれる以上、何らかの合目的的意思決定の枠組みは存在しうる訳である。大企業体制の確立が管理組織の発達をもたらし、経営意思決定の枠組みの客観化が要請されるようになってからは、計画策定技法の確立を中心に、計画における業績評価基準の明確化、業績測定

の制度的確立といった、意思決定過程をめぐる諸条件に関心が持たれ、それらが意思決定過程のコンピュータ化を促す基盤となって来た。ここで言う枠組みの客観化とは、結局、経営目的であるところの企業成長を達成するための、合目的な手段体系の整備ということになる。また、組織論的に見れば、これは価値判断を排除した、事実判断の体系化と言えるであろう。

経営計画策定において、果たしてどの程度コンピュータが入り込む余地があるかについては、既に述べたように経営意思決定の客観化、制度化といった条件、そして特に計画策定技法の確立が決定的な影響を持つが、これらは計画策定技法をルーチン化し、代替案の析出と選択の枠組みに、一定の問題解決手順を見いだす素地を与える。

サイモンの言う満足基準についても、結局はヒューリスティックス (heuristics)²⁹⁾として、問題解決の手順に収まる問題であり、経営計画策定は一見、コントロールと同じようにコンピュータ化への道を歩んでいるかのようである。ただ、現時点では、BES (Business Expert Systems)³⁰⁾があれば、意思決定は自動化出来るという性質のもでもなく、やはりコントロール過程とは異なり、計画策定過程のコンピュータ化は、単なる支援以上のものを期待するには限界があると考えた方が良い。

5. 結 び

以上のように、本稿では先ず、コンピュータ・ソフトウェアとプログラムの区別と関連について論じ、その上で経営意思決定過程のコンピュータ化について考察を進めて来た。ソフトウェアとプログラムの問題については、かねてより、検討課題として考えて来たところであり、本稿においてその基本的な分析視角を提示することにより、以前に行なったハードウェア主体のコンピュータ技術の研究³¹⁾と合わせて、コンピュータ化の問題をより掘り下げて検討するための糸口を得たと思う。意思決定過程については、人間の精

神的活動全般を扱うのではなく、経済的社会的制約諸条件のもとでの経営意思決定に問題を限定することにより、コンピュータ化の問題を、より具体的に解明していきたいという立場を、今回は強調しておいた。

先ずコンピュータ・ソフトウェアについては、いわゆるソフト化という用語の意味するところと、コンピュータに即したソフトウェアの理解との関係を論じた上で、ソフトウェア技術をプログラムをも含む、まとまったデータ表現を労働手段上に可能にする、媒体の技術として捉える立場を明らかにした。これは、要するにさん孔紙や磁気テープ、あるいはLSIをデータ保存、つまり記憶の媒体としてどう用いるかという技術を指し、ソフト化という用語に比べれば明らかに狭義のものである。ソフトウェアの捉え方としては、狭すぎるという批判があるかも知れないが、プログラムだけを指す用語法よりは範囲が広く、様々なデータを処理するコンピュータ関連諸技術全般に通用する、ソフトウェアの捉え方であるので、以後の分析においても十分通じる定義であると、本稿では考えた。

そして、ソフトウェアとほぼ同義に捉えられていた、コンピュータ・プログラムを、労働手段の一連の動作手順に関する構想にもとづく、一種の設計書として定義し、プログラム技法を、記号・人工言語表現を用いた、その具体化のための技法として位置づけた。これによりプログラムは、設計書の一種であるということになったが、プログラムを機構の運動として実現した場合には、プログラムは単に設計者の頭の中の産物に留まり、直接にプログラムを問題とする契機にはならないという点、そして、やはりプログラムをソフトウェアとして実現することにより、より自然言語に近い人工言語表現によるプログラムが可能となるという点を重視した。

さらに、プログラムと計画策定、設計との関連を取り上げ、現時点で経営意思決定のコンピュータ化が、どのような意味を持つかについて検討した。今日では生産過程のコントロールはもとより組織的な活動全般のコントロー

ルを、コンピュータ化することが可能となっているが、計画策定、設計の方面においてもこれらの諸活動の組織化と専門化により、コンピュータ化が著しいという指摘をした。計画策定過程のコンピュータ化について見ると、根底において、新しい構想の評価³²⁾については、人間独自のものではないかという見解を示す一方で、計画策定技法を確立し計画策定過程の客観化、制度化を図ることにより、意思決定の価値的要素を排除した、計画策定過程の存在も考えられ、微妙な問題があることを指摘した。

こうした点を明らかにするには、より詳細に計画策定とコントロールの問題に踏み込み、経済的社会的構造と経営意思決定との具体的な関連を検討する必要がある。本稿においてはその重要性を指摘するに留まったので、今後の研究課題として重視していくつもりである。

〔注〕

- 1) 制御という用語は、本質的にはコントロールと同義語として扱って差し支えないと考えている。ただ、労働手段自体の持つ生産過程のコントロール機能が問題とされるときに、本稿では制御という用語を使い分けることがある。
- 2) 情報の持つ意味については拙稿「経営意思決定と情報」『岐阜経済大学論集』22巻1号、1988年6月、41—43頁を参照。

- 3) 林雄二郎著『情報化社会——ハードな社会からソフトな社会へ』講談社現代新書、1969年5月。

本書で林氏は、今日広義にソフトと呼ばれているものについて、情動的機能という用語を提唱されている。また、まえがきで、これを選択的機能とした方が良いとも述べられている。

- 4) AI研究の概要を紹介した文献は数多い。特にAIの実用化の展望を示したものとして、下記の文献をあげておく。

P. H. Winston, K. A. Prendergast ed., *The AI Business—The Commercial Uses of Artificial Intelligence*, MIT Press, 1984. (P. H. ウィンストン・K. A. プレンダギャスト共編、森健一・南川忠利・麻田治男共訳『AIビジネス』近代科学社、1986年6月。)

- 5) 高橋秀俊著『情報科学の歩み』岩波書店、1983年5月、169—172頁。

このことは、当初ソフトウェアという用語が、特定のプログラムを指していたことを意味する。しかし、今日では(後出の)OSやシステム・プログラムのみを指

してソフトウェアとは呼ばず、代わりに基本ソフトウェアと呼び、ソフトウェアはプログラム一般という考えの方が広まっている。そのため、当初のソフトウェアの定義は、実情に合わなくなってしまっている。本稿で、ソフトウェアとプログラムについて、改めて区別と関連を問題にするのも、このような理由があるからなのである。

- 6) 同上書, 171-172 頁。
- 7) 今日では ROM (Read Only Memory) に書かれたプログラムをファームウェア (firmware) と呼ぶことが多い。マイクロ・プログラムはその原型とされている。
マイクロ・プログラムについての詳細は、矢島脩三著『計算機の機能と構造』岩波書店, 1982 年 2 月, 246-249 頁参照。
- 8) ファジィ理論を体系的に叙述したものとして、差し当り下記の文献を参照のこと。
水本雅晴著『ファジィ理論とその応用』サイエンス社, 1988 年 2 月。
- 9) ファジィ理論の、ハードウェア、ソフトウェア両面における応用と実用化については、『Computer Today』(特集「ファジィコンピュータ」) 5 巻 3 号, サイエンス社, 1988 年 5 月を参照。
- 10) 石沢篤郎著『コンピュータ科学と社会科学』大月書店, 1987 年 2 月, 83-88 頁。
- 11) 機械語のプログラムは、この命令セットにもとづいており、アセンブリ語のプログラムも、基本的にはそれに対応している。また、いわゆる手続き型と呼ばれるプログラミング言語の仕様についても、命令セットとの対応を意識した部分が多い。
- 12) アルゴリズムを厳密に考えると、人工知能的コンセプトにもとづくプログラムは、アルゴリズムによらない部分を持つことになる。この点では、むしろ問題解決手順と呼ぶ方が、適切ではないかと考えられる。
- 13) 命令はコード化されればデータとして扱われる。従ってプログラムがアルゴリズムとデータ構造の 2 大要素からなるとする見解が、広く認められているものの、データの方が、ソフトウェアの、より一般的な形態であると言うことが出来る。
アルゴリズム+データ構造=プログラムという見解について、詳しくは、H. Levine, H. Rheingold, *The Cognitive Connection*, Prentice-Hall Press, 1987. (H. レヴァイン・H. ラインゴールド共著, 椋田直子訳『コンピュータ言語進化論』アスキー出版局, 1988 年 3 月, 173-208 頁。)
- 14) 石沢, 前掲書, 47-48 頁。
ただし、条件つき分岐命令は、プログラム自体よりも、むしろ論理回路の特徴であるように思われる。つまり、条件つき分岐が可能であるかどうかは、論理回路そのものの問題であり、プログラムでは、むしろ、いつ分岐するかが問題となるだけの筈である。

なお、ワイゼンバウムの見解については、J. Weizenbaum, *Computer Power and Human Reason*, W. H. Freeman and Company, 1976. (J. ワイゼンバウム著、秋葉忠利訳『コンピュータ・パワー——その驚異と脅威』サイマル出版会、1979年、111頁参照。)

- 15) プログラム配線式のコンピュータとしては、1946年のENIAC (Electronic Numerical Integration and Calculation), プログラム内蔵式のコンピュータの最初のものとして、EDSAC (Electronic Delay Storage Automatic Calculator) が、有名である。
- 16) 石沢, 前掲書, 88—90頁。
- 17) OSの目的には、ハードウェア資源の管理が基礎にあるが、次第にソフトウェアを含めたコンピュータ資源全体の管理へと機能の拡大が進んだ。
高橋延匡・土居範久・益田隆司共著『オペレーティング・システムの機能と構成』岩波書店, 1983年9月, 6—8頁参照。
- 18) 言語プロセッサには、先ず特定のプログラミング言語で書かれたソース・プログラムから、コンピュータで実行可能なオブジェクト・プログラムを生成することの出来る、トランスレータがある。トランスレータには、コンパイラとアセンブラが含まれる。トランスレータと異なり、このような変換を行わずに、プログラムを逐次実行する言語プロセッサが、インタプリタである。機械語以外のプログラムは、言語プロセッサを介しなければ、実行出来ない。
- 19) 高水準(つまり、より人間向き)のプログラミング言語による、プログラム技法の習得は、実際、問題解決技法の習得にはほぼ等しい。しかし、プログラムの効率化、例えば同じ問題を、より速く解くプログラムの作成といった点を追求すれば、低水準(つまり、よりコンピュータ向き)のプログラミングに戻らざるを得ない。
- 20) 意思決定過程の、基本的な位置づけについては、前掲拙稿, 37—38頁参照。
- 21) 拙稿「コンピュータ技術の形成過程」野口祐教授還暦記念論文編集委員会編『現代企業における技術と経営の展開』森山書店, 1987年11月, 139—153頁。機械操作労働については、153頁の註11参照。
- 22) なお、この点については、事務機や事務機に向かう「作業」が事務であるとすると、旧来の事務労働観に問題があるように思われる。新旧の事務論について検討した文献として、涌田宏昭著『増訂 経営事務管理論』白桃書房, 1970年11月, 第3章, 21—39頁参照。
- 23) 勿論、人間自身が生命体としての客観的物的存在であるが、ここでの問題は、労働手段が社会的な精神的活動を担う主体たりうるか否かという、より高次の事柄なのである。
- 24) 石沢, 前掲書, 92頁。

- 25) DSS は、通常、データベースと、モデル分析手法のセットから成り立っている。詳細については、広内哲夫・小坂武共著『意思決定支援システム——DSS 構築の方法論』竹内書店新社、1983年6月参照。
- 26) C. I. Barnard, *The Functions of the Executive*, Thirtieth Anniversary Edition, Harvard University Press, 1968. (C. I. バーナード著、山本安次郎・田杉鏡・飯野春樹共訳『新訳 経営者の役割』ダイヤモンド社、1968年8月、第14章、209—222頁および第17章、269—297頁参照。)
- 27) H. A. Simon, *Administrative Behavior—A Study of Decision-Making Processes in Administrative Organization*, Third Edition, The Free Press, 1976. (H. A. サイモン著、松田武彦・高柳暁・二村敏子共訳『経営行動——経営組織における意思決定プロセスの研究』ダイヤモンド社、1989年2月、7—10頁、および第3章、56—76頁参照。)
- 28) いわゆる同族企業を指す。
- 29) H. A. Simon, *The Science of the Artificial*, Second Edition, MIT Press, 1981. (H. A. サイモン著、稲葉元吉・吉原英樹共訳『新版 システムの科学』パーソナルメディア、1987年12月、46頁参照。)
- 30) パーソナル・コンピュータを用いた BES の例としては、下記の文献を参照のこと。前出のファジィ理論の応用についても紹介されている。
C. W. Holsapple, A. B. Winston, *Manager's Guide to Expert Systems Using Guru*, Dow Jones-Irwin, 1986. (C. W. ホルスアップル・A. B. ウィンストン共著、オカムラ・コンピュータ・システム訳『ビジネス・エキスパート・システム——管理者のための人工知能入門』ダイヤモンド社、1986年10月。)
- 31) 前掲拙稿、参照。
- 32) コンピュータによる織物の模様のデザインや、作曲の実現については、人工知能研究に関連して、よく議論の対象となるところである。この議論については、結局、ある構想の善し悪しの評価は、社会的に人間が行なう筈であるということしか、今のところ言えない。例えば、人工知能研究で有名な、ウィンストンの見解では、社会をどう見るかについて全く触れていない。本稿では、ウィンストンとは異なる立場をとっているが、何れにせよ、この問題はコンピュータ技術論の核心部分であるので、今後十分に検討していきたい。
P. H. Winston, *Artificial Intelligence*, Addison-Wesley Publishing Company, 1977. (P. H. ウィンストン著、長尾真・白井良明共訳『人工知能』培風館、1980年9月、267—271頁参照。)