

研究所における実験，測定の自動化システム*

須 貝 哲 也**

1. ま え が き

研究所で実験や測定を計算機に結びつけて自動化する試みは、いわゆるセンサーベースの応用の一種である。ほかに工場の生産設備の自動化、製品の品質測定などがあり、これらはすでに充実の時期を迎えている。しかし、ここで問題にしようとしている例では、かならずしもこの段階にはないのが実状であろう。それにもかかわらず、企業内においては、研究開発力の増大はその死活の問題であるとする考えが強まったり、将来、さらに深刻になるであろう人手不足は研究の分野も例外でありえないなど、この分野での計算機利用への期待、緊急度に不足はない。

この遅れの理由は何か、これを考えることは、とりもなおさず、生産設備の自動化などは大部異なった、このシステムのもつ性格を考えることになるようである。実験、測定の自動化といっても、この場合にはそれが自己目的ではないのであって、研究者という主体との結合が問題となる。それに続く情報の処理と呼ばれることについても同じで、ここではむしろ吟味という言葉が適切となろう。いずれにしても、自動化とはいうものの無人化した研究所は矛盾なのである。まず、この由来について考えてみよう。

2. 研究所の自動化システム

研究所における計算機の利用はいまさらのことではない。数値解析の目的では計算機が生まれたときからの利用者である。またセンサーベースの応用の分野においても、個々の具体的テーマについては多くの成果もあがってきている。しかし、これらすべては計算機なしでは考えられないような分野、固有の限られた研究分野についてであって、これを行なう人々も計算機については半玄人のような方々ばかりといって過言ではないであろう。いまわれわれが問題とし、当面してい

るのはこのような限られたものでなく、一般の多数の目的に計算機を利用し、メリットをあげることである点、最初に断わっておく必要がある。いままで計算機なしになされてきた多くの研究に計算機を導入し、これによって仕事を能率化し、研究達成の期間を短縮するなどのメリットをあげようとするのが、われわれの当面している問題で、以前の視点とは大部異なったみかたが必要である。われわれのシステムのもつ特質の多くは、このことからでてくるのである。

かりに研究の活動を流れ図化したとしよう。ここにみられるのは、研究者が関与して形成されるループが大きな比重を占めていることにちがいない。この部分は本質的であって、ループ配列の合理化によって消去できない性質のものである。このため自動化とはいうものの、それにはおのずから限度があり、力点はむしろ仕事の能率向上と質の向上に置かれることになる。

当然のこととして、研究機器は電気回路のレベルで計算機に結びつくことになる。そして、これは、自身が測定対象の物理量である時間によって厳しく拘束された状態で動作するものでなければならない。これは実験をおこない、実験データを集める側面であるが、ここでは人間の介入があまりない。しかし、次に、得られたデータの解析がシステムの機能の半分である。ここでは、人間の介入が比重を増し、いくつかの処理が前に述べた吟味のみに値する程度、くり返しなされるようになる。処理の結果もできるだけ迅速に人間が理解できる形でなされなければならないだろうし、また処理の内容も多彩であり、使用されるプログラムも数回の使用で使命を終える場合も少なくない。したがって新しいプログラムを作る活動が活発であろうと予想される。発生するデータも流れ去って消えるものもあるだろうが、研究結果の記録として保存される必要のあるものも多い。このようなデータやプログラムを資産として蓄える問題も起る。以上が研究所システムの素描である。

活動の面からみると、一部には固定的業務も含まれるだろうが、研究所内部でなされる仕事は、常に企画

* Automation of laboratory experiments and measurements, by Tetsuya Sugai (Fundamental Research Laboratories of Nippon Steel Corp.)

** 新日本製鉄株式会社，研究開発本部，基礎研究所

があり、これに基づいて計画され、実行され、結果が分析され、結論が出ればこれは次の企画に結びついてゆく。計算機に電気回路のレベルで結びついていた機器は撤収され、また新しい仕事の開始によりこの逆のことがなされる。そして、このときセンサーベースの新しいプログラムが必要になるが、これは一般の科学技術計算のプログラムと同じ程度の労力で作成できるものでなければならない。このように、研究所のセンサーベースの応用には、一般の工場における同種の応用にはみられない質の機動性が要求されることがわかる。

このような活動は、また、人間集団の活動の一部としてなされる点をまえに強調したが、この部面でも問題が指摘される。一般に工場におけるセンサーベースの応用では、計算機専門家、インターフェース・ハードの専門家と生産に携わる技術者との共同作業が可能であるが、研究所においてはこれが困難である。工場では、このことは主に工場設備の建設期のみに限られ、システムの稼動後は保守が中心になり、このためシステムには変革より安定が強く要求されるのであるが、一方、研究所の応用ではソフト、ハード両面の修正は、むしろ、常習的であり、一面では、これは研究自体の進展にはかならない。一般に、センサーベースの応用では、実時的プログラムが要求されることから多くのシステムの知識が使用者に負われ、計算機専門家の助力が不可欠とならざるを得ないのが実状であろう。しかし、両専門分野間の障壁は重大であって、この克服には多くのエネルギーが必要となる。まえにのべた工場の例では、これも止むを得ないであろう。しかし、急テンポで変化する研究所の場合は、この障壁は致命的とさえいえる。特に最近のように実験技術の急速な進歩のまえには、この障壁は一層克服困難なものとなろう。このような実状そのままでは、したがって、研究所システムは望み得ないものとなる。研究者という素人が十分容易に、少なくとも一般の解析計算の程度に、扱えるセンサー・ベース・システムの開発が、不可欠となる。

3. システムにもちいる計算機

これまでは研究所システムの特質、満すべき要件とといったことを、やや抽象的に述べてきたのであるが、われわれの問題は、いまずぐ入手可能な機材をもちいて、これをどのように実現するかである。

最近の計算機の進歩はめざましいもので、利用者と

しても、その選択に迷うほどの有様である。特に、小型ないし超小型計算機の普及が著しい。これは、性能に対比する価格も低く、各種の測定器などに構成要素として利用されている。測定器メーカーの今後の開発に期待される部分が多い。しかし、これを一般の研究所システムの用途からみた場合には、いくつかの難点がある。これはきわめて単純なことで、問題はコア容量の絶対的不足である。このため、この種の計算機の利用形態は、原始そのものから一步も踏みだすことがない。一方、この種の計算機にもディスク装置がつくようになり、記憶容量の拡大は、隘路を除くことになるのだが、一般にその場合のコア容量は小型とはいえない程度のもが必要となるようである。これはディスク制御のためのコアの増加による。

一方、前二節で述べた要件を満すようなシステムを得ようとすれば、これを保つためのソフトウェアの量は、かなりの量になることが予想される。これには、高い水準の言語、ローダー、実行の制御、ファイルの統括などのプログラムが含まれる。その結果、計算機の規模は中型以上のものとならざるを得ないであろう。また費用の点からいって、一台の計算機で複数の実験をサポートせざるを得ないことになる。これには二つの方式が考えられる。一つは中型ないし大型を中心に、超小型を衛星計算機として配するやり方で、もう一つは、この中型ないし大型を分割して複数の実験に利用するやり方である。前者では、親と子の間は密接な連結が必要であるが、直接データ伝送ラインで結合するほかに、紙テープなどで間接的に結合する場合もあろう。いずれにしても、子はプログラム実行の専用となり、プログラムは親から供給を受ける。このプログラムは親の豊富なソフトウェアのもとで用意されることになる。この方式の利点は、各実験が一応独立した計算機を各自一台ずつ使用できる点で、厳しい実時性の要求される用途では重要なことであろう。計算機の故障のとき累のおよぶ範囲の限定されることも重要で、このことは後者の分割使用の致命的な欠点でもある。特に、システムの規模の拡大にともない、このことは深刻になる。しかしながら、この親子方式は、現在いくつかの隘路をもっている。その最大のもの、クロスコンパイラなどの親子間を前提としたソフトウェアの開発の遅れていることである。当面のつなぎとしては、後者の分割利用にたよらざるを得ない。

4. 計算機の分割利用技術

計算機の分割利用の分野では、すでに時分割の方法が実用されている。しかしながら、これは今のような厳しい実時性の要求される例では、ほぼ無力といってよい。計算機をこの用途に分割する場合の最大の関心事は、分割された各実験のプログラムがこのような実時性のもとでいかにしてそれぞれの独立性を保つかにある。その結果、コア領域は共同利用というわけにはゆかず、それぞれ割り当てられた領域をもつことになる。また、プログラム自身も実行時の優先度に応じて細分化され、これらの実行順序や実行所要時間の合理化を十分に進める必要がでてくる。それぞれ独立な実験のプログラムが、実行時に矛盾をひき起こしたり、たがいに干渉したりすることを避けるための制御が必要となる。また一個のオペレーティングシステムのもとでは、それに備わるタイマー機構などのソフトウェア上のファシリティーの数には限りがあるので、これをうまく配分して使用する技術も必要となってくる。共同利用されるのは、この他に、中央処理装置、ディスク装置、コアの共同利用域、データ処理用入出力装置、それに A/D 変換器などのインターフェース・ハードの大部分が含まれることになり、比較的少ない費用で多数の実験を賄うメリットを実現できる。

現在、大型機と称するものでこのようなセンサーベースの利用ができるものがないので、この用途にはもっぱら中型機しか利用できない。しかし、中型といっても使用できるソフトは比較的豊富に開発されてくるようになったので、この程度の集約規模でも、集約化のメリットは十分に期待できるようである。このような方式での応用の一例を次に述べよう。

5. 研究所システムの例

このシステムは、新日本製鉄、基礎研究所において使われている FLANC-1 である。使用機材としては、計算機は IBM 1800 DACS、コア記憶容量は 32kW (16 ビット)、ディスク記憶装置を使用して 512kW のディスク・カートリッジを 3面使っている (アクセス時間最大 75 ミリ秒)。ソフトウェアとしては、IBM 1800 TSX を基に、これを大幅に改造、新ソフトを追加した F2 と呼ばれるものを使用している。現在は研究用機器、分析用装置など計 17 台の装置がこのシステムのもとで動いている。これらの装置と計算機とのインターフェーシング、装置を動かしたり結果の

解析を行なうプログラムなど、すべてが各実験担当者の自主的な活動によって賄われている。この担当者は特に計算機に詳しい者でなく、計算機の知識としては FORTRAN の理解の程度で十分である。インターフェース・ハードは担当者が中心となり、当所のエレクトロニクス・ショップの助力を受けて実施された。

現在もちいられている F2 の開発に当っては、センサーベースの実時的プログラムを作成するとき、システム的な知識をとりたてて必要でなくすることに最大の考慮が払われている。そのためには、厳しい実時性のもとで各プログラムの独立性を系統的にどう保証するかに苦心が必要である。

F2 の最も特徴とするところは、実行優先順位の制御であろう。一本のプログラムは、各部分が、その必要な緊急さの度合に応じて多数のレベル上でモザイク的に実行される。実験装置を制御するプログラムは、一連の手順をできるだけ一本のプログラムにまとめて書き下した方が、理解しやすいものとなる。その場合に問題となるのは、プログラム中にさまざまな異なる緊急度の部分が混り合ってくることである。これらのうち最大の緊急度に合わせて、プログラム全体を実行すると、複数のプログラムの並行処理の場合には種々の遅延の問題が起ってくる。F2 においては、プログラマーは各自に必要な実行の優先度をプログラム中に自由に指定できる。緊急度は 11 のレベルに分けられていて、最も低い 11 番目のレベルの指定をうけた部分はディスク常駐プログラムとして締め出せるようになっている。このようにプログラムをレベル上に細分化することで、最大緊急度の部分は全体のごく一部分になってしまう。これによって最高レベルの占有時間は大幅に削減され、二つの要求がこのレベル上でかち合う確率を実用上無視できる程度にすることができる。実際、このレベルの使用に当っては、どのレベルでは何ステートメント以内しか実行できないとか、時間精度が何ミリ秒必要な場合には、それはどのレベルで実行することとか、制限が使用者に課される。しかし、いずれにしてもこの理由でプログラムをいくつかのタスクに分ける必要はなくなる。

センサーベースで実時的なプログラムでは、それぞれ、外部割込みとタイマー割込みの取扱いが中心となる。一般に、これらの処理は、プログラムは小さなタスクに分割され、このタスクを単位としてなされる。例えば、実験装置の制御プログラムでは多数回のタイマーの使用が必要であるが、このプログラムはその回

数に等しい数のタスクに分割されなければならないことになる。このことは原理的には、別に問題になることでないのであるが、プログラミングの手数としてはかなり面倒なことである。またリアルタイム・モニタが基本的に備えているタイマの数には、一般に制限があり、これらを直接に各使用者が利用するのは多分に問題である。その理由は、これを利用できるプログラム数に制限があること、いまのように多数のプログラムが雑居している場合には各プログラム間でその使用を調整する必要から、それぞれの間にややこしい相互作用が現れることなどである。F2 ではこのような問題をすべて吸収してしまい、使用者はこの問題から開放されるようにした。またタイマ、外部割込みは FORTRAN ステートメントのレベルで使用できるようにし、タスクへの分割を不要にした。つまり、これは F2-CALL と呼ばれる形で利用される。例えば、F2-CALL の一つに CALL MSEC (L) というのがある。利用者はこれを FORTRAN プログラム中で使用し、L ミリ秒の休止と理解するのである。図1はこの使用例で、ある DO ループを示す。ループ中には CALL MSEC (10) が含まれ、したがって、この DO ループは 10 ミリ秒に一回の割りで回ることになる。また CALL L1, CALL L10 とあるが、これはまえに述べたレベル切換えのための F2-CALL で、DO ループ中で CALL L1 から CALL L10 の間の部分はレベル1で実行され、その外側の部分はレベ

```

// JOB      X      Y      Z
// FOR
*LIST ALL
SUBROUTINE TEST
.
.
.
DO 100 I=1,33
.
CALL L1
CALL MSEC(10)
.
CALL L10
.
.
100 CONTINUE
.
.
.
END
VARIABLE ALLOCATIONS
(I I )=0000

INVALID STATEMENTS

```

図 1

ル 10 で実行されるのである。外部割り込みもタイマーのときと似た形で理解できる F2-CALL になっている。このような F2-CALL はそれ自体でいくらかの時間を消費するが、タイマーの場合が最長で 2.5 ミリ秒となっている。この F2-CALL はそれを使用する回数に制限がなく、またこれが使用できるプログラム数には制限があるが、これはシステム・ジェネレーションのときに自由に変わることができる。F2-CALL がなされると、一般に、プログラムは休止に入るが、この時間に計算機は他の用途に開放されている。使用者のプログラムは、通常の TSX のローダーにより実行可能な形で計算機にロードされる。ただし、実行は F2 の制御のもとでなされる。この実行開始は特別な F2 制御用の CALL でなされるが、これは計算機中のいかなる領域、いかなる順位からも可能である。

以上が FLANC-1/F2 のあらましであるが、ここでわかるように、各プログラムの独立性が保証され計算機の分割利用が可能となったこと、一般の利用者に特別のシステム知識を要求しないことなどが特徴である。この点については、一般のメーカーが提供するシステムは多くの約束ごとを要求する。そのため導入に当たっては、その教育という一項が課され、使用者にエネルギーの消耗を強いるのであるが、しかもその内容は必然性の理解が困難なものばかりである。これを止むをえない出費とする理解も多いのであるが、ここでみるように、個別のシステムの性格に応じた整理によって、このような制約は大幅に軽減できるのである。

6. 実験の操作, インターフェース

実際の実験の操作で必要なことから整理してみると比較的平易なことからで尽されるのがわかる。主な要求は、押しボタンなどで必要なプログラムを起動させること、プログラム中の変数に外部から数値を入力すること、プログラム中の変数の現在の内容を知ること、などの三つである。このためには、一般には、タイプライタ・キーボード、紙テープ装置などが使用されるが、かならずしも、これらが最も便利かというところでない。特に、タイプライタは日本人にとって苦手のようである。また上で述べた分割利用の場合などは、計算機がスタンダードなフィーチャーとして備えているこれらの装置には、数も限られているし、システムの制約も多い場合が多い。それで、むしろ、これらの機能を平易な形で備えた操作盤を工夫する方

が、使用も便利だし、費用も安くなる。例えば、一つのプログラムを起動するのに、略号をもちいたシステムの的な手続をキーボード上で実行するよりも、そのプログラムに直結した押ボタンを一個用意の方が便利であることは明らかである。また現在の実験の進行状態を実験者に知らせるのには、数個の数字表示管で十分である。この場合、略号の定義などにはシステムはできるだけ介入せず、実験者（プログラムの企画者）が自由に定義できるようにすべきであろう。またデータの入力についても同じである。

7. 実験データの解析

一般に、実験、測定自動化などの議論では、得られた結果の解析の問題はあまり問題とされない。しかし、この分野における計算機利用でえられるメリットの多くはこの部分にあり、研究所システムを問題とする限り、これを含めて考えないと片手落になるだろう。この部門をサポートするには、実験装置の制御に用いる計算機の余力を利用する考えがある。しかし、経験的には、この余力の質にはかなり問題がある。実験側の負荷によっては、解析の処理速度が大きく左右され常に研究者の不満を残すことになる。また、計算の規模もこの余力では対処し切れないようなものも多い。この問題を解析するには、データ・ファイルを共通する形の別計算機を導入するか、また所外の巨大機を利用することが魅力的である。この余力には計算機の普及とか、プログラムの開発などの適当な用途を探すべきであろう。

研究所においては、結果の出力がグラフなどの図形が多いことは、一つの特徴であるといえるだろう。このため X-Y プロッタが多くの計算機のフィーチャーとなっているが、最近はむしろ CRT (cathode ray tube) がスピードの点で便利である。この場合、管面

上でグラフの座標を指定し、これを計算機に読み取らせるような対話形の利用は、データ解析の手段として有用であろう。

実験の結果えられたデータは、データの収集段階ですでに適当な処理を受け、量が減らされたのち使用される場合もあるが、そうでない場合も多い。これは原始データが、それ自体として現象の記録としての意味をもつ例が、研究所では多いためであるが、別の理由として、同じ原データに数回の異なる解析を行ないたい場合が多いことによる。このため、データの量は増大するのが一般である。このような解析は実時的には不可能でバッチ的になされるので、一時的にしろかなりの量のデータを保存する必要がでてくる。これにはディスク装置が用いられるが、そのためにはいろいろなプログラムから容易にかつ確実に利用できるファイル・システムが不可欠となる。また、データの解析に研究所外の計算機を使用する場合には、このデータ量は一つの問題点である。FLANC-1の経験では、日常の実験で発生するデータのサイズは 1kW ないし 100 kW 程度であるが、これを所外の計算機に転送してスピーディに仕事を運ぶためには、伝送線の能力としてはかなりの速度が要求される。

8. む す び

現在はまだ研究所システムとして標準的のシステムを議論するほど、この分野の利用が充実している時期でない。この報文では、著者の考えるあるべき姿を強調し過ぎた憾みが多少あるが、研究所システムがその研究所で存在理由をもちうるのは、その研究所独自の要求に裏うちされて始めて可能となるのであろう。多くの個性をもったシステムの開発が望まれるゆえである。

(昭和 47 年 3 月 21 日受付)