CN 20 − 3
(1977. 5. 25)

分散型情報システム・シミュレータ

西原義之　真名垣昌夫　金子朝男　服部光宏
（日本電気　中央研究所）

1. はじめに

コンピュータおよびコンピュータ利用技術の発展にともない、大型コンピュータを中心とした集中型情報システムが1つの有効な情報システムの処理形態として確立されつつある。一方、最近になり、コンピュータ・ネットワーク技術を基盤として、分散処理システム、分散型データベースシステム、機能分散システムなどの一連の名称で呼ばれる分散型情報システムが議論されている[1,2]。またこのような処理形態をもつシステムは、すでに一部の先導的なユーザの間で稼動しており、今後も大きく発展するものと思われるが、その構築技術は今後の課題である[3]。

本資料では、分散型情報システムについて検討する場合、あるいはその導入や設計を行なう場合に問題となる分散型情報システムの性能予測方式に関して提案する。分散型情報システムの性能予測技術は、以下に示すように、すでにいくつか報告されている。

性能評価方式の一つであるハードウェア、ソフトウェアによるモニタリング方式[4,5]は、分散型情報システムの性能評価データの収集用として有効な手法であるが、性能予測用に使用するには不充分である。また解析モデルによるアプローチ[6,7]は、全体的な解析に有効であるが、分散型情報システムの多様性、複雑さにきめこまかく対処するには充分でない。一方、シミュレーションによる性能予測システムも提案されている[8,9]。これらは前記の欠点を解決しているが、分散型情報システムのシミュレータとして利用するには機能、性能面に不充分な点がある。第1の点として、分散型情報システムの性能予測は、アプリケーション、システムに依存する項目が多く、アプリケーション、サイドの設計者、管理者にも使用可能とすべきである。筆者らは、多端末を中心とした性能予測システム[10]の開発および利用を通し、エンドユーザ指向の重要性を認識しまた実現の可能性を示した。第2の点として、分散型情報システムの性能予測システムは、その対象とするシステム規模が大きくなり、シミュレーション時間の増大が予想されることから、時間短縮のための技術が必須となる。

本資料で述べる分散型情報システム・シミュレータDISS（Distributed Information Systems Simulator）は、分散したデータ資源を有する分散型情報システムの性能予測システムとして以下の点を目標にしている。

（1）OSの設計者、研究者や一部の専門家だけが利用するのでなく、主として一般的なシステム設計者やEDPS担当者を利用対象者として考慮する。（2）従って対象モデルに対する柔軟性があり、（3）対象モデルが簡易に記述でき、（4）シミュレーション環境が優れること。（5）またシミュレーション・システム自体の構造が柔軟で、部分的にミクロ／マクロなシミュレーションを行ない、（6）段階的連続シミュレーションを可能とする。

本資料では、上記目標を実現するための可変構造シミュレーション方式と疑似シミュレーション方式を中心に、DISSの概念とシステム構造について述べる。

2. DISS の目標

(1) 多様な利用目的

DISS はシステム・アナリストからユーザまでの広範囲な利用者を対象とし、分散型情報システムの OS 実現方式の検討、EDPS の最適設計のための機器構成、プロセス/データの編成などのシステム評価、アプリケーション、システムの運用評価などの各フェーズで利用する性能予測システムである。

(2) 対象モデルに対する柔軟性

一般のシミュレーションでは、対象システムの規模に比例して指定項目が増大し、シミュレーション時間の増加を招き、実際にはシミュレーションが不可能な状態となる。DISS では、少ない定義でかつ合理的時間で大規模なシステムのシミュレーションを行なう。このため対象とする分散型情報システムの規模、構造、構成要素に対して、何らの制限をかさない。また、(分散型) データベースに関して特に注意がはらわれている。

(3) 対象モデルの簡易な記述

分散型情報システムの特性要素には、ネットワーク形態、通信特性、ホストの処理能力、OS の特性、アプリケーション、プロセスの分布特性、プロセス間コミュニケーションなど非常に多くの項目があり、利用者がこれらの全てを詳細に記述することは不可能に近い。DISS ではこれらに対する簡易な記述言語を提供する。また部分ごとに種々のレベルの指定が行なえ、業務名やコンピュータ名などの抽象化レベルでも指定が行なえる。

(4) 優れたシミュレーション環境

DISS では、対象システム記述言語を用いて利用者が作成した対象システム記述をもとに、シミュレータを生成し、これを用いてシミュレーションを行なう。このシステム記述から解析用のデータ作成までの間、利用者は特殊な知識を持たなくとも DISS を使用できる。また実行マシンにおいても、特殊なコンピュータやデバイスを必要としない。

(5) 構造の柔軟性

シミュレーションでは、一般に精度とコスト(シミュレーション時間)は比例する。DISS では、シミュレーションの階層化を進め、各構成要素の独立性を高めることにより、部分的に精度の異なるシミュレーションを可能とする。

(6) 段階的連続シミュレーション

上記機能を応用し、DISS では局所的なシミュレーションから広範囲なシステムのシミュレーションへと対象システム規模の拡張性を与え、また粗いシミュレーションから詳細なシミュレーションへとシミュレーション内容の高密度化を与える。これらの段階的シミュレーションにおいて、前回までのシミュレーション結果を累積し、次のシミュレーションの動作特性のパラメータとし、次のステップでは全てを詳細にシミュレーションすることなく、詳細に行なうのと同様の結果を与える。このため、この段階的連続シミュレーションに対する各種の機能をそなえている。

3. DISS の実現方式

DISS では、前記特徴の実現を目指し、可変構造シミュレーション方式と、疑似シミュレーション方式を提案している。以下にこの二方式について説明する。

3.1 可変構造シミュレーション方式

シミュレーションによる性能予測は、まず、実システムをモデル化し、これをシミュレートすることにより性能測定を行なう。分散型情報システムにおいては、実システムが大規模となり、そのまま忠実にシミュレーションを行なうのは不可能に近く、多くの部分で簡易化が必要となる。簡易化の対象には、表1の右項に示したように多くの対象があり、利用者の目的に合ったレベル（部分的に異なるレベル）での簡易化が要請される。この要請を実現したのが可変構造シミュレーション方式である。

簡易化対象を、DISSでは表1に示す2レベルの構成及び特性に分類してシミュレーションを行なう。

表1. シミュレーション対象の簡易化レベル

レベル		簡易化の対象
ネットワーク・レベル	構成	・ネットワーク編成 ・サブネットワーク ・ホスト群 ・ホスト，CCP，回線，端末
	特性	・ハードウェア特性（CCP速度，回線速度，メモリサイズ，メモリ管理方式） ・システムソフトウェア特性（ルーティング方式，プロセス制御方式，DB管理方式）
プロセス・レベル	構成	・プロセス群 ・アプリケーションプロセス ・システムプロセス ・データベースプロセス ・チャネルプロセス
	特性	・プログラム記述 ・データ編成 ・プロセス負荷

（1）ネットワーク・レベルの構成

ネットワークの構成に従って全体を分割するレベルであり、サブネットワークやホストなどに分割する。例えば、図1に示す実システムにおいて、ホスト2.1の端末からホスト1.1のデータベースを使用するアプリケーション、プロセスの性能予測を行なう場合、全ての構成を詳細に記述する必要はない。ネットワークによる遅延を詳細にシミュレーションする場合においても、モデル化1に示す例のように、CCP 1および2以外は、サブネットワークとできる。またCCPに属する他のホストをホスト群としてまとめ、さらに、ホスト内の他の端末も端末群としてまとめられる。このように1つの部分としてまとめた単位を以後モジュールと呼ぶ。

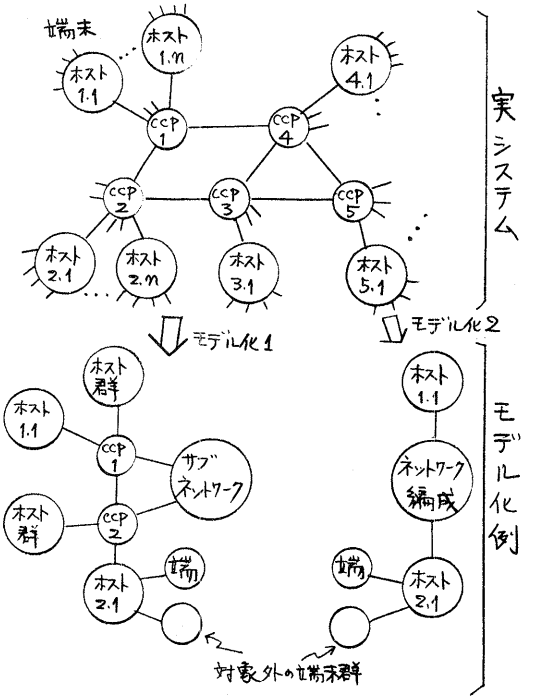

図1. ネットワーク構成のモデル化例

また、データベースの負荷に注目した場合には、モデル化2の例で示すように、ネットワーク部分を1つのモジュールとすることもある。なお、この程度の規模であれば、全てを詳細に記述してもシミュレーションできるが、更に大規模化した場合には、前述のモジュール化が必要となる。

（2）ネットワーク・レベルの特性

各モジュールにおけるモジュール内のプロセスの管理、制御、性能をシミュレーションするレベルである。ホストを記述したモジュールの場合には、CPU速度、メモリサイズと管理方式、プロセス制御方式、DBMS方式などをパラメータとして、プロセスへのCPU割合てや各種のオーバヘッドの算出を行なう。

（3）プロセス・レベル

モジュールをプロセスのレベルで分割する。このレベルの分割単位をサブモジュールと呼ぶ。サブモジュールには、プロセス群、チャネルプロセス、データベースプロセス、システムプロセス、アプリケーションプロセスがある。各サブモジュールごとに、プロセス特性に従って、アプリケーション、プログラムやサブネットワークなどの処理をシミュレートする。

これらの関係を図2に示す。

DISSでは、これらの異なるモジュールまたはサブモジュール間のインタフェースを、プロセス・レベルのインタフェースで行なう。例えば、ネットワーク構成レベルのマクロであるサブネットワークを1つのモジュールと指定した場合、自動的に1つのサブモジュールが生成される。これは、サブネットワークを詳細に指定した時と見かけ上のインタフェースを等価にするためである。しかしながらサブモジュールの内部処理方式は、その指定レベルや指定内容により大きく異なる。（次節でこの内部処理形態について述べる。）このようなレベルの異なる内部処理形態の結合を円滑に行なうため、サブモジュール間インタフェースには以下の考慮がはらわれている。

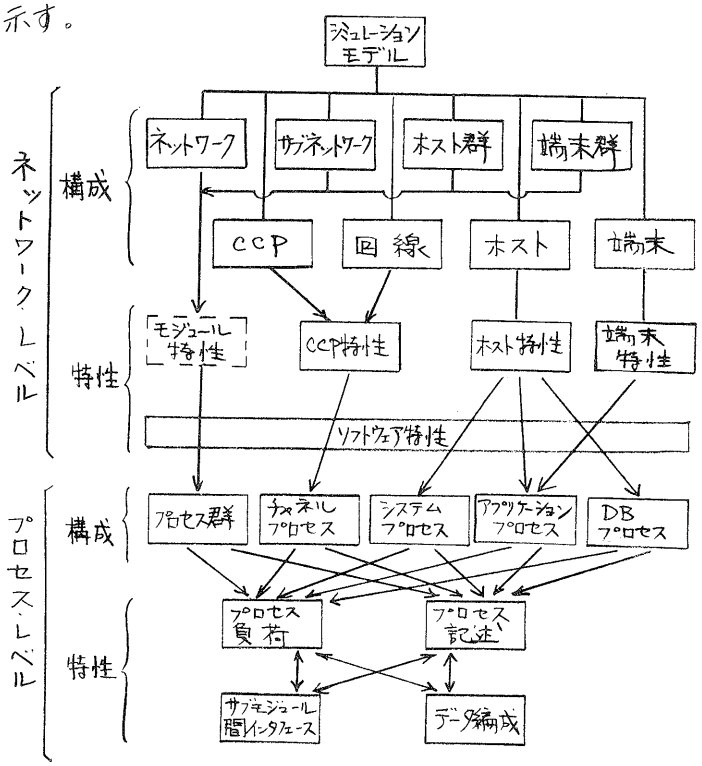

図2. シミュレーション・モデルの階層化。

（1）プロセス特性には、サブモジュール間に関係するものと、サブモジュール内の処理を表現するものがある。

（2）サブモジュール内の処理を表わすものに、プロセスの処理フローを正確に記述する外に、プロセス群の特性を1つのプロセスとして表現するプロセス負荷を示すものがある。

（3）サブモジュール間は、処理の要求側と処理側を明確に分離し、処理の要求およびその処理終了の手順によりインタフェースをとる。

（4）処理の要求側の記述は、ネットワーク、アクセスとデータベース・アクセ

スの2種のみとする。但し、それぞれの記述には、処理の内容を示したパラメータがあり、自らのサブモジュールの内部処理方式にのっとりパラメータを設定する。

（5）処理側は、自らのサブモジュールの内部処理方式にのっとり、上記パラメータを解釈・処理する。

DISSでは、上記したように、サブモジュールの内部処理レベルの相違を、パラメータの指定および解釈の所で吸収する。例えば、ネットワーク構成をマクロに指定し、これを使用するサブモジュールを詳細に記述した場合、処理要求側はネットワーク・アクセスにおいてデータ長まで指定するが、処理側はデータ長に関係なく、一定の転送時間として処理する。この逆の場合には、データ長が指定されておらず、処理側は一定のデータ長として処理する。

3.2 疑似シミュレーション方式

DISSで提案している疑似シミュレーション方式とは、前記サブモジュールの特性を、その外的特性で表現し、その外的特性を簡単な内部構成で実現する方式である。このサブモジュールを特にブラックボックスと呼ぶ。ブラックボックスの対象には、ネットワーク、ホスト、データベースおよびアプリケーション・プログラム（単数も複数でもよい）があり、またブラックボックスの内部処理方式には、利用者が指定する精度要請に対応して3種のレベルがある。

（1）制御フローレベル

実システムの制御フローに従って、詳細なシミュレーションを行なう。このレベルでは、内部パラメータを詳細に指定したサブモジュールを前もってシステムに登録しておき、利用者はそのシンボル名で利用する。このレベルは利用方式の簡易化が目的であり、シミュレーション時間は短縮しない。

（2）動作特性レベル

サブモジュール内の全プロセスを1つのプロセスと見なして、サブモジュールに対する入からの処理を実行する。段階的連続シミュレーションに有効なブラックボックスである。

（3）抽象化レベル

サブモジュールの内部処理はシミュレートせずこのサブモジュールに対する入からの処理要求に対し、ある時間経過後に応答を行なう。この処理時間には、平均値である一定時間で応答する場合と、分布した処理時間の場合とがある。これらのブラックボックスを外部から指定する場合の指定項目について表2に示す。

表2. ブラックボックスの指定項目例

	ネットワーク		ホスト	データベース	アプリケーションプログラム
	ネットワーク編成	サブネットワーク			
制御フローレベル	編成名	サブネットワーク名	ACOS700(*)	ADBS(*)	プログラム名
動作動性レベル	コマンド処理時間	コマンド処理時間 内部プロセス負荷	同左	コマンド処理時間	CPU DRアクセス ファイルI/O ネットワークI/O 比率
抽象化レベル	分布応答，一定時間応答				

(*) システムで持つキーワードを指定することも可能。

また、これらのブラックスの生成は、ユーティリティにより容易に行なえる。このユーティリティはブラックボックスの対象およびシミュレーション・レベルごとに、その核となる処理ルーチンを有し、利用者からの記述言語による指定、

または前もって シミュレーションされた測定データを集計して、ブラックボックスの外部特性をセットする。

ブラックボックスの内部処理の例を図3に示す。図3は、抽象化レベルのブラックボックスであり、応答時刻テーブルと応答時間テーブルを使用して、外部からの割込みに対し、分布した応答時間で応答する。上記ユーティリティは、この応答時間テーブルに数値をセットすればよい。

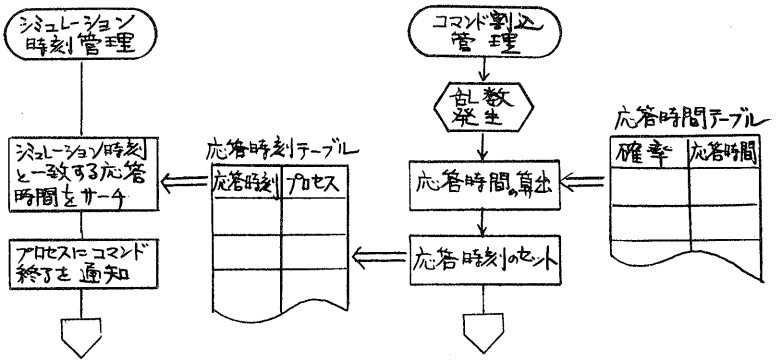

図3. ブラックボックスの内部処理例.

この疑似シミュレーション方式により、利用者は附随的部分の詳細を意識することなくシミュレーションを行なえる。またブラックボックス内の処理が単純なため、シミュレーション時間が短縮する。前もって行なった詳細なシミュレーション結果をブラックボックスのパラメータとして使用することにより、次回からは短時間で高精度な段階的連続シミュレーションも行なえる。

4. DISSモデル

前述したようにDISSは、利用者の目的に応じてシミュレーション精度、対象システムの任意性を提供し、段階的連続シミュレーションを可能としたシミュレータである。以下にDISSモデルについて述べる。

4.1 シミュレーション対象

DISSでは、スタンドアロンのTASSシステムから、データ、ソフトウェア、ハードウェアの分散した分散型情報システムまでを対象としている。この対象をシミュレータ側から分類すると図4に示す各モデルに分割できる。これらはシステムモデルとユーザモデルに大別される。システムモデルは、分散型情報システムを構成するハードウェアの特性、ネットワークの編成およびソフトウェアによる制御方式をモデル化する。一方ユーザモデルは、応用システム特性として、データベースの分散形態およびアプリケーション・プログラムなどをモデル化する。

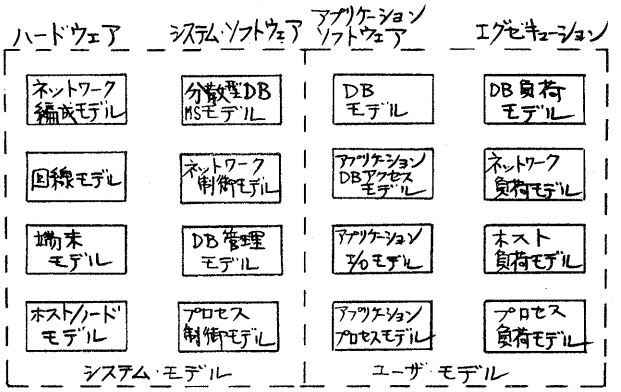

図4. DISS対象モデル

また、評価項目としては、表3に示すように、エンドユーザに容易に理解できる項目から、システム、アナリスト用の詳細な項目まであり、利用者の指定に従って要求された統計情報を表示する。

4.2 DISS 利用手順

DISS 利用者は、前記各モデルを選択、定義することにより シミュレータを生成し実行する。このシミュレーション実行は図5に示すように4つのフェーズからなる。

（1）DISS学習フェーズ

DISS利用法および標準化システムの照会を行なうフェーズである。

（2）シミュレータ定義フェーズ

対象システムの構造および特性を定義するフェーズであり、システム定義、分散型データベース定義、アプリケーション特性定義および出力定義からなる。この対象システムの定義フェーズにおいて、DISSでは利用者インタフェースとして、パラメトリックな記述言語を提供する。

（3）シミュレーション生成、実行フェーズ

前フェーズにおいて定義された各モデルを入力として、各モデル間のインタフェースをとるとともに、DISSライブラリ中の各モデルとのリンクをとり、シミュレータ・オブジェクト・イメージを生成する。このシステム生成時にDISS利用者は、任意の対象システム規模、シミュレーション精度を指定する。

シミュレーションの実行は、使用者がDISSのプロセス・コントローラを駆動して実行する。

（4）シミュレータ評価フェーズ

シミュレータ定義あるいはシミュレーション実行フェーズにおいて指定する出力項目に対するシミュレーション結果を解析するフェーズである。DISSにおいては、段階的連続シミュレ

表3. 評価項目例

		内容
実行環境		ネットワーク構成，ホスト/ハード/回線個別性能特性，DB分散形態/管理方式，ネットワーク制御方式，ホストの制御方式，アプリケーション特性
性能	スループット	システム・スループット，個別ホスト・スループット
	応答時間	端末応答時間，プロセス間通信応答時間，
	資源使用率	CPU，周辺装置，回線，CCP，システムプログラム使用率
	詳細情報	アプリケーション・プログラムのチェックポイント間通過時間，メッセージ処理時間，メッセージ待行列の統計，資源使用時の待時間，DBマクロ処理時間（最大値，最小値，平均，標準偏差，分布）
コスト		トータルコスト，アプリケーションコスト

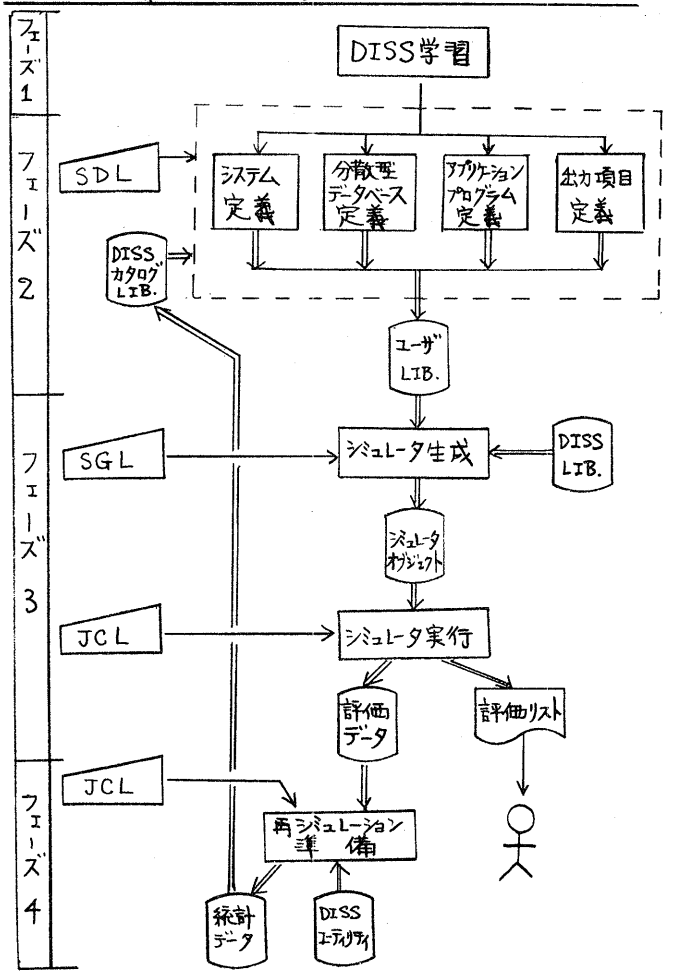

図5. DISS 実行モデル

ションとして、評価項目の検討後、DITSユーティリティを利用して部分的に詳細な、あるいはシミュレーション規模の拡大を計る再シミュレーションが行なえる。

4.3 DITSによるモデル記述

シミュレータの定義は表4に示すように、システム定義部、分散型データベース定義部、プログラム・モデル定義部、出力定義部の4定義部からなる。

表4に示したように、ホスト、ネットワーク、分散型データベースおよびプログラムの各モデルでは、使い易さを目的とし、図6に示す3クラスの指定方式を提供する。

クラス1では記述対象を1つの仮想システムとして使用者は考え、DITS側で標準特性を提供する。

クラス2では記述対象を機能別要素として考え、要素別に特性記述を行う。

クラス3では記述対象の詳特性、動特性制御方式を詳細に記述し、高精度シミュレーションを行なう。

このレベルにおける指定の例として、プログラム記述の主要コマンド一覧を表5に示す。

また、DITSによるシミュレータ定義例を図7に示す。ここでは特に図7に示すモデル化1のシステム定義部をとり上げて記述している。

表4. モデル記述言語

			主要記述項目
システム定義部	ホストモデル記述	マシンタイプ指定	超大型，大型，中型，小型，超小型
		機種指定	マシン名，メモリサイズ，OSタイプ，DB名 端末名，
		特性記述	CPU速度，メモリサイズ，OS領域サイズ メモリ管理方式，ジョブ管理方式，プログラム管理方式，
	ネットワークモデル記述	ネットワークタイプ指定	結合形態，性能特性
		ネットワーク構成記述	ネットワークノード名，結合形態，回線特性， 負荷特性，
		ネットワーク制御記述	ネットワークノード名，ネットワークノード特性， 結合形態，回線特性，ネットワーク制御方式，負荷特性
	端末ノード記述		端末名，入出力特性，接続ホスト名， インテリジェンス規模，
分散型DB定義部	DBモデル記述	応用システムタイプ指定	財務，販売，人事，生産業務 など
		DB特性記述	DBシステム名，性能特性
		DB構造記述	DBサイズ，DB格納内構造，論理構造、 性能特性，デバイス特性
	データアドミニストレーション記述		ディスクリプタ管理方式（集中型，階層型，分散型） 性能特性，共有データ特性，DBMS間特性 DBMS間通信方式，障害発生特性
アプリケーションプログラムモデル定義部	業務タイプ指定		情報検索，伝票発行，元帳更新，科学計算
	動特性記述		CPU、DB，ファイルI/O，ネットワークI/O，端末I/Oの 比率，動作特性
	プログラム記述		表5参照
出力定義部			表3参照，

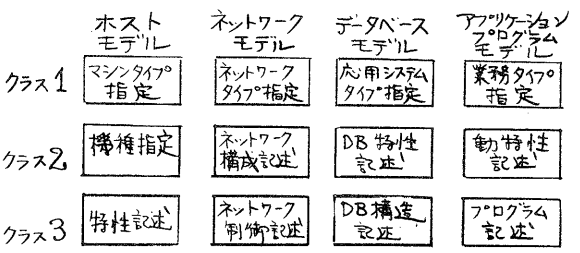

図6. モデル記述の階層

```
SYSTEM DEFINITION DIVISION .
HOST NODE DESCRIPTION SECTION .
    HOST-1 IS LARGE COMPUTER .
    HOST-2 IS NEAC SYSTEM100 ,
      MEMORY SIZE # 32K ,
      OS TYPE # MONITOR 4M ,
      MULTIPLICITY # 7 .
    HOSTS-1 ,
      SIMULATION LEVEL IS CLASS-1 .
    HOSTS-2 ,
      SIMULATION LEVEL IS CLASS-2 .
NETWORK DESCRIPTION SECTION .
  NETWORK NODE DESCRIPTION .
    CCP-1 IS NEAC 3200/70 .
    CCP-2 ,
      CPU SPEED # 1 N-SEC ,
      MEMORY SIZE # 64K ,
        :
      MEMORY MANAGEMENT # RESIDENT .
    SUBNETWORK-1 ,
      SIMULATION LEVEL IS CLASS-1 .
  LINE DESCRIPTION .
    LINE-1 IS 4800 BPS ,
          CONNECTS HOST-1 TO CCP-1 .
    LINE-2 IS 1200 BPS ,
          CONNECTS HOST-2 TO CCP-2 .
    LINE-3 IS 4800 BPS ,
          CONNECTS CCP-1 TO SUBNETWORK-1 .
        :
  NETWORK CONTROL DESCRIPTION .
    ALL CONTROL TYPES ARE ARPA .
  LOAD DESCRIPTION .
    SUBNETWORK-1 LOAD TO CCP-1 IS TABLE-1 .
      HOSTS-1 LOAD TO CCP-1 IS 50 MPS .
TERMINAL DESCRIPTION SECTION .
    HOST-2 TERMINAL .
      TERMINAL-1 IS DISPLAY .
        :
    HOST-1 TERMINAL .
      :
```

図7. DISS記述例.

5. DISSの構成

DISSは図8に示すように、6サブシステムから構成される。

（1）アプリケーションプログラム・トランスレータ

本サブシステムは、利用者の指定に従ってアプリケーション・プログラムを解析し、プログラムステートメントに変換する。

（2）システム記述言語トランスレータ

利用者の指定したシステム記述言語を解析し、モジュール/サブモジュールを作成する。

（3）システム・ジェネレータ

各モジュール/サブモジュールをリンクし、シミュレータを生成する。

（4）シミュレータ

擬シミュレーションの時間軸でシミュレーションを実行し、性能予測のための中間データを作成する。

表5. プログラム記述ステートメント

命令	形式	パラメータ
CPU実行	PROC n	n: 実行ステップ数
ファイルI/O	FIO d,c,l	d: デバイスタイプ, c: 入出力モード, l: データ長
DBアクセス	DBA n,c,l	n: DB名, c: コマンドタイプ, l: レコード長
ネットワークI/O	NW n,c,l	n: ホスト名, c: コマンドタイプ, l: メッセージ長
端末I/O	TIO n,c,l	n: 端末名, c: 入出力モード, l: データ長
無条件分岐	GOTO S	S: レーベル
確率的分岐	PRGOTO P_1, S_1 P_2, S_2 ⋮	P_i: 確率 S_i: レーベル

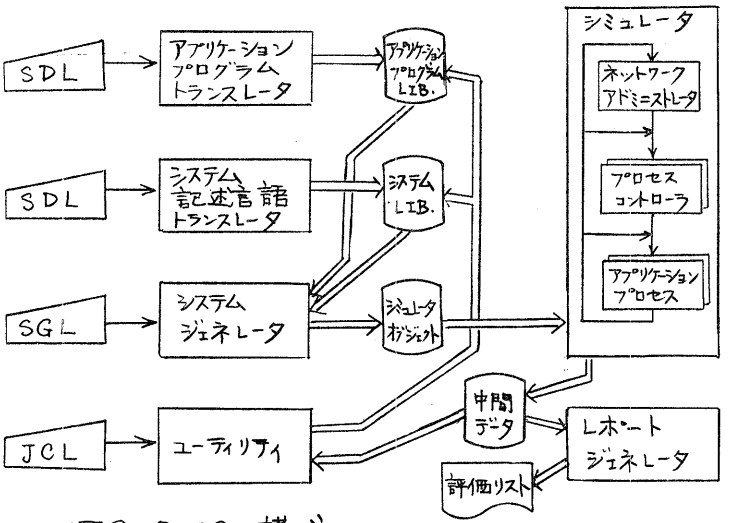

図8. DISSの構成.

（5）レポート・ジェネレータ
シミュレータが作成した中間データを集計し最終評価データを出力する。
（6）ユーティリティ
学習フェーズにおける利用者のガイダンス機能と、評価フェーズにおける段階的連続シミュレーションの準備をする機能とがある。

6、おわりに

分散型情報システムのシミュレータとしては、ユーザの目的に即した利用ができ、シミュレーション時間の短縮を計ることが重要な課題である。これを実現するために、本資料では、異なる処理レベルのモジュール／サブモジュールのリンクを行なう可変構造シミュレーション方式と、（サブ）モジュールの内部処理を疑似的に行なう疑似シミュレーション方式を提案した。これらの方式は、利用者に対し簡易なインタフェースを与え、またシミュレーション時間の短縮を可能とする。特に、前もって行なったシミュレーション結果を利用して、効率的にシミュレーションを行なう段階的連続シミュレーション機能は、利用者にとって有益な機能である。

全ての対象を詳細に記述した時のDISSの精度目標を±10%としている。DISSが最も使用されると予想されるのは、段階的連続シミュレーションの場合であり、この時の精度が問題となる。正確な数値はDISS完成後の検証例をまつ必要があるが、動作特性レベルのブラックボックスであれば±20%，抽象化レベルであれば±50%以内におさまると予想している。またシステムモデル（OS制御方式，ルーティング方式など）のブラックボックス化であれば、更に精度の向上が望め、ユーザに提供される特殊化されたDISSは、分散型情報システムの予測システムとして充分な効果が期待される。

参考文献

[1] G.M. Booth: Distributed Information System, Proc. NCC, pp787-794, 1976
[2] F.G. Withington: Future Computer Technology, ACM SIGBDP, Vol 17 No4 pp7-14, 1976
[3] T.N. Pyke: Assuring User Service Quality in a Distributed Computer Network, Compcon-spring pp73-76, 1976
[4] G.H. Wedberg & L.W. Hauschild: The General Electric Network Monitor System, Information Processing 74, pp24-28 1974
[5] D.E Morgan etal: A Performance Measurement System for Computer Networks Information Processing 74 pp29-33, 1974
[6] W.W Chu: Performance of File Directory Systems for Databases in Star and Distributed Networks, NCC-76, pp577-587, 1976
[7] K.M. Chandy & J.E. Hewes: File Allocation in Distributed System, Proc. of the Int. Symp. Computer Performance Modeling Measurement and Evaluation, pp10-13, 1976
[8] M. Irland: Simulation of CIGALE 1974, Fourth Data Communication Symposium pp5・13-5・19 1975
[9] 水谷：コンピュータ・システム・シミュレーション，情処システム性能評価研究会資料，SE14-1, 1976-3月
[10] 金子，西原，阪田：エキゾーザシステムにおける性能予測システム，情処 Vol.18, No6, 1977（予定）