# 1. 実時間制御とその問題点

穂坂　衛（東大）

## 1. 実時間制御とその必要性

電子計算機の初期の使用目的は数式の計算であつた．それにつづいてビジネスへの応用が試みられ，その面でのデータ・プロセシングに大きな成果をあげてきた．これらは人が行えば非常に時間を要する仕事を，限りられた時間内に，人手をかりずに実行できるから用いられてきたのであるが，その時間的制約は非常に強いものではなく，ある程度の時間おくれは許容できるものであつた．したがつて計算機使用効率向上のために，同一種類の仕事を集めて機械にかけるバッチ・プロセシングのやり方がビジネスデータ処理に広く用いられてきた．

計算機使用の目的がある対象を管理し，制御することであるならば，計算機は対象の状態をつねに把握して，計算機内部および外部から発生する要求に対して，限られた時間内に出力を発生し，それによる行動，あるいは処置をとることが必要である．それが許容時間を越えるならば，種々の不都合が生じたり，制御できなくなつたりする．計算機が時間的制限の強いシステムに用いられて，要求の発生や現状のデータ取得が計算機に直結しており，出力も時間遅れ少く直接に必要な箇所に出されるような使い方が直結(on line)，あるいは実時間(real time)の使い方といわれ，そのようなシステムをreal time on lineシステムとしばしば呼ばれる．

例えば鉄道や航空会社における座席予約業務は，多数箇所から発生する要求呼をもとにして限られた時間内にフアイルの処理を行い，回答を返すと同時にフアイルの修正を行う実時間の在庫管理である．銀行の窓口業務も幾分似ている．時間的に遙かにきつい実時間制御は，航空や鉄道における交通管制や運転司令業務である．それらは多量の交通の流れを，安全に迅速に制御しなければならない．個々の列車や飛行機の現在位置，速度，目的地，経路等の関係より将来の位置を推定し，もし計画よりのはずれや，近接や衝突の恐れがあるときには，それを避けるような警報と修正の司令を発生しなければならない．工場における生産ラインの制御，プラントのプロセス制御等においても，時々刻々の現状のデータは計算機に送られ，それによる指令の発生を行わせ，最良の稼動状態に保つといつたことは次第に取り入れられるようになつてきた．さらに規模が大きくなつて企業全体としての在庫，生産の実時間管理や，シミュレーションによる予測管理というやり方が取り入れられるのも遠い将来ではないであろう．

計算機がシステムの制御に用いられるようになつたのは，システムを合理化するにあたつて，多くの機能を綜合すること，地域的な壁をとりはずすことが考えられると同時に，計算

機の能力が十分に要求に対応できるようになり，しかも実時間で用いても無駄のない使用法が可能となつたためである．

さらに綜合インホーメーション・システムとしての考への発展，ランダム・アクセスの大容量のフアイル処理が可能となつたこと，通信回線を利用して直接計算機と多数の入出力が結びつけられるようになつたこと，要求の発生点での入出機器が備わつてきたこと，さらに実時間直結方式ができる信頼性の高い，高速な処理装置や処理方法が次第に確立してきたことなどが実時間制御をおし進めてきた．これらを導入し，あるいは導入を考えている企業をあげてみると，政府機関，航空会社，銀行，電話会社，ホテル，証券取引所，各種製造工業，鉄道，公共企業，卸小売業など広範な分野にわたつており，分後十年の間には，ビジネスにおける計算機利用の大半は実時間制御に向いていくであろう．

## 2. 実時間制御の問題点

### 2.1 システム分析と合成

前節にオンライン実時間制御の必要性と，それを可能ならしめる技術的な可能性ができたことを述べたが，実際のシステムを作るとなると，対象が一般に複雑であるため相当な計画と決意をもつて対処しなければ，よい結果は得られないであろう．先づ現状と将来の形を心に画いて，それを取りまく条件と環境の分析から始まり，利用できる技術，組織，開発すべき技術等も調査し，また現在要求されている問題を明確にしなければならない．それにはシステム制御するに必要なデータの収集，分析，使用者側の要求，費用，政策，システムの入出力等があり，その結果によつてさらに目的を明確に選定することが必要になる．それに従つて，目的を達成すべきいくつかの方法を考案し，それらの性能，コスト，等を評価し，最もよいと思われるものについて開発を行うことが定められるのである．大きなシステムを取扱かうには，そのシステムをよく理解することが第一歩である．しばしばシステム内部の相互の関係を明確にするために，また新しく提案した制御の方法を評価するために，対象となるシステムを計算機でシミュレートすることを考える．計算機でシミュレートするためには，対象の複雑な内部と外部の関係が明確に表現されなければならない．それ故システムを具体的に考えることになりシステムについて理解を深めるのである．さらに計算機で実験することによつて，モデルの修正を行つて制御される対象を計算機の中に正しく作りあげ，さらにそれを制御する手段も計算機内でシミュレートして各要素間の情報の量的な関係，システム制御の効率，あるいはシステム内部の弱点や余分な箇所の摘出，また実際には稀にしか生じない事態になつたときのシステムの応答等のことを知ることができる．したがつて制御が計算機が主体となる場合は，それに要求される性能，また処理の方法，判新の基準，フアイルの形式，人との関係，システムの構成におけるバランス等多くの事が評価される．

このような各種の手段を用いて，複雑なシステムの実時間制御の計画はかたまり，実際の

機器の設計，製作，またそれらのテスト，人員訓練の計画が立てられるのである．

2.2　人間と計算機との関係

計算機が単なるルーチンの仕事を処理するだけでなく制御や管理の中枢に入つてくると，人間の役割を改めて考えなくてはならない．すべて起り得る事態が捕えられ，あらかじめプログラムしておくことができればよいが，そのようなことが不可能であり，また機械に常識を期待することもできない．したがつて非常に重要な，また複雑な決定は計算機にまかすことは適当でないかも知れない．そのときは計算機は人が意思決定を行うのに必要なデータを，急速に人に与えることであり，また人の要求に応じてテストを行つたり，あるいは簡単な規則に従つて機械が決定を示し人間がそれを監視することも必要であろう．また人間が決定を行つたなら，計算機にそれをフィードバックする必要があろう．

人の決定のし方は，与えられたデータが同じであつても変わることもあり得るのであるから決定を行う規則を明確にし，その規則のどれを採用すべきかを計算機に知らせる方法もあるであろう．

計算機と多数の人間とが協力する場合，人の仕事の様式も考慮されなければならない．人間は人の取扱える速度，また量の範囲でしか機械と共に働けないのである．計算機が人の決定を急がせたら，人は機械と同じように機械的な決定しかできなくなるか，人が隘路となつて仕事がはかどらなくなる．このため仕事を多数の係員に分けるにあたつて，係員をシリアルに多数配置するよりも，一貫した仕事を多数の係員でパラレルに行つた方が，判断を要する仕事をするのに適していることも知られている．

実時間で情報処理を行うシステムにおいては，仕事の発生がランダムな間隔であれば，処理する計算機の入出力部や内部，および関係する係員の所の平均の仕事処理量が，ただそこにおける平均の処理能力に見合つたのでは，大変な待ち合せを生ずる．そのため限られた時間内に処理が終ることが必要ならば，処理能力を入力レートよりかなり大きくして，待ち合せの起る確率を減らさなければならない．これには一般に数学的解析よりもモンテ・カルロ法を用いるシミュレーションによつて，その待ち合せの性質をしらべなければならなくなる．

計算機と人との通信はタイプライタ，ボタンやスイッチ，ランプの他にブラウン管表示，音声等が用いられるが何を人に伝えるべきかは，制御においては重要な問題である．人には何が必要かということが明確にわかるならば，制御は人より計算機に移すことができるであろうし，必要らしいと思われる情報をすべて出すことは，人がそれを整理して判断することが困難になる．この表示の問題自身システムの問題であつて，操作員の仕事の分析とシステムの他の部分との関係で決定されるべきことである．

計算機制御システムでは，この他でも多く人間との関係がある．制御を自動化する目的の一には，人を減らすことであつたが，実際には他の種類の人が増加する．計算機を保守する仕事，システム・プログラムを常に最新にして，装置が改善されたり，操作方式が変わつた

りすることに対処するシステム・プログラマの仕事は何時までも残る．自動化が進む程，複雑なプログラムが必要となり，それの保守に高級な人がいるようになる．同様に操作員の選定，訓練も必要となる．日常の操作は必ずしも最もよい訓練を与えるものではないから，すでにいる操作員の訓練にも，その評価にも制御計算機自体を用いる必要がでてくるのである．

2.3　信頼性

システムの制御を行うときには，システム全体の信頼性について十分の対策がとられなければならないのは当然である：複雑なシステムでは多くの機能を行うために，関係する機器は数多く，故障を起す機会も多いのであるが，そのためにシステムの機能が失われるならば，その影響する所は大きく，大変な損失を招く．ことに，その中心となる計算機の信頼性は最も大切で，かつむづかしい所である．システムの信頼性を考えるときは，稼動率，MTBF，あるいはダウン・タイムの時間等が問題になるが，用いる機器は設計の段階から十分に信頼性実績のある部品と確実な手法を用いることにおいても信頼性向上のための製造過程およびその後の保守体制の確立等が大切であるが，さらに冗長度を設け，誤りを発見，訂正すること，一部が故障しても，システムの綜合機能には影響がないようにすべきである．通信や計算機内部の情報移送には各種のチエックを行うこと，予備機器をもつこと，プログラムによる誤りのチエックおよび処理を行うことはもちろんであるが，実時間制御プログラムが破壊されないこと，および過去からのデータを集積しているフアイルは失われたら復原できないのであるから，何らの形式で常に2重にフアイルをもち，いつも現状に訂正されていることが必要であろう．複雑なシステムの保守はやつかいであるから，できるだけ計算機の機能を利用して，計算機自身およびそれに連なる機器のテスト，診断を自動化することが大切で，このためにもプログラムは複雑なものとなる．

実時間制御において，計算機の動作の信頼性を確保するために，二台を並列に運転し，おのおのの結果をチエックし合うデユアル計算機においては，面どうな問題が幾つかある．一般に大容量のフアイルが存在するため，その機械的な運動は同期することはできないし，またプログラム自体，両方が同じステツプで進んで行かず各種のチエックのため異なつた道を通過する．両計算機での処理の途中および最後の結果を照合しようとすると，遅い処理に時間を合わせるため処理時間が長びく方向に行くだけでなく，一致しなかつたときの処置や，一方の機械を停止，発進させるときの処置の問題，さらにもしマルチ・プログラミングやマルチ・プロセシングが行われていると非常に複雑な問題を発生する．

制御計算機を二重にし，並列照合運転を行うのか，待機させるのか，待機中にオフラインの仕事をさせるのかは，制御対象のダウンのときの影響に強く関係する．フアイルの破壊を防ぐことが必要ならば，二重並列運転が最も安全である．いづれにせよ実時間の場合，MTBFと平均の故障からの回復時間の比が非常に大きいことが大切な条件で，それを確保することに努めなければならない．

2.4　マルチプログラミングとマルチ・プロセシング

実時間制御において，制御要求の発生は始終連続的に生じているわけではない．したがつて計算機に対する負荷も一様ではない．その負荷が軽いとき，計算機はオフラインの仕事を処理するのに用いることができる．オン・ラインの仕事が発生して割り込み信号を発生したら，今行つている仕事を一時中止して，より優先性の高い仕事に取りかかる．このような外部からの割り込みと内部要因による割り出しは制御計算機には必要である．また制御要求のピークがきて，単一の処理機では時間的に処理不可能である場合も生ずる．そのときには複数個の同一の処理部によつて，またはそれぞれ機能の異なつた処理機構で仕事を分けて，パラレルに処理しその結果を綜合するマルチ・プロセシングあるいはマルチ・コンピューター方式が考えられることもある．これらの場合には，割り込みの制御と，そのサービスのプログラム，およびマルチ・プロセシングのマスターコントロール・プログラムが必要となる．これではどの仕事をどのプロセサーに割り当てるか，その仕事が終了したらどのプロセサに仕事を渡すか，また各プロセサの負荷を考慮し，全体としての生産性をあげるかを定める．この他実時間の仕事では要求発生が一定しないから，動的な方法で各機器を各仕事に割り当て，磁心記憶装置，テープ，デイスク等の記憶場所の使用割り当てを考えないと不経済になる場合がある．

2.5　実時間処理プログラム開発中の記録管理

実時間システムのプログラム作製は，前節までに述べてきたように，かなり複雑であり，命令数はすぐ数万，あるいはそれ以上ということになる．しかも一般にバッチ・プロセシングと異なつて，プログラム実行上，run　とかpassとかに分けられないで，各種の実行段階のものが混在するのと，それらはコア・メモリに入りきれないため，分割やリンク，呼び出し，退避等も非常に多くなる．プログラム作製に関係する人も多く，完成まで時間も長く経過する．このようなプロジエクトでは，要求の変更はもとより，金物もプログラムも，また参加する人品も変更，増減がしばしば生ずるのであるから，プログラム開発に当つては，各種記録を整理し，updateすることが大切でしかも大きな仕事となる．その第1は使用者側の要求事項はできるだけ明文化されている必要がある．例えば座席予約装置に例をとるならば，端局の窓口の操作盤だけについても，予約事務に必要な押ボタンの数とその機能，人が誤つて押したときの処置，相続く要求における押ボタンの処置，回答の表示機能，乗車券印刷の内容と形式，回答に要する時間の制限等々から始まつて，中央処理装置においては，列車の編成とは，車輛形式とは，空席割当の順序は，その変更は，発売開始，保留とは，その条件は何かといつた事が非常に多くまた細かに明確に定めていかなければならない．これは意志の疎通だけの問題ではなく，使用者側においても意見が異なつたり，あいまいであつたりするので，明文化する段階において，次第に明確になつて使用者側の考も定まつてくる．また同時にプログラミングの記録の仕様をきめそれに従つて，プログラムとそのデータ，表

の記録を作つておくことが大切である．その目的は，与えられた問題を分割し，個々にプログラムできる所までもつてくること，プログラマ内の情報交換を標準化し，相互のコントロールの受けわたしや，その形式を明確にすること．これによつて自分および他のプログラマが後にプログラムを変更しなければならなくなつたときにも，経験の浅いプログラマを効果的に仕事に参加させるにも必要である．これらの記録により，各種の表やデータの形式，ファイルの内部形式，アクセスの仕方等も統一されるし，プログラム内の不明確や矛盾する所，あるいは相互に協力を要する点がはつきりしてくるであろう．このような記録は作業の進行中に，かなりの変更を受けるから常にupdateできていることが大切である．

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング―シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の捜索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者捜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日 〜 2021 年 3 月 19 日
掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程
https://www.ipsj.or.jp/copyright/ronbun/copyright.html