AN APPROACH TO FLEXIBLE MODELING AND SIMULATION FOR CONTROL PROCESSOR ANALYSIS

O. G. SOTO, J. A. G. HIGUERA, C. D. BERZOSA, L. M. MIGUEZ
Laboratorios ITT de Standard Eléctrica, S. A., Madrid, Spain

ABSTRACT

A methodology for the analysis of stored program control systems by simulation is presented. The modeling is based on the representation of traffic flow and control flow. The traffic flow modeling allows an equivalent system to be built by establishing a correspondence relation between the system activities and their equivalent tasks in the model. These tasks are allocated in a parameterized form to represent different call types and task allocations. The control flow modeling establishes a model structure associated with the hierarchical organization of the equivalent tasks by modular strata. This provides a high reusability of the simulation for many systems. The models obtained in this way may be easily adapted to different details of traffic and system representation. This is very important for the efficient traffic analysis of modern control structures. Examples of the application of the method to centralized and distributed SPC systems are presented.

1. INTRODUCTION

Traffic analysis of stored program control switching systems are performed in order to attain four basic objectives: (ref. 1).

a) Estimate call handling capacity
b) Determine grade of service
c) Optimize internal traffic flow, avoiding potential bottlenecks.
d) Analyze system behavior under overload conditions (either due to high offered load or failures) and identify the optimum controls.

1.1 Justification

The time-true simulation technique (ref. 1) is a necessary tool for the following reasons:

- The influence of new technologies on the architecture of control structures increases the complexity of function allocation and of the communication flow among system components. Normally, complexity cannot be represented by sufficiently simple statistical assumptions to be treated by means of analytical methods. It is necessary to introduce a greater degree of detail, and a greater number of measured performance parameters and associated statistical characteristics. Therefore the simulation technique must be used.

- Adaptive modeling ability is necessary to reflect the frequent and easy system changes in hardware and software that, in general, are simpler to implement in a simulation model than analytically.

- There is a need to test the transient behavior of the system and the different traffic flows before an exchange is implemented.

1.2 Previous models used to analyze SPC systems

For the simplex and duplex control structures, three models with different degrees of approximation have been developed in the past:

A) Load-type model
Assumptions:
- Steady state for all traffic sources.

- Calls themselves are not simulated, and therefore the dependencies and feedback effects between different processes are ignored.
- Process time is statistically generated by a probability distribution.
- Priority structure for levels and processes is represented in detail.
(ref. 1).

B) Subcall-type model
Assumptions:
- Steady state traffic conditions.
- Call attempts are broken down into separate independent "subcalls", normally preselection, selection, answer and release.
- Dialing and signaling phases are treated as independent stochastic processes.
(ref. 2).

C) Call-type model
Assumptions:
- All phases and their dependence are represented in detail.
- The traffic-dependent memory requirements are taken into account.
- Valid for transient and overload situations.

The application of these models has to be done for each particular system and isolated case, without reusability among them. Furthermore, their flexibility to follow system changes was low.

The evolution of the system design techniques (using top-down strategy, modularity, etc.), the expensive cost of using the preceding models, and the flexibility of control structures, require a modeling technique able to follow the system design and to cover the objectives needed throughout the system life.

1.3 Present approach

The approach presented in this paper consists in the development of a flexible method that permits model adaptability in two aspects: traffic flow and system detail. The bases of the approach are: (1) a continuous traffic modeling that allows an "equivalent" system to be built, and (2) a modular system modeling that facilitates the translation of the model to a simulation program.

The continuous traffic modeling is solved by means of a parameterized model that establishes the relationship between the telephone activities and their equivalent model activities with the degree of detail necessary in each phase of application (ref. 5). The generation of model activities is specified in a parameterized form by data external to the program. This representation permits an easy analysis of the system traffic behavior for different call-types and for centralized and distributed systems.

The modular system modeling is accomplished by means of a correspondence between the activities defined by traffic modeling and the model resources which handle the activities...
so that they are executed according to their service discipline. The model resources structure permits modular implementation in a simulation program and, in most of the cases, parameterized handling of activities. This approach achieves an important saving in programming effort and a quick response to the design needs.

2. BRIEF CONTROL STRUCTURE DESCRIPTION

The control structure in an SFC system has many functions to execute for call handling, system testing, system management, etc. The main contribution to the traffic behaviour is produced by the call handling, with functions like call detection and acceptance, digit analysis, route translation, subscriber class analysis, path search, etc. These functions are executed for each call by means of code pieces, each of which executes an activity.

2.1 Descriptive activity

In order to unify the concept of activity for modeling purposes from the multiplicity of code implementations in different systems, we consider the basic "descriptive activity" as the set of operations with an origin, a destination and with external interfaces only through the origin and destination. Note that the description of a system by basic activities can be made at different levels of detail like instruction stream, process, processor, etc. These levels of detail will be called strata. This concept matches very well with the description of a modern structured system. As example we have the ITT Digital Communication System 12 which is implemented by *Finite Message Machines*. The occurrence of a Finite Message Machine is a basic descriptive activity.

For a non-structured system, the same idea of basic descriptive activity can be abstracted. The guarantee of existence is based on the Bohm & Jacopini (ref. 4) theorem which demonstrates the existence of an equivalent structured program for any existing one.

2.2 Control structures

The basic activities are organized in the system by modules and implemented in one or more processors. Due to the fact that they consume time, in the same resource, a competition is produced by their execution request. In order to allocate the activity execution on each processor and fulfill the grade of service requirements, several processor configurations are implemented. The history of the design shows a multiplicity of control structures. The ones that we have analyzed can be summarized as follows:

a) Monoprocessing system with possible duplication of the processor for reliability reasons (fig. 1). 

b) Multiprocessing system with load and/or function sharing between processors and an overdimensioning for reliability reasons (fig. 2).

c) Hierarchically distributed system with load and/or function sharing between processors and communication in horizontal and vertical directions (fig. 3).

2.3 Service discipline

The way in which the activities are implemented within each processor depends on the specific urgency or need for each activity, which implies the assignment of priorities between different activity types and also between calls arriving at the same activity. The optimum allocation of the activities has great significance. It is important to have efficient use of the processor and to allow adequate treatment for each call type, as well as to be able to implement the system and network management actions as necessary (e.g., give priority to a specified call type). The following are the source disciplines that we have found most common in our studies:

- FIFO, LIFO, random order, round-robin, and priority by class for the basic activities.
- Head of the line, interrupt priority, cyclic for activity group type.

Of course, in a complex system, mixes of those priorities coexist, giving great interaction between activities and call service. The most common case that we have treated is the multilevel interrupt priority mixed with head-of-the-line priority for activity groups. For simplicity the following terminology and structure will be referred to this case.

3. BASIC DEFINITIONS

In order to provide a homogeneous set of concepts and terminology for the modeling of the control structure, the following definitions are given. Also, new terms used in this paper are introduced.

3.1 System definitions

**Descriptive Activity**: A set of time-consuming operations with two external interfaces only through their origin and destination.

**Descriptive Process**: A set of descriptive activities that are executed by the system. A set competes with equal priority with other sets for a common resource.

**Processor Level**: A set of descriptive processes with the same interrupt priority.

**Processor**: A set of levels which compete for a common resource.

**Interprocessor Communication**: The logic associated with the activities which relate different processors.

3.2 Model definitions

**Equivalent Traffic Task (ETT)**: A basic entity in the model with a defined correspondence relation to the descriptive activity in the real system. The correspondence relationship preserves the execution time, significant state changes, and predecessor and successor activities.

The correspondence relationship is defined in Section 5.

**Equivalent Traffic Process (ETP)**: The set of Equivalent Traffic Tasks obtained from a given descriptive process by the correspondence relationship.

**Equivalent Traffic Level (ETL)**: The set of Equivalent Traffic Processes obtained from the processor level by the correspondence relationship.

---

*SOTO / HIGUERA / BERZOSA / MIGUEZ-2*  

---

Fig. 1

Fig. 2

---

*ITT trade mark*
Equivalent Processor (EP): The set of Equivalent Traffic Levels obtained from the processor by the correspondence relationship.

Model Resource: The entity associated to each Equivalent Traffic Task, Equivalent Traffic Process, etc. which handles their execution according to the service discipline. It consists of:
- A logic: Model resource handler
- A finite set of variables and corresponding values: Model resource state.

Sequence: A set of Equivalent Traffic Tasks ordered by the same succession relations of the telephone call activities.

Overhead: It is defined by the Equivalent Processor occupancy time when there are no living calls in the Equivalent Processor.

4. APPROACH FOLLOWED

The obtaining of the traffic objectives mentioned in section 1 by a unique single model implies the use of the most complex one to fulfill all of them. This is impractical due to cost, unavailability of information during the system development phase and the frequent changes that occur during development. The use of several independent models for achieving these objectives does not allow sufficient speed to reflect system changes and is costly in manpower due to the low degree of reusability. The use of an independent model creates more problems when analyzing new control structures (mainly with a degree of distribution) which have the following characteristics:

- There is flexibility for allocating activities to the different modules.
- During design phase, system changes become easier and quicker to implement due to the degree of modularity.
- The top-down design technique implies an evolution of system development by hierarchical levels of detail.
- Design phase analysis is more important in SPC switching systems than in earlier types.

To fulfill the traffic objectives in section 1 in those control structures that are characterized by the above mentioned features, the following key points are considered:

- To cover the complete system development phase.
- To provide quick response to system changes and evaluation of several system design alternatives.
- To implement the simulation models with high degree of reusability to maintain low cost and adequate adaptation speed.
- To ensure close communication between traffic engineers and design teams through the model descriptive language.

In order to cover all the objectives, the following methodology of flexible modeling has been developed.

4.1 Methodology

- The system control structure is abstracted in Equivalent Traffic Tasks which are obtained by a defined correspondence relationship. The ETTs are subject to two membership and ordering relations, one due to the telephone flow and the other due to the control flow. (Fig.4 gives an example containing 4 strata.)
- A predecessor and successor relation is built on the ETT which is used to preserve the original flow and membership relations of the telephone call and to represent the traffic generation by means of the sequence. (The traffic flow modeling is described in section 5.)
- For the ETTs, a resource structure is created which comprises the system strata represented (e.g.: ETP, ETL, EP, etc.), and which includes a correspondence used to define the resource which handles the specified set of ETTs. This resource structure preserves the control membership relation, and it decides the service order of execution of the ETTs by the confluence of the two flows. (Details of system flow modeling are described in section 6.)

In the authors’ opinion, the system and traffic modeling must be structured in such a way that the specific subsets of the previously mentioned objectives can be fulfilled.

The corresponding models for each subset would be adaptable and reusable to accomplish the above mentioned key points.

To attain the traffic objectives, it is necessary to take some parameter measurements in the simulation. The number and accuracy of the results depends on the group of selected objectives. As an example, table 1 shows the necessary results as a function of three specific traffic objectives:

- Maximum Capacity Determination, as defined in reference 1. This objective is very important at early design stages.
- Knowledge of system behaviour under normal traffic conditions. This includes knowing the Nominal Capacity of a system already designed.
- Knowledge of system behaviour under Abnormal Traffic Conditions which include the influence of overload situations, effects of different overload control strategies, the influence of subsystem failures, etc.

<table>
<thead>
<tr>
<th>TABLE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULTS</td>
</tr>
<tr>
<td>Aver activ.</td>
</tr>
<tr>
<td>Aver. process delay</td>
</tr>
<tr>
<td>Number of interrupts</td>
</tr>
<tr>
<td>Number of offered BHCA</td>
</tr>
<tr>
<td>GOS delays (Aver.)</td>
</tr>
<tr>
<td>Specific Resource Waiting &amp; Response Time</td>
</tr>
<tr>
<td>activ.</td>
</tr>
<tr>
<td>queue length distr.</td>
</tr>
<tr>
<td>GOS delay distr.</td>
</tr>
<tr>
<td>Number of timed-out activ.</td>
</tr>
<tr>
<td>Distr. process delay</td>
</tr>
<tr>
<td>Distr. process real cycle</td>
</tr>
<tr>
<td>Number of accepted BHCA</td>
</tr>
<tr>
<td>Overload reaction &amp; absorption time</td>
</tr>
<tr>
<td>Performance parameters in critical periods</td>
</tr>
<tr>
<td>Overload indicators</td>
</tr>
</tbody>
</table>

M and H indicate a medium and high accuracy degree respectively.

The quantity of results, parameters, and their accuracy determines the traffic degree of representation and the system degree of representation. They are modeled as explained in section 5 and 6 respectively.
5. TRAFFIC FLOW MODELING

5.1 Traffic Assumptions

To model external events due to either the subscriber or the adjacent exchanges, the following main traffic behaviour characteristics have to be considered:

- Arrival law
- Successive call event interarrival
- Call-type distribution
- Call holding times

These traffic features have been treated for the modeling under the following traffic assumptions:

a) The external events which are statistically independent, are generated as Markovian. Generally, Poisson arrival is used. In some cases (e.g. for incoming traffic in a toll exchange, abnormal system condition, etc.) a general distribution with a predefined peakness is used.

b) The external events which are statistically interdependent like digit appearance, premature release, etc., are generated by the predecessor external event after an external dependent time which is created by a general distribution (erlang-k in most cases) with predefined average and peakness.

c) The call-type distribution, defined in ref.1, is used to generate the specific traffic to each type. This term includes the call completion degree commonly called call mix (ref.3) and the call pattern (which includes signaling type, origin and destination, facilities involved in the call, etc.).

d) The holding times of devices are generated by a general distribution for each one. The most frequent distribution used is the erlang-k (from k=1 exponential to k=0 constant). Also, a specific histogram is used when the resource behaviour is well known in detail. For the ETT holding time, the most common case is a constant value associated to a code piece. Nevertheless, there are some traffic dependent cases (e.g. path search time) which follow a given distribution.

e) The activities which produce a completely negligible control processor occupancy are not considered in the call flow representation. Anyhow, the total resource occupancy produced may be represented in a global manner by means of a statistical distribution.

5.2 Call Flow Representation

The application of assumption "e" above, drastically reduces the number of descriptive activities to be considered. The representation of the remaining activities is made in a way that preserves the same time ordering as the original activities by the definition of a telephone membership and ordering relations.

5.2.1 ETT Correspondence Relation

The flow modeling is based on the creation of the ETTs that are the basic entities, abstracted from the descriptive activities by means of the following correspondence relation.

A descriptive activity produces as many ETTs as trios of the following basic parameters can be found: (1) execution time, (2) predecessor activity identifier, and (3) successor activity identifier.

The use of this correspondence relation allows the handling of the telephone flow externally to the simulation program. Thus a parameterization is obtained, which simplifies and speeds up the application to many different call types and degrees of correlation.

Figure 5 gives an example of activity to ETT correspondence relation for a call release of a complete call (A1-A2-A3-A4) and a premature release (A5-A3-A4). It can be observed that activities A3-A4 are converted to two ETTs each.
5.2.2 ETT associated parameters

In addition to the trio of basic parameters, other parameters are associated to the ETT to represent telephone and system ordering.

5.2.2.1 Telephone membership and ordering relation-associated parameters

In order to preserve the original telephone membership and ordering of the activities, the following two entities have been developed to organize the ETTS:

A **sequence** is a set of consecutive ETTS with time dependence among them. The criteria employed to form the specific sequence depend on the degree of time correlation needed for each model and will be explained in the 5.2.4 section, below.

The **model call** is the ordered set of all ETTS corresponding to a telephone call type. Also, it can be viewed as an ordered set of all the sequences associated with the same telephone call type.

The parameters associated to the ETT derived from the telephone membership and ordering relations are:

- \( n \): relative number of the ETT within its sequence
- \( s \): sequence identifier within the model call
- \( m \): model call type
- \( c \): relative number of the ETT within its sequence
- \( r_j \): inter-request time, associated with the ETT, which is the elapsed time to require the next ETT due to the telephone ordering. (See fig. 6.)

5.2.2.2 System membership relation-ETT associated parameters

The ETTS are subject to a second membership relation which corresponds to the system structure. The fact of an ETT being executed at the time axes" the execution of a given ETT already belongs to a process, the process to a level, etc. defines the system membership relation. This structure "fixes in the time axes" the execution of a given ETT already requested. The associated parameters of the ETT derived from the system structure are:

- \( k \): relative number of the ETT within the process to which it belongs.
- \( j \): process identifier within its level
- \( i \): level identifier within its processor
- \( p \): processor identifier within the system
- \( \text{ET}_{pijk} \): time between the entries of two ETTS to which the consecutive ETTS belong. Figure 6 graphically represents both RT and ET parameters.

5.2.3 Types of generation

To represent the flow of the system, ETT copies have to be generated. Two types of ETT copy generation are considered:

**Type 1**: Traffic generators create the first or "header" ETT copy of each independent sequence with the corresponding interarrival time of the sequence, using Poissonian arrival law. The call-type distribution and the offered traffic determine the interarrival rates for each sequence "header" copy.

**Type 2**: When each ETT copy is executed, its dependent ETT copy after \( \text{RT}_{csn} \) time units is created. Thus, beginning with sequence "header" copy, each ETT copy execution generates its successor. In this way complete correlation between ETTS of the same sequence is maintained. Each dependent sequence is generated by the previous one.

5.2.4 Different degrees of correlation. Sequence building examples

The selection of the sequences \( S_{csn} \) for all the possible subsets of the model call can be made in various ways according to the objectives of the simulation. Three typical examples of the different desired degrees of correlation are explained below:

a) **No Flow (NF)**

Each sequence \( S_{csn} \) is associated with one and only one ETT and is considered as independent. A traffic simulation is obtained with no correlation between consecutive ETTS. The NF representation is valid only for analysis of the steady state and normal traffic conditions.

b) **Partial Flow (PF)**

In the PF, a limited correlation of ETTS is maintained excluding the dependence between the ETTS with a long interarrival time (low correlation). In the PF, two different degrees of correlation are given as typical cases:

i) In the call phases degree of representation, the independent sequences are associated with different call phases, e.g. preselection, answer, release.

ii) In the degree of representation by ETT burst, the comparison between the interrequest and interentry times (\( \text{RT}_{csn} \) and \( \text{ET}_{pijk} \)) is used as the criterion to select the sequence. (Notice that \( c,s,n \) and \( p,i,j,k \) are parameters which belong to the same ETT under study). This criterion is applied to every pair of consecutive ETTS from the telephone ordering point of view and can be summarized as follows:

- i) When \( \text{RT}_{csn} > \text{ET}_{pijk} \), independence between them, can be assumed, and the second ETT will be the "header" of the new independent sequence. For example, we have the end of selection task and subscriber answer which are considered as independent.
- ii) When \( \text{RT}_{csn} \leq \text{ET}_{pijk} \), they are considered as dependent ETTS belonging to the same sequence.
- iii) When \( \text{RT}_{csn} > \text{ET}_{pijk} \), they are considered as dependent ETTS belonging to consecutive dependent sequences.

This type of representation is a very economical one because the removal of long-term dependence considerably reduces the simulation run time; however, it maintains the strong correlations of the ETTS that are close in time.

c) **Total Flow (TF)**

In this representation, total correlation between ETTS is maintained except for the case of a complete call release in which the correlation is maintained only through the updating of the global call release generation rate. The TF gives greater accuracy than the previous representations and allows the analysis of the system under transient and abnormal conditions like system failure, overload, etc.

It has to be borne in mind that due to the ETT correspondence defined, the change from one degree of correlation to another can be externally supplied to the simulation program. Hence, the feature of traffic modeling parameterization.

---

**Fig. 6** INTER-TIMES FOR BOTH ORDERING RELATIONS

ITC-9

SOTO / HIGUERA / BERZOSA / MIGUEZ-5
6. SYSTEM FLOW MODELING

In order to build the system model, the system characteristics that influence the results to be obtained and their accuracy must be selected. Thus the degree of detail to be considered is defined. These system characteristics are reflected in the model state and its changes. The model state is defined by the values taken with a finite set of variables, which are associated to each hierarchical stratum of ETTs.

The basic entity that simulates changes in the system state is the ETT. From the system flow point of view, it is necessary to note two important factors: (1) it takes time, and (2) it produces (potentially) a significant change in the state of the model. An ETT is bounded by two events, the beginning and the end of the task. Changes in the system taking place during the tasks are updated in the events. This way of simulating implies that no change occurs in the model system state until the completion of an ETT.

In the system, the descriptive activities are executed according to a certain policy applying to groups of them. These groups form the system structure. The ETTs defined from the traffic model are grouped by their membership relations which reflect in the model the system structure, through the following hierarchical strata: Equivalent Traffic Process, Equivalent Traffic Level, Equivalent Processor (defined in section 3.). Associated with each of these hierarchical strata, a model resource is created which handles the stratum components and orders their execution according to their service discipline.

Each model resource must be considered under two aspects: (1) the resource state, and (2) the resource handler logic. All the resource states and the values needed to take the statistics form the model state.

6.1 ETT Resource

The variables and logic which may be associated to each resource are as follows.

6.1.1 ETT Resource State (ETTR State)

The most typical variables for the ETTR state are:

- ETTR overhead time
- ETTR execution ETT counter
- ETTR queue associated variables (number, first, last, etc.)
- ETTR statistical parameters (i.e. arrival laws, mean, STD, etc.)

Depending on the service discipline to be modeled in a particular ETTR, only some of these variables are needed. Therefore the ETTR indicator is defined as the set of variables necessary to model a predetermined service discipline. Although, the number of indicator types seems to be very large, our simulation experience has shown that most of the cases may be covered by a small number of types: overhead, scanning, timing, batch, individualized - ETT and statistical.

6.1.2 ETT Resource Handler (ETTR Handler)

Each ETTR has two associated jobs which handle the corresponding indicator. The first calculates the execution time of the ETT in order to schedule the ETT's end event. The second executes the state changes that correspond to the ETT handled by the ETTR. The changes are:

- To destroy executed ETT copies by updating the indicator of the simulated ETTR.
- To request the execution of the next ETTR copies that follow in the traffic call sequence, updating the next ETTR indicator.
- To communicate to higher resources the new state of the ETTR.

The limited number of indicator types means that the number of jobs necessary to cover most of the cases is also low.

6.2 ETP, ETL, EP Resources

6.2.1 State

Each model resource must handle the components of the stratum with which it is associated. To treat them it is necessary to have several variables that reflect the state of the resource members. For instance, the processor must have the state of its levels; the level, the state of its processes, etc. The states that must be reflected for each member are:

- Execution priority, including several parameters to handle it.
- Requested, when the member must be executed because its entry has been reached or an interruption asking its execution has been generated.
- Running, when the control unit is occupied in carrying out the member.
- Interrupted, when its execution has been stopped to execute another member with higher interrupt priority.

6.2.2 Resource Handler

The handler of each resource has one associated job which handles the previous variables and decides which member of the stratum is the next to be executed.

6.3 Equivalent System Resource (ES Resource)

6.3.1 E. System Resource State

For each Equivalent Processor Resource belonging to the system, the following variables are used:

- Processor occupied condition (busy, free)
- Processor identifier
- Processor functional condition (on, off, stand by, function shared assignment)

6.3.2 E. System Resource Handler

It has the following functions:

- Handle the processor priority to solve potential competition.
- Change the function shared and/or load sharing in abnormal conditions.
- Assign the ETT among different resources when several processors may handle the ETT according to a specific rule.

Fig.7 SYSTEM SIMULATION FLOW

The base of the resource structure is the ETT Resource which is always represented, whatever the degree of detail at which the model reproduces the system. The upper stratum, ES Resource, is always necessary too. According to the objectives to be fulfilled (see section 4), the ETTs are defined on the lowest stratum to be represented. Also the number of strata and which of them are represented must be selected. For instance:

- When the ETT is defined on the Processor Stratum, only the ES Resource and ETT Resource exist.
- When the ETT is defined on the Processes Stratum, the ES Resource, EP Resource, ETL Resource, and ETT Resource are obtained.
The confluence of the two flows, traffic and system, gives the actual execution in a specific point-of-time of the ETTs belonging to each call. The results to be obtained for the objectives identified in the section 4 impose a traffic flow selection following section 5. The confluence with the system representation (section 6) and the degree of accuracy needed gives a multiplicity of models, the most common of which are shown in the table 2.

### Table 1

<table>
<thead>
<tr>
<th>Traffic flow degree of representation</th>
<th>NO FLOW</th>
<th>PARTIAL FLOW (PP)</th>
<th>TOTAL FLOW (TF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM R.</td>
<td>N/A</td>
<td>M-D</td>
<td>M-D</td>
</tr>
<tr>
<td>P'SOR. R.</td>
<td>M-D</td>
<td>M-D</td>
<td>M-D</td>
</tr>
<tr>
<td>LEVEL R.</td>
<td>M-D</td>
<td>M-D</td>
<td>M-D</td>
</tr>
<tr>
<td>PROCESS R.</td>
<td>M-D</td>
<td>M-D</td>
<td>M-D</td>
</tr>
<tr>
<td>ETTR-TYPE</td>
<td>N/A</td>
<td>M-D</td>
<td>M-D</td>
</tr>
<tr>
<td>ETT/TYPE</td>
<td>N/A</td>
<td>M-D</td>
<td>M-D</td>
</tr>
</tbody>
</table>

The title (Equal Type) or (Different Type) for the ETT Resource refers to the representation of all the ETT disciplines by the worst case as a simplification of the case in which the specific differences among service disciplines are simulated.

The note "not applicable" (N/A) indicates that this particular combination on both details of representation does not allow the implementation of either of the two flows (e.g. the No Flow case does not allow the implementation of the service discipline required for ETTs with different service strategies.)

The note "D" means that the particular model is appropriate for some processor of a distributed configuration; "M" means that the model is appropriate for a mono or multiprocessing configuration.

The case which have an asterisk (*) in the corresponding position can be modeled but there is no interest in their simulation, either due to the simplicity of the interaction or the low number of results provided. Some of these are typical cases to be approached by analytical tools (e.g. the PROCESS R.- NO FLOW case may be analyzed by the M/G/1 model.)

The rest of the cases are the ones in which the simulation models may be useful depending on traffic objectives and the complexity of the processor configuration.

From the matrix of models needed for different objectives and processor configurations, some typical cases are identified as minimal models for each of the three main objectives given in the section 4. The term minimal refers to the simplest and most economical one in both degrees of representation that can be used for the set of objectives. Of course, any other model with higher degree of representation could be used, but would require more system information longer run time and application time, and higher cost.

### 7.1 Minimal models for maximum capacity objectives

The maximum capacity objectives do not include grade of service evaluation, but the ensuring of system stability under steady-state conditions. The most simple model for a multiprocessing or multiprocessor structure is the PROCESS R.-NO FLOW (also called load model.)

The same set of objectives for a distributed processor structure has to be covered by the PROCESS R.-ETT BURST PARTIAL FLOW model due to the need of analyzing the interprocess communication stability for the maximum capacity environment. When a certain hierarchical distribution gives low interaction to the processors (those which have less effect on system operation) not all the processors have to be represented at the same degree of detail. An appropriate representation is obtained with a mixed model of PROCESS R.-ETT BURST for the more relevant processors and PROCESS R.-ETT BURST for the less relevant ones.

### 7.2 Minimal model for the Nominal Capacity Objectives

These objectives include the grade of service delay distribution as an output; therefore the minimum degree of representation has to be given by an ETTR (EQUAL TYPE)-PARTIAL FLOW for the multiprocessing and multiprocessor configurations. Again for a case involving distribution of processors, and according to the function sharing among processors, a mix of models may be necessary with the above mentioned for multiprocessing applied to the relevant processors and the less detailed stratum of the same ETT BURST column (e.g. PROCESS R., LEVEL R. or PROCESSOR R. for less relevant ones.)

### 7.3 Minimal models for overload objectives

The set of objectives and results needed for an overload analysis and overload control testing impose the consideration of the TF OVERLOAD CONDITIONS representation of the column five in table 2 and the ETT Stratum for the monoprocessing and multiprocessor configurations. Also, in the case of distributed processor configurations, the low hierarchy ones may need a system representation of less detail with the PROCESS, LEVEL or PROCESSOR strata representation. The subcall and call models mentioned in section 1 correspond with the ETTR (DIFFERENT TYPE)-TOTAL PHASES and ETTR (DIFFERENT TYPE)-TOTAL FLOW NORMAL CONDITIONS, respectively.

The following section summarizes (1) the application of a model for two different hierarchical processor distributions and (2) several models for a conventional multiprocessing system.

### 8. SUMMARY OF APPLICATION TO THREE TYPICAL SYSTEMS

The modeling method presented above has been applied to several switching systems. The characteristics of three of these systems are summarized below, and the simulation features of six cases involving these three systems are given in table 4.

(1) The ITT 1220 DCS system has a hierarchical distributed-control structure which is a particular case of the type mentioned in fig.3 (ref.7). The objectives of the analysis were the obtention of the nominal capacity and the associated results of table 1, which were used in an iterative way for system optimization. The modeling used was a mix of ETTR (EQUAL TYPE)-ETT BURST model for the processor highest in the hierarchy and PROCESSOR R.-ETT BURST model for the impact of processors lower in the hierarchy (case a.)

As an example of the results obtained, figure 8 gives the overhead, preselection delay and through-connection delay as a function of the process cycles selected. This sensitivity allows the optimum selection of cycles for the trade-off between the maximum capacity and OOS delays.
II) Another processor structure with hierarchical distribution of control and a higher degree of distribution for the call-handling functions was analyzed. The objective was a comparison of Back-end/Front-end assignment of call handling activities observing the processor behavior and grade of service obtained.

Two alternatives were compared, with a 50/50 (case b) and 80/20 (case c) ratio of call-handling distribution among the FE and BE processors. Case b and case c consider 1 and 4 FE processors, respectively. Both cases assume 1 BE processor. In these cases the ETTR (EQUAL TYPE)-ETT BURST model was used for all of the system processors.

This comparison showed that the system configuration considered in case c has a capacity three times as high as that of the configuration of case b and a preselection delay 1.8 times shorter.

III) The Stored Program MultiRegister (SPMR) is a new control unit for PENTACONTA* systems (ref.8). SPMR is composed of two miniprocessors that work on a load-sharing basis, each being assigned certain registers to handle. (See fig.1). The organization of each processor is a multi-level interrupt preemptive priority.

In this system, a number of traffic studies have been performed, from design optimization to dimensioning of processors and overload control definition.

In order to fulfill the objectives of maximum capacity determination and detection of bottlenecks, a model of ETTR BURST-ETTR (DIFFERENT TYPE) was used (case d). The types of ETTR indicators that were needed to implement the different service disciplines are: overhead, scanning, timing, batch and individualized-ETTR. The bottleneck detected was the processor occupancy. In this case, the results show the necessity of specifying the service discipline of each process.

To estimate the grade of service it is necessary to measure the influence on it of Pentaconta network control devices, mainly the markers (case e). The model used was an ETTR BURST-ETTR (DIFFERENT TYPE), with some ETTR’s related with Pentaconta network control device operation being detailed to permit the simulation of the waiting time in front of these devices. An example of the results obtained is shown in figure 9.

In order to define optimum overload controls it was necessary to simulate the registration, because in this system, these devices are considered as a filter for the processors. The model used was a SP OVERLOAD CONDITIONS-ETTR (DIFFERENT TYPE) (case f). In the determination of overload controls an important parameter to be taken into account is the grade of service deterioration when overload occurs. Table 3 specifies the time interval after occupancy of the processor is equal to 1, until the grade of service, measured as the average setup time of the finalized preselection phase, reaches the indicated value. In this point we refer to the preselection phase call setups within one-second interval, not to setups completed since the start of the simulation. (Ref.9)

Table 4 summarizes the number of elements per stratum and simulation features for the previous six cases:

<p>| TABLE - 4 |
|------------|---------------|----------------|------------------|----------------|------------------|</p>
<table>
<thead>
<tr>
<th>PROCESSES</th>
<th>LEVELS</th>
<th>PROCESSES</th>
<th>LEVELS</th>
<th>PROCESSES</th>
<th>LEVELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case a</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>100</td>
<td>1:1</td>
</tr>
<tr>
<td>Case b</td>
<td>2</td>
<td>2</td>
<td>30</td>
<td>100</td>
<td>2:1</td>
</tr>
<tr>
<td>Case c</td>
<td>5</td>
<td>1</td>
<td>32</td>
<td>100</td>
<td>5:1:1</td>
</tr>
<tr>
<td>Case d</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>1220</td>
<td>1900</td>
</tr>
<tr>
<td>Case e</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>1610</td>
<td>18</td>
</tr>
<tr>
<td>Case f</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>2060</td>
<td>18</td>
</tr>
</tbody>
</table>

The ratio of simulation time to real time, given in the last column of table 4, is based in the use of an IBM 370/148, except for cases d and e, which used an IBM 370/145.

1. The difference in code between this and the previous case is mainly due to the addition of statistics and outputs.
2. This new code aimed at this case is due to the implementation of control actions for the overloading handling.

9. CONCLUSIONS

The modeling and simulation approach described has been found very useful for the analysis of the three examples summarized.

Because of the representation of both the traffic and the system could be varied as needed, it was possible to cover all the phases of analysis for each system as well as the application to systems with very different structures.

The traffic flow parameterization was very useful for application to completely different flows (due to either correlation or call-type) allowing the change of this characteristic without program recoding, only a few days being required for the building of all the generation sequences for each case.

The modular increase of system flow representation allows a great reusability of the functional modules for different service disciplines of the systems and makes it possible to add details as system design evolves.

Because of the flexibility of its application and speed with which the model can be adapted, this modeling procedure can be extrapolated to processor configurations other than the ones analyzed, and is thus an interesting tool for the traffic analysis of modern microprocessor-based systems.

10. ACKNOWLEDGMENTS

Thanks are given to the members of the TTLG traffic division for their interest and contribution to this area, and especially to Messrs. D. Gutierrez and A. Sanchez for their help in the application to cases a,b and c.

REFERENCES: