THE UNIVERSITY OF MICHIGAN
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Computer and Communication Sciences Department

SYSTEMS SIMULATEABLE BY THE DIGITAL COMPUTER

Part 1: Discrete Event Representable Models

Bernard P. Zeigler

The Logic of Computers Group
and
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel

April 1977

Technical Report No. 199

with assistance from:

National Science Foundation
Grant No. DCR71-01997
Washington, D.C.

SYSTEMS SIMULATEABLE BY THE DIGITAL COMPUTER

Part 1: Discrete Event Representable Models*

by

Bernard P. Zeigler

Department of Applied Mathematics

Weizmann Institute of Science

Rehovot, Israel

* This work was partially supported by NSF Grant No. DCR71-01997.

SYSTEMS SIMULATEABLE BY THE DIGITAL COMPUTER

Part 1: Discrete Event Representable Models

Contents

Abstract.

Discrete Event System Specifications (DEVS) formalize the concepts underlying discrete event simulation languages. More broadly, they embody the fundamental constraint of digital computer simulation that a model state can be updated only a finite number of times in a finite interval of time. This constraint, together with the fact that in the cases of paramount interest, the computer is given a network, rather than a global, system specification, limits the class of systems which can be faithfully simulated by a digital computer. In this paper, we formulate a notion of DEVS simulation and characterize the class of systems simulateable in this manner. In a subsequent continuation study, we shall show how the present formulation embodies both the above mentioned constraints and leads to a network level characterization of digital simulateability. The given characterization provides a canonical means of discrete event model construction and an example, drawn from a large scale simulation study is given to illustrate its applicability.

SYSTEMS SIMULATEABLE BY THE DIGITAL COMPUTER:

Part 1: Discrete Event Representable Models

Introduction.

A fundamental constraint on digital computer simulation is that the sequentially acting computer can update a model state only a finite number of times in a finite interval of model time.

Discrete time, including automaton, formalisms embody this constraint in the form of an assumed underlying fixed time step. Asynchronous model formalisms relax the latter constraint, but do not explicitly represent the computer's ability to control the time step size. Discrete event formalisms (Zeigler, 1976) however, embody a continuous time base and formalize the scheduling of model updates realizable by discrete event simulation languages such as SIMSCRIPT, SIMULA, etc. The advantages of such formalisms in terms of conceptual expression and computational efficiency have been previously noted (Zeigler and Barto, 1977; Hogeweg, 1975; Babich et al, 1975). In this paper, we initiate the characterization of systems simulateable by discrete event models, which we take to represent the maximum capability of digital simulation.

To do this, we must recognize that in the cases of paramount interest, the computer is given only local descriptions and composes these to generate global behavior. It is the combination of the finite update and local description constraints that limits the class of systems faithfully simulateable by digital computer (Zeigler, 1976, Chapter 5). In the (present) first part of this work, we formulate discrete event simulateability from a local point of view in such a way as to facilitate the more general network characterization to be given in Part 2. We rely upon system theoretic concepts previously developed (Zeigler, 1976, Chapters 9 & 10). For convenience, these are briefly outlined in Appendices 1 and 3.

In particular, the notion of iterative system specification has been defined in order to characterize the computer's ability to iteratively generate a model's state and output trajectories, given a decomposition of its received input segment into a sequence of finite length generators. The discrete event system specification (DEVS) was developed as a special case of the iterative specification concept. Roughly, in a DEVS, external events arrive according to the finite number per finite interval rule. These cause the scheduling and rescheduling of internal events which effectuate jump-like state transitions. Between successive events, no activity takes place.

Thus the natural DEVS input generators are finite length segments consisting either of no events, or of one event arriving at the beginning of the segment.

In this paper, we show that under a realistic encoding notion, the DEVS generator sets canonically represent the full class of admissible generating sets for iterative specification. The encoding of input time functions into DEVS segments thus involves identifying event times (segmentation points) and event names (characteristics of the enclosed segments). Moreover, the fact that admissibility is also shown to be necessary for DEVS encodability, reduces the study of DEVS simulation of arbitrary systems to that of DEVS simulation of iteratively specifiable systems.

We then proceed to define an appropriate morphism for DEVS simulation and characterize the class of iterative specifications simulateable in this way. Besides giving the necessary and sufficient conditions for DEVS simulateability, this characterization has as a practical side effect, a canonical procedure for constructing discrete event models for systems satisfying the prerequisite conditions.

The efficacy of this approach was demonstrated in the design of a spatially structured ecosystem model (Zeigler, 1977) whose running time efficiency brought the study of such systems well within feasible limits (this feasibility contrasts strongly with the simulation time requirements of the original system of differential equations). An example drawn from this simulation model is given here.

The DEVS simulator keeps track of a model's state, but can update it at only external and internal event times. Internal events are scheduled according to predicted boundary crossing of model state trajectories. Necssary conditions for DEVS simulateability are thus shown to require autonomous operation of the model between successive external events and the existence of a partitioning of its state space which enables admissible segmentation of the state trajectories.

Realistic restrictions on DEVS simulation over and above the essential finite update constraint, are paralleled in corresponding restrictions on the DEVS simulateable systems. But the basic principles for converting models to DEVS form are shown to remain unchanged.

It is of theoretical interest to point out non DEVS simulateable systems since these establish inherent limitations on digital simulation. While we have not been able to construct such counter-examples in the unrestricted case, we do show here that the integrator (perhaps the most simple continuous system) is not simulateable by finite dimensional discrete event systems. This is a satisfying result in not contradicting the basic presupposition underlying all numerical methods of differential equation solution, but the inability to extend this result to the unrestricted DEVS case remains a puzzle. It should be noted that non-DEVS simulateability of the integrator does not imply the same is true for differential equation systems, which are integrators connected by instantaneous functions.

As indicated, the present treatment deals with DEVS simulation at the system level. The internal-external event distinction which we make here is however, crucial to the characterization of DEVS simulation at the network level. At this level, the constraint on local description of models for computer simulation may be formulated, as will be shown in Part 2, (Zeigler, In Preparation).

USEFUL ADMISSIBLY GENERATED SETS

We develop examples of admissibly generated sets which will be useful in the sequel.

1. Discrete Event Segments Ω_{DEVS}

   See Appendix 1.

2. Cover Generated Segments

Let $\Omega \subseteq (X,T)$ be a semigroup closed under right and left segmentation. A cover of X is a family π of blocks (subsets) whose union is X ; π is a finite intersection cover if every subset of every block of π is contained in at most a finite number of blocks of π .

By range $(\bar{\omega})$ we shall mean the set $\{\omega(t)/0 < t < \ell(\omega)\}$. For each subset B of π , define

$$\Gamma_B = \{\omega/\omega \in \Omega \ , \text{range } (\bar{\omega}) \subseteq B\}$$

i.e., Γ_B is the set of segments ω of Ω , whose values at points in the interior of dom(ω) lie in B . Let Γ_π be the union of the Γ_B, $B \in \pi$.

Proposition 1.

Γ_π admissibly generates Γ_π^+ .

Proof. Let $\omega = \omega_1, \omega_2 \ldots \omega_n$, each $\omega_i \in \Gamma_\pi$. Then range $(\bar{\omega}_1)$ is included in at least one B in π . By the finite intersection property, range $(\bar{\omega}_1)$ is included in at most a finite number of sets $B_1, \ldots, B_m$ in π . For each $i = 1, \ldots, m$, we claim $t^*(B_i) = \max\{t/\text{range } (\bar{\omega}_{t>}) \subseteq B_i\}$ exists. To see this let $s_1 \leq s_2 \leq s_3 \leq \ldots$ be a sequence with $s_1 = \ell(\omega_1)$ such that either

a) for some j , $s_j = s_{j+1} = s_{j+2} = \ldots$ and for no $t > s_j$ does range $(\bar{\omega}_{t>}) \subseteq B_i$, or

b) for all j , $s_{j+1} > s_j$, and range $(\bar{\omega}_{s_{j+1}>}) \subseteq B_i$.

If a) holds, then $s_j = t^*(B_i)$. Otherwise b) holds and we have a strictly increasing sequence in a finite interval, dom(ω), which must accumulate to some s from below. We claim $s = \ell.u.b.\{t/\text{range } (\bar{\omega}_{t>}) \subseteq B_i\}$, since for $t > s$, range $(\bar{\omega}_{t>}) \not\subseteq B_i$ [s is an upper bound on the sequence in b)] and for $t < s$, there is some $t_k > t$ in the sequence, so that range $(\bar{\omega}_{t>}) \subseteq$ range $(\bar{\omega}_{t_k>}) \subseteq B_i$. Now if range $(\bar{\omega}_{s>}) \not\subseteq B_i$ then there is some t , $0 < t < s$, such that $\omega(t) \notin B_i$. But then there is a t' , $t < t' < s$, such that range $(\bar{\omega}_{t'>}) \not\subseteq B_i$, a contradiction. Thus $t^*(B_i) = s$.

Now it is easy to show that $\max\{t/\omega_{t>} \in \Gamma_\pi\} = \max_{i=1,\ldots,m} \{t^*(B_i)\}$. Moreover, Γ_π is closed under right segmentation, so the conditions of Theorem 1, Appendix 1, sufficient for admissibility, are satisfied.

Denote $\Omega_\pi = \Gamma_\pi^+$. Then $\Omega_\pi = \{\omega \in \Omega |$ there exists a finite set $\{t_1,\ldots,t_n\}$, $0 = t_1 < t_2 \ldots t_{n-1} < t_n = \ell(\omega)$, such that for $i = 1,\ldots,n$, there is a $B_i \in \pi$ for which $\omega(t) \in B_i$ for $t_i < t < t_2\}$. Segments in Ω_π are said to be <u>π-generated.</u> These trajectories have the property that they remain in blocks of π for non zero durations. A non π-generated segment, on the other hand, is infinitely oscillating with respect to π in the sense that for each $B \in \pi$ there is a convergent sequence $t_1, t_2, t_3, \ldots$ such that $\omega(t_i)$ is not in B for all i , or $\omega(t_i)$ is alternately in B and not in B .

<u>Example</u>. Let (X,T) be (R,R) (R denotes the reals). Let $\ldots b_{-2} < b_{-1} < b_0 < b_1 < b_2 \ldots$ be a countable sequence of "threshold" or "quantization" levels. Let π be the cover whose blocks are $B_i = [b_i, b_{i+1}]$ (where if the series terminates on the left at b_ℓ then $B_{\ell-1} = (-\infty, b_\ell]$, and similarly, for the right). Let Ω be the piecewise continuous bounded

segments which are finitely oscillating in the sense that for each $x \in$ range (ω) , $\omega^{-1}(x)$ consists of a finite number of isolated points and/or intervals. Then $\Omega_\pi = \Omega$.

Note that Γ_π has been defined in reference to a semigroup Ω . If we denote this situation by $\Gamma_\pi(\Omega)$, then $\Gamma_\pi(\Omega) = \Gamma_\pi(X,T) \cap \Omega$.

3. Polynomial and Analytic Generated Segments

For $(X,T) = (R,R)$, let Γ be the set of all finite degree polynomials. Then Γ^+ are the piecewise polynomial segments. Let C_m be the segments which are differentiable at least to order m , and let $\Gamma_m = C_m \cap \Gamma^+$. Then $\Gamma^+ \supset \Gamma_0 \supset \Gamma_1 \supset \Gamma_2 \ldots \supset \Gamma_\infty = \Gamma$ is a strictly decreasing chain of admissible generating set for Γ^+ . Break points in a $m\ell s$ decomposition by Γ_m are points at which polynomial segments of finite degree are patched together, so that at most the first $m-1$ derivatives agree. Thus "events" for decomposition by Γ_0 are jumps in the function value; by Γ_1 these are jumps in either function value or in derivative, etc.

Now let Γ be the set of segments analytic on their domains. Then Γ^+ is the set of piecewise analytic segments; ω , μ and $\omega\mu \in \Gamma$ implies that μ is the unique extension of m to $\langle \ell(\omega), \ell(\omega)+\ell(\mu) \rangle$ and the breakpoints of a decomposition by Γ are points at which analytic continuation is impossible. (See Veech, 1967, Chapter 1.)

DEVS ENCODABILITY OF SEGMENTS

From now on, all segment sets will be assumed to be closed under (right and left) segmentation.

Let $g:\Omega' \to \Omega$ be an encoding. To avoid triviality, we will require that g be onto in what follows.

Let Ω_G admissibly generate Ω . We say that g preserves right segmentation at breakpoints if whenever s is the first breakpoint in the mls decomposition of $g(\omega)$ then $g(\omega_{<MATCH^{-1}(s)}) = g(\omega)_{<s}$.

By induction on the size of $g(\omega)$, it can be seen that the above property holds for all breakpoints of $g(\omega)$ if it holds for the first.

We say the g is simple if it is invertable and preserves right segmentation at breakpoints.

In Appendix 2 we show that under reasonable conditions, invertability and homomorphism implies simplicity, but the converse is not true. This is important to note since most of the useful encodings are not homomorphisms but yet take the weaker form of simplicity.

The following proposition shows that for a segment set to be simply encodable by an admissibly generated set, it must itself be admissibly generated.

Proposition 2.

Let $g:\Omega' \xrightarrow{\text{simple}} \Omega$ and let Ω_G admissibly generate Ω . Then $\Gamma = g^{-1}(\Omega_G)$ admissibly generates Ω' . Moreover, if Ω_G is a nontrivial generating set, so is Γ .

Proof:

Let $\omega \in \Omega'$ and $g(\omega) \in \Omega$. Let the breakpoints of the mls decompositio of $g(\omega)$ be $\{t_1, t_2, \ldots, t_n\}$. If this set is empty then $g(\omega) \in \Omega_G$ so $\omega \in \Gamma$. So assume it is not empty. We show that the set $\{MATCH^{-1}(t_1), \ldots, MATCH^{-1}(t_n)\}$ forms the breakpoints for a mls decomposition of ω by Γ . First, $\omega_{MATCH^{-1}(t_1)>} \in \Omega'$ by left segmentation closure. By Lemma A.2, $g(\omega_{MATCH^{-1}(t_1)>}) = g(\omega)_{t_1>}$. Since t_1 is an mls breakpoint, $g(\omega)_{t_1>} \in \Omega_G$ and so $\omega_{MATCH^{-1}(t_1)>} \in \Gamma$. For $t > MATCH^{-1}(t_1)$, $MATCH(t) > t_1$ (monotonic property) and since $\omega_{MATCH^{-1}(t_1)>}$ is a left segment of $\omega_{t>}$, we have $g(\omega)_{t_1>}$ is a left segment of $g(\omega)_{MATCH(t)>}$ (Lemma A.2). If $\omega_{t>} \in \Gamma$, then $g(\omega)_{MATCH(t)>} \in \Omega_G$ and t_1 is not an mls breakpoint as assumed. Thus $MATCH^{-1}(t_1)$ will be the first mls breakpoint in the decomposition of ω by Γ provided we can show that such a decomposition exists. (Note that $t_1 < \ell(g(\omega))$ so $MATCH^{-1}(t_1) < \ell(\omega)$ and $\omega_{MATCH^{-1}(t_1)>}$ is a proper left segment of ω , thus Γ is nontrivial if Ω_G is nontrivial and g is onto.) But $\omega_{<MATCH^{-1}(t_1)} \in \Omega'$ by right segmentation closure, and $g(\omega_{<MATCH^{-1}(t_1)}) = g(\omega)_{<t_1}$ since g is simple. We continue the induction for $i = 2, \ldots, n-1$, in this way, substituting $\omega_{<MATCH^{-1}(t_i)}$ and $g(\omega)_{<t_i}$ for ω and $g(\omega)$ respectively, thus obtaining $MATCH^{-1}\{t_1, \ldots, t_n\}$ as breakpoints in the mls decomposition of ω by Γ .

Q.E.D.

Applying this result to the case of DEVS segments yields

Theorem 3.

Let $g: \Omega' \to \Omega_{DEVS}$ be a simple encoding. Then $\Gamma = g^{-1}(\Gamma_{DEVS})$ non-trivially admissbly generates Ω' , and $\omega \in \Gamma \Rightarrow g(\omega) = x_\tau$ where $x = g(\omega_{t>})(0)$ for all $t \in (0, \ell(\omega))$ and $\tau = MATCH(\ell(\omega))$.

Proof.

Since Γ_{DEVS} admissibly generates Ω_{DEVS} , we have that $\Gamma = g^{-1}(\Gamma_{DEVS})$

admissibly generates Ω' by Proposition 2.

Now for $\omega \in \Gamma$, $g(\omega) = x_\tau \in \Gamma_x \cup \Gamma$. Since g is invertable $\tau = \text{MATCH}(\ell(\omega))$, and also $g(\omega_{t>}) = g(\omega)_{\text{MATCH}(t)>} = (x_\tau)_{\text{MATCH}(t)>} = x_{\text{MATCH}(t}$

Q.E.D.

The theorem indicates that in a simple encoding into DEVS segments, the code of a generator ω is determined by its initial portion no matter how small we take that portion to be. This leads us to define the relation $\underset{0}{\equiv}$ on Ω', where

$$\omega \underset{0}{\equiv} \omega' \Leftrightarrow (\exists\ t > 0)\ (\omega_{t>} = \omega'_{t>})$$

which turns out to be an equivalence relation. (Reflexivity and symmetry are obvious. For transitivity, note that $\omega_{t>} = \omega'_{t>}$ and $\omega'_{\tau>} = \omega''_{\tau>}$ implies $\omega_{\sigma>} = \omega'_{\sigma>}$ where $\sigma = \min\{t,\tau\}$.)

Called the <u>initial segment relation</u>, $\underset{0}{\equiv}$ partitions segments into equival classes, each sharing a common initial segment in the limit of small t. In Appendix 3, we consider the characterization of this fundamental relation for interesting segment sets and note the striking connection to analytic functions.

The degree of coarseness which $\underset{0}{\equiv}$ imposes on a set of generators Γ determines a lower bound of information which must be lost in encoding Γ into DEVS segments. This is evident in the following corollary of Theorem 3.

<u>Corollary 4.</u>

Let $g:\Omega \xrightarrow{\text{simple}} \Omega_{\text{DEVS}}$. Define $\underset{g}{\equiv}$ on Ω, by $\omega \underset{g}{\equiv} \omega' \Leftrightarrow g(\omega) = g(\omega')$

Define $\underset{0,g}{\equiv}$ on Ω by $\omega \underset{0,g}{\equiv} \omega' \Leftrightarrow (\exists\ t > 0)(g(\omega_{t>})(0) = g(\omega'_{t>}(0))$.

Then for ω, $\omega' \in g^{-1}(\Gamma_{\text{DEVS}})$

a. $\omega \underset{g}{\equiv} \omega' \Leftrightarrow \omega \underset{0,g}{\equiv} \omega'$ and $\ell(\omega) = \ell(\omega')$

b. $\omega \underset{0}{\equiv} \omega' \Rightarrow \omega \underset{0,g}{\equiv} \omega'$

c. $\underset{g}{\equiv}$, $\underset{0,g}{\equiv}$ and $\underset{0}{\equiv}$ are equivalence relations on $g^{-1}(\Gamma_{\text{DEVS}})$.

Proof.

Directly from Theorem 3.

Q.E.D.

The equivalence relation $\underset{0,g}{\equiv}$ partitions the generators of Ω into classes of segments which are assigned to the same event, so call it the event relation. The corollary states that $\underset{0}{\equiv}$ refines $\underset{0,g}{\equiv}$, i.e., all segments in a block of the initial segment partition are coded as the same event.

We now show that this refinement is also sufficient for encodability.

Theorem 5.

Let Ω be admissibly generated by Γ . Let $\equiv$ be an equivalence relation on Γ refined by $\underset{0}{\equiv}$.

Then there is a simple encoding $g:\Omega \to \Omega_{DEVS(X)}$ where $X = \Gamma/\equiv$, and g is the extension to Ω of $\not{g}:\Gamma \to \Gamma_{DEVS(X)}$ where $\not{g}(\omega) = x_\tau$ and $x = [\omega]$ and $\tau = \ell(\omega)$. Call this the standard encoding

Proof.

That $\not{g}$ preserves left segmentation is shown by noting that if ω is a left segment of ω' , then $\omega \underset{0}{\equiv} \omega'$ and so by hypothesis $\omega \equiv \omega'$ i.e., $[\omega] = [\omega']$. Since also $\ell(\omega) \leq \ell(\omega')$ we have $\not{g}(\omega)$ is a left segment of $\not{g}(\omega')$. Preservation of left segmentation by $\not{g}$ extends to that of g by induction on the size of ω .

Invertability of g then follows readily from the fact that $\ell(g(\omega)) = \ell(\omega)$.

To show that g preserves right segmentation at breakpoints, note that for $g(\omega) = g(\omega_1)\ldots g(\omega_n)$ we have that $\not{g}(\omega_1),\ldots\not{g}(\omega_n)$ is the mls decomposition of $g(\omega)$ (since the range of $\not g$ does not include Γ_ϕ). Thus $\ell(g(\omega_1) = \ell(\omega_1))$ is the first breakpoint and $g(\omega)_{<\ell(\omega_1)} = \not g(\omega_2)\ldots\not g(\omega_n) = g(\omega_2\ldots\omega_n) = g(\omega_{<\ell(\omega_1)})$.

Q.E.D.

Combining Theorems 3 and 5 gives

Theorem 6 (Characterization of Simple DEVS Encodability)

A segment set Ω is simply encodable onto DEVS segments if, and only if, it has a nontrivial admissible generating set.

Proof

To encode we may always use the inital segment equivalence relation as $\equiv$ in Theorem 5.

Q.E.D.

We interpret the above results intuitively as follows:

Let Ω be the input segment set for a system. We wish to decompose any given $\omega \in \Omega$ into segments $\omega_1, \omega_2, \ldots, \omega_n$ such that the switching from ω_i to ω_{i+1} represents as "event". The criterion for segmentation is that of obtaining maximal length segments relative to some generator set Γ . This determines the event times (breakpoints of the mls decomposition). The event names are determined by the initial segments just following each event time. Subject to certain constraints (Γ must be admissible, and the event naming relation $\equiv$ must be refined by the initial segment relation $\equiv_0$ on Γ), we have freedom of choice in the parameters, Γ and $\equiv$, of the event identifica[tion] process.

SYSTEM STIMULATION BY DISCRETE EVENT MODELS

We wish to characterize the class of systems simulateable by discrete event systems. We start at the system level, and in Part 2, will apply the results to the network level formulation of the problem. In Appendix 4, we show that it is enough to consider the simulation of iterative specifications by discrete event specifications. We modify slightly the notion of iterative specification given by Zeigler (1976), (Appendix 3), by replacing the output set and function by the more general notion of finite intersection cover of Q . This will enable us to more easily deal later with the network case.

An iterative specification $G = \langle T,X,\Omega_G,Q,Y,\delta,\pi \rangle$ is DEVS-like if the following hold:

1. Autonomy between external events:

   For all $\omega,\mu \in \Omega_G$, $q \in Q$

   $$\omega \underset{0}{\equiv} \mu \Rightarrow \delta(q,\omega_{t>}) = \delta(q,\mu_{t>}) \quad \text{for all} \quad t \leq \min\{\ell(\omega),\ell(\mu)\} \quad .$$

2. State trajectories are π-generated.

   For all $\omega \in \Omega_G$, $q \in Q$

   $$STRAJ_{q,\omega} \in \Gamma_\pi^+$$

   where $\Gamma_\pi = \Gamma_\pi(Q,T)$.

Condition 1) requires that the system respond indifferently to generators belonging to the same event class. Roughly, between external events, the system is insensitive to its input, i.e., functions autonomously, conditioned only by the nature of the last external event. Thus we shall also refer to a class $[\omega]$ of $\Omega_G/\underset{0}{\equiv}$ as an input regime.

Condition 2) requires that state trajectories be mls decomposable into generators determined by π . This requires a finite number of events -

roughly crossings of π block boundaries - be identifiable in a finite length state trajectory.

Remark.

Let the iterative specification induced by DEVS $M = <X_M, S, \delta, t>$ be $G(M) = <T, X, \Gamma_{DEVS(X)}, Q_M, \delta_G, \pi_M>$ where $T, X, \Omega_{DEVS(X)}, Q_M, \delta_G$ are defined as in Appendix 3, and π_M is the partition defined by

$$(s,e)\ \pi_M\ (s',e') \Leftrightarrow s = s'$$

for all $(s,e),(s,e') \in Q_M$.

It is readily checked that $G(M)$ is DEVS-like.

Let G be an iterative specification and M a DEVS. A pair (g,h) is a DEVS morphism from $G(M)$ to G if

a) g extends to a simple encoding of Ω_G^+ to $\Omega_{DEVS(X)}$

b) (g,h) is a specification morphism from $G(M)$ to G (See Appendix 3)

c) States of M represent block of π :

For every $s \in S$ there is a $B_s \in \pi$ such that $h(s,e) \in B_s$ for $0 < e < t(s)$.

We say that an iterative specification G is DEVS simulateable if there is a DEVS M and a DEVS morphism from $G(M)$ to G .

We shall prove that the class of DEVS simulateable systems is precisely the class of DEVS-like systems. To do so we shall show how to construct a DEVS simulation for an iterative specification with DEVS-like properties.

In what follows, we shall assume that there is no upper bound on the lengths of segments any class $[\omega]$ of $\Omega_G/\equiv_0$. The case where this is not true involves no new principles but is notationally messy. More specifically, we define for each class $[\omega]$ a representative $\bar{\omega}$ with domain T_0^+ such that for each t , $\bar{\omega}_{t>} \in [\omega]$ and assume $\bar{\omega}$ is well defined.

Given a DEVS-like specification construct a DEVS M called the <u>standard simulator</u> of G as follows:

$$M = \langle X, S, \delta, t \rangle \qquad \ldots 1)$$

where $X = \Omega_G / \equiv_0$,

$$S = Q \times X \ , \qquad \ldots 2)$$

$t: S \to R^+_\infty$ is defined for $(q,[\omega]) \in S$ by

$$t(q,[\omega]) = \max\{t \mid \delta(q,\bar{\omega}_{t>}) \in \Gamma_\pi\} \qquad \ldots 3)$$

(the first breakpoint in the mls decomposition of $STRAJ_{q,\bar{\omega}}$)

$$\delta: Q_M \times (X \cup \{Q\}) \to S$$

is defined for $(q,[\omega]) \in S$ by

$$\delta_\phi(q,[\omega]) = (\delta(q,\bar{\omega}_{t(q,[\omega]}) \ , \ [\bar{\omega}_{<t(q[\omega])}]) \qquad \ldots 4)$$

and for $(q,[\omega],e) \in Q_M$ and $[\mu] \in X$ by

$$\delta(q,[\omega],e,[\mu]) = (\delta(q,\bar{\omega}_{e>}),[\mu]) \qquad \ldots 5)$$

Let us interpret the elements of the standard simulator.

The external event set X is the set of "events" determined by the initial segment relation; a typical "event" is an equivalence class (or its name) $[\omega] \in \Omega_G/\equiv_0$ (line 1).

The sequential state set S consists of all pairs $(q,[\omega])$ where q is a state of G the simulated model G and $[\omega]$ is an external event. The pair $(q,[\omega])$ will be employed by the standard simulator to keep track of the state of G , q at the time of the last change in regime and the current regime $[\omega]$ (line 2). Thus M "knows" the state of G and its input regime at all times.

The time advance function determines for each sequential state $(q,[\omega])$ the time until the next internal event; $t(q,[\omega])$ is the time it would take the system G to reach a π block boundary from state q operating under input regime $[\omega]$ (line 3).

The internal event transition function determines for each sequential state $(q,[\omega])$ the sequential state pertaining just after the next internal event: the new remembered is the state (on a π block boundary) G would have reached from state q in time $t(q,[\omega])$ under input regime $[\omega]$, the new remembered regime is that which would be seen by G when reaching this boundary (line 4).

The external event transition function determines for each sequential state $(q,[\omega])$ and elapsed time e , $0 < e \leq t(q,[\omega])$, the sequential state pertaining just after a change in regimes to $[\mu]$ occurs after having been in regime $[\omega]$ a time e : the new remembered state is the state G would have reached in these circumstances and the new remembered event is just $[\mu]$.

We shall now show that the standard simulator is well defined and simulates G .

<u>Proposition 7.</u>

The standard simulation M of a DEVS-like specification G is a legitimate DEVS . Moreover, there is DEVS morphism from G(M) to G .

<u>Proof</u>

Define $STRAJ_{q,\overline{\omega}}$ also denoted $\tilde{\delta}(q,\overline{\omega})$ with domain T_0^+ by

$$STRAJ_{q,\overline{\omega}}(t) = \tilde{\delta}(q,\overline{\omega})(t) = \begin{cases} \delta(q,\overline{\omega}_{t>}) & t > 0 \\ q & t = 0 \end{cases}$$

Then $STRAJ_{q,\bar{\omega}}$ represents the state trajectory associated with state $q \in Q$ under input regime $[\omega]$. This is so, since for any $t > 0$ and any $\omega' \in [\omega]$ of length greater or equal to t , we have $\omega \underset{0}{\equiv} \bar{\omega}_{t>}$ and by autonomy (condition 1)) $\delta(q,\omega_{t>}) = \delta(q,\bar{\omega}_{t>})$.

Since state trajectories are π-generated (condition 2)), $STRAJ_{q,\bar{\omega}}$ has a mls decomposition by Γ_π (Proposition 1) and its first breakpoint is thus well defined and identical to the definition of $t(q,[\omega];\delta$ is now easily seen to be well defined.

Now define a mapping $h: Q_M \to Q$ by

$$h(q,[\omega],e) = \begin{cases} q & \text{if } e = 0 \\ \delta(q,\bar{\omega}_{e>}) & \text{otherwise} \end{cases}$$

for all $(q,[\omega],e) \in Q_M$. We shall presently employ h as the state mapping in the DEVS simulation of G . For this we require the following

Lemma 7

The standard simulator M is legitimate. Moreover, for $\omega \in \Omega_G$, $q \in Q$

$$h(\delta_G(q,[\omega],0,\phi_{\ell(\omega)}) = \delta(q,\omega)$$

where $\delta_G : Q_M \times \Omega_{DEVS(X)} \to Q_M$ is the single segment transition function belonging to the iterative specification $G(M)$.

Proof

The proof proceeds by induction on the length of the mls decomposition of state trajectories by Γ_π . The proposition for interger n is

$$P(n) \equiv \Big[(\forall q \in Q)(\forall \omega \in \Omega_G)(\text{the mls decomposition of } STRAJ_{q,\omega} \text{ by } \Gamma_\pi \text{ has length } n \Rightarrow m_{q,[\omega],0,\ell(\omega)} = n \text{ and } h(\delta_G(q,[\omega],0,\phi_{\ell(\omega)}) = \delta(q,\omega) \Big] .$$

Here $m_{q,[\omega],0,\ell(\omega)}$ is the step counting function (Appendix 3). Showing that P(n) is true for all n amounts to proving the lemma.

The proof of P(0) and $P(n) \Rightarrow P(n+1)$ is a straightforward use of the explicit definition of δ_G (ibid) and the fact that if $\omega_1,\ldots,\omega_{n+1}$ is an mls decomposition of length n+1 , then $\omega_2,\ldots,\omega_n$ is an mls decomposition of length n .

Q.E.D.

Now define a candidate morphism $(\not{g},h)$ from G(M) to G where h is as just defined and $\not{g}$ is standard encoding of Ω_G to $\Gamma_{DEVS(X)}$ given by

$$\not{g}(\omega) = x_\tau \quad \text{where} \quad x = [\omega] \quad \text{and} \quad \tau = \ell(\omega) \quad .$$

Then by Theorem 5 , $\not{g}$ extends to a simple encoding of Ω_G^+ to $\Omega_{DEVS(X)}$ Thus condition a) of the requirement for DEVS morphism is satisfied. For the specification morphism of condition b), we have for $(q,[u],e) \in Q_M$, and $\omega \in \Omega_G$

$$h(\delta_G(q,[u],e,\not{g}(\omega))$$

$$= h(\delta_G(q,[u],e,[\omega]_{\ell(\omega)})) \quad \text{(definition of } \not{g}\text{)}$$

$$= h(\delta_G(\delta(q,[u],e,[\omega],0,\phi_{\ell(\omega)})) \quad \text{(definition of } \delta_G\text{)}$$

$$= h(\delta_G(\delta(q,\bar{u}_{e>}),[\omega],0,\phi_{\ell(\omega)})) \quad \text{(definition of } \delta\text{)}$$

$$= \delta(\delta(q,\bar{u}_{e>}),\omega) \quad \text{(lemma 7)}$$

$$= \delta(h(q,[u],e),\omega) \quad \text{(definition of } h\text{)}$$

as required for specification morphism.

For condition c) requiring states of M to represent blocks of π , we have that for every $(q,[\omega]) \in S$ and $0 < e < t(q,[\omega])$

$$h(q,[\omega],e) = \delta(q,\bar{\omega}_{e>}) \quad .$$

But by definition of $t(q,[\omega])$, there is a block B of π such that

$\delta(q,\bar{\omega}_{e>}) \in B$ for $0 < e < t(q,[\omega])$ (by definition of mls decomposition by Γ_{π} , $t(q[\omega])$ is the maximum τ for which there is a block B such that $STRAJ_{q,\bar{\omega}}$ is in B during $(0,\tau)$).

Thus (g,h) is a DEVS morphism from the legitimate simulator M to G as claimed in the theorem.

Q.E.D.

We now show that the necessary condition for DEVS simulateability is DEVS-likeness.

Proposition 8

Let $(\not{g},h)$ be a DEVS morphism from a DEVS specification G(M) to an iterative specification G . Then G is DEVS-like.

Proof

We shall need the following more general observation:

Lemma 9

Let $(\not{g},h)$ be a specification morphism from $\bar{G}$ to G . Then for all $\omega,\mu \in \Omega_G$

$$\omega \underset{\not{g}}{\equiv} \mu \Rightarrow \delta(q,\omega) = \delta(q,\mu) \quad \text{for all } q \in Q .$$

Proof

Let $\omega \underset{\not{g}}{\equiv} \mu$ (i.e., $\not{g}(\omega) = \not{g}(\mu)$.

Since h is onto, for each $q \in Q$, there is a $\bar{q} \in \bar{Q}$ such that $h(\bar{q}) = q$.

Then

$$\begin{aligned} \delta(h(\bar{q}),\omega) &= h(\bar{\delta}(\bar{q},\not{g}(\omega)) \\ &= h(\bar{\delta}(\bar{q},\not{g}(\mu)) \\ &= \delta(h(\bar{q}),\mu) \end{aligned}$$

Q.E.D.

Since $\not{g}$ of a DEVS morphism extends to a simple encoding, we have from Corollary 4 that $\omega \underset{\not{g}}{\equiv} \mu \Leftrightarrow \omega \underset{0,g}{\equiv} \mu$ and $\ell(\omega) = \ell(\mu)$ and $\omega \underset{0}{\equiv} u \Rightarrow \omega \underset{0,g}{\equiv} \mu$. Thus we have from the lemma 7

$$\omega \underset{0}{\equiv} \mu \text{ and } \ell(\omega) = \ell(\mu) \Rightarrow \delta(q,\omega) = \delta(q,\mu)$$

or equivalently

$$\omega \underset{0}{\equiv} \mu \Rightarrow \delta(q,\omega_{t>}) = \delta(q,\mu_{t>}) \quad \text{for} \quad t \leq \min\{\ell(\omega),\ell(\mu)\}$$

$$(\omega \underset{0}{\equiv} \mu \Rightarrow \omega_{t>} \underset{0}{\equiv} \mu_{t>} \;;\; \ell(\omega_{t>}) = \ell(\mu_{t>})\;) \;.$$

Thus the autonomy condition (1) of DEVS-likeness is satisfied.

Since h is onto, for any $q \in Q$, there is $(s,e) \in Q_M$ such that $h(s,e) = q$. Since the DEVS M is DEVS-like, any state trajectory $STRAJ_{s,e,\not{g}(\omega)}$ is π_M-generated (by the Remark following the DEVS-like definition). $STRAJ_{s,e,\not{g}(\omega)}$ is thus a composition of finite number of segments such that each segment is characterized by a constant s and e , $0 < e \leq t(s)$. Since by conditions a) and b) of the DEVS morphism, $\not{g}$ is an invertable, it is readily shown that $STRAJ_{h(s,e),\omega}$ is a composition of a finite number of segments, each being pointwise mappings under h of the constant s segments (domains being matched up by the MATCH function induced by $\not{g}$). (Appendix 3). By condition c) of DEVS morphism, each such segment maps to a generator in Γ_π. Thus $STRAJ_{h(s,e),\omega}$ is composed of a finite number of segments from Γ_π , i.e., is π-generated. Since this is true for all possible pairs $q \in Q$, $\omega \in \Omega_G$, condition 2) for the DEVS-likeness is established.

Q.E.D.

Summarizing Propositions 6 and 8.

Theorem 9 (Characterization of DEVS simulateable systems)

An iterative specification G is DEVS simulateable if, and only if, it is DEVS-like.

REALISTIC DEVS SIMULATION

Until now, our characterization of computer simulateability has only encorporated the "finite number of computation instants in a finite interval" constraint. This is because the DEVS concept formalizes precisely this, and only this, notion. Various constraints can be imposed on the DEVS structure and on the DEVS morphism which reflect further "realistic" limitations on computer simulation. We shall discuss two possibilites involving finite dimensional and finite state constraints.

The finite dimension restriction requires that all spaces employed in the simulation be finite dimensional. Many simulation models constructed independently of discrete event theory satisfy this restriction and their representation by DEVS would also be finite dimensional. The finite state constraint requires that all spaces, except that of the clock, be finite. This can be viewed as a further restriction on the finite dimensional class.

The effect of these constraints is to narrow the class of simulateable systems in an easily understood manner. In fact, we can modify the definition of DEVS-like systems accordingly and maintain the equivalence of the two concepts.

We shall be able to point out limitations in DEVS simulateability when the above finiteness conditions are applied. In contrast, in the unrestricted case, we have not been able to construct examples of non-simulateable systems.

FINITE DIMENSION DEVS SIMULATION

A DEVS $M = \langle X, S, t, \delta \rangle$ is <u>finite dimensional</u> if the external event set X and sequential state set S are finite dimensional linear spaces. We convert $\Omega_{DEVS(X)}$ into a linear space by pointwise addition, treating ϕ as zero in the obvious way.

An iterative specification G is <u>finite dimension DEVS simulateable</u> if there is a finite dimensional DEVS M and a DEVS morphism (g,h) from $G(M)$ to G for which g is a linear map (this presumes Ω_G is a linear space).

An iterative specification $G = \langle T, X, \Omega_G, Q, \delta, \pi \rangle$ is <u>finite dimension DEVS-like</u> if:

1. <u>Autonomy with respect to an equivalence on Ω_G</u>

There is a finite dimensional linear equivalence $\equiv$ on Ω_G (i.e., $\Omega_G/\equiv$ is a finite dimensional linear space) such that for all $\omega, \mu \in \Omega_G$, $q \in Q$

$$\omega \underset{0}{\equiv} \mu \Rightarrow \omega \equiv \mu \Rightarrow \delta(q, \omega_{t>}) = \delta(q, \mu_{t>}) \quad \text{for all} \quad t \leq \min\{\ell(\omega), \ell(\mu)\}$$

(In particular, $\underset{0}{\equiv}$ refines $\equiv$ and $\Omega_G/\equiv$ is a factor space of $\Omega_G/\underset{0}{\equiv}$).

and

2. <u>State trajectories are π-generated</u> (unchanged from DEVS-like definition).

<u>Theorem 10</u> (Characterization of finite dimension DEVS simulateability).

An iterative specification is finite dimension DEVS simulateable if, and only if, it is finite dimension DEVS-like.

<u>Proof</u> ($\Rightarrow$)

Let (g,h) be a DEVS morphism from $M = \langle X, S, \delta, t \rangle$ to G. As in

Proposition 8, we find that for all $\omega,\mu \in \Omega_G$,

$$\omega \underset{0,g}{\equiv} \mu \Leftrightarrow \omega_{t>} \underset{\not{g}}{\equiv} \mu_{t>} \Rightarrow \delta(q,\omega_{t>}) = \delta(q,\mu_{t>})$$

for all $t \leq \min\{\ell(\omega)\ell(\mu)\}$ and $q \in Q$ and also since $\underset{\not{g}}{\equiv}$ is linear, so is $\underset{0,\not{g}}{\equiv}$. Thus the autonomy condition 1 is satisfied with $\equiv$ taken to be $\underset{0,\not{g}}{\equiv}$, noting that $\Omega_G/\underset{0,\not{g}}{\equiv}$ is isomorphic to X .

($\Leftarrow$)

Let G be finite dimension DEVS-like. We construct the standard simulator M employing $\equiv$ instead of $\underset{0}{\equiv}$. Thus $X = \Omega_G/\equiv$ is finite dimensional and $S = Q \times X$ is finite dimensional, as required. We define a candidate morphism $(\not{g},h)$ as in Proposition 6, except that $\not{g}$ is defined using $\equiv$ rather than $\underset{0}{\equiv}$. Since $\equiv$ is linear, it is easily shown that $\not{g}$ is linear. All other aspects of the proof that $(\not{g},h)$ a DEVS morphism remain unchanged.

Q.E.D.

We can demonstrate that there are systems which are not <u>finite dimensional</u> DEVS simulateable although the same question for unrestricted DEVS simulateability remains open.

Say that a system $S = <T,X,\Omega,Q,\delta,\pi>$ is <u>(finite dimension) simulateable</u> if there is a (finite dimensional) DEVS M and a system morphism (g,h) from $S_{G(M)}$ to S such that g is a simple (linear) encoding.

<u>Theorem 11</u>

There are systems which are not finite dimension DEVS simulateable. In particular, the integrator is not finite dimension DEVS simulateable.

<u>Proof</u>

Let S be finite dimension simulateable via morphism (g,h) . Then there is an iterative specification G of S is DEVS simulateable (Theorem A8,

Appendix 4).

The generator set $\Omega_G = g^{-1}(\Gamma_{DEVS})$ is a linear subspace of Ω (since Γ_{DEVS} is a linear subspace of Ω_{DEVS} amd g is linear). Moreover, the restriction $\not{g} = g/\Omega_G$ is linear. Thus if S is finite dimension DEVS simulateable, so is G finite dimensional DEVS simulateable. By Theorem 10, there is a finite dimensional equivalence $\equiv$ on Ω_G which is refined by $\underset{0}{\equiv}$. In particular, if Ω is the set of piecewise analytic segments, then by Theorem A.5 of Appendix 5, $\underset{0}{\equiv}$ is infinite dimensional on Ω_G , so $\underset{0}{\equiv}$ strictly refines $\equiv$. Thus there is an $\omega \in \Omega_G$ such that ω is analytic and not identically zero, i.e. $\omega \underset{0}{\not\equiv} 0$, for which $\omega \equiv 0$. By autonomy it is also necessary that for all $q \in Q$, $\delta(q,\omega_{t>}) = \delta(q,0_{t>})$ for all $t \leq \ell(\omega)$. In particular, if S represents an integrator, then $\delta(q,\omega_{t>}) = q + \int_0^t \omega(t')dt'$. If ω is analytic and not identically zero, then $\delta(0,\omega_{t>}) = \int_0^t \omega(t')dt'$ is also no zero for all $t \leq \text{dom}(\omega)$. But $\delta(0,0_{t>})$ is clearly always zero, thus yielding a contradiction.

Since piecewise analytic implies piecewise continuous, the same conclusion applies in the case that Ω is piecewise continuous.

Q.E.D.

Remark

The integrator is unrestricted DEVS simulateable for the choice Ω = the piecewise (finite degree) polynomial segments, and π = any quantization cover (Example 1). To see this take as admissible generator set, the polynomial segments, on which $\underset{0}{\equiv}$ is the identity. The autonomy condition for DEVS-likeness is thus easily satisfied. Moreover, the finite degree polynomial segments and their integrals are finitely oscillatory, so π-generability of the integrator's state trajectories is also satisfied (Example, Section 2).

More surprising, perhaps, is that the integrator with piecewise continuous segments as input may be "arbitrarily closely simulated" by a DEVS. By the Weierstrass Approximation Theorem, every continuous function on a finite interval can be approximated arbitrarily closely by a finite degree polynomial. Choose as admissible generators the continuous segments, and let $g(\omega) = x_{\ell(\omega)}$ where $x = [\text{approx}_{\varepsilon}(\text{int}(\omega))]_0$ where $\text{int}(\omega)(t) = \int_0^t \omega(t')dt'$ and $\text{approx}_{\varepsilon}(\omega)$ is a polynomial approximating ω with maximum error ε. The DEVS in this case updates its memory of the integrator state by evaluating the polynomial named by x (a finite list of coefficients) at external event times, and internal event times (state trajectory boundary crossings predeterminable in principle as a z crossing problem). The error in state trajectory simulation is at most $n\varepsilon$ where n is the number of discontinuities in the input segment.

The reader is invited to provide the details of simulation.

FINITE STATE DEVS SIMULATION

An obvious finiteness restriction on a DEVS $M = \langle X,S,t,\delta\rangle$ is to require that the sets X and S be finite. This turns out not to be "architecturally" satisfying since it forces quantization of the elapsed time clock (see Appendix 5). We shall work with a weaker and useful restriction based on the possibility of adjoining a "time-left" component to a finite state space.

A DEVS M is explicit form (Zeigler, 1976, p.245), if there is a set $\bar{S}$ and a function $T:\bar{S} \to R^+_\infty$ such that

$$S = \{(\bar{s},\sigma) \mid \bar{s} \in \bar{S} \ , \ 0 \leq \sigma \leq T(\bar{s})\}$$

and such that

$$t(\bar{s},\sigma) = \sigma \qquad \text{for all} \quad (\bar{s},\sigma) \in S \ .$$

The situation $(\bar{s},\sigma)$ can be interpreted as indicating that there is a time σ left for the system to remain in state $\bar{s}$. $T(\bar{s})$ is the maximum such time such time $(T(\bar{s}) = \infty$ will cause the system to wait until an external event occurs before leaving state $\bar{s}$).

A DEVS in explicit form is finite if the sets X and $\bar{S}$ are finite.

Every DEVS can be DEVS simulated by an explicit form equivalent (obvious). This holds for finite DEVS and finite explicit form DEVS as well. But the state space of a (nontrivial) finite explicit form DEVS is infinite, in contrast to that of a finite DEVS, and thus adds extra power (Appendix 5). In terms of simulation language concepts, the finite explicit form DEVS assumes an infintely exact clock but otherwise finite memory and finite events.

An iterative specification G is finite explicit DEVS simulateable if there is a finite explicit form DEVS M and a DEVS morphism $(\not{g},h)$ from $G(M)$ to G for which h satisfies

*) $h(\bar{s},\sigma,e) = h(\bar{s},\sigma-e,0)$

for all $(\bar{s},\sigma) \in S$ and $0 \leq e \leq \sigma$.

This restriction on the state decoding h ensures that the σ component truly plays the role of a time left indicator.

An iterative specification $G = \langle T,X,\Omega_G,Q,\delta,\pi \rangle$ is finite explicit DEVS-like if:

1. Autonomy with respect to a finite equivalence of Ω_G .

(Same as finite dimension case, except that $\equiv$ is finite, i.e., $\Omega_G/\equiv$ is finite).

2. State trajectories are finitely π-generated.

There is a finite subcover $\bar{\pi}$ of π and a finite subset Γ of $\Gamma_{\bar{\pi}}(Q,T)$ such that for all $q \in Q$, $\omega \in \Omega_G$,

$STRAJ_{q,\omega} \in \text{right.seg}(\Gamma)^{+}\text{seg}(\Gamma)$

where right.seg(Γ) is the closure of Γ under right segmentation and seg( ) is the closure under both right and left segmentation.

We note that $\text{right.seg}(\Gamma)^{+} \subseteq \Gamma_{\bar{\pi}}(Q,T)$ so that $\Gamma_{\bar{\pi}}(\text{right.seg}(\Gamma)^{+}) = \text{right.seg}(\Gamma)^{+}$. Thus condition 2) requires that each state trajectory be decomposable into mls segments $\omega_1,\ldots,\omega_n$ such that each ω_i , $i=1,\ldots,n-1$ is a right segment of some segment in Γ (Proposition 1) and ω_n is subsegment of some segment in Γ .

Theorem 12 (Characterization of finite explicit DEVS simulateability).

An iterative specification G is finite implicit DEVS simulateable, if, and only if, it is finite implicit DEVS-like.

Proof ($\Rightarrow$)

Let M be a finite implicit form DEVS which simulates G using $(\not{g},h)$. Consider the segments $\{\eta_{\bar{s}} \mid \bar{s} \in \bar{S}\}$ where

$\eta_{\bar{s}}(t) = h(\bar{s},t,0)$ for $0 \leq t \leq T(\bar{s})$.

Since each $(\bar{s},\sigma)$ represents a block of π we have $h(\bar{s},\sigma,e) \in B_{\bar{s},\sigma}$ for $0 \leq e < \sigma$. But since also $h(\bar{s},\sigma,e) = h(\bar{s},\sigma-e,0)$ we have $\{h(\bar{s},\sigma,e) \mid 0 \leq e \leq \sigma\} \subseteq B_{\bar{s},T(\bar{s})}$. Since h is onto, $\pi = \{B_{\bar{s},T(\bar{s})} \mid \bar{s} \in \bar{S}\}$ is a finite subcover of π and $\eta_{\bar{s}} \in B_{\bar{s},T(\bar{s})}$ for each $\bar{s} \in \bar{S}$.

As in Proposition 6, it is easy to show that for each $q \in Q$, $\omega \in \Omega$, $STRAJ_{q,\omega}$ has decomposition consisting of a finite number of right segments of elements in $\{\eta_{\bar{s}}\}$ followed by a subsegment of some $\eta_{\bar{s}}$. Thus we may set $\Gamma = \{\eta_s\}$.

($\Leftarrow$)

Let G be finite implicit DEVS-like. Construct a DEVS similar to the standard simulator of Proposition 6 as follows:

$X = \Omega_G/\equiv$

$S = \{(\eta,[\omega],\sigma) \mid \eta \in \bar{\Gamma}_{\bar{\pi}},\ [\omega] \in X,\ \sigma \in [\ ,\ell(\eta)]\}$

$t: S \to R^+_\infty$ is defined by $t(\eta,[\omega],\sigma) = \sigma$

$\delta_\phi: S \to S$ is defined by $\delta_\phi(\eta,[\omega],\sigma) = (\eta',[\omega_{<\sigma}],\sigma')$ where $\eta' \in \Gamma$ is such that $\eta'_{<-\sigma'} = (STRAJ_{\eta(\ell(\eta)),\bar{\omega}_{<\sigma}})_{\sigma'>}$ and σ' is the first breakpoint in the mls decomposition of $STRAJ_{\eta(\ell(\eta)),\bar{\omega}_{<\sigma}}$ by Γ.

$\delta: Q_M \times X \to S$ is defined by $\delta(\eta,[\omega],\sigma,e,[\mu]) = (\eta',[\mu],\sigma')$ where $\eta' \in \Gamma$ is such that $\eta'_{<-\sigma'} = (STRAJ_{\eta_{<-\sigma}(e),\bar{\mu}})_{\sigma'>}$ and σ'

is the first breakpoint in the mls decomposition of

$STRAJ_{\eta_{<\sigma}(e),\bar{\mu}}$ by Γ .

We interpret $-\sigma$ as the point which is σ units to the left of the right end of η . Thus $\eta_{<-\sigma} = \eta_{<\ell(\eta)-\sigma}$ if $\ell(\eta)$ is finite. If $\ell(\eta)=\infty$, we assume that η has domain $(-\infty,0]$ and $\eta_{<-\sigma} = \eta|<-\sigma,0>$.

The candidate morphism $(\not{g},h)$ has $\not{g}$ defined as in Theorem 10, and h defined by

$$h(\eta,[\omega],\sigma,e) = \begin{cases} \eta(\ell(\eta)) & \text{if } \sigma = e = 0 \\ \eta_{<\ell(\eta)-\sigma}(e) & \text{for } 0 \leq e \leq \sigma \quad \text{otherwise} \end{cases}$$

Clearly $\eta_{<\ell(\eta)-\sigma}(e) = \eta_{<\ell(\eta)-(\sigma-e)}(0) = \eta(\ell(\eta)-\sigma+e)$, so h satisfies condit:

The state $(\eta,[\omega],\sigma)$ represents the situation in G where operating in regime $[\omega]$, time σ is left to the end of the generator state trajectory η . If no external event occurs in the meantime, the DEVS selects the next generator and the appropriate point from its right end at which to start. If a change in regimes occurs after elapsed time e , the DEVS determines the current state of G , viz. $\eta_{<\ell(\eta)-\sigma}(e)$, and from this determines the new, appropriate right segment of a generator under the new input regime.

We omit the proof, which follows the lines of Theorem 10.

Q.E.D.

EXAMPLE: A DEVS SIMULATION

We provide an example of a DEVS simulateable system drawn from a much larger simulation model built on the principles of this paper (for further information see Zeigler, 1977,).

Consider a "patch" in which may live a prey population feeding on resources in the patch. Let $p(t)$ and $r(t)$ denote the population size, and resource amount available at time t, respectively. Initially there are no prey and some resources. This state persists until some prey immigrate into the patch. The resources are then consumed to exhaustion (without replenishment) at which time some prey emigrate from the patch, while the remainder die out.

The state space of the system is $Q = \{(r,p) \mid r \geq 0 \ , \ p \geq 0\}$ the first quadrant in the p,r plane. The dynamics not including migration are given by:

$$\frac{dp}{dt} = (b \ pos(r) - d)p$$

$$\frac{dr}{dt} = -u \ pos(r)p$$

where $pos(r) = 1$ if $r > 0$ and $pos(0) = 0$. The parameters b, d and u give the birth rate (when there is food), death rate and resource consumption rate respectively.

The input set X is the set $\{x / x \geq 0\}$ and the input segment set Ω consists of all finite segments $\omega \in (X,R)$ such that $\omega(t) > 0$ for only a finite number of points t in its domain; $\omega(t) = x$ means that x prey arrive at the patch at time t. Ω is isomorphic to the DEVS segments $\Omega_{DEVS(X)}$ so that we may take $\Gamma_{DEVS(X)}$ as an admissible set of generators for it (identifying ϕ with 0).

The system is autonomous with respect to $\underset{0}{=}$, since for

$\omega, \mu \in \Gamma_{DEVS(X)}$

$\omega \underset{0}{\equiv} \mu \iff \omega(0) = \mu(0) \iff [\omega$ is a left segment of μ or conversely].

Thus the regime $[\omega]$ consists of essentially one segment and represents the arrival of $\omega(0)$ prey followed by an arbitrarily long period of no immigration during which the system evolves independently of its environment.

Q is divided by a partition $\pi = \{B_1, B_2, B_3\}$ where $B_1 = (0,\infty)$, $B_2 = \{(r,p) \;/\; r > 0 \;,\; p > 0\}$, and $B_3 = (\infty, 0]$ (Figure 1).

State trajectories are π-generated as we see as follows:

For $\omega \in \Gamma_{DEVS(X)}$, $\omega(0) = 0$ and

a) for $(r,0) \in B_1$

$$STRAJ_{(r,0),[0]}(t) = (r,0) \in B_1 \quad , \quad t \geq 0$$

b) for $(0,p) \in B_3$

$$STRAJ_{(0,p),[0]}(t) = (0 \;,\; pe^{-dt}) \in B_3 \quad , \quad t \geq 0$$

c) for $(r,p) \in B_2$

$$STRAJ_{(r,p),[0]}(t) = (r - \frac{up}{a}[e^{at} - 1] \;,\; pe^{at}) \in B_2$$

for $0 \leq t < \tau(r,p) = \frac{1}{a} \ln [\frac{ar}{up} + 1]$ where $a = b - d$ and

$$STRAJ_{(r,p),[0]}(t) = (0 \;,\; fp(\tau)e^{-d(t-\tau)}) \in B_3$$

for $t \geq \tau = \tau(r,p)$, where $p(\tau) = p + \frac{ar}{u}$

and f is the fraction remaining after emmigration at time τ.

For $\omega \in \Gamma_{DEVS}$, $\omega(0) \neq 0$, we define

$$\delta(r,p,\omega) = \delta(r \;,\; p + \omega(0) \;,\; 0_{\ell(\omega)}) \quad \text{for all} \quad (r,p) \in Q \;,$$

i.e., we specify that the effect of immigration is to add the incoming prey instantaneously to the current population. Thus

$$STRAJ_{(r,p),[\omega]} = STRAJ_{(r \;,\; p + \omega(0)),[0]}$$

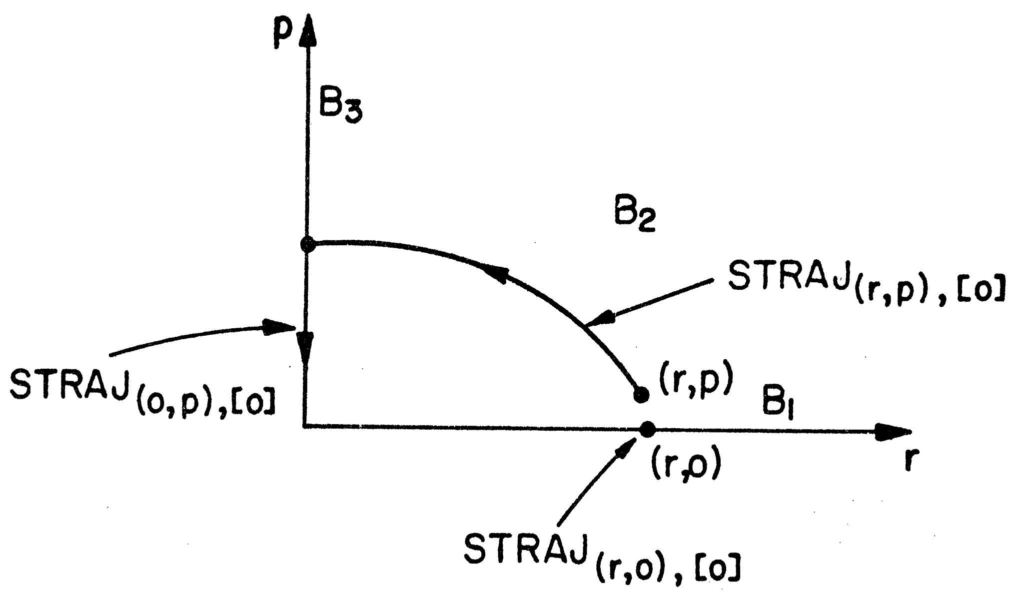


Fig. 1

and it is clear that every state trajectory consists of a finite number of segments lying total within B_1 , B_2 or B_3 , i.e., is π-generated.

Thus the constructed system S is DEVS-like.

The standard DEVS simulator has input event set $\Gamma_{DEVS(X)}\big|_0^{\equiv}$ isomorphic with X , sequential states $S = Q \times X = \{(r,p,[\omega])\}$, with

$$t(r,p,[\omega]) = \begin{cases} \tau(r,p+\omega(0)) & \text{if } (r,p+\omega(0)) \in B_2 \\ \infty & \text{otherwise} \end{cases} \qquad \ldots 1)$$

$$\delta_\phi(r,p,[\omega]) = (0, f(p+\omega(0)+\frac{ar}{u}),[0]) \quad \text{if } (r,p+\omega(0)) \in B_2$$

(since $t = \infty$ on $B_1 \cup B_2$, δ_ϕ need not be defined there) ...2)

and

$$\begin{aligned} \delta(r,p,[\omega],e,[u]) &= (\delta(r,p,\omega_{e>}),[u]) \\ &= (\delta(r,p+\omega(0),0_e),[u]) \\ &= \begin{cases} (r,p+\omega(0),[u]) & \text{if } (r,p+\omega(0)) \in B_1 \\ (r - \frac{u(p+\omega(0))}{a}[e^{ae}-1] , (p+\omega(0))e^{ae} , [u]) & \\ & \text{if } (r,p+\omega(0)) \in B_2 \\ (r,(p+\omega(0))e^{-de},[u]) & \text{if } (r,p+\omega(0)) \in B_3 \end{cases} \end{aligned} \qquad \ldots 3)$$

An output event occuring in the crossing from B_2 to B_3 (prey emigration) can be defined by adding an output set $Y = \{y/y \geq 0\}$ and an output function

$$\lambda(r,p,[\omega],e) = \begin{cases} (1-f)(p+\omega(0)+\frac{ar}{u}) & \text{if } e = \tau(r,p+\omega(0)) \\ 0 & \text{otherwise} \end{cases}$$

The simulator is in fact a finite dimension DEVS. Realized by a discrete event simulation language such as SIMULA , it requires memory for three real variables r,p, and $[\omega]$ (actually, in this case $[\omega]$ can be dispensed with), one internal process for the B_2 phase (prey growth and emigration) and one external process (for the prey arrival). Thus it is realizable exactly to the precision with which real numbers are represented on the particular computer.

Supposing that the model always starts with r_0 food supply and no prey, the following sketches a simulation program realization:

Internal Process

state variables are: r , p , last.time

B_1 Set $r = r_0$, $p = 0$

Wait

B_2 Hold for a time $\tau(r,p)$

Cause emigration of $(1-f)(p+\frac{ar}{u})$ prey

Set $r = 0$, $p = f(p+\frac{ar}{u})$, last.time = time.now

B_3 Wait

External Process

1 Cause immigration of x prey

If Internal Process is in B_1 , Set $p = x$, last.time = time.now

Start Internal Process from B_2

If Internal Process is in B_2 , Set e = time.now - last.time

Set $r = r - \frac{up}{a}[e^{ae}-1]$, $p = pe^{ae} + x$, last.time = time.now

Start Internal Process from B_2

If Internal Process is in B_3 , Set e = time.now - last.time

Set $p = pe^{-de} + x$

Start Internal Process from B_1

Hold for interarrival time go to 1.

Finite Explicit DEVS Simulation

We shall give an example of a finite explicit form DEVS approximation to the model just developed. Let us start by noting that because of the exponential prey growth and limited resources, the effect of the actual number of prey in migrating is small.

This is seen in lines 1) and 2) with $\frac{ar}{u}$ large compared to $\omega(0)$. For the same reason, subsequent prey invasions have little effect once a colony has been established. This is seen in lines 1) and 2) with p (the population size at a subsequence prey invasion) large compared to $\omega(0)$. Thus our first simplification is to reduce the input event set to a single event indicating that at least one prey is arriving.

Formally, we are assuming that the system is autonomous with respect to the equivalence $\equiv$, where $\Omega_G|\equiv$ consists of two classes:

$[0] = \{\omega/\omega(0) = 0\}$

and $[1] = \{\omega/\omega(0) \geq 1\}$

Note that the autonomy condition for generators in [1] is precisely a statement of the fact that the system is sensitive at most to the arrival of prey and never to the exact number arriving.

Now let us (reasonably) suppose that there is an upper bound on the resources available; set it equal to 1 by normalization. To make things more interesting we shall allow the resources to grow to the maximum in logistic fashion: $\frac{dr}{dt} = gr(1-r)$ in the absence of prey. Let us recognize 3 intervals of resources: low, medium and high (Figure). Placing r_{low}, r_{med} and r_{high} at the center of the respective intervals, we shall assume that the state trajectory $STRAJ_{(r_i,1),[0]}$ (i $\in$ {low, med, high}) adequately approximates the state trajectories $STRAJ_{(r,1)[0]}$ from interval i

Thus our model has 5 generator trajectories as follows:

$\eta^1 = STRAJ_{(.01,0),[0]}$ (resource growth in the absence of prey, assuming an initial value $\geq .01$ in B_1)

$\eta^i = STRAJ_{(r_i,1),[0]}$ $i \in \{low, med, high\}$ (restricted to B_2)

$\eta^3 = STRAJ_{(0,p_{max}),[0]}$ (prey decay from a maximum $p_{max} = \frac{a}{u} + 1$ in B_3)

All state trajectories associated with input generators are $m\ell s$ decomposable into a number $n \geq 0$ of right segments of the generators $\Gamma = \{\eta^1, \eta^{low}, \eta^{med}, \eta^{high}, \eta^3\}$ followed by a subsegment of one of them. For example, for $(0,r) \in B_1$, $STRAJ_{(p,r),[0]}$ is of the form $\eta^0_{t>}$; and $STRAJ_{(0,r),[1]}$ is of the form $\eta^i_{t>}$, or $\eta^i \eta^3_{<\ t>}$ or $\eta^i \eta^3_{<t}$ where i depends on r and σ depends on i (Figure 2). Thus the state trajectories are finitely π-generable and the system is finite explicit DEVS-like.

The standard simulator has five finite states with associated maximum times as follows:

$\bar{s}$	$T(\bar{s})$	
1	∞	
low	T_{low}	(time to reach B_3 along η^{low})
med	T_{med}	(similar)
high	T_{high}	(similar)
3	σ_{max}	(time to reach $p=1$ starting from p_{max})

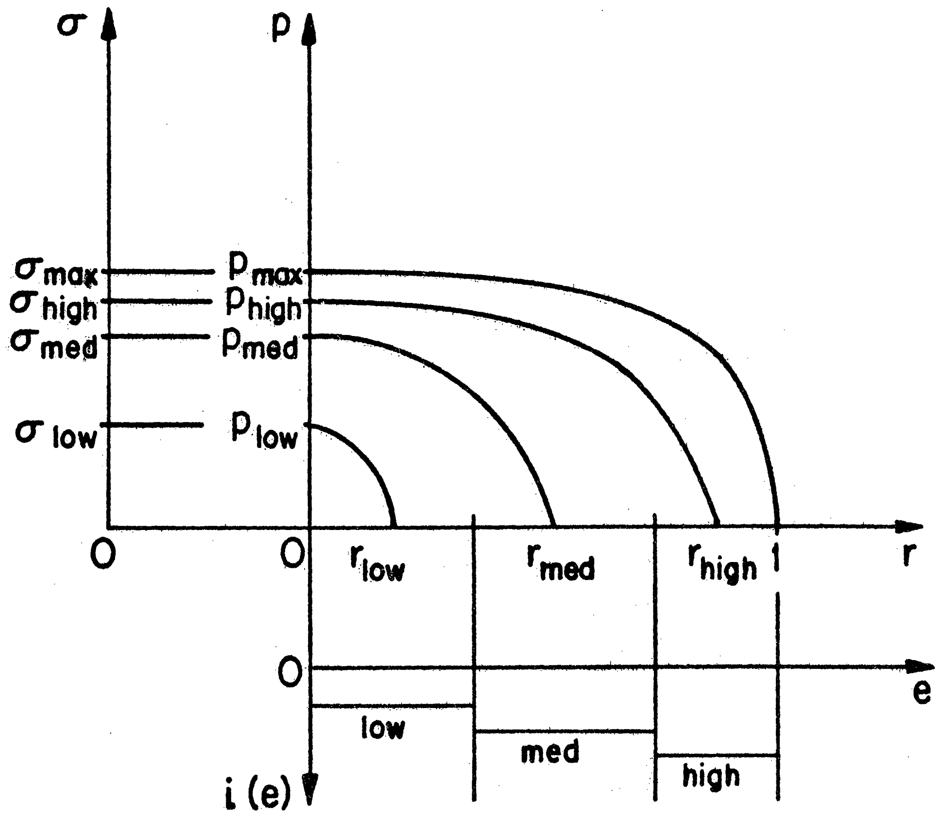
Then, referring to Figure 2, δ_ϕ is defined by

$\delta_\phi(i,\sigma) = (3,\sigma_i)$ for $i \in \{low, med, high\}$ and

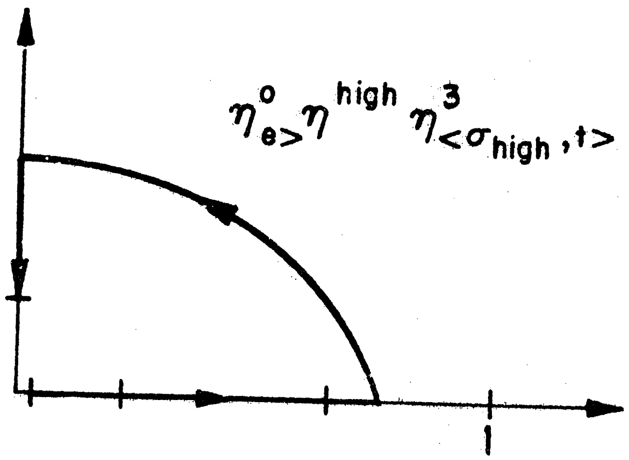
$\delta_\phi(3,\sigma) = (1,\infty)$

and δ is defined by

$\delta(1,\infty,e,1) = (i(e),T_{i(e)})$


σ
p
σmax
σhigh
σmed
σlow
pmax
phigh
pmed
plow
O
O
rlow
rmed
rhigh
1
r
O
e
low
med
high
i (e)

a)


$\eta^{o}_{e>}\eta^{high}\,\eta^{3}_{<\sigma_{high},t>}$
1

b)

Fig. 2

and

$\delta(i,0,e,1) = (i,\sigma-e)$ in all other cases (invading prey ignored)

As a simulation program this takes the form:

1. Wait until a prey invades
2. Obtain elapsed time e

   Hold for time $T_{i(e)}$

   Cause emigration of $f_{i(e)}$ prey
3. Hold for time $\sigma_{i(e)}$
4. Go to 1

APPENDIX 1

REVIEW OF INPUT SET RELATED CONCEPTS

(Z,T), where Z is a set and T is a time base (reals or integrals), denotes the set of all segments $\omega:\langle 0,\tau\rangle \to Z$ where $\tau > 0$. The angular brackets represent a fixed choice from the set $\{(\ ],[\ ),[\ ]\}$. (Z,T) is a semigroup under composition; $\omega\cdot\omega'$ is the segment obtained by translating ω' so that as to be contiguous with ω and concatenating.

For $\omega:\langle 0,\tau\rangle \to Z$, $\mathrm{dom}(\omega) = \langle 0,\tau\rangle$, $\ell(\omega) = \tau$; $\omega_{t\rangle} = \omega|\langle 0,t\rangle$ and $\omega_{\langle t} = \omega|t,\tau\rangle$ are left and right segments of ω at t respectively.

ADMISSIBLE GENERATING SETS

For a subset $\Gamma \subseteq (Z,T)$, the semigroup generated by Γ is denoted Γ^+. A decomposition $\omega_1,\omega_2,\ldots,\omega_n$ of ω by Γ is a maximal length segment (mls) decomposition, if each ω_i is the longest generator in Γ which is a left segment of $\omega_i\ldots\omega_n$. The points $\{\ell(\omega_1),\ell(\omega_1)+\ell(\omega_2),\ldots,\sum_{i=1}^{n-1}\ell(\omega_i)\}$ are called the breakpoints of the decomposition $\omega_1,\omega_2,\ldots,\omega_n$. The length of this decomposition is n.

Γ generates Ω if $\Gamma^+ = \Omega$ and admissibly generates Ω if moreover each $\omega \in \Omega$ has an mls decomposition by Γ (it is unique if it exists).

A semigroup Ω is admissibly generated by itself. Γ non trivially admissibly generates Ω if Γ is a proper subset of Ω which admissibly generates it.

Sufficient, but not necessary conditions, for admissibility are:

Theorem A.1 (P. 220, [Zeigler, 1976]).

Γ admissibly generates Γ^+ if

a) $\omega \in \Gamma^+ \Rightarrow \max\{t \mid \omega_{t>} \in \Gamma\}$ exists

b) $\omega \in \Gamma \Rightarrow \omega_{<t} \in \Gamma$ for all $\tau > 0$ in $\mathrm{dom}(\omega)$.

[Zeigler, 1976], P.220 gives an example where b) fails. An example where a) fails is the following:

Let $\Gamma = \{a_t \mid t \in \text{irrationals}$; here a_t denotes the segment of length t and constant value a . Then $\Gamma^+ = \{a_t \mid t \in \text{reals}\}$. Γ does not admissibly generate Γ^+ since for integer τ , the set $\{t \mid t \text{ is irrational} < \tau\}$ has no maximum.

DISCRETE EVENT SEGMENTS

Let X be a set of "events" and T be the reals.

$\Gamma_{DEVS(X)} = \Omega_\phi \cup \Omega_X$ where Ω_ϕ is the set of no event segments $\{\phi_\tau \mid \tau > 0\}$ where $\phi_\tau(t) = \phi$ for all $t \in \langle 0,\tau\rangle$ and Ω_X is the set of one event segments $\{x_\tau \mid \tau > 0\}$, where $x_\tau(t) = \phi$ for $0 < t < \tau$ and $x_\tau(0) = x$. Here $\langle\ \rangle$ is fixed to be $[\ )$.

$\Gamma_{DEVS(X)}$ and $\Omega_{DEVS(X)} = \Gamma^+_{DEVS(X)}$ are called discrete event generators and segment sets, respectively. $\Gamma_{DEVS(X)}$ admissibly generates $\Omega_{DEVS(X)}$, (P. 238, [Zeigler, 1976]).

ENCODINGS

Let $\Omega \subseteq (X,T)$ and $\Omega' \subseteq (X',T')$ be semigroups. A mapping $g:\Omega' \to \Omega$ is called an encoding of Ω' by Ω .

An encoding $g:\Omega' \to \Omega$ is invertable if

1. g preserves left segmentation : ω' a left segment of ω implies $g(\omega')$ is a left segment of $g(\omega)$.

2. g is one-one on lengths : $\ell(\omega) = \ell(\omega') \Leftrightarrow \ell(g(\omega)) = \ell(g(\omega))$.

Lemma A.2 (P. 271, [Zeigler, 1976]).

g is invertable $\Longleftrightarrow$ there is a one-to-one function MATCH such that

$$\text{MATCH}(\ell(\omega)) = \ell(g(\omega))$$

$$g(\omega_{t>}) = g(\omega)_{\text{MATCH}(t)>}$$

Let Ω and Ω' be closed under right and left segmentation. Then MATCH is a monotonically increasing function with inverse MATCH^{-1} defined for all positive values of T .

An encoding g is a homomorphism if $g(\omega\,\omega') = g(\omega)g(\omega')$ for all $\omega,\omega' \in \Omega'$.

Let Γ admissibly generate Ω' and $\not{g}:\Gamma \to \Omega$. Then g is the unique extension of $\not{g}$ to Ω' where $g(\omega) = \not{g}(\omega_1)\ldots\not{g}(\omega_n)$ and $\omega_1,\ldots\omega_n$ is the mls decomposition of ω by Γ .

APPENDIX 2

RELATION BETWEEN HOMOMORPHISM AND SIMPLICITY

We remark that the properties of invertability and homomorphism are completely independent. For example, the map g is invertable, but not a homomorphism, where $g(\omega)(t)=\int_0^t \omega(t')dt'$, and the map g is homomorphism but not invertable, where $\Omega' = \{\text{piecewise constant segments}\}$ and g is the unique extension to Ω' of $\not g : \{\text{constant segments}\} \to \Omega$, where

$$\not g(a_\tau) = \begin{cases} a_\tau & \text{if } a \geq 0 \\ a_{2\tau} & \text{otherwise} \end{cases}$$

We say that g is a <u>homomorphic invertable</u> encoding if it is both invertable and a homomorphism.

A homomorphic invertable encoding allows only length dependent time scale change (the function MATCH), and memoryless segment by segment mapping. A class of examples of homomorphic invertable encodings is given by the point-by-point, uniform dilation encoding g , where $g(\omega)(t) = \not g(\omega(t/\tau))$ where $\not g : X' \to X$ and τ are arbitrary.

Ω is <u>left cancellable</u> if $\omega = \mu\eta \Rightarrow \quad = \omega_{<\ell(\mu)}$. The choice $\langle\ \rangle = [\ )$ with the natural composition, render Ω left cancellable. So does the choice $[\ ]$ with right segment's endpoint, determining the value at the joint.

<u>Assertion A.3</u>

Let Ω be left cancellable. If g is a homomorphic invertable encoding into Ω then g is a simple encoding.

<u>Proof</u>

If it is readily shown that g preserves both left and right segmentation at all points, including breakpoints, i.e.,

$g(\omega_{t>}) = g(\omega)_{MATCH(t)>}$ and $g(\omega_{<t}) = g(\omega)_{<MATCH(t)}$.

Q.E.D.

It is important to note that a simple encoding need not be a homomorphism. For example, let $\Omega' = \Omega_\pi$ of Example, Section 2, and let $\Omega = \Omega_{DEVS}$. Let $g:\Gamma_\pi \to \Gamma_{DEVS}$ be defined by $\not{g}(\omega) = x_\tau$ where $x = \omega(0)$ and $\tau = \ell(\omega)$. Let $g:\Omega_\pi \to \Omega_{DEVS}$ be the unique extension of $\not{g}$ defined by $g(\omega) = \not{g}(\omega_1)\ldots\not{g}(\omega_n)$, where $\omega_1,\ldots,\omega_n$ is the mls decomposition of ω . It is readily shown that g is invertable and preserves right segmentation at breakpoints, but is not a homomorphism since if $\omega,\mu \in \Gamma_B$, then $\omega\mu \in \Gamma_B$ and $g(\omega\mu) = \omega(0)_{\ell(\omega)+\ell(\mu)}$. But $g(\omega)g(\mu) = \omega(0)_{\ell(\omega)}\mu(0)_{\ell(\mu)}$, and $g(\omega\mu)(\ell(\omega)) = \phi \neq \mu(0) = g(\omega)g(\mu)(\ell(\omega))$.

APPENDIX 3

THE EVENT RELATION

Consider the sets Γ = {finite degree polynomial} and Γ_m {piecewise finite degree polynomial having derivatives at leat to order m}. Then for $\omega, \mu \in \Gamma$,

$\omega \underset{0}{\equiv} \mu \Leftrightarrow \omega$ is a left segment of μ or conversely

(polynomials which agree on a non zero interval are identical; Levi, 1968), and for $\omega, \mu \in \Gamma_m$,

$\omega \underset{0}{\equiv} \mu \Leftrightarrow \omega$ and μ start with the same polynomial segment.

Thus $\Gamma_m/\underset{0}{\equiv}$ is an infinite dimensional vector space isomorphic to Γ. We can show this is true for any subvector space of Γ^+ which generates it.

Proposition A.4

Let Ω_G be a subvector space of Γ^+ which generates Γ^+. Then $\Omega_G/\underset{0}{\equiv}$ is an infinite dimensional vector space isomorphic to Γ.

Proof

It is readily shown that $\Omega_G/\underset{0}{\equiv}$ is a vector space. Since $\Omega_G \subseteq \Gamma^+$, for every $\omega \in \Omega_G$ there is a τ such that $\omega_{\tau>} \in \Gamma$. Moreover, $\omega \underset{0}{\equiv} \omega_{\tau>}$. Thus there is a linear transformation taking $\Omega_G/\underset{0}{\equiv}$ into Γ (or better its representation) given by mapping $[\omega]_0$ onto the sequence of derivatives $(\omega(0), \omega'(0), \omega''(0), \ldots)$.

Conversely, since $\Gamma \subseteq \Omega_G^+$, for every $\omega \in \Gamma$, there is a τ such that $\omega_{\tau>} \in \Omega_G$. Again, since $\omega \underset{0}{\equiv} \omega_{\tau>}$ there is a linear transformation mapping Γ into $\Omega_G/\underset{0}{\equiv}$ given by mapping ω into $[\omega_{\tau>}]_0$. Thus $\Omega_G/\underset{0}{\equiv}$ and Γ are isomorphic as vector spaces.

Q.E.D.

Theorem A.5

Let Ω be the piecewise analytic segments or the piecewise continuous segments. Then for any vector space generator set Ω_G of Ω , $\Omega_G/\equiv_0$ is infinite dimensional.

Proof

$\Gamma^+ \subseteq \Omega$ in both cases. $\Omega_G^+ \supseteq \Gamma^+$ and in fact it is easy to show that $\Omega_G \cap \Gamma$ generates Γ^+ . Since $\Omega_G \cap \Gamma$ is infinite dimensional, so is Ω_G .

APPENDIX 4

REVIEW OF SYSTEM SPECIFICATIONS AND MORPHISMS

$G = \langle T, X, \Omega_G, Q, \delta_G, \pi \rangle$ is an iterative specification if $\Omega_G \subseteq (X,T)$ is an admissible generating set, and $\delta_G : Q \times \Omega_G \to Q$ extends uniquely to a transition function $\bar{\delta} : Q \times \Omega_G^+ \to Q$ having the composition property. We have modified the definition of Zeigler 1976, P. 223, by omitting Y and λ and requiring instead the specification of a finite intersection cover of Q .

DISCRETE EVENT SPECIFICATION

A DEVS $M = \langle X_M, S, \delta, t \rangle$ where X_M and S are sets (external events and sequential states, respectively); $t : S \to R^+_{0,\infty} = [0,\infty]$ is the time advance function; and $\delta : Q_M \times (X_M \cup \{\phi\}) \to S$ where $Q_M = \{(s,e)\ s \in S,$ $0 \leq e \leq t(s)\}$ and $\delta(s,e,\phi) = \delta_\phi(s)$ where $\delta_\phi : S \to S$ is called the internal event transition function.

A DEVS M is legitimate if for each $(s,e) \in Q_M$ and $\tau > 0$, the number of transitions in the interval $(0,\tau)$, made starting form (s,e) , $m_{s,e,\tau}$ is finite. (See Zeigler, 1976, P.237 for formal definition.)

Theorem A.6 (Zeigler, 1976, P. 238)

A legitimate DEVS M induces an iterative specification $G(M) = \langle T, X, \Gamma_{DEVS(X)}, Q_M, \delta_G \rangle$ where $T = R$, $X = X_M \cup \{\phi\}$ and $\delta_G : Q_M \times \Gamma_{DEVS(X)} \to Q_M$ is given by

$$\delta_G(s,e,\phi_\tau) = \begin{cases} (s, e+\tau) & \text{if } e+\tau \leq t(s) \\ \delta_G(\delta_\phi(s), 0, \phi_{e+\tau-t(s)}) & \text{else} \end{cases}$$

and

$$\delta_G(s,e,x_\tau) = \delta_G(\delta(s,e,x),0,\phi_\tau) \quad .$$

SYSTEM

A system $S = <T,X,\Omega,Q,\delta>$ where T is the time base (R or I) X is the input value set, $\Omega \subseteq (X,T)$ is the input segment set, Q the state set and $\delta : Q \times \Omega \rightarrow Q$, the transition function. We require that Ω is a semigroup under composition and δ has the composition property : $\delta(q,\omega\omega') = (\delta(q,\omega),\omega')$

An iterative specification G specifies a system S_G where $\Omega = \Omega_G^+$ and $\delta = \bar{\delta}_G$.

MORPHISMS

A system morphism from S to S' is a pair (g,h) where $g:\Omega' \rightarrow \Omega$ and $h:\bar{Q} \xrightarrow{\text{onto}} Q'$ where $\bar{Q} \subseteq Q$ such that for all $q \in \bar{Q}$, $\omega \in \Omega'$,

$$h(\delta(q,g(\omega))) = \delta'(h(q),\omega) \quad .$$

A specification morphism from G to G' is a pair $(\emptyset,h)$ where $\emptyset : \Omega'_G \rightarrow \Omega_G^+$, h is as in the system morphism, and the commutative relation is required to hold only for $q \in \bar{Q}$.

Theorem A.7 (Zeigler, 1976, P. 271, 274)

A specification morphism $(\emptyset,h)$ from G to G' induces a system morphism (g,h) from S_G to $S_{G'}$ where g is the extension of $\emptyset$ to $\Omega_G'^+$. If moreover, g is invertable,

$STRAJ_{h(q),\omega}(t) = h(STRAJ_{q,g(\omega)}(MATCH(t)))$ for all $q \in \bar{Q}$, $\omega \in \Omega_G'^+$ . .

APPENDIX 5

EQUIVALENCE OF DEVS SYSTEM AND SPECIFICATION MORPHISMS

Theorem A.8

Let M be a DEVS, S a system and (g,h) a system morphism from $S_{G(M)}$ to S. If g is a simple encoding, then (g,h) is induced by a specification morphism (ϕ,h) form G(M) to G where $S_G = S$.

Proof

Since g is simple, by Proposition 2, $\Gamma = g^{-1}(\Gamma_{DEVS})$ admissibly generates Ω. Let G have the same objects as S except that $\Omega_G = \Gamma$ and $\delta_G = \delta|\Gamma$. It is easily shown that G is an iterative specification and $S_G = S$. Also (ϕ,h) is a specification morphism from G(M) to G, where $\phi = g|\Gamma$ which induces the morphism (g,h) from $S_{G(M)}$ to S.

Q.E.D.

Since every specification morphism (ϕ,h) from G(M) to G induces a system morphism (g,h) from $S_{G(M)}$ to S_G, we have

Corollary A.9

A system S is DEVS simulateable if, and only if, it has an interative specification which is DEVS simulateable.

(By a system S being DEVS simulateable, we mean that there is a system morphism (g,h) from some $S_{G(M)}$ to S such that g is simple and states of M represent blocks of π as in the DEVS morphism definition.)

APPENDIX 6

FINITE DEVS LIMITATIONS

Theorem A.10

For a finite DEVS M let the set $E_{s,x,s'} = \{e | \delta(s,x,e) = s'\}$ be a union of a finite number of intervals for each s,x,s' for which $E_{s,x,s'}$ is not empty. Then M is DEVS simulateable by a finite DEVS $\hat{M}$ for which $\hat{\delta}(s,e,x)$ is independent of e for all $s \in S$, $x \in X$.

Proof

The union of $E_{s,x,s'}$ over all $x \in X$ and $s' \in S$ is just $[0,t(s)]$. Since each $E_{s,x,s'}$ is a finite union of intervals and there are finitely many pairs (x , s) , the intersection all these sets finitely partitions $[0,t(s)]$ and refines every $E_{s,x,s'}$. Let $0 < t_1 < t_2 \ldots < t_n = t(s)$ be this partition. In $\hat{M}$, replace s by states $(s,1),\ldots,(s,n)$ such that $\hat{\delta}_\phi(s,i) = (s,i+1)$ for $i = 1,\ldots,n-1$, and $\hat{\delta}_\phi(s,n) = (\delta_\phi(s),1)$; $t(s,i) = t_{i+1} - t_i$; and $\hat{\delta}((s,i),e,x) = \delta(s,t_i,x)$. Clearly, $\hat{\delta}((s,i),e,x) = \hat{\delta}((s,i),e',x)$ for all $e,e' \in [0,t(s,i)]$ as claimed. It is easy to show that $\hat{M}$ DEVS simulates M .

Q.E.D.

The theorem indicates that a finite DEVS with a reasonable clock dependence, can be only finitely sensitive to the arrival time of an external event. Any timing of arrivals that such a device can do, can also be done by a succession of a finite number of states in which the elapsed time within a state plays no role in determining the next state. In effect, the simulation clock has been replaced by a finite counter.

Note that the proof depends on the finite number of sets $E_{s,x,s'}$

so would not go through for a finite explicit form DEVS which may have an infinity of sets $E_{(\bar{s},\sigma),x,(\bar{s}',\sigma')}$

While the foregoing indicates the "architectural" weakness of finite DEVS systems, it does not establish that there are I-O behaviors "tuned" to this weakness. We now provide an example of an I-O behavior not finite DEVS realizable, but finite explicit form DEVS realizable.

Theorem A.11

There are I-O function behaviors of finite explicit form DEVS systems which are not realizable by any finite DEVS system.

Proof

Consider a finite explicit form DEVS $M = \langle X,S,Y,\delta,\lambda, \rangle$ where $X = \{1\} = Y$, $\bar{S} = \{\underline{ref},\underline{act}\}$, $T(\underline{ref}) = T_{ref} > 0$, $T(\underline{act}) = \infty$ and

$$\delta_{\phi}(\underline{ref},\sigma) = (act,\infty)$$

$$\delta(\underline{ref},\sigma,e,1) = (ref, \sigma - e)$$

$$\delta(\underline{act},\infty,e,1) = (\underline{ref},T_{ref})$$

$$\lambda(\bar{s},\sigma,e) = \begin{cases} 1 & \text{if } s = \underline{ref} \text{ and } \sigma = e \\ \phi & \text{otherwise} \end{cases}$$

The I-O behavior of this M is characterized by the fact that in state act (active) it waits until receiving an input pulse, whence it switches to state ref (refractory). It remains in state ref for a finite refractory period T_{ref} , during which all input pulses are ignored, and then reverts to state act while signaling this event by an output pulse (Figure 3).

Now let $\hat{M}$ be a finite DEVS which realizes that behavior of M at the I-O function level (Zeigler, 1976, P. 205). In particular, let $(\bar{s},\bar{e})$

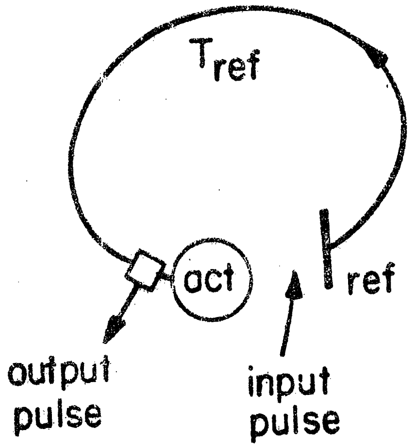


Fig. 3

realize the input segment to output segment mapping from state $(ref, T_{ref}, 0)$ in $S_{G(M)}$. Since $\hat{M}$ has a finite set $\hat{S}$, $\min\{t(\hat{s}) \mid \hat{s} \in \hat{S}\}$ exists and is not zero, (assume $\hat{M}$ has no transitory states, without loss of generality, Zeigler, 1976, P.24). Let $\varepsilon = \frac{1}{2} \min\{t(s)\}$ and apply any segment $\phi_\tau 1_e$ for $0 < \tau < T_{ref} - \varepsilon$ to $(\bar{s}, \bar{e})$. By the DEVS operation the state at the end of this segment is of the form $(\hat{s}, \varepsilon)$ (the pulse at time τ causes an immediate transition to some "unseen" state $(\hat{s}, 0)$ and since $\varepsilon < t(\hat{s})$, we are in state $(\hat{s}, \varepsilon)$, ε seconds later). Since $\hat{S}$ is finite, there are τ_1, τ_2, $0 < \tau_1 < \tau_2 < T_{ref} - \varepsilon$ such that $S_{G(\hat{M})}$ is in the same state $(\hat{s}, \varepsilon)$ at both τ_1 and τ_2. The behavior from this state cannot be different in both cases. But as can be seen by applying a segment $\phi_{T_{ref} - \tau_2}$, this contradicts the behavior of $S_{G(M)}$.

Q.E.D.

REFERENCES

Bablich, A.F., J. Grason and D.L. Parnas (1975) "Significant Event Simulation, C.A.C.M., 18:323-329.

Hogeweg, P. (1975) "Concepts For Simulation, A Comparison Based on Simulation Languages", In: Systeemleer, Oud and Buffort (eds.), Stenford Kroese, Leiden.

Levi, H. (1968) Polynomials, Power Series and Calculus, D. Van Nostrand, Princeton, N.Y.

Veech, W.A. (1967) A Second Course in Complex Analysis, W.A. Benjamin, New York.

Zeigler, B.P. (1976) Theory of Modelling and Simulation, Wiley & Sons, N.Y.

Zeigler, B.P. (In Preparation) "Systems Simulateable by the Digital Computer, Part 2: Discrete Event Network Models".

Zeigler, B.P. and A. Barto (1977) "Alternative Formalisms for Bio- and Eco-Systems Modelling", Simulation (In Press).

Zeigler, B.P. (1977) "Multi-Level, Multi-Formalism Modelling: An Ecosystem Example", Proc. First Int. Conf. on Math. Modelling, St. Louis, Mo.