"SIMULATION GRAPHS FOR DESIGN AND ANALYSIS OF DISCRETE EVENT SIMULATION MODELS"

by

Enver YUCESAN*
and
Lee SCHRUBEN**

N° 89 / 60 (TM)

* Assistant Professor of Operations Management, INSEAD, Boulevard de Constance, 77305 Fontainebleau, France

** School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, U.S.A.

Printed at INSEAD,
Fontainebleau, France

SIMULATION GRAPHS FOR DESIGN AND ANALYSIS OF DISCRETE EVENT SIMULATION MODELS

by

Enver Yucesan
INSEAD
Boulevard de Constance
77305 Fontainebleau
FRANCE

Lee Schruben
School of Operations Research and Industrial Engineering
Cornell University
Ithaca, New York 14853, U.S.A.

This report was partially supported by the National Science Foundation under grant ECS-8810517 to Cornell University.

ABSTRACT

This paper contains a rigorous definition of discrete event dynamical systems using the mathematics of graph theory. The generality of these graphs is established. This result has implications for the design of discrete event simulation modeling languages as well as for the analysis of discrete event dynamical models.

1. INTRODUCTION

Many intuitively appealing ways of representing the relationships between events in a discrete **event dynamical system** (DEDS) exist. Evans et al. (1967) use schematic chains of event occurrences to depict the general logic flow in a discrete event simulation model implementation. Gordon (1969) uses flow charts to model the behavior of a DEDS. Suri (1984) uses a tree representation to describe the logic for a specific simulation run.

A DEDS network model, that gives quantitative attributes to the elements of a relationship graph, is presented in Schruben (1983). This graph structure, called an *event graph*, has logical and temporal expressions attributed to the edges, while state transition expressions are attributed to the vertices. Several variations of this event graph have appeared in the literature [Pegden, 1987; Sargent, 1988; Law and Kelton, 1989].

In this paper, we present a rigorous mathematical definition of DEDS using graphs. It is shown that these graphs can represent any discrete event simulation. Indeed, they can represent any computer program whatsoever. This result permits us to define a *Simulation Graph* that can be used in the design of discrete event simulation languages as well as the analysis of discrete event simulation models.

The goal of research on the mathematical foundations of discrete event simulation modeling is the development of a general and practically useful mathematical base for this important class of models. The lack of such a base accounts in part for the apparent *ad-hoc* and complex reputation of these models. In a recent report of the *Panel on Future Directions in Control Theory* (1988), Wendel H. Fleming, the panel's chairman, writes: "discrete event dynamical systems exist in many technological applications, but there are no models of discrete event systems that are as concise or computationally as feasible as are differential equations for continuous variable dynamical systems. (...) A complete theory of discrete event dynamical systems may require fundamentally new approaches, involving techniques from algebra, graph theory, discrete mathematics, computational mathematics, and combinatorics. More specifically, the following modeling questions should be answered: how does one formally specify and reason about discrete event dynamical systems?" Here, we propose graph theory as an effective base for discrete event systems.

There have been recent advances in obtaining a general understanding discrete event models. See, for instance, Nance (1981), Overstreet (1982), Radiya and Sargent (1987) and Zeigler (1984). While theoretical structures for discrete event simulations and practical model building environments have been developed [eg., TESS (Pritsker and Standridge, 1987) and CINEMA (Systems Modeling Corp., 1985)], an explicit general mathematical formulation of discrete event simulations with practical utility has not yet emerged. *This paper formalizes the concept of using graph structures to represent DEDS models.*

Modeling tools can be evaluated by two characteristics: their generality and their utility. Generality refers to the ability to construct a *valid* representation of a wide class of systems. Utility is the ability to analyze specific instances of a modeled system and make valid inferences. In the parlance of theoretical Computer Science, generality of a modeling tool is more precisely defined as its *modeling power* whereas its utility is known as its *decision power* [Peterson, 1977]. These characteristics usually represent a trade-off. As we increase the generality of our modeling tools, they tend to become more abstract and conceptual. This, in turn, makes it more difficult to determine various properties of specific systems, reducing the tool's utility.

This trade-off is more explicitly depicted in Figure 1. The Turing Machine [Hopcroft and Ullman, 1979] is general; it is accepted as being able to model any modern computational procedure ("Church's thesis"). However, it has limited utility, since even the simplest system can be difficult to represent as a Turing Machine and most specific questions are undecidable. The highest levels of Zeigler's abstract conceptual DEVS hierarchy represents an extreme example of this trade-off in favor of generality [Zeigler, 1984]. Examples of modeling tools with limited modeling power but excellent decision power include most of the analytical modeling tools of Operations Research (OR) and Computer Science (for example, game theory, queueing theory, linear programming, finite automata). Typically, OR methodologies have high decision power but are limited in generality. Much of OR research is directed at increasing the applicability of these tools. Most analytical techniques can be observed to have evolved from being very specific to being more general (eg., M/M/1 queues to GI/GI/s systems). However, as these techniques have become more general, they have become more complex often impairing their decision power. The development of simulation languages, on the other hand, has been typically driven by the needs of

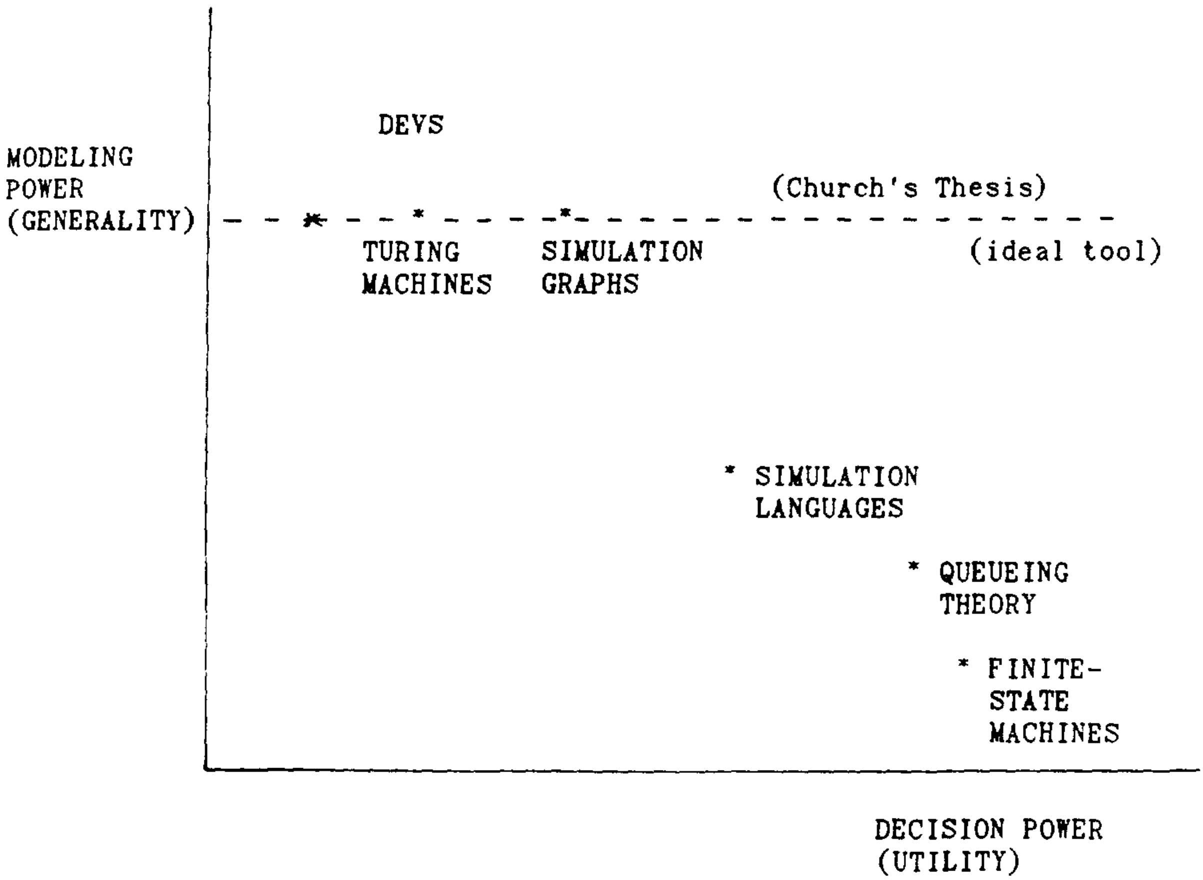


FIGURE 1: MODELING POWER vs DECISION POWER

the users (see, for instance, Gordon (1979)). As more and more features were built into these languages, both their modeling and decision power increased.

We will show that the Simulation Graphs presented here retain the modeling power of Turing Machines while achieving considerably better decision power for a special class of models: discrete event dynamical simulations.

This paper is organized as follows: Section 2 defines Simulation Graphs and exhibits its decision power. An example is also included. The modeling power of these graphs is discussed in Section 3. Section 4 Section 4 discusses some implications. Conclusions are presented in Section 5.

2. THE DECISION POWER OF THE SIMULATION GRAPHS

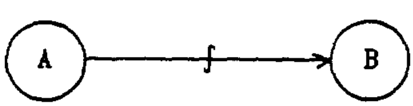
2.1. Simulation Graphs:

The elements of a DEDS model are state variables that describe the system, events that change the values of state variables, and the relationships between events. A Simulation Graph is a structure of the objects in a discrete event system that facilitates the development of a correct simulation model. Hence, the emphasis is directly on system events; system entities are represented implicitly.

Events are represented on the graph as vertices. Here, each vertex is associated with a set of changes to state variables.

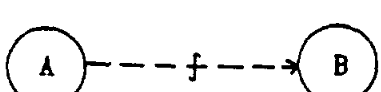
Relationships between events are represented in a Simulation Graph as directed edges between pairs of vertices. Each edge is associated with sets of logical and temporal expressions. Basically, the edges define under what conditions and after how much of a time delay one event will schedule another event to occur.

Two classes of edges are distinguished: a (solid) *scheduling* edge and a (dashed) *cancelling* edge. Specifically,



indicates that *event A may schedule event B with some time delay*.

Similarly,



indicates that *event A may cancel any currently scheduled occurrence of event B*.

In defining a Simulation Graph ***Model*** in Section 2.4, we will associate with each edge a set of conditions that must be true in order for the edge origination vertex to schedule or cancel the edge termination vertex. Also associated with each edge will be an expression or function that will give a delay time telling how long until the scheduled event will occur. There can be multiple edges between any pair of vertices; the edges can point in either direction or may simply point from a vertex to itself. To simplify the notation, if $t \equiv 0$ (no time delay) or if $i \equiv (1=1)$ (condition always true), then these parameters are omitted from the graph. Similarly, if $k \equiv 1$ (same event instances), then j is also omitted from the graph.

In a Simulation Graph Model, one vertex may also pass a string of parameters to another vertex. This parameter string carries information pertaining to a particular event instance as well as the execution order priority of that event.

Note that the customary notion of an *event procedure* in a discrete event simulation program may involve a subgraph, typically a set of vertices connected by edges with no time delays.

Before we introduce a mathematical definition of Simulation Graphs, we present an example.

2.2. An Example: Oil Tanker Problem:

A sample study problem from the texts by Schriber (1977), and Law and Kelton (1989) is used to illustrate the construction of a Simulation Graph. This system is represented as a process block diagram in Schriber, as a process network in Pritsker (1984) and as a modified Petri net graph in Torn (1981). The system description is adapted from Law and Kelton (Problem 2.23):

> A port with three docks is used to load oil. Three different types of tanker ships use the port. The distribution for the random time between arrivals of a type j tanker, $t_a(j)$, is known. The random time required to load a type j tanker, $t_l(j)$, also has a known distribution. There is a single tugboat in the harbor that is required for all tanker berthing or deberthing. It takes one hour for a tanker to be berthed or deberthed. The area experiences storms of random frequency and random duration. The time between storm occurrences is t_s and the duration of a storm is t_d. No berthing or deberthing operation can be initiated while the storm is in progress and any berthing operation in progress when the storm hits is aborted. However, the storm does not interrupt any deberthing in progress. A fleet of another tanker type wishes to begin using the port for shipment of oil on a specific route. The loading times, $t_l(4)$, and (random) round-trip time, $t_a(4)$, for this fourth type of tanker have known distributions. The travel time for the tug when not towing a tanker is insignificant. The tug will attend to tankers requesting berthing first.

The model is taken from Schruben (1982). The state variables we selected for this model are:

1. B represents the number of tankers waiting for berthing; the range of B is the set of non-negative integers.

2. D represents the number of tankers waiting for deberthing; the range of D is also the set of non-negative integers.

3. P represents the number of empty docks at the port; the range of P is the set $\{0,1,2,3\}$.

4. T denotes the status of the tug; it has a range of $\{-2,-1,0,1\}$ with $-2 \equiv$ berthing aborted by storm, $-1 \equiv$ deberthing a tanker, $0 \equiv$ berthing a tanker, $1 \equiv$ available.

5. S denotes the status of the storm; it has a range of $\{0,1\}$ with $0 \equiv$ storm in progress and $1 \equiv$ no storm.

6. QB and QD are lists of tankers requesting berthing and deberthing, respectively. Both variables are single dimensional arrays denoting the type of the tanker. More precisely, QB[i] = j (QD[i] = j) denotes that the i^{th} tanker requesting berthing (deberthing) is of type j. Here, $i \in \{1,2,3,...\}$ and $j \in \{1,2,3,4\}$.

The edge conditions for the model are:

(i) (No storm in progress, a berth is available, the tug is available) S = 1, P > 0, T = 1, i.e., (S*P*T>0) (* denotes multiplication).

(ii) (No storm in progress, a berth is available, berthing queue is non-empty) S = 1, P > 0, B > 0, i.e., (S*P*B>0).

(iii) (No storm in progress, deberthing queue is non-empty, berthing queue is empty) S = 1, D > 0, B = 0, i.e., (S*D>0)&(B=0).

(iv) (No storm in progress, the tug is available, berthing queue is empty) S = 1, T = 1, B = 0, i.e., (S*T=1)&(B=0).

(v) (No storm in progress, berthing queue is empty, deberthing queue is non-empty) S = 1, B = 0, D > 0, i.e., (S*D>0)&(B=0).

(vi) (No storm in progress, berthing queue is non-empty) S = 1, B > 0, i.e., (S*B>0).

(vii) (The tug is berthing a tanker) (T=0).

(viii) (Berthing is aborted by a storm) (T=-2).

(ix) (The tug is available, berthing queue is non-empty, a berth is available) T = 1, B > 0, P > 0, i.e., (T*B*P>0).

(x) (The tug is available, berthing queue is empty, deberthing queue is non-empty) T = 1, B = 0, D > 0, i.e., (T*D>0)&(B=0).

The event descriptions are presented in Table 1.

TABLE 1

Event Type	Event Description	Parameters	State Changes
AR(j)	Type j tanker arrival	j = k	B = B + 1 QB[tail] = j
BB(j)	Begin berthing tanker of type j		T = 0 B = B - 1 k = QB[head]
EB(j)	End berthing tanker of type j	j = k	P = P - 1 T = 1
BL(j)	Begin loading tanker of type j	j = k	
EL(j)	End loading tanker of type j	j = k	D = D + 1 QD[tail] = j
BD(j)	Begin deberthing tanker of type j		T = -1 D = D - 1 P = P + 1 k = QD[head]
ED(j)	End deberthing tanker	j = k	T = 1
BS	Begin storming		S = 0
ES	End storming		S = 1
BS2	Abort berthing tanker		T = -2

The associated Simulation Graph is presented in Figure 2. Note that there may be several possible representations of this model as a Simulation Graph. For example, by applying the "rules of thumb" in Schruben (1983), this model can be represented as in Figure 7 in Schruben (1982).

2.3. Definition of Simulation Graph Models:

A directed graph is characterized as an ordered triple $(V(G),E(G),\Psi_G)$ consisting of a nonempty set $V(G)$ of vertices, a set $E(G)$, disjoint from $V(G)$, of edges, and an incidence function Ψ_G that associates with each edge of G an ordered pair of (not necessarily distinct) vertices of G. If e is a directed edge joining vertices u and v, then $\Psi_G(e) = uv$ [Bondy and Murty, 1976].

We define a *Simulation Graph* as an ordered quadruple $\mathcal{G} = (\mathcal{V}(\mathcal{G}),\mathcal{E}_s(\mathcal{G}),\mathcal{E}_c(\mathcal{G}),\Psi_\mathcal{G})$, where $\mathcal{V}(\mathcal{G})$ is the *vertex set* of $\mathcal{G}$, not necessarily finite, $\mathcal{E}_s(\mathcal{G})$ is the *set of scheduling edges* of $\mathcal{G}$, $\mathcal{E}_c(\mathcal{G})$ is the *set of cancelling edges* of $\mathcal{G}$, and $\Psi_\mathcal{G}$ is the *incidence function*.

We then define the objects in a simulation model as the following ordered sets:

$\mathcal{F} = \{ f_v : v \in \mathcal{V}(\mathcal{G}) \}$, which is the set of *state transition functions* associated with (event) vertex v;

$\mathcal{C} = \{ C_e : e \in \mathcal{E}_s(\mathcal{G}) \cup \mathcal{E}_c(\mathcal{G}) \}$, which is the set of *edge conditions*.

$\mathcal{T} = \{ t_e : e \in \mathcal{E}_s(\mathcal{G}) \}$, which is the set of *edge delay times*; note that $t_e \in R^+_{0,\infty}$ for all edges.

$\Gamma = \{ \gamma_e : e \in \mathcal{E}_s(\mathcal{G}) \}$, which is the set of *event vertex execution*

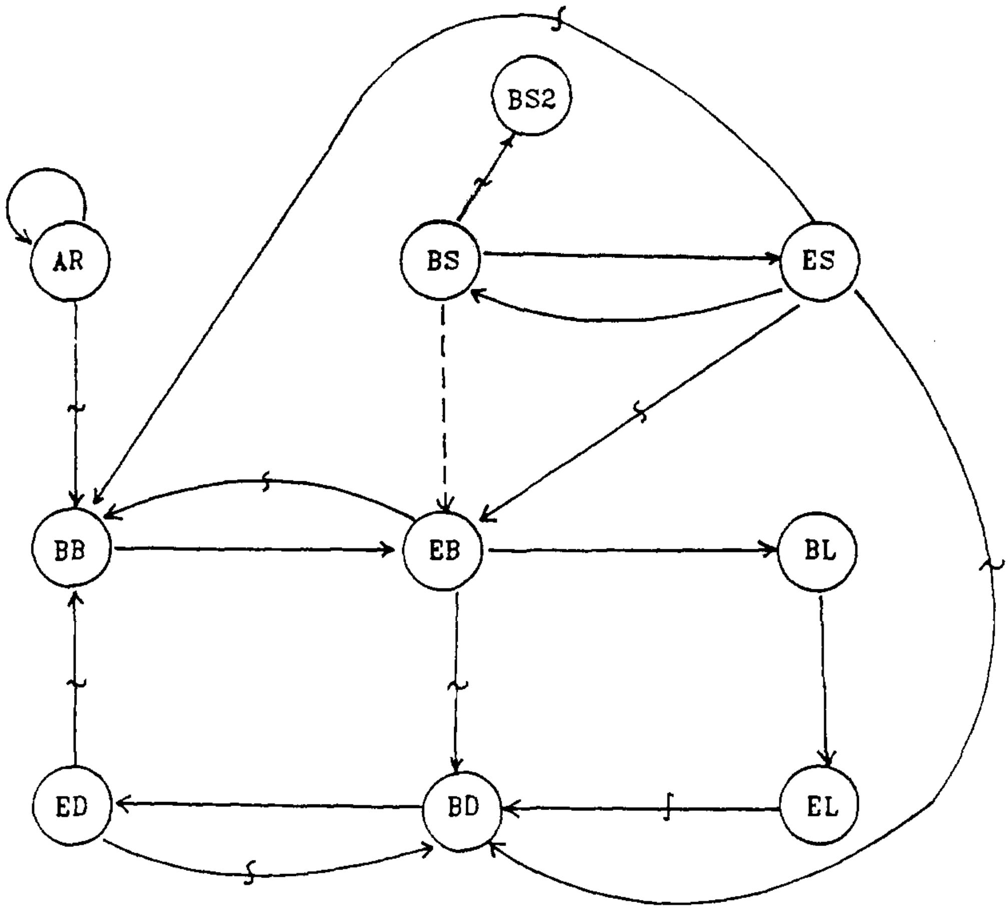


FIGURE 2: SIMULATION GRAPH FOR THE OIL TANKER PROBLEM

priorities. γ_e is a scalar assuming values from the set of non-negative real numbers extented to include positive infinity. Notice that the event execution priorities are associated with edges, not vertices. Therefore, the priority of a vertex can change dynamically depending on when and how it is scheduled. This assignment yields a *complete* ordering of execution priorities. Schemes of static event execution priorities, attaching static priorities to event vertices (Som and Sargent, 1988), are ambiguous in that they provide only a partial ordering of event execution. Each vertex has an execution priority with smaller values corresponding to higher priorities, and with 0 representing the highest possible event execution priority.

The key idea here is that a Simulation Graph specifies the relationships between the elements of the sets of objects in a simulation model. Then, we can formally define a Simulation Graph *Model* as:

$$\mathcal{S} = (\mathcal{F}, \mathcal{C}, \mathcal{T}, \Gamma, \mathcal{G})$$

The first four sets in the above five-tuple define the objects in a simulation. The role played by the Simulation Graph, $\mathcal{G}$, in the definition of a Simulation Graph Model, $\mathcal{S}$, is analogous to the role of the incidence function in the definition of a directed graph. The incidence function organizes the individual objects (vertices and edges) into a meaningful graph. The Simulation Graph, G (a graph of *indices*), organizes the objects ($\mathcal{F}$, $\mathcal{C}$, $\mathcal{T}$, Γ) into a meaningful simulation model.

In a Simulation Graph Model, parameter strings can be passed from one vertex to another through vertex and edge attributes. A *vertex attribute list* is a string of state variables associated with a particular vertex. An *edge attribute list*, on the other hand, is a string of expressions associated with a particular edge. When the origination vertex of an edge is executed, the expressions in the edge attribute list are evaluated. When the destination vertex is subsequently executed, the state variables in its attribute list take on the values that had been computed for the expressions in the scheduling edge's attribute list. The vertex and edge attribute lists are useful in scheduling or cancelling specific instances of system events.

Example: The Oil Tanker Problem (Continued):

We will use our example for a concrete illustration of the sets in a simulation model, $\mathcal{S}$. The following objects make up the model (for exposition purposes, the vertex indices are not included):

$\mathcal{V}(\mathcal{G}) = \{v_1, v_2, \ldots, v_{10}\} = \{AR, BB, EB, BL, EL, BD, ED, BS, ES, BS2\}$,

$\mathcal{E}_s(\mathcal{G}) = \{e_1, e_2, \ldots, e_{17}\} = \{(AR,AR), (AR,BB), (BB,EB), (EB,BB), (EB,BL), (EB,BD), (BL,EL), (EL,BD), (BD,ED), (ED,BD), (ED,BB), (BS,ES), (ES,BS), (BS,BS2), (ES,BB), (ES,EB), (ES,BD)\}$,

$\mathcal{E}_c(\mathcal{G}) = \{e_{14}\} = \{(BS,EB)\}$,

$\Psi_{\mathcal{G}}$: see figure 2,

$\mathcal{F}$: State Transition Functions

$f_{AR} = \{ j = k, B = B + 1, QB[tail] = j \}$,

$f_{BB} = \{ B = B - 1, T = 0, k = QB[head] \}$,

$f_{EB} = \{ j = k, P = P - 1, T = 1 \}$,

$f_{BL} = \{ j = k \}$,

$f_{EL} = \{ j = k, D = D + 1, QD[tail] = j \}$,

$f_{BD} = \{ D = D - 1, T = -1, P = P + 1, k = QD[head] \}$,

$f_{ED} = \{ j = k, T = 1 \}$,

$f_{BS} = \{ S = 0 \}$,

$f_{ES} = \{ S = 1 \}$,

$f_{BS2} = \{ T = -2 \}$.

$\mathcal{C}$: Edge Conditions

$C_{AR,BB} = \{ S * P * T > 0 \}$,

$C_{EB,BB} = \{ S * P * B > 0 \}$,

$C_{EB,BD} = \{ S * D > 0, B = 0 \}$,

$C_{EL,BD} = \{ S * T = 1, B = 0 \}$,

$C_{ED,BD} = \{ S * D > 0, B = 0 \}$,

$C_{ED,BB} = \{ S * B > 0 \}$,

$C_{BS,BS2} = \{ T = 0 \}$,

$C_{ES,EB} = \{ T = -2 \}$,

$C_{ES,BB} = \{ T*B*P > 0 \}$,

$C_{ES,BD} = \{ T*D > 0, B = 0 \}$.

$\mathcal{T}$: Edge Delay Times

$t_{AR,AR} = t_a$,

$t_{BB,EB} = 1$,

$t_{BL,EL} = t_1$,

$t_{BD,ED} = 1$,

$t_{BS,ES} = t_d$,

$t_{ES,BS} = t_s$,

$t_{ES,EB} = 1.$

$\Gamma = \{\gamma_{AR}, \gamma_{BB}, \gamma_{EB}, \gamma_{BL}, \gamma_{EL}, \gamma_{BD}, \gamma_{ED}, \gamma_{BS}, \gamma_{ES}, \gamma_{BS2}\} = \{2, 1, 2, 2, 2, 2, 2, 1, 2, 1\}$.

This model can be executed using the algorithm presented in the next section.

2.4. Execution of a Discrete-Event Simulation

The simulation (execution) of a Simulation Graph model requires the services of a support routine, which will be referred to as the *Event Scheduling Function* (ESF). Basically, ESF maintains two crucial variables: τ, the global simulation clock, and $\mathcal{L}$, the list of scheduled events (the events list). In addition, it also supports a scalar stochastic process of independent uniform random variables, U, on [0,1]. Appropriate transformations of U provide any random process needed in the simulation. Notice that the *non-model-specific objects* such as the events list and the global simulation clock are *not* included in the definition of a Simulation Graph Model, $\mathcal{S}$.

The events list is an ordered set of triples. That is,

$$\mathcal{L} = \{ (t_1,\gamma_1,v_1(k)),\ (t_2,\gamma_2,v_2(k)), \ldots.\ \}.$$

where t_i represents the *event execution time*, γ_i represents the *event execution priority* and v_i represents the associated event vertex. The index k denotes specific instances of these events. The triples, which will be referred to as *event records*, are ordered by the event execution time. Those events that are scheduled to occur at the same simulated time are further ranked by their event execution priorities. Recall that event execution priorities can be assigned dynamically using the event and edge attribute lists.

For convenience, we also define the following sets:

S_v is the set of state variables possibly altered at vertex v ($v \in \mathcal{V}(G)$);

E_v is the set of state variables involved in the conditions on the edges emanating from vertex v ($v \in \mathcal{V}(G)$);

A_v, *the vertex attribute list*, is a set of state variables associated with vertex v ($v \in \mathcal{V}(G)$). Similarly, A_e, the *edge attribute list*, is simply a string of expressions associated with edge e ($e \in \mathcal{E}_s(G) \cup \mathcal{E}_c(G)$).

Based on the above framework, the execution of a Simulation Graph model is carried out in the following manner:

<u>INITIALIZE (Run Initialization)</u>

<u>Step 1.</u> Initialize global simulation clock:

$$\tau := 0$$

<u>Step 2.</u> Insert the first event record into the events list: this is usually the event where the initial condition of the model is specified by assigning the initial values to the state variables:

$$\mathcal{L} := \mathcal{L} \cup \{ (0, \Upsilon_0, v_0) \}$$

<u>EXECUTE (Execution of the Model)</u>

<u>Step 1.</u> Remove the first event record from $\mathcal{L}$; this is the event with the smallest event execution time (ties are resolved based on the

event execution priorities; remaining ties are broken arbitrarily):

$$\mathcal{L} := \mathcal{L} \setminus \{(t_i, \gamma_i, v_i(k))\}$$

Step 2. Update the simulation clock:

$$\tau := t_i$$

Step 3.1. Assign the values of the state variables in the vertex attribute list, $A_{v_i(k)}$, if the list is not empty.

Step 3.2. Evaluate the state variables in $S_{v_i(k)}$:

$$S_{v_i(k)} := f_{v_i(k)}(S_{v_i(k)})$$

Step 4. Schedule and/or cancel further events:

for all edges emanating from vertex v_i:

if $C_{v_i v_j}(E_{v_i}) = 1$, then

compute values of expressions in the

corresponding edge attribute list

(including execution priorities),

generate the inter-event time, t_j and

$$\mathcal{L} := \mathcal{L} \cup \{(\tau + t_j, \gamma_j, v_j(k))\}$$

(Note that for cancellations, we have

$$\mathcal{L} := \mathcal{L} \setminus \{(\tau, \gamma_j, v_j(k))\} \text{ .)}$$

For completeness, in Step 4, any event scheduling is done before any event cancellation.

Step 5. Terminate the execution of the simulation if any of the following situation is reached:

(i) $\tau \geq T_{STOP}$,

(ii) $(t_{end}, \gamma_{end}, v_{end})$ has just been executed,

Otherwise, go to Step 1 of EXECUTE.

Here, T_{STOP} represents a pre-determined stopping time for the simulation and v_{end} represents an end-of-simulation event.

2.5 Computer Implementation of Simulation Graphs

Simulation Graph models can be implemented using a high level programming language or a general purpose simulation modeling language by possibly coding subgraphs into separate procedures [Som and Sargent, 1988]. Alternatively, they can be directly implemented using Σ [Schruben and Briskman, 1988]. Σ is an interactive graphics program specifically designed to build, test, and experiment with discrete event dynamical systems on personal computers using Simulation Graphs. The development of Σ has been different from other discrete event simulation languages. For the latter, the addition of built-in functions has increased both their modeling and decision power, moving in the northeastern direction in Figure 1. Σ, however, starts with a basic structure having complete modeling power; any addition simply increases its decision power, moving in the eastern direction along the horizontal dotted line in Figure 1. Building a model, implementing the model, and experimenting with its implmentation are some of the steps in a simulation study. Σ is intended to facilitate model implementation by allowing the user to construct executable models by drawing their graphs with a mouse.

3. THE MODELING POWER OF SIMULATION GRAPHS

In Figure 1 of Section 1, it is claimed that Simulation Graph Models have the same modeling power as Turing Machines. In this section, we will formalize this assertion. This has implications both on the design of discrete event simulation languages and the analysis of simulation models as we will illustrate.

3.1 Turing Machine Representation:

We first define the basic Turing Machine. For other variants, the reader is referred to Hopcroft and Ullman (1979). The Turing machine is a simple mathematical model of a computer. Despite its simple structure, it captures the essential features of real computing machines, and is accepted (by Church's Thesis) as the formalization of any algorithm or computational procedure.

Following Hopcroft and Ullman (1979), we define a basic Turing Machine as consisting of a finite control, an input tape that is divided into cells and a tape head that scans one cell of the tape at a time. The tape has a leftmost cell, but is infinite to the right. Each cell of the tape may hold exactly one of a finite number of tape symbols. Formally, the Turing Machine is defined as:

$$\mathcal{M} = (Q, \Sigma, T, \delta, q_0, B, F)$$

where

Q is the *finite set of states*,

T is the *finite set of allowable tape symbols*,

B is the symbol for *blank*, $(B \in T)$,

Σ is the *set of input symbols*, $(\Sigma \subset T \setminus \{B\})$,

δ is the *next-move function*, $(\delta : Q \times T \to Q \times T \times \{L,R\})$,

q_0 is the *start state*, $(q_0 \in Q)$,

F is the *set of final states*, $(F \subset Q)$.

To demonstrate that Simulation Graphs and Turing Machines have the same modeling power, one needs to show that every system specified in one formalism can be simulated by a system specified in the other one.

The fact that a Turing Machine, $\mathcal{M}$, can simulate a Simulation Graph Model, $\mathcal{S}$, is established through Church's Thesis. A constructive proof is also presented in Schruben and Yucesan (1987). We next establish the fact that a Simulation Graph Model, $\mathcal{S}$, can simulate a Turing Machine, $\mathcal{M}$.

This is achieved through the construction of a a Simulation Graph Model of a basic Turing Machine. The *tape symbols* of $\mathcal{M}$ will be represented by *state variables* in $\mathcal{S}$. Hence, the *input symbols* for $\mathcal{M}$ will simply consist of a subset of the state variables of $\mathcal{S}$. Another state variable will be used to denote the *state of the finite control* of the Turing Machine. Hence, STATES of $\mathcal{S}$ will completely describe the set of states of $\mathcal{M}$, Q. Table 3 summarizes the correspondance between the elements of $\mathcal{M}$ and $\mathcal{S}$. Here, β_i denotes the i^{th} state variable of the model, whereas RANGE[β_i] depicts the set of possible values that may be assumed by that state variable. Figure 3 depicts the Simulation Graph representation of a Turing Machine. Here, vertex INT is the *initialization* vertex, where the initial state of the finite control, FC, is established by assigning the initial values to appropriate state variables. An input string of length L is also read in. Notice that the

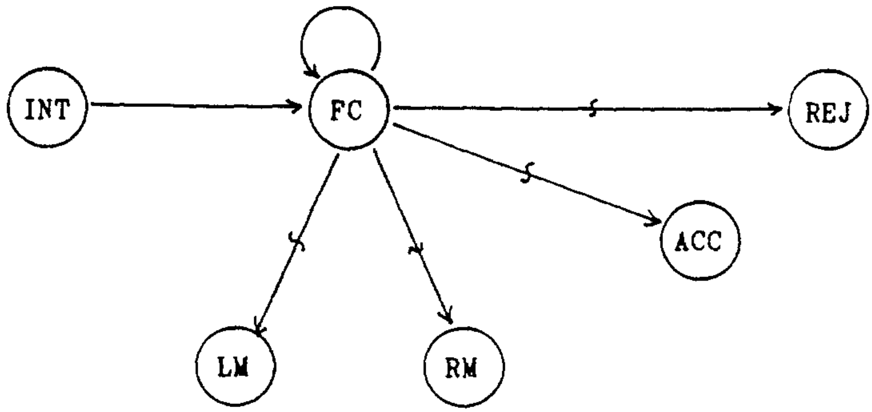


FIGURE 3: SIMULATION GRAPH REPRESENTATION OF TURING MACHINES

tape of $\mathcal{M}$ is represented in $\mathcal{S}$ by a one-dimensional array (a state variable) of infinite length. Each element of the array will correspond to a cell of the tape. The input string will be stored in this state variable, ARRY(.). Another state variable, say HEAD, will also keep track of the current position of the tape head. Finally, the first element of the array, ARRY(1), is read and HEAD is assigned the value of 1 corresponding to the placement of the tape head on the first cell of the input tape.

The vertex FC represents the *finite control*. Note that the *next move function* of $\mathcal{M}$

$$\delta : Q \times T \rightarrow Q \times T \times \{L,R\}$$

is represented *implicitly* in $\mathcal{M}$. That is, depending on the current state of the finite control, which is depicted by the current values of the state variables, and the current input symbol, given by the array element just read, the edge conditions, $\mathcal{C}$, determine the next vertex to be scheduled. This vertex specifies the move of the tape head as well as the new state of the finite control.

Vertices LM and RM represent *the left move (L)* and *the right move (R)*, respectively. At these vertices, the values of the state variables, hence, the state of the finite control, are updated. The tape head is also moved one cell to the right or to the left by pointing at the proper array element. For instance, a right move will be represented by HEAD = ARRY(CURRENT + 1), where CURRENT denotes the array element that was read last.

Vertices ACC and REJ represent the set of *final states*. More specifically, vertex ACC represents the final state where $\mathcal{M}$ stops with a "YES" answer, accepting the input string. Vertex REJ, on the other hand, represents a final state where the input string is rejected.

The execution of the above described Simulation Graph model, $\mathcal{S}$, of a Turing Machine, $\mathcal{M}$, is carried out until the first execution of either vertex ACC or vertex REJ.

Thus, with the above construction, we have shown the simulation of a Turing Machine by a Simulation Graph model.

TABLE 3

$\mathcal{M}$	$\mathcal{S}$
Q	STATES $\subset$ RANGE[$\mathcal{E}_1$]X..XRANGE[$\mathcal{E}_m$]
T	$\cup_i$RANGE[$\mathcal{E}_i$] ($1 \leq i \leq m$), where $\mathcal{E}_i$ is a state variable
Σ	Subset of $\cup_i$RANGE[$\mathcal{E}_i$] ($1 \leq i \leq m$)
δ	$\mathcal{C}$, set of edge conditions
q_0	Set of initial values for the state variables
F	Set of final values for the state variables

This result shows that any simulation, indeed any computer program as we know it today, can be modeled using Simulation Graphs.

An example of a Simulation Graph Model of a Turing Machine used for bit parity checking executed using Σ is presented in Schruben and Yucesan (1987).

3.2 DEVS Representation

As Zeigler (1984) points out, "the concept of *simulation* employed here is a rather weak one, and states little more than that a [Turing Machine] can be used as a computer to generate the behavior" of a Simulation Graph Model, $\mathcal{S}$. "It leaves the burden of writing and interpreting this simulation to the user."

A perspective on this result can be obtained by considering the hierarchical system specification scheme that Zeigler introduces (1984). At the "lower" levels of this hierarchy, the system is considered as a black box. Hence, only the system's input/output (I/O) behavior is apparent to the observer/user. As we move "up" the hierarchy, this I/O behavior is more precisely defined, leading us to the "top" level of the hierarchy where the internal structure of the system is completely apparent. This internal structure, in turn, fully accounts for the observed I/O behavior of the system. For a detailed discussion of the hierarchical system specification, the reader is referred to Zeigler (1984). Zeigler also discusses the DEVS (Discrete Event System Specification) formalism in this hierarchical approach. DEVS is the following structure:

$$\langle \text{INPUTS, STATES, OUTPUTS}, \delta, \lambda, ta \rangle.$$

From our previous discussion, it is clear that Simulation Graph Models comfortably fit into the DEVS framework. These models are autonomous. Then, with slight modifications for this case, the definition of STATES, OUTPUTS, δ, and λ directly carry over from DEVS to $\mathcal{S}$. In the context of Simulation Graph Models, INPUTS = $\emptyset$. The time advance mechanism needed to execute (simulate) the model is provided by the support routine, the

Event Scheduling Function (ESF). ESF also maintains the global simulation clock, τ, as well as the events list, $\mathcal{L}$ (see Schruben and Yucesan (1987) for details).

Since we have just discussed how Simulation Graphs fit into the DEVS framework, we can conclude that they are not only useful for depicting the I/O behavior of any system (Turing machine representation), but also sufficiently powerful to describe the internal structure of any autonomous discrete event system (DEVS representation).

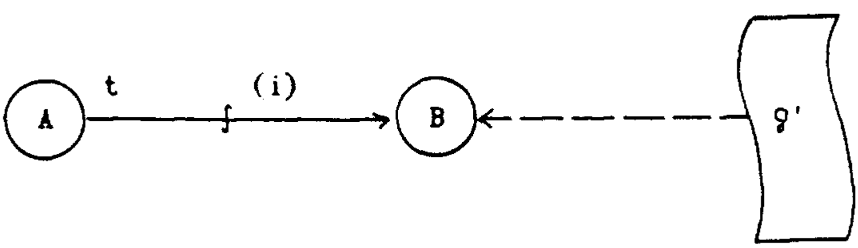
4. THE DESIGN OF SIMULATION LANGUAGES

4.1. Cancelling Edges

The correspondance between Simulation Graphs and Turing Machines permits the classification of the various features of a simulation language (or environment) as being *"necessary"* or *"a modeling convenience."* We consider here three examples: event cancelling, event attribute passing, and event execution priorities.

For instance, in establishing the equivalence between $\mathcal{S}$ and $\mathcal{M}$, it is observed that cancelling edges in Simulation Graphs are not necessary. Cancelling edges are typically used to model *disruptions* in a process. A machine breakdown that shuts down a production line or a storm that closes an airport are examples. When such a disruption occurs, it may be that certain events that were scheduled will be cancelled. Event cancellation is a feature of most modern simulation languages. It is implemented using a "CANCEL" statement in Simscript II.5 [CACI, 1976], an "RMOVE" subroutine or a "PREEMPT" node in SLAM II [Pritsker, 1984], a "PREEMPT" or "ABORT" node in Insight [Roberts, 1983] and an "REMOVE" block in SIMAN [Pegden, 1985].

A Simulation Graph Model of such a system takes the following form:



Here $\mathcal{G}'$ denotes the subgraph which eventually causes the cancellation of event B.

Instead of using cancelling edges as shown above, one can use a "look-ahead" feature along with a secondary list of *pending events*, where decisions are based on what is going to happen in the future. This is due to the unique advantage of computer simulation where the user has complete control over the events throughout simulated time. Therefore, in the generic situation depicted above, the use of cancelling edges can be eliminated as follows: let T denote the current value of the simulation clock (τ = T). Before scheduling event B to occur at T+t, one simulates the subgraph, $\mathcal{G}$, up to that point in simulated time. Within this *test window*, [T, T+t], event B is kept in this secondary list of pending events, say $\mathcal{L}_p$. If the cancellation of event B would indeed occur within this test window, then one avoids the scheduling of event B altogether. If event B is never cancelled within this period, than its event record is transferred from the secondary list, $\mathcal{L}_p$, to the future events list, $\mathcal{L}$.

As this "look-ahead" feature requires the execution of a subgraph within a test window and the maintenance of a secondary list structure, it tends to be far less appealing than the use of cancelling edges. Thus, cancelling edges are indeed a valuable modeling convenience. This feature of various simulation languages, while affording a great modeling convenience, is technically not necessary.

4.2. Vertex and Edge Attribute List

As discussed earlier, vertex and edge attribute lists are simple but powerful modeling tools. Recall that an edge attribute list is a string of expressions. The vertex attribute list, on the other hand, is a string of state variables. The main value of the attribute lists is in defining the particular system entities to which an event pertains. For example, suppose that there are two identical machines in a simulated factory. The same "start processing" event vertex may be used for both machines if the event has an attribute telling which machine is to start. In fact, it is possible to simulate a factory with n (n > 1) machines with only three event vertices provided that we define vertex and attribute lists in the model. Another common use for attribute lists is in initializing a simulation program. In Σ, for instance, the use of an edge attribute list for the "initialization vertex" creates an interactive environment where the user can specify the starting conditions for the model. For example, the user can specify the number of servers and their service rates while initializing a multi-server queueing simulation.

A Simulation Graph model with vertex and edge attributes really represents an *array of graphs*. For example the Simulation Graph model of the Oil Tanker Problem actually embodies four copies of the same graph, one for each type of tanker. During the execution of the model, for instance, vertex AR may execute the arrival event of a tanker of type 1, while vertex BB executes the berthing of another tanker of type 3. Hence, the use of vertex and edge attributes eliminates unnecessary duplication of similar event vertices; and in so doing, may prevent a

Simulation Graph from becoming infinite (consider a Simulation Graph of an $M/M/\infty$ queue without attributes where the identity of individual servers must be preserved).

It is possible to construct simulation models without using vertex and edge attributes. In this case:

1. It would be necessary to duplicate the vertices (together with their edges) to accomodate all of the different classes of entities that were previously distinguished by a vertex and/or edge attribute. However, this may result in a very large (possibly infinite) Simulation Graph.

2. It would be necessary to code the initial conditions of the model within the body of the simulation program. This would further necessitate the alteration (and subsequent recompilation) of the simulation program every time a different set of initial conditions need to be incorporated into the model (see the discussion of the "experimental frames" in Zeigler).

We, therefore, observe that vertex and edge attributes are indeed valuable modeling conveniences. Moreover, most of the modern simulation languages mentioned above provide facilities to index event routines as well as other entities in simulation models. For example, the expression "WITH" in Simscript, "entity attributes" in GPSS, event and queue subscripts in SLAM, and edge and vertex attributes in Σ are such modeling conveniences.

4.3 Event Execution Priorities

In establishing the equivalence between $\mathcal{S}$ and $\mathcal{M}$, it is also observed that event execution priorities in Simulation Graph Models are not necessary. In fact, many simulation programming languages do not offer such a facility. One exception is SIMSCRIPT [CACI, 1976] which has the "PRIORITY ORDER" statement for breaking ties among simultaneously scheduled events. This is a static priority scheme that must be established *apriori* in the so-called "PREAMBLE" whereas Simulation Graph models allow dynamic priority assignments which can be altered during the execution of a model implementation. As we will see in the next section, even though it is not technically necessary, an event execution priority scheme is useful in the construction of an unambiguous model.

4.4 Further Implications

The fact that every Turing Machine can be represented as a Simulation Graph Model further implies that the *undecidability results* pertaining to Turing Machines also apply to Simulation Graph Models. These results, which have important implications for model analysis, were first reported by Overstreet (1982). Here, we will review some of the issues in the context of Simulation Graph Models.

First, we recast some preliminary definitions in the context of Simulation Graphs. A Simulation Graph Model, $\mathcal{S}$, is said to be *finite*, if, in any implementation of it, when provided with valid initial conditions, only a finite number of events are executed before the termination condition is satisfied. $\mathcal{S}$ is said to be *complete*, if, at each instant during the execution of any implementation of it, either the termination condition is satisfied or the events list contains at least one more event record. An event vertex is *accessible* if it can be executed during the execution of some implementation of the Simulation Graph Model. Two events are *order independent* if their execution in any particular sequence yields identical results. If two implementations of a Simulation Graph Model, $\mathcal{S}$, are possible in which their output behavior would differ, then $\mathcal{S}$ is said to be *ambiguous*.

Overstreet has shown that *"one cannot construct (by Church's thesis) a computer program that can determine,"* for every Simulation Graph Model, *"whether it is complete, finite, accessible or ambiguous."* He has also shown that no general algorithm exists to determine whether two events are order independent. The importance of these results is that no algorithmic solution exists to determine some of the most vital properties of simulation models. Another implication is that the *"rules of thumb"* offered by Schruben (1983) for "event initialization" and "simultaneous event precedence" as well as procedures constructed by Sargent (1988) are the best heuristic tools we can hope for. In other words, most of the rules given for model analysis contain necessary, but not sufficient, conditions for their application.

5. CONCLUDING REMARKS

As Nance (1981) points out, "... simulation still carries the label of an expensive and uncertain problem solving technique that represents 'the court of last resort.' (...) The concern with simulation extends beyond those who use the technique or those who underwrite the project costs. Others, admittedly a much smaller group, are concerned with the lack of recognition of a fundamental structure - a theory - after so many years of practice."

This paper represents a discrete event simulation modeling structure. The graphs described here offer a general and practical means of mathematically defining a discrete event dynamical system and its simulation model as a language free structure. Furthermore, Simulation Graphs are designed to offer the decision power of high level simulation languages as well as the modeling (expressive) power of the abstract formalisms.

Acknowledgements: *The authors are grateful to David Heath, David Tate and Arnold Buss for their comments on an earlier version of this work.*

6. REFERENCES

[1] Bratley, P. Fox, B. and Schrage, L. (1987) *A Guide to Simulation* (2nd. Edition) Springer-Verlag

[2] Bondy, J.A. and Murty, U.S.R. (1976) *Graph Theory with Applications*. North-Holland. New York

[3] Consolidated Analysis Centers, Inc. (1976) *Simscript II.5 Reference Handbook* Los Angeles, CA

[4] Evans, G.W. Wallace, G.F. Sutherland, G.L. (1967) *Simulation Using Digital Computers* Prentice Hall. Englewood Cliffs, N.J.

[5] Fleming, W.H. (1988) *Report of the Panel on Future Directions in Control Theory Mathematical Perspective* Society for Industrial and Applied Mathematics. Philadelphia, PA

[6] Gordon, G. (1969) *System Simulation* Prentice Hall. Englewood Cliffs, N.J.

[7] Gordon, G. (1979) *The Design of the GPSS Language* Current Issues in Computer Simulation (Adam and Dogramaci, eds.) Academic Press. New York

[8] Hopcroft, J.E. and Ullman, J.D. (1979) *Introduction to Automata Theory, Languages and Computation*. Addison Wesley. Reading, MA.

[9] Law, A.M. and Kelton, W.D. (1989) *Simulation Modeling and Analysis* (2nd. Edition) (forthcoming)

[10] Nance, R.E. (1981) *The Time and Relationships in Simulation Modeling*. Comm ACM. Vol. 24.4

[11] Overstreet, C.M. (1982) *Model Specification and Analysis for Discrete Event Simulations* Ph.D Dissertation. Virginia Tech. Blacksburg,

VA.

[12] Pegden, C.D. (1985) *Introduction to SIMAN* Systems Modeling Corporation. State College, PA

[13] Peterson, J.L. (1977) *Petri Nets* Computing Surveys. Vol.9.3

[14] Pritsker, A.A.B. (1984) *Introduction to Simulation and SLAM II* John Wiley and Sons, New York

[15] Pritsker, A.A.B. and Standridge, C.R. (1987) *The Extended Simulation Support System* John Wiley, New York

[16] Radiya A. and Sargent, R.G. (1987) *A New Formalism for Discrete Event Simulation* Proceedings of 1987 Winter Simulation Conference. (Thesen, Grant and Kelton, eds.) Atlanta, GA

[17] Roberts, S.D. (1983) *Simulation Modeling and Analysis with INSIGHT* Regenstrief Institute. Indianapolis, IN

[18] Sargent, R.G. (1988) *Event Graph Modeling For Simulation With an Application to Flexible Manufacturing Systems* Management Science. Vol.34.10

[19] Schriber, T. (1974) *Simulation Using GPSS* John Wiley New York

[20] Schruben, L.W. (1982) *System Modeling With Event Graphs* Technical Report #498. School of Operations Research and Industrial Engineering, Cornell University

[21] Schruben, L.W. (1983) *Simulation Modeling with Event Graphs.* Comm ACM. Vol. 29.11

[22] Schruben, L.W. and Briskman, D. (1988) *Teaching Simulation with SiGMA (Σ)* Proceedings of the 1988 Winter Simulation Conference (Abrahms, Haigh and Comfort, eds.) San Diego, CA

[23] Schruben, L.W. and Yucesan, E. (1987) *On The Generality of*

Simulation Graphs. Technical Report #773. School of Operations Research and Industrial Engineering, Cornell University

[24] Som, T.K. and Sargent, R.G. (1988) *A Formal Development of Event Graphs as an Aid to Structured and Efficient Simulation Programs* Working Paper #87-004 Simulation Research Group, Syracuse University Syracuse, N.Y.

[25] Suri, R. (1984) *A General Theory for Infinitesimal Perturbation Analysis of Discrete Event Dynamic Systems* IEEE Transactions on Autom. Control

[26] Systems Modeling Corporation (1985) *CINEMA User's Manual* State College, PA

[27] Torn, A.A. (1981) *Simulation Graphs: A General Tool For Modeling Simulation Designs* Simulation. Vol. 37 (December)

[28] Zeigler, B.P. (1984) *Multifacetted Modelling and Discrete-Event Simulation*. Academic Press. New York

INSEAD WORKING PAPERS SERIES

1986

86/01	Arnoud DE MEYER	"The R & D/Production interface".
86/02	Philippe A. NAERT Marcel WEVERBERGH and Guido VERSWIJVEL	"Subjective estimation in integrating communication budget and allocation decisions: a case study", January 1986.
86/03	Michael BRIMM	"Sponsorship and the diffusion of organizational innovation: a preliminary view".
86/04	Spyros MAKRIDAKIS and Michèle HIBON	"Confidence intervals: an empirical investigation for the series in the M-Competition" .
86/05	Charles A. WYPLOSZ	"A note on the reduction of the workweek", July 1985.
86/06	Francesco GIAVAZZI, Jeff R. SHEEN and Charles A. WYPLOSZ	"The real exchange rate and the fiscal aspects of a natural resource discovery", Revised version: February 1986.
86/07	Douglas L. MacLACHLAN and Spyros MAKRIDAKIS	"Judgmental biases in sales forecasting", February 1986.
86/08	José de la TORRE and David H. NECKAR	"Forecasting political risks for international operations", Second Draft: March 3, 1986.
86/09	Philippe C. HASPESLAGH	"Conceptualizing the strategic process in diversified firms: the role and nature of the corporate influence process", February 1986.
86/10	R. MOENART, Arnoud DE MEYER, J. BARBE and D. DESCHOOLMEESTER.	"Analysing the issues concerning technological de-maturity".
86/11	Philippe A. NAERT and Alain BULTEZ	"From "Lydiametry" to "Pinkhamization": misspecifying advertising dynamics rarely affects profitability".
86/12	Roger BETANCOURT and David GAUTSCHI	"The economics of retail firms", Revised April 1986.
86/13	S.P. ANDERSON and Damien J. NEVEN	"Spatial competition à la Cournot".
86/14	Charles WALDMAN	"Comparaison internationale des marges brutes du commerce", June 1985.
86/15	Mihkel TOMBAK and Arnoud DE MEYER	"How the managerial attitudes of firms with FMS differ from other manufacturing firms: survey results". June 1986.
86/16	B. Espen ECKBO and Herwig M. LANGOHR	"Les primes des offres publiques, la note d'information et le marché des transferts de contrôle des sociétés".
86/17	David B. JEMISON	"Strategic capability transfer in acquisition integration", May 1986.
86/18	James TEBOUL and V. MALLERET	"Towards an operational definition of services", 1986.
86/19	Rob R. WEITZ	"Nostradamus: a knowledge-based forecasting advisor".
86/20	Albert CORHAY, Gabriel HAWAWINI and Pierre A. MICHEL	"The pricing of equity on the London stock exchange: seasonality and size premium", June 1986.
86/21	Albert CORHAY, Gabriel A. HAWAWINI and Pierre A. MICHEL	"Risk-premia seasonality in U.S. and European equity markets", February 1986.
86/22	Albert CORHAY, Gabriel A. HAWAWINI and Pierre A. MICHEL	"Seasonality in the risk-return relationships some international evidence", July 1986.
86/23	Arnoud DE MEYER	"An exploratory study on the integration of information systems in manufacturing", July 1986.
86/24	David GAUTSCHI and Vithala R. RAO	"A methodology for specification and aggregation in product concept testing", July 1986.
86/25	H. Peter GRAY and Ingo WALTER	"Protection", August 1986.
86/26	Barry EICHENGREEN and Charles WYPLOSZ	"The economic consequences of the Franc Poincare", September 1986.
86/27	Karel COOL and Ingemar DIERICKX	"Negative risk-return relationships in business strategy: paradox or truism?", October 1986.
86/28	Manfred KETS DE VRIES and Danny MILLER	"Interpreting organizational texts.
86/29	Manfred KETS DE VRIES	"Why follow the leader?".
86/30	Manfred KETS DE VRIES	"The succession game: the real story.
86/31	Arnoud DE MEYER	"Flexibility: the next competitive battle", October 1986.
86/31	Arnoud DE MEYER, Jinichiro NAKANE, Jeffrey G. MILLER and Kasra FERDOWS	"Flexibility: the next competitive battle", Revised Version: March 1987
86/32	Karel COOL and Dan SCHENDEL	Performance differences among strategic group members", October 1986.

86/33	Ernst BALTENSPERGER and Jean DERMINE	"The role of public policy in insuring financial stability: a cross-country, comparative perspective", August 1986, Revised November 1986.
86/34	Philippe HASPESLAGH and David JEMISON	"Acquisitions: myths and reality", July 1986.
86/35	Jean DERMINE	"Measuring the market value of a bank, a primer", November 1986.
86/36	Albert CORHAY and Gabriel HAVAVINI	"Seasonality in the risk-return relationship: some international evidence", July 1986.
86/37	David GAUTSCHI and Roger BETANCOURT	"The evolution of retailing: a suggested economic interpretation".
86/38	Gabriel HAVAVINI	"Financial innovation and recent developments in the French capital markets", Updated: September 1986.
86/39	Gabriel HAVAVINI Pierre MICHEL and Albert CORHAY	"The pricing of common stocks on the Brussels stock exchange: a re-examination of the evidence", November 1986.
86/40	Charles WYPLOSZ	"Capital flows liberalization and the EMS, a French perspective", December 1986.
86/41	Kasra FERDOWS and Wickham SKINNER	"Manufacturing in a new perspective", July 1986.
86/42	Kasra FERDOWS and Per LINDBERG	"FMS as indicator of manufacturing strategy", December 1986.
86/43	Damien NEVEN	"On the existence of equilibrium in hotelling's model", November 1986.
86/44	Ingemar DIERICKX Carmen MATUTES and Damien NEVEN	"Value added tax and competition", December 1986.

1987

87/01	Manfred KETS DE VRIES	"Prisoners of leadership".
87/02	Claude VIALLET	"An empirical investigation of international asset pricing", November 1986.
87/03	David GAUTSCHI and Vithala RAO	"A methodology for specification and aggregation in product concept testing", Revised Version: January 1987.
87/04	Sumantra GHOSHAL and Christopher BARTLETT	"Organizing for innovations: case of the multinational corporation", February 1987.
87/05	Arnoud DE MEYER and Kasra FERDOWS	"Managerial focal points in manufacturing strategy", February 1987.
87/06	Arun K. JAIN, Christian PINSON and Naresh K. MALHOTRA	"Customer loyalty as a construct in the marketing of banking services", July 1986.
87/07	Rolf BANZ and Gabriel HAVAVINI	"Equity pricing and stock market anomalies", February 1987.
87/08	Manfred KETS DE VRIES	"Leaders who can't manage", February 1987.
87/09	Lister VICKERY, Mark PILKINGTON and Paul READ	"Entrepreneurial activities of European MBAs", March 1987.
87/10	André LAURENT	"A cultural view of organizational change", March 1987
87/11	Robert FILDES and Spyros MAKRIDAKIS	"Forecasting and loss functions", March 1987.
87/12	Fernando BARTOLOME and André LAURENT	"The Janus Head: learning from the superior and subordinate faces of the manager's job", April 1987.
87/13	Sumantra GHOSHAL and Nitin NOHRIA	"Multinational corporations as differentiated networks", April 1987.
87/14	Landis GABEL	"Product Standards and Competitive Strategy: An Analysis of the Principles", May 1987.
87/15	Spyros MAKRIDAKIS	"METAFORECASTING: Ways of improving Forecasting. Accuracy and Usefulness", May 1987.
87/16	Susan SCHNEIDER and Roger DUNBAR	"Takeover attempts: what does the language tell us?, June 1987.
87/17	André LAURENT and Fernando BARTOLOME	"Managers' cognitive maps for upward and downward relationships", June 1987.
87/18	Reinhard ANGELMAR and Christoph LIEBSCHER	"Patents and the European biotechnology lag: a study of large European pharmaceutical firms", June 1987.
87/19	David BEGG and Charles WYPLOSZ	"Why the EMS? Dynamic games and the equilibrium policy regime, May 1987.
87/20	Spyros MAKRIDAKIS	"A new approach to statistical forecasting", June 1987.
87/21	Susan SCHNEIDER	"Strategy formulation: the impact of national culture", Revised: July 1987.
87/22	Susan SCHNEIDER	"Conflicting ideologies: structural and motivational consequences", August 1987.
87/23	Roger BETANCOURT David GAUTSCHI	"The demand for retail products and the household production model: new views on complementarity and substitutability".

87/24	C.B. DERR and André LAURENT	"The internal and external careers: a theoretical and cross-cultural perspective", Spring 1987.
87/25	A. K. JAIN, N. K. MALHOTRA and Christian PINSON	"The robustness of MDS configurations in the face of incomplete data", March 1987, Revised: July 1987.
87/26	Roger BETANCOURT and David GAUTSCHI	"Demand complementarities, household production and retail assortments", July 1987.
87/27	Michael BURDA	"Is there a capital shortage in Europe?", August 1987.
87/28	Gabriel HAVAVINI	"Controlling the interest-rate risk of bonds: an introduction to duration analysis and immunization strategies", September 1987.
87/29	Susan SCHNEIDER and Paul SHRIVASTAVA	"Interpreting strategic behavior: basic assumptions themes in organizations", September 1987
87/30	Jonathan HAMILTON V. Bentley MACLEOD and J. F. THISSE	"Spatial competition and the Core", August 1987.
87/31	Martine QUINZII and J. F. THISSE	"On the optimality of central places", September 1987.
87/32	Arnoud DE MEYER	"German, French and British manufacturing strategies less different than one thinks", September 1987.
87/33	Yves DOZ and Amy SHUEN	"A process framework for analyzing cooperation between firms", September 1987.
87/34	Kasra FERDOVS and Arnoud DE MEYER	"European manufacturers: the dangers of complacency. Insights from the 1987 European manufacturing futures survey, October 1987.
87/35	P. J. LEDERER and J. F. THISSE	"Competitive location on networks under discriminatory pricing", September 1987.
87/36	Manfred KETS DE VRIES	"Prisoners of leadership", Revised version October 1987.
87/37	Landis GABEL	"Privatization: its motives and likely consequences", October 1987.
87/38	Susan SCHNEIDER	"Strategy formulation: the impact of national culture", October 1987.
87/39	Manfred KETS DE VRIES	"The dark side of CEO succession", November 1987
87/40	Carmen MATUTES and Pierre REGIBEAU	"Product compatibility and the scope of entry", November 1987
87/41	Gavriel HAVAVINI and Claude VIALLET	"Seasonality, size premium and the relationship between the risk and the return of French common stocks", November 1987
87/42	Damien NEVEN and Jacques-F. THISSE	"Combining horizontal and vertical differentiation: the principle of max-min differentiation", December 1987
87/43	Jean GABSZEVICZ and Jacques-F. THISSE	"Location", December 1987
87/44	Jonathan HAMILTON, Jacques-F. THISSE and Anita VESKAMP	"Spatial discrimination: Bertrand vs. Cournot in a model of location choice", December 1987
87/45	Karel COOL, David JEMISON and Ingemar DIERICKX	"Business strategy, market structure and risk-return relationships: a causal interpretation", December 1987.
87/46	Ingemar DIERICKX and Karel COOL	"Asset stock accumulation and sustainability of competitive advantage", December 1987.

1988

88/01	Michael LAWRENCE and Spyros MAKRIDAKIS	"Factors affecting judgemental forecasts and confidence intervals", January 1988.
88/02	Spyros MAKRIDAKIS	"Predicting recessions and other turning points", January 1988.
88/03	James TEBOUL	"De-industrialize service for quality", January 1988.
88/04	Susan SCHNEIDER	"National vs. corporate culture: implications for human resource management", January 1988.
88/05	Charles WYPLOSZ	"The swinging dollar: is Europe out of step?", January 1988.
88/06	Reinhard ANGELMAR	"Les conflits dans les canaux de distribution", January 1988.
88/07	Ingemar DIERICKX and Karel COOL	"Competitive advantage: a resource based perspective", January 1988.
88/08	Reinhard ANGELMAR and Susan SCHNEIDER	"Issues in the study of organizational cognition", February 1988.
88/09	Bernard SINCLAIR-DESGAGNÉ	"Price formation and product design through bidding", February 1988.
88/10	Bernard SINCLAIR-DESGAGNÉ	"The robustness of some standard auction game forms", February 1988.
88/11	Bernard SINCLAIR-DESGAGNÉ	"When stationary strategies are equilibrium bidding strategy: The single-crossing property", February 1988.

88/12	Spyros MAKRIDAKIS	"Business firms and managers in the 21st century", February 1988
88/13	Manfred KETS DE VRIES	"Alexithymia in organizational life: the organization man revisited", February 1988.
88/14	Alain NOEL	"The interpretation of strategies: a study of the impact of CEOs on the corporation", March 1988.
88/15	Anil DEOLALIKAR and Lars-Hendrik ROLLER	"The production of and returns from industrial innovation: an econometric analysis for a developing country", December 1987.
88/16	Gabriel HAVAWINI	"Market efficiency and equity pricing: international evidence and implications for global investing", March 1988.
88/17	Michael BURDA	"Monopolistic competition, costs of adjustment and the behavior of European employment", September 1987.
88/18	Michael BURDA	"Reflections on "Wait Unemployment" in Europe", November 1987, revised February 1988.
88/19	M.J. LAWRENCE and Spyros MAKRIDAKIS	"Individual bias in judgements of confidence", March 1988.
88/20	Jean DERMINE, Damien NEVEN and J.F. THISSE	"Portfolio selection by mutual funds, an equilibrium model", March 1988.
88/21	James TEBOUL	"De-industrialize service for quality", March 1988 (88/03 Revised).
88/22	Lars-Hendrik RÖLLER	"Proper Quadratic Functions with an Application to AT&T", May 1987 (Revised March 1988).
88/23	Sjur Didrik FLAM and Georges ZACCOUR	"Equilibres de Nash-Cournot dans le marché européen du gaz: un cas où les solutions en boucle ouverte et en feedback coïncident", Mars 1988
88/24	B. Espen ECKBO and Herwig LANGOHR	"Information disclosure, means of payment, and takeover premia. Public and Private tender offers in France", July 1985, Sixth revision, April 1988.
88/25	Everette S. GARDNER and Spyros MAKRIDAKIS	"The future of forecasting", April 1988.
88/26	Sjur Didrik FLAM and Georges ZACCOUR	"Semi-competitive Cournot equilibrium in multistage oligopolies", April 1988.
88/27	Murugappa KRISHNAN Lars-Hendrik RÖLLER	"Entry game with resalable capacity", April 1988.
88/28	Sumantra GHOSHAL and C. A. BARTLETT	"The multinational corporation as a network: perspectives from interorganizational theory", May 1988
88/29	Naresh K. MALHOTRA, Christian PINSON and Arun K. JAIN	"Consumer cognitive complexity and the dimensionality of multidimensional scaling configurations", May 1988.
88/30	Catherine C. ECKEL and Theo VERMAELEN	"The financial fallout from Chernobyl: risk perceptions and regulatory response", May 1988.
88/31	Sumantra GHOSHAL and Christopher BARTLETT	"Creation, adoption, and diffusion of innovations by subsidiaries of multinational corporations", June 1988.
88/32	Kasra FERDOWS and David SACKRIDER	"International manufacturing: positioning plants for success", June 1988.
88/33	Mihkel M. TOMBAK	"The importance of flexibility in manufacturing", June 1988.
88/34	Mihkel M. TOMBAK	"Flexibility: an important dimension in manufacturing", June 1988.
88/35	Mihkel M. TOMBAK	"A strategic analysis of investment in flexible manufacturing systems", July 1988.
88/36	Vikas TIBREWALA and Bruce BUCHANAN	"A Predictive Test of the NBD Model that Controls for Non-stationarity", June 1988.
88/37	Murugappa KRISHNAN Lars-Hendrik RÖLLER	"Regulating Price-Liability Competition To Improve Welfare", July 1988.
88/38	Manfred KETS DE VRIES	"The Motivating Role of Envy : A Forgotten Factor in Management, April 88.
88/39	Manfred KETS DE VRIES	"The Leader as Mirror : Clinical Reflections", July 1988.
88/40	Josef LAKONISHOK and Theo VERMAELEN	"Anomalous price behavior around repurchase tender offers", August 1988.
88/41	Charles WYPLOSZ	"Assymetry in the EMS: intentional or systemic?", August 1988.
88/42	Paul EVANS	"Organizational development in the transnational enterprise", June 1988.
88/43	B. SINCLAIR-DESGAGNE	"Group decision support systems implement Bayesian rationality", September 1988.
88/44	Essam MAHMOUD and Spyros MAKRIDAKIS	"The state of the art and future directions in combining forecasts", September 1988.
88/45	Robert KORAJCZYK and Claude VIALLET	"An empirical investigation of international asset pricing", November 1986, revised August 1988.
88/46	Yves DOZ and Amy SHUEN	"From intent to outcome: a process framework for partnerships", August 1988.

88/47	Alain BULTEZ, Els GIJSBRECHTS, Philippe NAERT and Piet VANDEN ABEELE	"Asymmetric cannibalism between substitute items listed by retailers", September 1988.
88/48	Michael BURDA	"Reflections on 'Wait unemployment' in Europe, II", April 1988 revised September 1988.
88/49	Nathalie DIERKENS	"Information asymmetry and equity issues", September 1988.
88/50	Rob WEITZ and Arnoud DE MEYER	"Managing expert systems: from inception through updating", October 1987.
88/51	Rob WEITZ	"Technology, work, and the organization: the impact of expert systems", July 1988.
88/52	Susan SCHNEIDER and Reinhard ANGELMAR	"Cognition and organizational analysis: who's minding the store?", September 1988.
88/53	Manfred KETS DE VRIES	"Whatever happened to the philosopher-king: the leader's addiction to power, September 1988.
88/54	Lars-Hendrik RÖLLER and Mihkel M. TOMBAK	"Strategic choice of flexible production technologies and welfare implications", October 1988
88/55	Peter BOSSAERTS and Pierre HILLION	"Method of moments tests of contingent claims asset pricing models", October 1988.
88/56	Pierre HILLION	"Size-sorted portfolios and the violation of the random walk hypothesis: Additional empirical evidence and implication for tests of asset pricing models", June 1988.
88/57	Wilfried VANHONACKER and Lydia PRICE	"Data transferability: estimating the response effect of future events based on historical analogy", October 1988.
88/58	B. SINCLAIR-DESGAGNE and Mihkel M. TOMBAK	"Assessing economic inequality", November 1988.
88/59	Martin KILDUFF	"The interpersonal structure of decision making: a social comparison approach to organizational choice", November 1988.
88/60	Michael BURDA	"Is mismatch really the problem? Some estimates of the Chelwood Gate II model with US data", September 1988.
88/61	Lars-Hendrik RÖLLER	"Modelling cost structure: the Bell System revisited", November 1988.
88/62	Cynthia VAN HULLE, Theo VERMAELEN and Paul DE WOUTERS	"Regulation, taxes and the market for corporate control in Belgium", September 1988.
88/63	Fernando NASCIMENTO and Wilfried R. VANHONACKER	"Strategic pricing of differentiated consumer durables in a dynamic duopoly: a numerical analysis", October 1988.
88/64	Kasra FERDOWS	"Charting strategic roles for international factories", December 1988.
88/65	Arnoud DE MEYER and Kasra FERDOWS	"Quality up, technology down", October 1988.
88/66	Nathalie DIERKENS	"A discussion of exact measures of information assymetry: the example of Myers and Majluf model or the importance of the asset structure of the firm", December 1988.
88/67	Paul S. ADLER and Kasra FERDOWS	"The chief technology officer", December 1988.

1989

89/01	Joyce K. BYRER and Tawfik JELASSI	"The impact of language theories on DSS dialog", January 1989.
89/02	Louis A. LE BLANC and Tawfik JELASSI	"DSS software selection: a multiple criteria decision methodology", January 1989.
89/03	Beth H. JONES and Tawfik JELASSI	"Negotiation support: the effects of computer intervention and conflict level on bargaining outcome", January 1989.
89/04	Kasra FERDOWS and Arnoud DE MEYER	"Lasting improvement in manufacturing performance: In search of a new theory", January 1989.
89/05	Martin KILDUFF and Reinhard ANGELMAR	"Shared history or shared culture? The effects of time, culture, and performance on institutionalization in simulated organizations", January 1989.
89/06	Mihkel M. TOMBAK and B. SINCLAIR-DESGAGNE	"Coordinating manufacturing and business strategies: I", February 1989.
89/07	Damien J. NEVEN	"Structural adjustment in European retail banking. Some view from industrial organisation", January 1989.
89/08	Arnoud DE MEYER and Hellmut SCHÜTTE	"Trends in the development of technology and their effects on the production structure in the European Community", January 1989.
89/09	Damien NEVEN, Carmen MATUTES and Marcel CORSTJENS	"Brand proliferation and entry deterrence", February 1989.
89/10	Nathalie DIERKENS, Bruno GERARD and Pierre HILLION	"A market based approach to the valuation of the assets in place and the growth opportunities of the firm", December 1988.

89/11	Manfred KETS DE VRIES and Alain NOEL	"Understanding the leader-strategy interface: application of the strategic relationship interview method", February 1989.
89/12	Wilfried VANHONACKER	"Estimating dynamic response models when the data are subject to different temporal aggregation", January 1989.
89/13	Manfred KETS DE VRIES	"The impostor syndrome: a disquieting phenomenon in organizational life", February 1989.
89/14	Reinhard ANGELMAR	"Product innovation: a tool for competitive advantage", March 1989.
89/15	Reinhard ANGELMAR	"Evaluating a firm's product innovation performance", March 1989.
89/16	Wilfried VANHONACKER, Donald LEHMANN and Fareena SULTAN	"Combining related and sparse data in linear regression models", February 1989.
89/17	Gilles AMADO, Claude FAUCHEUX and André LAURENT	"Changement organisationnel et réalités culturelles: contrastes franco-américains", March 1989.
89/18	Srinivasan BALAK-RISHNAN and Mitchell KOZA	"Information asymmetry, market failure and joint-ventures: theory and evidence", March 1989
89/19	Wilfried VANHONACKER, Donald LEHMANN and Fareena SULTAN	"Combining related and sparse data in linear regression models", Revised March 1989
89/20	Wilfried VANHONACKER and Russell WINER	"A rational random behavior model of choice", Revised March 1989
89/21	Arnoud de MEYER and Kasra FERDOWS	"Influence of manufacturing improvement programmes on performance", April 1989
89/22	Manfred KETS DE VRIES and Sydney PERZOW	"What is the role of character in psychoanalysis? April 1989
89/23	Robert KORAJCZYK and Claude VIALLET	"Equity risk premia and the pricing of foreign exchange risk" April 1989
89/24	Martin KILDUFF and Mitchel ABOLAFIA	"The social destruction of reality: Organisational conflict as social drama" April 1989
89/25	Roger BETANCOURT and David GAUTSCHI	"Two essential characteristics of retail markets and their economic consequences" March 1989
89/26	Charles BEAN, Edmond MALINVAUD, Peter BERNHOLZ, Francesco GIAVAZZI and Charles WYPLOSZ	"Macroeconomic policies for 1992: the transition and after", April 1989
89/27	David KRACKHARDT and Martin KILDUFF	"Friendship patterns and cultural attributions: the control of organizational diversity", April 1989
89/28	Martin KILDUFF	"The interpersonal structure of decision making: a social comparison approach to organizational choice", Revised April 1989
89/29	Robert GOGEL and Jean-Claude LARRECHE	"The battlefield for 1992: product strength and geographic coverage", May 1989
89/30	Lars-Hendrik ROLLER and Mihkel M. TOMBAK	"Competition and Investment in Flexible Technologies", May 1989
89/31	Michael C. BURDA and Stefan GERLACH	"Durables and the US Trade Deficit", May 1989
89/32	Peter HAUG and Tawfik JELASSI	"Application and evaluation of a multi-criteria decision support system for the dynamic selection of U.S. manufacturing locations", May 1989
89/33	Bernard SINCLAIR-DESGAGNE	"Design flexibility in monopsonistic industries", May 1989
89/34	Sumantra GHOSHAL and Nittin NOHRIA	"Requisite variety versus shared values: managing corporate-division relationships in the M-Form organisation", May 1989
89/35	Jean DERMINE and Pierre HILLION	"Deposit rate ceilings and the market value of banks: The case of France 1971-1981", May 1989
89/36	Martin KILDUFF	"A dispositional approach to social networks: the case of organizational choice", May 1989
89/37	Manfred KETS DE VRIES	"The organisational fool: balancing a leader's hubris", May 1989
89/38	Manfrd KETS DE VRIES	"The CEO blues", June 1989
89/39	Robert KORAJCZYK and Claude VIALLET	"An empirical investigation of international asset pricing", (Revised June 1989)
89/40	Balaji CHAKRAVARTHY	"Management systems for innovation and productivity", June 1989
89/41	B. SINCLAIR-DESGAGNE and Nathalie DIERKENS	"The strategic supply of precisions", June 1989
89/42	Robert ANSON and Tawfik JELASSI	"A development framework for computer-supported conflict resolution", July 1989
89/43	Michael BURDA	"A note on firing costs and severance benefits in equilibrium unemployment", June 1989
89/44	Balaji CHAKRAVARTHY and Peter LORANGE	"Strategic adaptation in multi-business firms", June 1989
89/45	Rob WEITZ and Arnoud DE MEYER	"Managing expert systems: a framework and case study", June 1989

89/46	Marcel CORSTJENS, Carmen MATUTES and Damien NEVEN	"Entry Encouragement", July 1989
89/47	Manfred KETS DE VRIES and Christine MEAD	"The global dimension in leadership and organization: issues and controversies", April 1989
89/48	Damien NEVEN and Lars-Hendrik ROLLER	"European integration and trade flows", August 1989
89/49	Jean DERMINE	"Home country control and mutual recognition", July 1989
89/50	Jean DERMINE	"The specialization of financial institutions, the EEC model", August 1989
89/51	Spyros MAKRIDAKIS	"Sliding simulation: a new approach to time series forecasting", July 1989
89/52	Arnoud DE MEYER	"Shortening development cycle times: a manufacturer's perspective", August 1989
89/53	Spyros MAKRIDAKIS	"Why combining works?", July 1989
89/54	S. BALAKRISHNAN and Mitchell KOZA	"Organisation costs and a theory of joint ventures", September 1989
89/55	H. SCHUTTE	"Euro-Japanese cooperation in information technology", September 1989
89/56	Wilfried VANHONACKER and Lydia PRICE	"On the practical usefulness of meta-analysis results", September 1989
89/57	Taekwon KIM, Lars-Hendrik ROLLER and Mihkel TOMBAK	"Market growth and the diffusion of multiproduct technologies", September 1989
89/58	Lars-Hendrik ROLLER and Mihkel TOMBAK	"Strategic aspects of flexible production technologies", October 1989
89/59 (OB)	Manfred KETS DE VRIES, Daphna ZEVADI, Alain NOEL and Mihkel TOMBAK	"Locus of control and entrepreneurship: a three-country comparative study", October 1989