International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Discrete Event Models for Cell Space Simulation[1]

Bernard P. Zeigler

Department of Computer Science, Wayne State University, Detroit, Michigan 48202

Received May 7, 1981

Cellular automata are a well-known class of models that characterize many of the essential properties of spatially distributed parallel systems. At a yet more fundamental level, Burks has employed the cellular automaton modeling formalism to formalize what is meant by "law of nature." In this article, we introduce a broader class of spatially distributed models called discrete event cell space models. Comparing the old and new formalisms as adequate and convenient vehicles for representing various natural and artificial systems, we suggest the potential of the discrete event approach for modeling at the level of fundamental physics.

1. CELLULAR AUTOMATA AND THE LAWS OF NATURE

Arthur Burks, when explicating what is meant by a "law of nature" states that it has three main features: uniqueness, modality, and uniformity (Burks, 1977, p. 425). Very briefly, "uniqueness" means that the law makes unequivocal assertions, "modality," that it holds not only for actual, but also possible, situations, and "uniformity," that it applies uniformly to all points in space and time. Proceeding to formalize these concepts, Burks presents cellular automata as model systems in which such laws of this nature are readily comprehended. (Burks, 1977, p. 562). A cellular automa-

[1]This work was prepared with the assistance of NSF Grant No. MC578-26016. A more extended version under the title "Cellular Space Models: New Formalisms For Simulation and Science" will appear in a collection in honor of Arthur W. Burks, edited by M. Salmon and to be published by Reidel.

0020-7748/82/0600-0573$03.00/0

ton[2] is characterized by specifying three parameters: a set of states S, a neighborhood N, and a transition function T. The interpretation of the triple $\langle S, N, T \rangle$ is as follows:

> One imagines a checkerboard stretching out towards infinity in both north-south and east-west axes. At each square is located a cell with state set S. The neighbors of a cell located at square (i, j) (where as will be apparent "neighbors" is intended in an informational sense) are simply determined from the neighborhood N: in fact, N is a finite ordered set of integer pairs and the neighbors of this cell are located at the squares obtained by adding (i, j) to each pair of N (vector addition). Now imagine a global state of this system pertaining at some time instant t, i.e., assign to each cell a state from the set S. Then the system will move to a succeeding global state at time instant $t+1$ which is determined as follows: Simultaneously, for each cell apply the transition function T to the states of its neighbors and let the result be the state of the cell at $t+1$.

Within such a framework, Burks identifies a law of nature as asserting a property of the transition function. Indeed, further placing a contiguity requirement on such a law—that it assert a Humean direct spatiotemporal cause–effect relation—it can be identified with a statement of the following form:

> If the states of the neighbors of a cell satisfy P at time t then the state of the cell will satisfy Q at time $t+1$.

A somewhat less general way to capture this idea is that a law of nature reveals what the transition function T is for one or more of its arguments.

Let us see how cellular automata universes exemplify the three features of natural laws. "Uniqueness" is embodied in the transition relation being a function, i.e., specifying a unique next state for a cell as determined by the present states of its neighbors. In the system theoretic context this feature is called *local determinism*, and also implies *global determinism*: every global state of the system has a unique successor.

"Modality" is exemplified by the fact that there are many possible initial global states of the system, each initiating a state trajectory which can be postulated to be the actual history of our universe. Only one can in fact represent the actual universe while the others represent "causally possible"

[2]More precisely, we are restricting our discussion to two-dimensional, deterministic cellular automata. The reader wishing more background in the subject may consult Burks (1970), Barto (1975), and Zeigler (1976, Chap. 4).

universes, i.e., state histories that could have happened according to the laws of nature—the transition function—but did not because the initial state was as it was. Burks formalizes this "could have happened" (counterfactual) feature by means of the nonparadoxical causal implication operator of modal logic:

*) For all cells c and times t, the neighbor states of c satisfy P at t nonparadoxically causally implies that the state of c satisfies Q at $t+1$.

Such an implication asserts a transition relation which holds for all causally possible universes including the postulated actual one.

"Uniformity" is embodied in the universal quantification over space and time of the foregoing scheme. More specifically, every cell no matter where located, has the same state set and the same transition function. And while neighborhoods are not literally the same, they are all "isomorphic" in the sense that any one is a translation of any other [or indeed a translation of the set N which turns out to be the neighborhood of the origin $(0,0)$]. Thus a cellular automaton specifies a system which is both discrete and invariant in both time and space.

While cellular automata exemplify the features of natural laws, it does not follow that they are the only formal models which do so, or indeed, that these features are mandatory for any formalism to be useful for science. In this paper we shall examine some of the questions raised by the foregoing observations. We shall introduce a class of cellular models called discrete-event cell spaces which also exhibit some features of natural laws but which are more convenient to employ than the cellular automata in many simulations of natural and artificial systems. Indeed, we show that the representation of phenomena of interest to physics such as motion, collision, and "action at a distance" is natural in the discrete-event framework while stilted or impossible for cellular automata.

2. DISCRETE-EVENT CELL SPACE MODELS

Discrete-event cell space models preserve the spatial discreteness of cellular automata as well as their space and time invariance. However, the time base is no longer discrete but is continuous, that is to say, there is no intrinsic time step for such models. Events, i.e., cellular state transitions, may occur at irregularly spaced intervals, not necessarily synchronized to the beat of a clock as in the cellular automaton case. These events are all scheduled to occur as a consequence of the actions of cells: a cell may schedule events to occur to itself as well as to its neighbors, and these events may in turn schedule other events, and so on.

Next-event models form a subclass of the more general discrete-event models—other subclasses of the latter are the activity scanning and process interaction models.[3] Activity scanning and process interaction cell space models can be defined by combining the cell space structure with the appropriate dynamic structure, as we now shall do for the next-event models.

A *next event cell space* (NEVS) is specified by a quadruple $\langle S, N, T, \text{SELECT} \rangle$, where the first three parameters bear a resemblance to those of the cellular automaton, but their interpretation is quite different. As before, one imagines an infinite checkerboard, with each square containing a cell. However, the state of any cell is now a pair (s, σ), where s is an element of S and sigma is a nonnegative real number (it will be useful to allow sigma to take on the value of infinity ∞ as well). A cell in state (s, σ) will remain in "sequential" state s for a time σ before undergoing a self-induced transition. However, in the meantime its state may be altered as a result of some other cell's action. Thus s and σ are called the *sequential state* and *time left* components of the total state (s, σ). If cell c is in state (s, σ) at time t, this can be expressed as

*) Cell c is in sequential state s and is scheduled for a transition in time σ (or at time $t + \sigma$).

As before the neighborhood of a cell is computed by adding its coordinates to the prototype neighborhood N (a finite set of pairs as before). When a transition event occurs to a cell, it and and its neighbors undergo an immediate state change as dictated by the transition function T. Thus, in contrast to the automaton case, T takes a list of pairs (s, σ)—the total states of the cell and its neighbors just before the event—and produces a list of such pairs—giving the total states of these cells just after the transition.

In contrast to the cellular automaton, the updating of an NEVS model takes place at irregularly spaced computation instants. Let t_i be such an instant; then its successor t_{i+1} is computed as follows:

Imagine the global state at time t_i to be a list of pairs (s, σ) containing the total states of all the cells at this time. Let σ^* be the minimum of all the σ's on the list. Then $t_{i+1} = t_i + \sigma^*$.

The global state at time t_{i+1} is computed as follows:

Call the cells whose σ's at t_i were equal to σ^* the IMMINENT cells. These cells are all scheduled for a transition event at t_{i+1}. If there are more

[3]The discrete-event formalism and its subformalisms express the models implemented in discrete-event computer simulation languages. For background in this area, the reader is referred to Zeigler (1976, Chap. 9).

than one, apply the SELECT function to choose one of the IMMINENT cells, call it c^*. (Thus the SELECT function provides the desired tie breaking rules when more than one transition event is scheduled for activation at the same time.) The chosen cell c^* carries out the transition function as indicated above resulting in a new global state which is related to that pertaining at t_i as follows:

The total states of the neighbors of c^* are as dictated by T; the total state of every other cell is converted from (s, σ) to $(s, \sigma - \sigma^*)$.

3. EXAMPLES OF DISCRETE-EVENT CELL SPACE MODELS

Some examples of next-event cell space models will be given to illustrate their operation and versatility. They will also serve to point out the differences in expressive capability relative to cellular automata.

3.1. Example 1: Motion of a Single Body. Let us begin simply by expressing the motion of a single object in a NEVS. The object is to move to the right, at a speed of one square every MOVE_TIME seconds. An appropriate NEVS is defined as follows:

Every cell has two states, one to indicate the presence of the object and the other its absence: thus $S = \{0, 1\}$. The absence indicator 0 is a *passive* state, i.e., a cell in this state will never become active of its own accord. This is represented by the total state pair $(0, \infty)$ which indicates that the time left for a cell in sequential state 0 to change state is ∞. On the other hand, the presence indicator 1 is an *active* state (opposite of a passive state); indeed, it is the source of motion in this example.

To get the object to move we must arrange for a cell in state 1 to cause its right adjacent (physical) neighbor to enter state 1 after staying in this state for MOVE_TIME seconds. Thus for the neighborhood N we require only the two integer pairs $\{(0,0),(1,0)\}$—adding (i, j) to each we obtain $\{(i, j),(i+1, j)\}$, the coordinates of the cell at (i, j) and it's right neighbor.

In the light of the above comments, the transition function can be expressed as follows:

*) set your own sequential state to 0 and passivate (set your own time-left to ∞)

set right neighbor state to 1 and schedule a transition event there in MOVE TIME
(set neighbor's time-left to MOVE TIME)

The SELECT function is arbitrary in this example since at most one event is scheduled at any time.

Starting from an initial global state in which all but one cell are in state $(0, \infty)$, the exception being in state (1, MOVE TIME), successive global

states are generated every MOVE TIME seconds in which a 1 is seen to travel horizontally to the right.

3.2. Example 2: Traffic Congestion. Note that the above model does not specify what happens in the case of collisions—when two or more objects want to move into the same square. The present example exemplifies this phenomenon. Consider cars which move to the right at a constant speed when possible (traveling in an east-bound lane for example). When the car immediately ahead obstructs progress a car seeks to pass preferably to the right, but also to the left if the first is not possible. The patterns of traffic congestion which develop can be studied with NEVS models of which the following is prototypic:

The state set $S = \{0, 1\}$ to represent absence and presence of a car as in Example 1. We put in the neighborhood N all cells immediately above, below, and to the right of the center cell to allow for the possibility that a car may move in these directions if direct progress is blocked. Let N be ordered in the order of preferred motion as illustrated in Figure 1. Then the transition function T can be expressed as follows:

```
*) Find the first unoccupied neighbor cell
     (scanning in the preference order)
   set this cell to state 1 and schedule a
     transition event there in MOVE_TIME
   set your state to 0 and passivate.
   leave all other cells unaffected
     (let their sequential states remain unchanged and subtract
     your time-left (σ*) from theirs)[4]
   if no unoccupied neighbor exists
     reschedule yourself in time CHECK AGAIN TIME
     leave all other cells unaffected.
```

The SELECT will determine which of two or more cars contending for the same square at the same time will move into it. Thus, the order in which IMMINENT cells are selected does make a difference in this example. SELECTION, based on "right of way" principles, for example, therefore incorporates essential model hypotheses in this case.

An initial global state in this model consists of putting a finite set of cells in sequential state 1 with a time-left σ between 0 and MOVE TIME; all other cells are set to the passive $(0, \infty)$ state. The pattern then generally moves right with rearrangements as cars take diversionary maneuvers to bypass obstructions.

[4]This is to update the time left values appropriately as with all nonneighboring cells in the space.

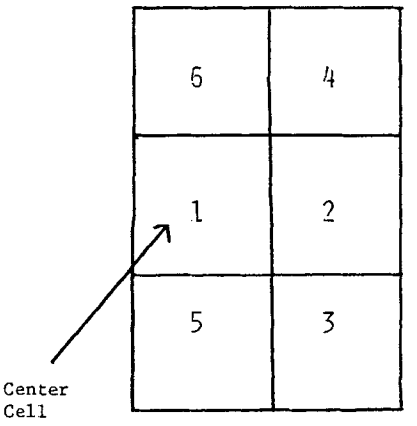

Fig. 1. Neighborhood for traffic congestion model

Note that to make the simulation more realistic, MOVE_TIME and CHECK_AGAIN_TIME can be made to be random variables so that cars move at randomly assigned rather than constant speeds.[5]

3.3. Example 3: Plant Growth: Gravity Signal. This example shows how next-event models can handle "action at a distance" in a very natural manner. Consider embedding a plant in a cell space as illustrated in Figure 2. Growth is made to occur at the tip of the plant by sprouting a new cell every GROWTH_TIME hours. The extra weight is transmitted to the stem and base cells by a gravity signal which travels instantaneously downward. Thus at the same computation instant at which the tip extends itself, the weights on the other cells are simultaneously increased—and this happens no matter how long the plant becomes. The following NEVS will exhibit this behavior:

Let the state set S be $\{0,1,2,3,\ldots\}$, where 0 is a passive environmental state and $i>0$ is a plant cell supporting a total weight of i units (itself plus $i-1$ cells above it). The neighborhood N consists of the center cell and the cells above and below it. The transition function T is expressed as follows:

*) If you are the tip (center cell sequential state = 1)
set the upper neighbor to state 1 and
schedule a transition event to
occur there in GROWTH_TIME
if the lower neighbor is a plant cell (state $= i > 0$)

[5] Zeigler (1976, Chap. 6) shows how this is done formally.

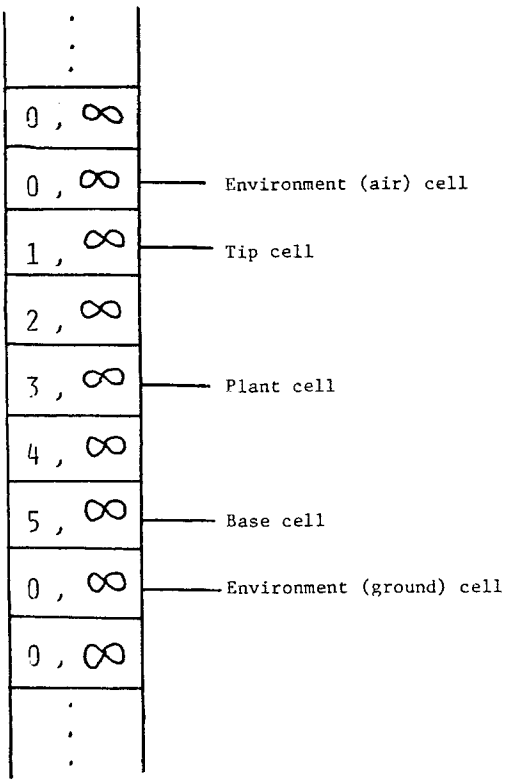

Fig. 2. Embedding of a plant in a next-event cell space.

add 1 to its state and schedule a
transition event to occur there in 0 time
set yourself to state 2 and passivate

Otherwise (you are a plant cell)
if the lower neighbor is a plant cell
add 1 to its state and schedule a
transition to occur there in 0 time.

Note that if the activated cell is a tip then it carries out three actions: sprout a new tip upward, start a gravity signal downward and transform itself to a passive plant cell; if the activated cell is a plant cell it passes on the gravity signal (unless it is a base cell known by the fact that its lower neighbor is an environmental cell).[6]

[6]Studies of plant growth using discrete event models may be found in Hogeweg (1980). Actually, it is interesting that a further extension of the cell space formalism is necessary to conveniently express growth in the interior of a plant—the neighborhood must be allowed to change (Lindenmeyer, 1968; Herman and Rosenberg, 1975).

When started in a global state such as illustrated in Figure 2, this NEVS will display plant extension with weight accumulation in "real time" (see later comparison with cellular automata).

3.4. Other Examples. Collisions which result in changes of state are naturally expressed in the NEVS framework. Such collisions occur, for example, when two molecules collide and combine chemically or when predators devour prey giving birth to new predator offspring. In an NEVS model, each cell may represent a "patch," a hospitable area isolated from other patches by inhospitable terrain. Prey species colonize such patches and are predated upon by predators. Both species tend to remain localized to a patch until forced by shortage of resources to migrate. How far migrants can get to is species dependent (depending on such factors as speed of travel, ability to survive without food, etc.). Studies have been done in which colonization and migration are made stochastic and the space is limited to a finite region. The evolution of the system is followed to see if global persistence of predators is maintained despite almost certain local extinction (Zeigler, 1977; Sampson and Dubreuil, 1979).

Cellular models are natural media for representing networks of computing devices. Indeed, von Neumann originally invented the cellular automaton framework to embed networks of logical elements with the purpose of exhibiting systems with self-reproductive ability (von Neumann, 1966). Subsequently, this framework under the rubric "tesselation or iterative arrays" has been extensively employed for hardware design. Extending the same principles to the discrete-event cell space, one can readily model networks of asynchronous elements for questions of logical correctness, coordination control, and reliability. Such networks may model computer systems at all levels, whether the components be electronic elements, hardware devices, or full-scale computers connected by communications links.

On the natural computing side, cellular automata have been used to model networks of neurons, and the extension to discrete-event models is natural (Barto, 1975, and citations therein).

4. CELL SPACE FORMALISMS: THE ADEQUACY ISSUES

We raised the issue of adequacy of model formalisms for expressing real-world relationships. This is a multisided question which we cannot hope to answer fully in this paper. But we can explore some of its aspects. One of these is a formal one: the relative expressive power of a formalism. We shall first comment upon this "hard" criterion as a backdrop for consideration of the "softer" questions such as convenience, versatility, naturalness, etc.

Expressive Power: A "Hard" Criterion. By the expressive power of a formalism is meant the class of systems it can specify (Zeigler, 1977b). We

can compare powers of two formalisms by asking whether every system specified by a model in one can also be specified by a model in the other. In this form, automata and discrete-event systems are incomparable since the time bases are distinct. Thus we are led to the more sophisticated formulation: can every system specified in one be "simulated" by a system specified in the other? Now the formal concept of "simulation" is as expressed in a hierarchy of system-preservation relations (articulated in Zeigler, 1976, Chap. 10). At the lowest level, we require only that one system reproduce the external behavior of another in order to grant it simulation capability; at the higher levels, we require that the first represent more and more of the internal structure of the second to be accorded such status. Thus as we ascend the hierarchy, simulation related systems will be "closer" together in structure and behavior. Conversely, as we descend, systems so related may be more and more different in their detailed operation. A consequence of this fact is that as we loosen our definition of simulation, we increase the expressive power of formalism but at the expense of burdening the observer with the task of making the translation required to interpret the behavior of the simulator as revealing that of the simulatee.

With this as prologue, let us compare the powers of the cellular automata and NEVS formalisms. We shall restrict our attention to computable models, i.e., models whose behavior can be computed by a(n) (idealized) digital computer.[7] Then it is easily seen that the cellular automata can simulate any NEVS in the sense of computing its behavior. The argument is simple: to be computable is more particularly to be computable by a Turing machine. Now any Turing machine can be embedded in a cellular automaton space [using von Neumann's construction or the very straightforward approach of Smith (1972)]. Composing the two simulations we get that any next-event cell space model can be simulated by some cellular automaton.

Of course, the concept of simulation employed in this result is a rather weak one (lies at the lower level of the hierarchy) and states little more than that a cellular automaton can be used as a computer to generate the behavior of a NEVS. It leaves the burden of writing and interpreting this simulation to the user and so says nothing of the ability of the automaton formalism to directly express NEVS-like models.

[7]For cellular automata, a cell state is "quiescent" if the global state in which all cells are assigned this state is an equilibrium state. A computable (or algorithmic) cellular automaton is one having at least one quiescent state called the "blank" state and whose initial global states are restricted to those in which only a finite number of cells are not in the blank state (Burks, 1977, pp. 566 and 567). In the NEVS case, the role of the blank state is played by a passive state (recall earlier definition). A computable NEVS has such a blank state and is initialized only from global states having a finite number of nonblank cells. We also require that all real numbers used must be representable by fixed precision rationals as they would be in a digital computer.

In fact, this ability is demonstrably limited when the simulation relation is tightened (hierarchical level increased) by requiring the simulation to be in "real time." This requires that the simulator reproduce the behavior of the simulatee in phase with it (allowing the simulator no intermediate computation[8]). Let us employ such a relation in which we require that each cell state of the simulator be interpretable as a cell state of the simulatee [we allow the simulator cells to have any (including infinite) number of states]. Under such a readout map we require that the simulator produce the same output trajectory as the simulatee when started in corresponding global states.

Using the above simulation relation we can show that every cellular automaton can be simulated by some NEVS but that the converse does not hold. Thus the next-event formalism is the more powerful in terms of direct model expression.[9] We sketch the proof of this result.

First the positive part. Let $\langle S, N, T \rangle$ specify a cellular automaton. We specify a NEVS $\langle S', N', T', \text{SELECT} \rangle$ as follows:

S' is composed of two copies of S, one to hold the present state of a cell after its next state is computed. Also there is a third component to S' which is a phase indicator: in phase 1, the next state of a cell is computed; in phase 2, the next state is transferred into the present state location and the inverse neighbors are activated (see below).

N' is the union of the neighborhood N and the inverse neighborhood $-N$. The inverse neighborhood consists of the set of cells directly influenced by the center cell: those cells to whose neighborhood the center cell belongs. The inverse neighborhood of the origin can easily shown to be $-N$, i.e., the negatives of each of the pairs of N (see Zeigler, 1976, Chap. 4).

The transition function T' is expressed as follows:

*) in phase 1:
apply T to the present states of the cells
in N and store the result as the next
state of the automaton cell

[8]In the case of cellular automata one can even allow a fixed number of intermediate-state transitions without increasing the power of the class, since all this does is effectively expand the neighborhood of the simulator, a free parameter in the cellular automaton class.

[9]However, as a consequence of this observation and the previous one, the unrestricted simulation capability of the two is the same, both being computation universal.

set the phase indicator to 2 and reschedule
an event here in 0.5
in phase 2:
transfer the next state to the present state
set the phase indicator to 1 and
reschedule an event here in 0.5.
if you truly changed state
(next state not = present state):
schedule an event to occur in
time 0.5 at all cells in $-N$ which are passive

The SELECT function is arbitrary: the result will not depend on the order in which scheduling ties are broken.

The readout map in this simulation just consists of paying attention to the present state component of S' and ignoring the rest. Initialize the NEVS to a global state in which all cells are in phase 1 and have time-left either 0.5 (representing initially nonblank cells and their inverse neighbors) or ∞ (all other blank cells). It will then output a trajectory which reproduces the global state transitions of the cellular automaton every 1 time unit.[10]

Now for the negative part: By virtue of its finite neighborhood the cellular automaton eschews "action at a distance" (Burks, 1977, p. 569). By allowing instantaneously propagating signals, as in the plant growth example, the NEVS formalism does not impose this restriction.[11] In particular the plant model can be shown not to be simulatable by any cellular automaton in the real time sense. To see this consider a plant configuration which has extended beyond the neighborhood of the base cell. To simulate the plant model, the base cell readout must increase by 1 at the next time step (for simplicity let GROWTH_TIME be unity). But now consider the same configuration in which the growth tip is removed. Then it can be readily checked that the NEVS model will not change state (no gravity signal will be sent down to the base). But in the purported simulation the base cell readout must still increase by 1 since its neighborhood configuration has not been affected by the removal of the growth tip.

In short, a cellular automaton realization of the plant model would require an infinite neighborhood to instantaneously pass the weight information generated at the arbitrarily extending tip to the base and stem cells.

[10] This construction is of more than theoretical interest: it parallels an efficient strategy for digital simulation of cellular automata (Zeigler, 1976, Chap. 4).

[11] Holland has developed an extension of the cell space concept which removes the action-at-a-distance limitation as well (Holland, 1970). His scheme lays down wires capable of passing instantaneous signals in such a way that no cycles or infinite paths are generated. In our scheme such conditions are possible and are the responsibility of the user to prevent. The criteria for well-defined discrete event specifications are discussed in Zeigler (1976, Chap. 9).

5. CELL SPACE MODELS: THE "SOFT" ADEQUACY CRITERIA

The expressive power of a formalism throws light on but does not close the question of its adequacy. We can see, for example, that the two formalisms under discussion are computationally equivalent (a loose notion of simulation) but differ quite drastically when more direct model representation is required—the cellular automaton being incapable of exhibiting immediate signal propagation in real time. And although any cellular automaton can be directly represented in next-event form, the representation $\langle S', N', T', \text{SELECT}\rangle$ contains much additional apparatus not found in the original specification $\langle S, N, T\rangle$. Thus if one had already developed a simple cell space model, one would have to complicate its description in order to convert it to next-event form. Moreover, the result would obscure the parameters of the original specification and would be more difficult to modify because of the interdependencies of the components of the new parameters (N', for example, consists of N and $-N$: if a change is made in N a corresponding change must be made in $-N$). The same considerations hold, in even more striking form, when expressing simple NEVS models as cellular automata. To make this point we shall consider expressing the models given in Example 1–4 as cellular automata.

5.1. Expression of Motion. Consider Example 1 modeling the motion of a single body. To represent this in a cellular automaton, we require a neighborhood consisting of the center cell and its *left* adjacent cell (rather than the right neighbor as in the NEVS case). The reason is that motion consists of two parts: leaving one cell and entering the next. Since all state transitions are made independently and simultaneously by automaton cells, both cells affected by the motion must do some computation. The cell containing the object must release it (change from state 1 to 0) and the cell receiving the object must accept it—look to the *left* to see if the neighbor is a 1 and if so change itself to state 1. Thus what is a single action in the NEVS case is redundantly expressed in two actions in the automaton case.

5.2. Handling of Collisions. The redundancy of computation required is greatly compounded when collisions must be resolved as in the traffic congestion model in Example 2. Consider that to compute its next state, the automaton cell must independently decide whether any car it contains will move right and also whether any other car will move into it. To see whether its car can move, it must look not only for unoccupied cells in its own preference order but also check that an unoccupied cell is not "desired" by a car with higher priority (otherwise both cars will attempt to move into the

same cell). To see whether a car will move into a cell, the cell must check whether any car which can move into it, will prefer to do so (a car below will move up only if all of its other possible moves are blocked). The reader can convince himself that this computation will require a neighborhood of approximately 20 cells (!) and is, of course, heavily redundant—the same checks being done independently by many cells.

5.3. Choice of Time Step. The actual value of the time step does not appear explicitly in the automaton formalism. However, when discrete time models are formulated for real systems, the choice of this value is always an issue. As discussed by Barto (1975) cellular automata can be viewed as providing the formal basis for the usual representation of partial differential equation models for computer simulation. Such models are based on continuous time and space variables which are discretized for numerical integration. The time step and spatial unit must be chosen carefully to conform to the rates of propagation expressed in the original model.

Similar considerations apply when the underlying model is a discrete-event type. The most direct way to simulate an NEVS by a cellular automaton is to let the automaton cell state be a pair (s, σ) where now σ is a nonnegative integer or ∞. The transition function sees to it that σ is treated as a time-left variable by decrementing it by 1 at every time step; when 0 is reached, a new total state pair is determined according to the transition function of the NEVS. Of course, the cell must also check at each time step for cells reaching $\sigma = 0$ for changes that they would cause to it.

The question is: what real value should the integer σ represent? Too small a discretization will result in much unnecessary recomputation of global states that essentially do not change; too large a value risks missing of events which would have occurred in the original. We can conceptualize this problem by limiting consideration to recurring cycles of a propagating nature (such as the motion of Examples 1 and 2 or the signals of Example 3) or of a local nature (such as in Section 3.4). Then if the cycle durations are constant the best step size is given by their greatest common divisor. However, in even the simplest of models, these durations may not be constant due to being random variables or due to interactions which determine the continuation of the cycle (this is especially evident in Section 3.4 where the cycle must await successive prey and predator migrations in order to proceed).

In conclusion, no general procedure exists for selecting the step size underlying cellular automaton representation of NEVS models. Even if such a procedure were employed, the step size would have to be recalculated, and the time-left variables appropriately rescaled, for every change in the NEVS

parameters. Thus a cellular automaton realization of models such as those of Examples 1–3 would be either inefficient or inflexible (or both).[12]

6. CONCLUSIONS: THE WORLD VIEWS OF FORMALISMS

Let us summarize the above findings in terms of the "world view" of the formalisms. The cellular automaton cell space is one in which each cell updates its state at each time step. The updating is done simultaneously and independently by the cells. The cell space thus expects a state change to occur at every cell even though a true change may not in fact take place. To obtain the information necessary for computing its next state, a cell interrogates the cells of its neighborhood—which must be sufficiently extensive to provide this information. Since cells cannot immediately act on one another, a cell must predict what effect other cells would have had on it in the more interactive NEVS, and this may require large neighborhoods and redundant computation. Since all cells use the information current at the same date (i.e., the previous global state) to compute their next state, the result is independent of any sequential order imposed by a sequential simulator.

The discrete-event cell space on the other hand, considers cells to be capable of acting upon their neighborhood as well as themselves. Cells undertake their activity in one-at-a-time fashion so that the result is deterministic but also may be dependent on the order in which simultaneously scheduled cells are serialized by tie-breaking rules. Updating of states occurs only via events which need not be uniformly distributed in time or space. The information required to schedule its transition event is contained within the cell state (in the time left component).

One's perception of the real system being modeled (presumably reflecting its intrinsic nature) will determine whether a formalism provides an appropriate "world view" for it. My own perception is that for most situations, the discrete-event formalism provides the more natural world view. That its direct expressive power is provably the greater provides support for this ultimately subjective assessment of adequacy but does not clinch it. One could, for example, reject instantaneously propagating signals as valid explanatory mechanisms—relying on the accepted postulate of the finite velocity of light—and so maintain that the extra expressive power is not of use. (An answer is that often the speed of light is so great compared

[12] It should be noted that the same difficulties *do not* apply to digital simulation of discrete event models (as they do to simulation of differential equation models). The reason is that the simulation strategies employed by such discrete event languages as GPSS, SIMSCRIPT, and SIMULA are based on event-driven, rather than fixed-step, time advance.

to other rates of interest that it is best modeled as infinite; a cellular automaton which represented a wide range of signal speeds would suffer from the time step problem discussed above.) Further support for my preference for the discrete-event view is supplied by the fact that even when discrete-event models are expressible in the automaton formalism, the resulting descriptions seem to be unnecessarily complex, redundant and not easily modifiable. Again, it could be countered that the reverse representation of cellular automata models in the NEVS framework involves a similar inelegance—but the *degree* of inelegance is significant.

Thus I believe that the discrete-event formalism is the more adequate in terms of range of expressive power, simplicity of expression, and flexibility. Discrete-event models may thus be of interest to those seeking more adequate representations of the universe at the level of basic physics.

REFERENCES

Barto, A. G. (1975). "Cellular Automata as Models of Natural Systems," doctoral dissertation, CCS Department, University of Michigan, Ann Arbor.

Burks, A. W. (ed.) (1970). *Essays on Cellular Automata*. University of Illinois Press, Urbana, Illinois.

Burks, A. W. (1977). *Cause, Chance, and Reason*. University of Chicago Press, Chicago, Illinois.

Holland, J. H. (1970). "A Universal Computer Capable of Executing an Arbitrary Number of Subprograms Simultaneously," in *Essays on Cellular Automata*, A. W. Burks, (ed.). University of Illinois Press, Urbana, Illinois.

Herman, G. T., and Rosenberg G., (1975). *Developmental Systems and Languages*. North-Holland, Amsterdam.

Hogeweg, P. H. (1980), "Locally Synchronised Developmental Systems," *International Journal of General Systems*, **6**, 57–73.

Lindenmeyer, A. (1968). "Mathematical Models for Cellular Interactions in Development," *Journal of Theoretical Biology*, **18**, 280–312.

Sampson, J. R., and Dubreuil, M. (1979). "Design of An Interactive Simulation System for Biological Modelling," in *Methodology in Systems Modelling and Simulation*, B. P. Zeigler et al., eds. North-Holland, Amsterdam.

Smith, A. R. (1972). "Simple Computation Universal Cellular Spaces," *Journal of the Association of Computing Machinery*, **18**, 339–353.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*, A. W. Burks (ed.). University of Illinois Press, Urbana, Illinois.

Zeigler, B. P. (1976). *Theory of Modelling and Simulation*, John Wiley and Sons, New York.

Zeigler, B. P. (1977). "Persistence and Patchiness of Predator–Prey Systems Induced by Discrete Event Population Exchange Mechanisms," *Journal of Theoretical Biology*, **67**, 687–714.

Zeigler, B. P. (1977b). "Systems Theoretical Description: a vehicle reconciling diverse modelling concepts," in *Proceedings of the International Conference on Applied General Systems Research*, (G. J. Klir), ed. Von Nostrand, New York.