

ー待ち行列網理論の応用の観点からー

京都大学工学部

長谷川 利治

1. まえがき

人々がシステムを構築しようとするとき、何らかの目標を必ず持っているものと考えられる。すなわち、ある目標を達成するためにシステムを構築しようとするのである。従って、システムを構築しようとするとき、設計しているシステムが初期の目標をどの程度達成できるのかを、推定し評価することが極めて重要な問題となる。

この種の問題は、対象となるシステムによって、また、システムに対する観点によって、性質が大いに異なる。例えば、高速道路などにおける構造物の耐震性の評価等においては、耐震性の評価手法がかなり信頼しうるものがあるため、どのような外力が構造物に加えられる可能性があるかを予測し、多くの条件を考慮に入れた結果、最大外力を設定することが問題となる。一方、高速道路の容量について考慮する場合は、最も簡単な短い区間、例えば料金収受ブースの容量を決定する場合においては、自動車の到着間隔、料金徴収時間等の平均や分布等が問題となり、複雑な確率計算を行わなくてはならず、解析的に解けない場合も多々ある。また、システムの振る舞いが、かなり直感とは異なったものとなることが多い。従って、システムの評価手法に一般的なものではなく、対象とする問題によって種々なものが考えられている。

システムの性能評価を十分に行わないことによってもたらされる問題は、2つに大別される。性能評価を十分行わなかったため、まず、予想した範囲の需要であったにもかかわらず、システムが働かなくなってしまう問題である。数年前に起こり、大きな社会問題となった郵便貯金や東京証券取引所のシステムダウン等がこの例と考えられる。もう一つの問題は、安全係数を大きくとりすぎたため、システムを過大に作ってしまうことである。この場合は、問題が表面に現れにくいという新たな問題を起こす。一見問題がないように見えても、内面に資源や資金の無駄が生じている。

ここで、システムの「適正」な構成をするためにはどのように考えればよいか、まず、重要となる。一般的且つ普遍的に「適正」はシステムを定義することは不可能である。なぜなら、宗教的なシステムは例外であるかもしれないが、あるシステムを構成するとき、そのシステムはある価値体系のもとで構成されるので、一般的且つ普遍的な価値体系が存在し得ないからである。また、あるシステムを規定する価値体系は、そのシステムで規定されないメタシステムである。このように考えると、システムの性能評価にあたっては、「適正な」システムを構築するためには、まず「適正」さの意味を限定して定義し、それに従って「適正さ」を評価するための測度を選択していかなければならない。続いて、この採択された評価測度に対して、対象とするシステムの挙動を知るためのモデル化を行うのである。

いくつかの要素の組合せによって構成されたシステムの挙動を推定する際、個々の構成要素の挙動を知ることは必須の事であるが、それらの挙動の単純な合成によって、システム全体の挙動を知りうることはごく希である。種々の挙動を持つ要素群から構成されたシステムの全体の挙動が、要素を組み合わせる方法の差、例えば順序関係の差により大きく異なることがあることは周知の事実である。このことは、個々の構成要素の挙動を明確にして置くことが、システム構成を行う際の必要条件ではあるが、十分条件ではなく、より高いレベルから広く考慮しなければならないことを示している。

システムの構成に当たって、どのレベルまでの挙動を評価し、目標達成度を推定する必要があるかを一般的に述べることは不可能であり、その時々々の技術レベルなど多くのものに依存している。従って、ここでは、待ち行列網理論によってモデル化できるシステムに性能評価に限って、その現状を述べることにする。このような性能評価手法は、阪神高速道路において、採用されていたことのある縦取り料金所の処理性能の評価に用いられ、かなり正確に現実を再現する評価モデルが構築された例もある。小文では、かなり一般的にモデル化された、情報伝送系を含む情報処理システムのモデル化と性能評価について述べ、他のシステム構成に対する参考となることを願うものである。

2. システムの性能評価測度

一般的なシステムは勿論、情報システムに限っても、性能評価測度をどの様を選択、定義するかは、対象とするシステムの構成目的、構成者あるいは管理者、さらにはシステム利用者などに依存する問題であり、統一的或は一意に取り扱うことは不可能である。評価測度を広く考えると、必ずしも定量的なものとは限らず、定性的なものも含まれるべきであろう。定性的な測度を客観的に扱うことは、重要ではあるが容易ではなく、ここでは定量的な測度に就いて考えることとする。

以上のような前提にしたがっていくと、まず考えられる立場として、システムの評価をシステムの利用者の立場と、システムの管理者の立場がある。単純に考えると、これらの立場が対立的なものであるように見えるかもしれないが、必ずしもそうではないことを次節で述べる。

利用者の立場に立てば、システムが利用者へ供給できるサービスの程度が、利用者の予期したものを上回るものであれば、そのシステムが「良い」性能を持ったものだと言うことが出来る。計算機システムであれば、あるジョブをシステムに投入してから回答を得られるまでの時間、すなわち「ターンアラウンド時間」、あるいは利用者がキャリッジリターンキーを押してから計算機からの応答がスクリーンに現れるまでの「応答時間」、などが短ければ良いシステムである。

また、一般にシステム内で同時に取り扱えるジョブ数には制限があるので、あるジョブがシステムに到着しても、システムに入ることを拒否されることが起こり得る。このブロッキング確率が評価測度となる。すなわちブロッキング確率が小さいシステムが良いシステムである。

上述の応答時間などの時間は、遅延と総称されるが、比較すべきシステムの間において、同一のものを定義しておくことが比較的容易であり、かつ利用者にとっても理解し易いものであるため、よく採用される。ブロッキング確率も含めて、これらの評価測度はすべて決定論的な値を持つものでなく、確率変数である。従って、これらを採用する際には、それらの分布や平均値、分散などを問題とすることになる。

管理者の立場を取れば、スループットや利用率が評価測度として考えられる。スループットは、単位時間あたりにシステムが処理し得る平均ジョブ数と定義され、システムのある意味での生産性を示すものである。しかし、ジョブのシステムへの到着がシステムの状態と独立であり、かつシステムが定常状態で稼働しているときは、スループットはジョブの到着率と等しくなり、従ってシステムの性能の評価測度としては不適当なものとなる。この場合は、スループットはシステムの性能ではなく負荷を示すものと考えられる。スループットが性能評価測度として適している場合は、次のいずれかの場合である。

(i) システムへの入力において、サービス待のジョブが常に（確率1で）存在する場合。

(ii) ジョブの到着がシステムの状態に依存する場合。

(i)の場合は不安定なシステムとなり、サービスを待っている行列の長さは時間と共にどこまでも増大するばかりである。しかし、実際には待行列が無限大の長さにならない有限の時間区間において、待行

列長が0でないという条件のもとで、スループットを定義し測定するので、システムの能力の評価測度として適切なものと考えられている。

(ii)の場合は、ジョブを生成する源の数が有限である場合が考えられる。たとえばシステム内の端末の数は有限であり、活動中の端末数すなわちシステムの使用状況によって、新たに活動を開始する端末が発生する確率が異なって来る。すなわちシステム内のジョブ数の増大によって到着は減少し、システムの状態と到着率に負のフィードバックがあることになり、システムを安定にする。このような場合にも、スループットは適切な性能評価測度となる。

利用率とは、処理装置〔たとえば中央処理装置(CPU)〕が稼働している確率、または有限の観測時間内に処理装置がジョブを処理している時間の割合のことである。いま、簡単な単一CPUのモデルを例にとる。CPUに到着する平均ジョブ数(定常状態を考えれば、負荷であるスループット)を毎秒 λ 個、ジョブ当りの平均処理時間を $1/\mu$ とすると、利用率 ρ は次式で与えられる。

$$\rho = \lambda / \mu$$

システムが定常な状態に留まるための条件は、

$$\rho < 1$$

である。このようなシステムにおいて、平均応答時間[T]は、システム内で待行列で待っているジョブと、処理を受けているジョブ数の和を[n]とすると、次式で与えられる。

$$\lambda [T] = [n]$$

以上に述べたもの以外にも、評価測度が考えられる場合がある。システムの性能評価の問題は、システム構成の価値体系の問題とかかわっており、システム構成に当たって広くかつ深く考慮しなければならない。

3. 評価測度のトレードオフ

前節で述べたように、評価測度は考える立場によって異なったものとなる。利用者にとっては、システムの応答時間が短ければ短いほどよいが、応答時間を短くするシステムにおいては、処理装置あるいは伝送装置の利用率を極端に下げおかなければならない。このことはシステムが本来持っている処理能力に比べて、スループットが極端に低くなることを意味している。一方、システムの管理者の立場に立てば、システムの規模に対してスループットすなわち収入をできるだけ大きくするシステムが、よい性能を持ったシステムであり、利用率の高いシステムがよいシステムである。この種の「良さ」を追求すれば、利用者にとって望ましい指標は低くなってしまふ。

以上のように単純に考えれば、利用者と管理者の立場は対立関係にあるように思われるが、必ずしもそうではない。たとえば、利用者の立場から応答時間を極端に短くすれば、システムの利用効率が下がり、ひいては使用費用がかさむ。また、システムのスループットや利用率を高くすることのみを考えれば、利用者にとって使用しがたいシステムとなり、システムの競争力を低下させる。

従って、これらの測度間のトレードオフを考え、適当な値を示すシステムを構成することが重要となる。妥協点の定量的な追求である。この種の評価法として、吉岡らのパワー関数によるものがある。パワー関数はスループットを遅延時間で除したものであり、スループットを適当に大きくしつつ遅延時間を適当に短くしている割合の評価測度となっている。

高速道路について考えると、旅行時間最短という利用者の目標を満たすためには、高速道路上での自由な交通流を実現させなければならない。このためには、高速道路上の交通密度を下げねばならず、交通量を減少させねばならない。交通量の減少は料金収入の低下を招き、それに対処するためには料金値

上げ、あるいは公的な収入補助が必要となる。この事実は、旅行時間最短をめざせば、旅行者の負担を増やさなければならなくなることを示している。

旅行時間と負担の重さのトレードオフを考えると、旅行時間最短という目標より、交通量を最大にする目標の方が、利用者と管理者双方にとって望ましいものであることと判断することが出来る。

しかし、一般にはトレードオフの形は無限に考えられるので、ある特定のものに評価を依存させるのは適当ではない場合もある。しかも、評価測度を決定することは、性能評価のレベルでできる、あるいは行うべきものではなく、より高いレベルにあるメタシステムにおいて決定されるべきものである。さらに、システムの設計段階から全ての評価法が確定しているとは限らないし、また固定してしまうことに問題がある場合もあることより、設計から構成段階へと進むにつれて、新たな問題が提起されることは複雑なシステムにおいてはよく起こることである。従って、性能評価を行うに当たって、先ずシステムの挙動をできる限り解明し、システム構成の意志決定者に可能な限りの資料を提出する手法を確立することが、もっとも重要なことである。

4. 情報システムのモデリング

システムの設計時において、種々の代替案に対して性能の比較評価を行う際、システムの挙動のモデル化が必要となる。この際、全ての挙動を正確に示すようなモデルを作ることは不可能であるし無駄でもあるので、いくつかに分割して考えることが通常である。分割の形としてはいくつも考えられるが、代表的なものとして図-1に示される階層化分割がある。

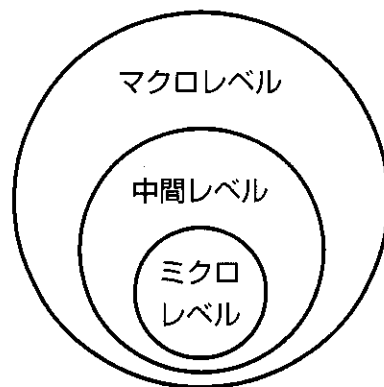


図-1 モデルの階層

図-1において、マイクロレベルでのモデル化からの解析結果が、中間レベルすなわち上位レベルでの解析に利用され、さらに中間レベルの解析結果がマクロレベルの解析に利用されるように階層構造を持つことが示されている。実際のモデルではこのように明白な形で境界が決定されることはまれであろうが、概念的にはこのように考えられる。

分割に際して重要なことは、各層で考えられる同一のパラメータの単位の大きさが、層の順序関係と同じ順序関係を保つことである。すなわち、事象の発生を捉える時間について考えれば、マイクロレベルからマクロレベルへ時間間隔が大きくなって行くことである。例えば、マイクロレベルで考える事象の生起時間間隔が、ナノあるいはマイクロ秒の単位で考える程度のものであるとすれば、中間レベルではミリ秒の単位で、マクロレベルでは秒の単位で考えられるようなものとしなければならない。

4.1 モデリングにおけるパラメータの選定

モデル化に際して考慮すべきシステムパラメータを、計算機システムの例について示すと、次のようなものが考えられる。

- ①主記憶装置、キャッシュメモリのメモリサイクル。
- ②CPUの処理速度。
- ③記憶装置のデータアクセス時間、データ転送時間。
- ④端末装置および通信装置の特性。
- ⑤ジョブスケジューリングアルゴリズム。
- ⑥メモリマネージメントアルゴリズム。
- ⑦ページサイズ。
- ⑧各機器へのトラフィック量。
- ⑨ジョブあたりのCPU時間。
- ⑩ページフォルトレイト。
- ⑪ディスクなどへのアクセスレイト。

上の例において、①～④はシステムのハードウェアに関するパラメータであり、⑤～⑦はソフトウェアに関するパラメータであるが、⑧～⑪は利用者についてのパラメータである。これらは比較的簡単に知り得るものである。

しかし、これらの多数の要素間の関数関係を知り、システム全体の性能に最もクリティカルに作用するものはどれかを知ることは簡単ではない。問題に即してパラメータを選択し、モデル化を行って行くのである。

図-1に示す階層モデルを、上のパラメータ群を対象にして考えると、マイクロレベルにおいては、ナノあるいはマイクロ秒単位の時間が問題となる主記憶キャッシュへのRead/Write要求などの生起が対象となる。従って、このレベルにおいては、メモリインタフェース、インタリーブの効果、キャッシュのマッピングアルゴリズムなどが解析される。

中間レベルにおいては、ミリ秒の単位で測られる事象の生起が問題となるマルチプログラミングのモデルに相当する。ジョブ（プロセス、プログラム）のI/O、ディスクへの転送、CPUでの処理時間および各サービスを受けるために、待行列で待つ時間などが対象となる。従って、このレベルでは、プロセッサスケジューリング、I/Oスケジューリング、メモリマネージメントなどが評価される。

マクロレベルにおいては、秒単位で測られる事象を対象とする。TSSによる入出力システムが、このレベルの例である。TSSユーザがキャリッジリターンキーを押し、システムへのリクエストを入力する時間間隔などがこのレベルで考えられる。TSS端末における応答時間が、このレベルでの典型的な評価測度である。

阪神高速道路の交通のモデル例について考えると、マクロレベルモデルでは、年間、月間あるいは1日交通量等の諸量を扱うモデルであり、中間レベルモデルは、交通制御のための5分間毎の各種パラメータを扱うものである。マイクロレベルのモデルとしては、突発事象の自動検出システムに見られるような、秒以下の時間間隔で測定される、個々の自動車の振る舞いを扱うモデルが考えられる。

4.2 負荷の特性と従来性能の評価

負荷は、システムに対して利用者が要求するサービスの量であり、これに関する重要なパラメータはジョブの到着率である。第2節で述べたように、システムへの到着率はシステムの混雑度に依存する場合がある。従って、一般に負荷とシステムの性能は独立ではない。

負荷の特性は、ジョブの到着率のみによって定められるものではない。このジョブが要求する仕事の内容も問題となる。計算機システムは種々のリソースから構成されているので、ジョブが要求する仕事は、少なくともCPUの使用量、要求記憶容量、入出力要求量、使用するソフトウェアなどによって特性づけられる。計算機システムにおいて、CPUが主な臨界的要素であった頃は性能評価も比較的簡単で、各項目について別々に考えることが出来た。現在においては、従来用いられていた性能評価では十分ではなくなっているが、これらが全く意味を無くしたというのではない。ある要素、例えばCPUまわりのみの性能を示すことは可能であるため、従来の性能評価について言及する。

従来、負荷の特性を示すためによく用いられてきたものに、命令ミックスがある。対象となる計算機システムに与えようとしている代表的なプログラムの組を考え、そこで使用される第*i*種の命令の出現頻度を*f_i*としたものを命令ミックスとする。あるCPUを用いて命令*i*を実行するのに要する時間を*t_i*とすると、演算毎の実行時間 τ は、

$$\tau = \sum_i f_i t_i$$

で与えられる。*f_i*の単位時間をマイクロ秒とすると $1/\tau$ はそのCPUのミップス(MIPS: Million Instructions Per Second)値による能力を与える。ミップスは重要な測度ではあるが、CPUの性能の1つの評価測度であり、システム全体の評価には十分ではない。

負荷の特性を示す他のものに、核プログラムがある。これは、対象とするシステムにおいてよく使用されるプログラム内の内部ループを表すプログラム部分である。この核プログラムをシステムで実行することにより、CPU、記憶装置、コンパイラなどの性能評価が可能であり、命令ミックスよりは広い性能評価が出来るが、I/Oなどの評価に対しては十分ではない。

さら広い性能評価が可能なものに、ベンチマークテストがある。ベンチマークとは、高級言語で書かれた完全なプログラムで、あるアプリケーションプログラム群の代表的なものと考えられるプログラムである。ソーティングプログラムやアップデートプログラムなどがよく選ばれている。対象とするシステムにおいて、いくつかのベンチマークを実行させ、システム全体の性能を評価するものであり、計算機機種選定などの段階でよく使用されている。実際のプログラムで性能を評価するものであるから、命令ミックスや核プログラムなどを用いる方法に比べて、より現実的な評価を与えるが、一方どの様なベンチマークが評価のために選ばれたかによって、公平な性能評価に問題をもたらすことがあることに注意しなければならない。

実際のプログラムからベンチマークを選ぶのではなく、人工的に構成されたジョブを性能評価に用いることがある。対象とするシステムにおいて使用されると考えられるジョブ群を十分代表できるような人工的なジョブを構成するので、仮想的なものも含み得ると言う点で、ベンチマークとは少し異なった評価が可能であるが、ジョブ群を旨く代表できるプログラムの作成は簡単な仕事ではない。

4.3 性能評価のためのモデリング

ベンチマークなどによる計算機システムの性能評価は、使用するプログラムが現実のジョブをうまく代表しているか、と言う問題に加えて、実システムを用いて性能評価をしなければならないと言う問題がある。対象とするシステムの完成されたものを用いて評価しなければならないと言うことは大きな制限となる。実際の問題としては、これから構成しようとしているシステムの性能を推定しなければならないことが一般的である。このためにモデル化が必要となるわけである。

すでに述べたように、システムを構成する個々の要素の性能から直ちに全システムの性能が評価されることはほとんどなく、カタログ性能などは単に手がかりに過ぎない。各構成機器のパラメータ、オ

ペレーティングシステムの構成要素、人力特性などは互いに強く関連しあっている。

システム全体の性能評価に際して、重要な概念となるものにシステムボトルネックと不安定性がある。システムボトルネックは、システムの負荷が増大していく過程において、最初に飽和状態に達する部分である。飽和状態というのは、ジョブ間の干渉を各ジョブが強く受けるようになる混雑状態であり、動的でも確率的でもないモデルにおいて予測される混雑度より、かなり低いレベルにおいて起こることが一般的である。

システムのボトルネックは、ある場合はCPUであったり、I/O部であったりするが、システムボトルネックを知ることにより、システム全体の性能を制限している部分を知ることが出来、システムの処理能力を向上させようとしているとき、どの部分の性能を向上させればよいかを知ることができる。

不安定な状態は、前述のように利用率 ρ が1以上となる範囲において生ずる。この場合、システムの混雑状態と到着率が独立であれば、待行列長は増加の一途をたどる。注意しなければならないのは、たとえ $\rho < 1$ であったとしても、瞬時的な入力量の増大によりシステムが不安定になり得ることである。

システムが不安定状態に入ろうとしているとき、システム内のジョブ数が増大するにつれて到着率が下がる、あるいは到着がシステムに入ることを拒否される損失率が高くなるように、負のフィードバックループが存在する場合と、新たな到着がますます混雑を助長し、ひいてはスループットを0にしてしまうような、正のフィードバックループを持つ場合とがある。これらはいずれも、モデル作成にあたって、とくに注意しなければならない点である。

5. 性能評価の手法

ベンチマークテストなどによって、システムの性能評価を行う際、前述のように実際に使用されるものと同じシステムが必要であるが、性能評価モデルを作る際は、性能評価の手法のどれを適用するのかを考えておかなければならない。考えられる手法は、解析的手法とシミュレーション手法である。ここでは、基本的な注意について言及するにとどめる。

5.1 解析的手法

解析的手法が用いられるのは、システムパラメータと考えている評価測度との関数関係を記述できる場合である。例えば、システムへのジョブの到着率(λ)、およびジョブの平均処理時間($1/\mu$)が与えられたとき、ジョブのターンアラウンド時間(w)が計算で求められる場合などである。このような場合、重要な役割を果たすのが待行列理論を含む待行列網理論である。しかし、待行列網理論は任意の性能評価モデルに適用可能であるということではない。また、待行列網理論の適用を念頭においたモデルを作り得たとしても、厳密な解、しかも取扱いやすい形が得られるようなモデルとなることはごくまれである。

従って、厳密解が得られるような形のモデル化を、現実のシステムとの乖離には目をつぶって行うか、現実に近いモデルを作り厳密解をあきらめ近似解を求める方法がとられる。どちらの方法をとるにしても、一般にかなり大胆な近似をいろいろなレベルで導入することになる。このような近似が、システムの性能評価に最終的に与える影響について、十分考察を加えておかなければならない。

近似解析に満足が得られない場合は、一般的な解析をあきらめ、個別のシステムを対象とするシミュレーションに頼らざるを得ない。

5.2 シミュレーション技法

対象とするシステムが数学的、解析的に解くには複雑あるいは大規模すぎる場合、そのモデルを計算機上で作り、統計的な実験をそのモデル上で行うのがシミュレーションの実行であるが、ここで注意が

必要なことは、作成したモデルの正当性を一般的に証明することは不可能であるという点である。しかし、ある特定の目的に対しては、その正当性をかなりの説得力を持って示すことを怠ってはならない。

シミュレーション技法を採用することの長所と短所を次に要約する。

長所：*次のような対照実験が可能であり、解析的な解が得られないような問題の解が得られる。

- (1) 多くの要素を考慮に入れることが出来る。
- (2) 多くの要素を個々に扱うことが出来る。
- (3) 代替策を考慮することが出来る。
- (4) 実システムに殆ど影響を与えない。

*システム関連要員の効果的な訓練が出来る。

*システム運用に対する洞察力を与える。

*多くの予備知識を持たない人々にでもモデルの説明が容易に出来るので、関係者に理解されやすい。従って、有益なコメントを得やすい。

短所：*解に統計的誤差が必ず伴う。

*解の正当性の検証を一般的に行うことが出来ない。

*重要な仮定を見落とす可能性がある。

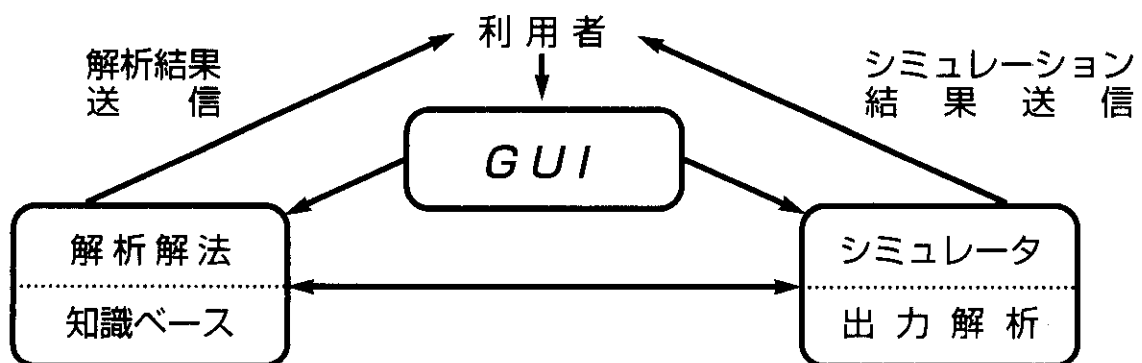
*実データ収集などに手数をかけなければならない。

*数値解は得られても、解の一般的な性質は得られない。

*シミュレーションの準備、モデルの作成、プログラミング等に手数がかかる。

待行列網理論による解析とシミュレーション手法を備えたプログラムパッケージの優れたものが開発されている。例えば、フランスの国立情報科学および自動制御研究所で開発されたQNA P、IBMのRESQ、日本電気のQM-Xなどは注目に値する。

一方、筆者らの研究室においても、ASQと称するハイブリッド型性能評価支援システムを開発している。(1)これは、解析解法とシミュレーション解法を並列で実行するシステムであり、ハードウェアシステムとしても分散型を採用している。この支援システムをブロック図で示したものが、図-2である。支援システムはグラフィックユーザインタフェース(GUI)、オブジェクト志向型シミュレータおよび解析解法部の3部分に大別される。利用者がGUIを介して、評価しようとしているシステムの待ち行列網モデルを入力すると、知識ベースによってそのモデルが解析的に解けるか否かを判断し、可能なら



GUI：グラフィックユーザインタフェース

図-2 ハイブリッド型性能評価支援システムの概要

解析解法が動き出し、解を出力する。否なら、モデルがシミュレータに送られ、シミュレーションを実行する。また、シミュレーションを実行する時間を制御するサブシステムを支援システムが備えている。

6. おわりに

以上、情報システムの性能評価のためのモデル化と手法について、ごく概略のみを、計算機システムを例として述べ、高速道路の例についても一部言及した。情報システムの性能評価についてくわしくは、例えば文献(2)-(5)を参照されたい。ここで強調したいことは、次のように要約される。

- (I) 従来の性能評価は、どちらかといえば静的、決定論的なものであったが、確率論的、さらには動的な性能評価の重要性は増加の一途をたどっている。
- (II) 性能を評価する測度はいろいろなものが考えられ、しかもそれらは互いに複雑に関連しあっていることが多い。性能を評価する手法から、どのような測度を選定すべきかは決定不能であるので、性能評価の手法の段階では、システムの挙動をうまく表すモデルを作り、考えられる限りの評価測度を求められるようにすべきである。評価測度は、システム設計者のみではなく、システム管理者・利用者などとともに決定されるべきである。
- (III) システムの性能評価においては、諸手法を熟知していることも必要ではあるが、そのシステムを熟知していることのほうがもっと大切である。

情報システムの社会的プレゼンスが増大するにつれて、システムの性能評価はますます重要となる。これに対処するためには、さらに多くの人々が待行列網理論に興味を示して頂きたい。

小文を準備するに当たり、筆者が近畿日本鉄道技術研究所技報第20巻1989年3月発行に寄稿させて頂いた報告「情報システムの性能評価」に加筆した。

参考文献

- (1)三好、河野、高橋、長谷川：「分散環境下における性能評価支援知識ベースシステムに関する考察」、1993年度日本オペレーションズ・リサーチ学会春期研究発表会アブストラクト集、PP.148-149.
- (2)長谷川、宮原、高橋：「情報システムの性能評価Ⅰ－評価測度とモデリング－」、システムと制御、第27巻2号、pp.114-121、1983年.
- (3)高橋、宮原、長谷川：「情報システムの性能評価Ⅱ－待ち行列システム－」、システムと制御、第27巻4号、pp.260-267、1983年.
- (4)宮原、高橋、長谷川：「情報システムの性能評価Ⅲ－近似解法－」、システムと制御、第27巻6号、pp.385-390、1983年.
- (5)宮原、高橋、長谷川：「情報システムの性能評価Ⅳ－ローカルエリアネットワーク－」、システムと制御、第27巻7号、pp.464-468、1983年.