P. Hogeweg
Bioinformatica, Padualaan 8, 3584 CH Utrecht, the Netherlands

MIRROR beyond MIRROR, Puddles of LIFE

PRELUDE

We call our approach to Artificial Life MIRROR modeling to indicate that Artificial Life should:

- Be "As large as life and twice as natural,"
- "Reflect our reflections" on life rather than life itself,
- Form almost independent, partly overlayed, sometimes interacting "puddles" which are represented at multiple levels of detail,
- Repeat simple structures,
- "Shape, unshape and reshape," and
- Be interesting to observe.

Artificial Life, SFI Studies in the Sciences of Complexity,
Ed. C. Langton, Addison-Wesley Publishing Company, 1988

1. BIOLOGICAL MODELING, BIOINFORMATICS AND ARTIFICIAL LIFE

Artificial Life could mean many different things. In this paper, we delineate a class of models, which in our opinion could profitably be termed "Artificial Life" models. "Artificial" because they are not designed to be in close "one-to-one" correspondence with previously observed life forms; "life" because they expand our observable universe with entities "which live their life" in which we can observe patterns normally preeminently associated with life. We sketch the position of such models among other biological and bioinformatic models.

1.1 OUTPUT-ORIENTED VS. STRUCTURE-ORIENTED MODELS

In conventional biological models, one usually takes as starting point some more or less well-described phenomenon, and tries to find a representation which reproduces this phenomenon. We call such an approach "output-oriented" modeling. An alternative approach, which we call "structure-oriented modeling," is to start with an *a priori*-defined model structure and study the types of behavior it generates. In the first approach, constraints may be (and usually are) added to the structure of the model; in the latter case, ideas usually exist about the type of behavior it at least should generate. Nevertheless, the two approaches are conceptually sufficiently different to warrant the distinction.

For the purpose of bioinformatic research (i.e., for studying informatic processes in biotic systems), we have emphasized the need for structure-oriented modeling[15,16] because:

1. Bioinformatic processes are inherently "local," and bioinformatic models should be in terms of local information processing.
2. Our insight in local/global transformation is entirely insufficient to "predict" the outcome of local rules; output-oriented modeling, therefore, is impractical, and if tried, will usually lead to much too complicated models.
3. Local micro-interactions can generate a set of qualitatively different macrophenomena. These phenomena would seem to be unrelated if studied at the macro level only. Thus, in an output-oriented approach, we would probably construct models for each of the phenomena separately, and fail to recognize their interrelationships.
4. It is not true that we "know" macro-level phenomena much better then micro interactions.

Thus, in bioinformatic modeling, we define the structure of a universe, requiring that the information processing of this universe is a "reasonable" representation of the informatic processes in the system under consideration, and "observe" the ensuing macro behavior of the system; this behavior may or may not coincide with behavior previously observed in biotic systems. Observation of the model system becomes the crucial, nontrivial part of modeling.[11,15]

"Artificial Life" models can belong to either class of models. In the output-oriented approach, one has to set up some (partial) definition of "life" and generate systems which do satisfy this definition of life, possibly in a way structurally, not only materially, quite different from the way biotic life does. In the structure-oriented approach, one can set up information processing universes and study the behavior. Both approaches were used in the pioneering work on cellular automata: Von Neumann's[28] self-reproducing cellular automaton is clearly an output-oriented model (and, consequently, turned out to be far more complex than necessary,[3,24] whereas Ulam's Cellular Auxology models[36] are clearly structure-oriented models (Ulam coined the term "imaginary physics"). Both approaches still flourish in cellular automaton and related contexts. Output-oriented models, for example, are cellular automaton models which are designed to "self-reproduce,"[24] network models designed for recognizing certain types of patterns (e.g., symmetry groups[32]) and L-systems designed to resemble certain plant species.[21,26] Structure-oriented models are the studies on the the patterns generated by classes of cellular automata,[6,25,30,37] networks[22,23] or L-systems.[7,9,33]

In this paper we will focus on structure-oriented Artificial Life models: the reasons mentioned above for bioinformatic models apply here as well. Moreover, when generating artificial life it seems more interesting to generate and observe new (lifelike but yet unknown) behavior patterns than to generate "more of the same," i.e., *a priori* definable behavior patterns from new ingredients.

1.2 MODELS, PARADIGM SYSTEMS AND ARTIFICIAL LIFE

The shift we are presently experiencing from analytically treated models toward models studied by simulation is much more profound than is usually acknowledged, and has profound methodological (and ontological, if we are to make Artificial Life) consequences. An important side effect of this shift from models allowing for analytical solutions to models which are studied by simulation is that instead of studying a class of model systems (i.e., the different parameter values and different initial conditions), one studies just one, fully specified system at a time. This is, of course, why analytical solutions are supposed to be more valuable than simulation results. However, there are a number of reasons why this latter conclusion is not necessarily correct:

1. The class of models studied in analytical solutions is shaped primarily by solvability, and not by problem-oriented considerations, and its relevance therefore can be limited.

2. Such a model selection is not at all an unbiased sample: this is forcefully illustrated by the late recognition of the chaotic behavior of very many systems which are simple generalizations of earlier studied models. Moreover, generalizations are demanded by the object studied. For example, before 1970 many

discretizations of the classical logistic-growth model were proposed for species with discrete generations, all of which avoided chaotic behavior. Nevertheless, such density-dependent growth models were criticized because of the absence of severe fluctuations which were observed in such populations.

3. Some behavior patterns only occur in models which do not allow for general "shortcuts;" their fate can only be resolved by letting a fully specified system "live its life" (e.g., behavior patterns associated with universal computation).
4. Parameter values are just as important in representing some other system as are structural properties and they are often better known.[2]

The usefulness of studying fully specified systems is evident when the model is supposed to represent one particular other system(e.g., a particular lake). Such "one-to-one" models are particularly jmportant in semi-engineering context (ecosystem management, medicine, building airplanes), but are usually not the goal of scientific modeling. Fully specified model systems then function as "paradigm systems,"[11,15] i.e., they are specific examples of an almost always loosely circumscribed class of systems. Such paradigm systems should be located at "interesting" points within this class of systems. A paradigm system can derive its interest in several ways:

1. It represents closely some other ("real world") example, but is made up of different ingredients.
2. It represents some sort of "average" of the class of systems, i.e., is located close to the middle of the class of systems envisaged.
3. It is a very simple (the most simple known so far) example of the class, i.e., is located near the margin of the class of systems envisaged.
4. It lies clearly outside the class of systems but is related to it in an interesting manner: for example, it is very similar to a paradigm system which does lie within the class.
5. It exhibits unique features not known from any other representative of the class.

In fact, most scientific models also take the form of paradigm systems if parameter-independent solutions are sought: their structure and thereby the implicit parameters are paradigm-like. Simplicity (3 above) is very often the prime source of interest, for which close similarity is readily sacrificed.

Artificial Life models are clearly paradigm systems: nothing can "live" without being fully specified, and "life" is surely a loosely specified class of phenomena, if abstracted from its physicochemical basis. Artificial Life models can be "interesting" because of all the reasons mentioned above (maybe except 2, whose usefulness is doubtful in any case when none of the other points are satisfied), although points 4 and 5. (i.e., those exhibiting hitherto unknown behavior) are of special interest in this context, because these are often not recognized as interesting in "normal science." Nevertheless, studying examples outside the known class of living systems is very valuable for understanding life.

We conclude that "Artificial Life" models differ only gradually from other bioinformatic paradigm systems, and are indispensible in bioinformatic research. At the beginning of modern science Bruno, having heard Galileos theories, concluded, "Then there are many worlds like our own." Now at the beginning of flexible media to make paradigm systems, which once again revolutionizes science, we should consider that we may be able to make many (artificial) worlds, some quite unlike our own ("real") world which may yet support a form of "life," as worthy of observation as the biotic ("real") variety of life is.

2. MIRROR MODELING

2.1 BASIC STRUCTURAL CHARACTERISTICS OF MIRROR WORLDS

In our opinion there are, apart from the structural requirements set by any particular problem, a number of structural requirements which should be satisfied in any system in order to qualify as (structurally) lifelike (artificial or otherwise), although systems may be behaviorally lifelike without satisfying them. MIRROR modeling methodology is developed to make it easy to satisfy these requirements. They are discussed in this section.

2.1.1 LOCAL DESCRIPTION The importance of local description is widely recognized, and it is the main motivation for cellular automata-like formalisms. Nevertheless, these formalisms do not realize a fully local description: implicit global structure may be responsible for the observed behavior. Such implicit global factors are, in particular, a global timing regime and a fixed network, which are discussed in turn.

1. Global Timing Regime (Synchronicity). A number of authors have shown that synchronicity generates global structure and that, therefore, cellular automata cannot directly model certain processes, e.g., morphological developments[9] or Ising systems,[37] because unrealistic long-range correlations are introduced by the synchronicity. Asynchronous versions may generate interesting macro structures, but for other local rules than their synchronous counterparts.[20,27,31] However, synchronicity changes what is "simple" and therefore profoundly influences our mapping of possible worlds. A local timing regime implies asynchronicity, but does not introduce this in a global way: timing depends on local circumstances, e.g., internal kinetics[34,35] or triggering by events.[8-10] In MIRROR modeling, a local timing regime is enforced: there is no global monitor and entities either schedule their own next activity on a timing list (implementing internal kinetics), are explicitly activated by other entities, or are implicitly triggered by events through DEMONs, which are posted by themselves

or other entities. The self-structuring properties of MIRROR worlds make use of the local timing regime. In particular, when a (semi)invariant structure is recognized, which has a predictable behavior until something special happens, the initially explicitly happening but predictable events may be moved to the "internal dynamics" of a new entity. This new entity replaces (temporariiy) the set of entities involved in the invariant structure. Henceforward, only the macroevents, i.e., the "something special" which changes invariant structures, take place explicitly and are timed on the basis of the experience gathered about the invariant structure when it was still explicitly simulated. We call these macro events "interesting events,"[8,10,14] interesting because they are not predictable.

2. Global Interaction Structure (Fixed Network). In many modeling formalisms which incorporate local dynamics, the interaction structure is either local but fixed and uniform (e.g., cellular automata), or global (every entity can interact with every other entity) and only quantitatively modifiable (e.g., (neural) network models). This restricts the choice of entities severely.[15] In the case of cellular automata, the fixed local structure enforces that the information processing units represent patches of space instead of biological information processing entities.[17] We think the latter should be the unit of description in bioinformatic models and also in (some) artificial life models; we call such models "individual oriented" (see section 2.1.2). In the case of network models, it implies the preexistence of all cells and interconnections, which is only partially true for neural systems and is not only unrealistic, but leads to begging the question in evolving systems. Thus, models of evolving systems should include the evolution of interaction structure. This is also true for models which simply involve a variable number of information processing units. We call such models "variable structure models" (compare references [4,5]; see section 2.1.3.).

2.1.2 INDIVIDUAL-ORIENTED MODELS Although it may be an ultimate goal to understand "life" as a property (state) of space/time (as is attempted in cellular automata models; see 2.1.1), it may be more directly interesting to understand how complex biological information-processing structures emerge in assemblies of simpler biological information processing units. Moreover, we think that the process of individualizing (and deindividualizing) is fundamental in lifelike processes, and therefore should be fundamental in (artificial) life models. Stated more simply, if "life" is generated as a property of space/time (in a manner described by Langton[25]), it still seems to need us to recognize it as alive: if complex individuals are generated from more simple ones, they themselves may recognize themselves as living individuals in space/time.

Thus, according to the first point, the basic entities in MIRROR models are "individuals," possessing an (extendable) behavioral repertoire. These individuals live in spaces. Their behavioral repertoire includes sensing their local environment, changing the environment locally, and changing their own position in the space (i.e., changing what constitutes their local environment). The space in which the individuals live can take the form of (non-sychronized) cellular automata. The state of the cells of the cellular automata can include, apart from regular state variables of the "patches" (cells), the individuals which inhabit it. Thus, cellular automata are incorporated as a MIRROR world without individuals (as yet).

MIRROR worlds can contain separate structure-recognizing entities (DWARFs). DWARFs recognize invariant structures, e.g., recognize that a certain set of individuals interact exclusively among each other. As a still primitive form of individualizing (2 and 3 above), DWARFs can replace such invariant structure with a new individual, whose behavioral repertoire corresponds to the behavior of the invariant structure it replaces (incorporates). However, certain events ("interesting events," see 1.1.1) can disrupt the invariance, and if such events occur, the new individual is again replaced by the set of interacting individuals whose behavior it incorporates, i.e., deindividualizing occurs.[10,16]

2.1.3 VARIABLE STRUCTURE MODELS In order to be able to generate structure in a non-trivial way, (artificial) life models should be variable structured. Engaging in an interaction is part of the behavioral repertoire of an individual. However, in order to interact, it should "know" (sense, have a pointer to) the other individual. In MIRROR worlds, such knowledge is obtained by

1. Spatial Embedding. The individuals live in spaces. Through this spatial embedding they can select potential interaction partners (those that are "nearby"). Who is nearby depends, of course, on the behavior of many entities and on the topology of the space. Spatial embedding is the most important structure-generating device in MIRROR models. MIRROR worlds typically contain several spaces in which individuals live, and the above applies to each of them. Primarily there is "SPACE SPACE," i.e., the space in which the "organisms" (or cells or molecules) move about. Obviously nearness is very important for interaction among them. There are a variety of other represent national spaces, most importantly SKIN SPACES: with each individual a space may be associated which contains its representation of its world. It is inhabited by individuals which interact in ways similar to that of space space (see section 3.2). The topology of the latter spaces, of course, can be non-euclidean.
2. Acquaintance. The pointer to an individual (once obtained by spatial proximity) can be stored in its memory.
3. Ancestry Based. Such stored acquaintances may be passed on to the "offspring." Note that this is the way variable structures are realized in L-systems.
4. Pattern Based. Interactions are based on pattern similarity. This can be seen as a special case of spatial embedding, without movement through space. It is used by Farmer et al.[5] and in many message passing algorithms.

An important side effect of using variable structure models is that at all times the interaction structure can be minimized. Thus, relevant interactions can be more easily observed. Observation of the relevant interactions, both within the model and by us, is what life and modeling is about.

2.2 ENTITIES OF MIRROR WORLDS

MIRROR worlds are not only characterized by the types of interaction and control structure as discussed in the previous section, but also by the (proto)types of entities which form the world. These prototype entities are an important heuristic for shaping models.

MIRROR worlds consist minimally of a SPACE, subdivided in PATCH(es), in which DWELLERs live. DWELLERs posses a set of sensors with which they obtain information of their immediate surroundings, and can perform actions on the basis of this information. The sensors include those which react on signals created by the environment on an action of a DWELLER, e.g., an environment may generate "cracks" when a certain type of DWELLER moves, which warns other DWELLERs of its approach. These actions include changing position in space, changing local features of the space or of DWELLERS they meet, and creating new DWELLERs of the same type as itself (reproduction) or of other space inhabiting entities (like PATHs or ODOUR)s.

A MIRROR world often consists of several SPACES inhabited by DWELLERs. These SPACEs are largely independent, but are interrelated via DWELLERs and their behavior: with each DWELLER, a SPACE may be associated (its SKINSPACE) in which DWELLERs dwell. The behavior of the DWELLER can be dependent on the configuration of DWELLERs in its SKINSPACE (which it "observes") and thereby influences the configuration of DWELLERs in the SKINSPACE of other DWELLERs. An extremely simple example of the use of such multiple spaces is given in section 3 and Figure 1. We think, however, that more complex implementations of this structure may go a long way to creating the multiple, almost independent "puddles" which make biotic life fascinating.

Apart from this multiple level "real" world, MIRROR universes incorporate a "shadow world." This shadow world is also defined in terms of locally interacting individuals, although "local" may be defined in a way quite different from the "real" world. Entities of the shadow world include DEMONs, DWARFs, OBSERVERs and WIZARDs.[6,11,15,16] DEMONs activate entities on certain clues: they are extensively used by all types of entities of the MIRROR world (including the shadow world entities) to "notice" relevant events; also the above-mentioned "cracks" are generated by a specialized DEMON (CRACKER).

The other shadow world entities are more specialized: DWARFs detect invariant relations (see Hogeweg and Hesper[16,18] for an explanation of how they go about doing it). OBSERVERs find (e.g., by nonsupervised learning methods) interesting patterns in MIRROR worlds, and WIZARDs may change worlds, e.g., to maximize or minimize a certain type of invariant relation. Although presently DWARFs, OBSERVERs and WIZARDs are mainly used to generate multiple representations of the MIRROR world to the user, they are intended to be used by the entities of the "real" world (and the shadow world itself) for the same purpose.

2.3 NOTES ON THE IMPLEMENTATION OF MIRROR

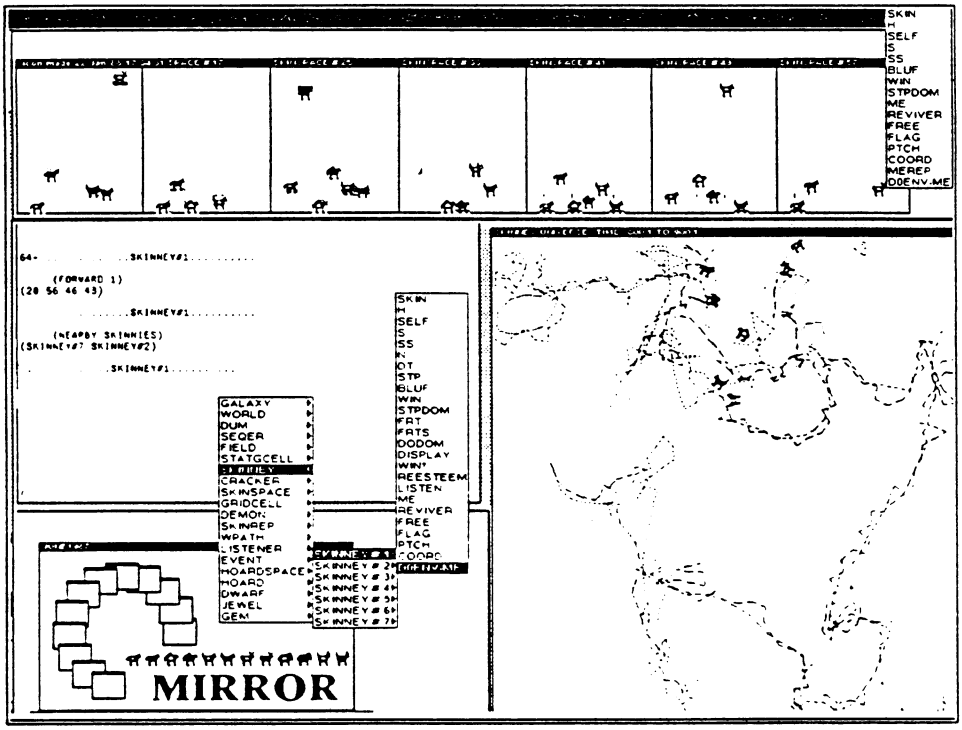
The current implementation of our ideas on MIRROR modeling is called MIRSYS, and runs on the XEROX-1186 INTERLISP-D workstation.

2.3.1 INDIVIDUALS The individuals are (parallel) invocations of INTERLISP functions. Using the Spaghetti stack facilities (and downloading whenever possible), the entities exist simultaneously in their respective stack environments. An individual is represented in the system by an atom whose value is the stack pointer. Typically a DWELLER has access directly to its own stack environment and further up to the stack environment of the space in which it lives. Thus it inherits the space-defining properties which determine its sensing and acting on its environment (e.g., the same DWELLER can live in a two- or three-dimensional environment). Moreover, an entity can examine (and initiate functions in) the stack environment of its interaction partners.

2.3.2 ENTITY-DEFINING FUNCTIONS The function defining a type of entity can take any form: for example, it can be a regular function or it can be "rule based." It can be modified by adding rules, or by using the advice facilities of INTERLISP. The behavior of the individuals, of course, is modified all the time by the modification of the information used by the function. Moreover, individuals typically "extend" themselves by generating other entities, in particular DEMONs which are attached to certain variables or procedures of other individuals and "revive" the individual when this variable is accessed or changed or the procedure is called (by certain entities and/or with certain parameters).

2.3.3 REVIVAL OF INDIVIDUALS Control is passed from one individual to the next by explicit REVIVALS, DEMON-based REVIVALS, and time-based REVIVALS. Revivals pass control to the stack pointer representing the revived individual and flag the cause of revival. In explicit revivals, this flag refers to the individual doing the revival and possibly any "message" it cares to send, in DEMON-based revivals the event on which it was activated (DEMONs themselves are transparent for the other entities, although they can add "messsages" to further specify the event and can delay the activation following the event) and in time-based REVIVALs just time. Time-based revivals take place whenever nothing else is happening: the time then proceeds to the time at which some event is scheduled.

2.3.4 IMPLEMENTATION OF DEMONS The DEMONs are placed in property lists of individuals under the name of the variable or procedure which they "haunt." The basic MIRSYS procedures check the property lists of the individuals forming their current stack environment for property names corresponding to their parameters and revive the DEMONs stored there. For example, the MIRSYS function SETENV sets a variable in some explicitly referenced individual (stack environment) and, if the value of the variable is indeed changed, revives the DEMONS in the property <variable name> of the referenced individual as well as the individuals (e.g., its SPACE) in whose stack environment it occurs. Thus, DEMONs can haunt events at several levels of generality (e.g., eating of a specific individual, eating of any individual in a SPACE, etc.). Standard INTERLISP functions can also be "haunted" without being accessed via the MIRSYS local-function call functions: if so, these functions are automatically advised to check for DEMONs in



```
64-  . .  .  .   . .SKINNEY#1. . . . . . .

    (FORWARD 1)
(20 56 46 43)

        . . . . . . .SKINNEY#1. . . . . . . . .

    (NEARBY SKINNIES)
(SKINNEY#7 SKINNEY#2)

 .        . .SKINNEY#1. . . . . . .
```

FIGURE 1 Interactive facilities of MIRSYS as functioning in the SKINNY world. MIRROR window (lower left) gives access to all entities of the MIRROR world through a hierarchical popup menu: all types of entities, all individuals of the clicked type, all variables of the individual clicked. Interactive window (middle left): By clicking DOENV-ME user control is passed to the stack environment of the clicked individual and can "do" things,e.g., move forward (returns the new coordinates and generates as side-effects the moving of the icon representing the individual, etc.) and inspect the world from the viewpoint of the clicked individual, e.g., inspect the world for nearby SKINNIES (returns the list). ICON popup menu (upper right): the same interactive facilities are available when clicking an icon representing an individual; here a SKINREP is clicked, i.e., an individual representing a SKINNY in mental space.

the property <function name> of the individuals in its own stack environment. DEMONs typically check for some conditions before passing on the revival to the individual tied to it (minimally it checks for its existence, if it does not exist anymore the DEMON kills itself).

2.3.5 GENERATION OF OUTPUT All the generation of "output" is done by specialized entities (RECORDERS, REPORTERS and OBSERVERs), which are activated by the DEMON mechanism: in this way the model entities are not contaminated with output generation and a very flexible output-generating structure is established: any collection of information-gathering entities can be let loose in the world. RECORDERs are the simplest of these entities, they gather statistics on certain types of events and display them in real time, periodically or on user request. REPORTERS and OBSERVERs are progressively more versatile: REPORTERs report on properties of the system not explicitly represented in the model formulation but which can be gathered fairly easily, whereas OBSERVERs use their own "judgment" on interesting phenomena and on when (and how) to tell about them. The graphics representation of the various spaces (see Figures 1 and 2) is also done in this way: whenever an individual moves in a space, a DEMON activates a "DISPLAY PATH" entity which moves an icon representing the individual in the display space.

2.3.6 INTERACTIVE FACILITIES MIRSYS makes full use of the INTERLISP-D graphics and interactive facilities (see Figures 1 and 2). At any time, a MIRROR universe can be interrupted and examined at any level of detail, can be changed (by changing the values of variables of individuals, by adding new individuals to the universe, or by changing the definition of (types of) entities) and subsequently can continue its operation. For example, a standard MIRROR window gives, via a hierarchical pop-up menu, access to all entities in the MIRROR world (Figure 1): all types of entities (i.e., entities being defined by a specific function), all individuals of that type, and all local variables of those individuals. The values of each of these can be displayed and changed by mouse clicking (and typing the new value). Moreover, by clicking the DOENV box, user control is passed to the stack environment of the specified individual; the user then can view the world from the viewpoint of the entity, e.g., by typing (NEARBY SKINNIES), a list of nearby SKINNIES being considered by the "possessed" individual is displayed. The user can also make the individual do things, e.g., by typing: (FORWARD 1), the possessed individual will move forward a unit length. This may cause several DEMONs to be activated; for example, a DEMON tied to the DISPLAY-PATH entity mentioned above, so that the icon representing the possessed individual will move, and the CRACKER which activates possibly other individuals who begin to fight with the individual possessed by the user (etc.). The same interactive facilities are available by directly clicking into an icon representing an entity in a display space.

3. FROM BIOINFORMATIC MODELS TO ARTIFICIAL LIFE, AN EXAMPLE; SPATIAL AND SOCIAL STRUCTURE FROM LOCAL INTERACTIONS

Our bioinformatic research on the emergence of social structures due to pairwise interactions between initially identical individuals[12,13,15] illustrates the use of several paradigm systems to map a set of behavior patterns found in a large variety of biotic systems. Some of these paradigm systems proved to represent the observed behavior of certain animals closely whereas others were never intended to do so, but were chosen so as to be representative for certain basic socioinformatic processes. Such "artificial life" models, apart from being interesting in themselves, generated the knowledge needed to attempt to construct paradigm systems for creating specific behavior patterns. In other words, this research illustrates the concerted use of models and artificial life models in bioinformatic research.

3.1 A MODEL OF BUMBLEBEES: A SOCIALLY REGULATED "CLOCK"

A pattern analysis study on the interactions of live bumblebees has shown that the workers in bumblebee colonies can be subdivided into two groups: "common workers" and "elite workers."[19] Once a worker has entered the elite, she remains in it until the end of the season when the queen is killed or kicked off the nest. After that, she will lay unfertilized (drone) eggs. We set up a paradigm system to find the requirements for the formation of the two types of workers under the assumption that all workers are identical when hatching.[12,13] To this end, we set up a MIRROR world consisting of a nest space in which BUMBLEs dwell. The behavior of the BUMBLEs was derived from:

1. The known population dynamic properties of bumblebees (i.e., development time of eggs, larvae, pupae)
2. The TODO principle, i.e., the BUMBLE's do what there is to do, not what they "intend" to do. Thus, if an adult BUMBLE meets a larval BUMBLE, it feeds it; if it meets a pupae of the right age, it starts building an eggcell; etc.
3. All social (i.e., non-maintenance) interactions are of the DODOM type. DODOM interactions involve three stages: (1) displaying/observing mutual dominance; (2) win/lose, determined on the basis of the mutual dominance, local factors and chance; (3) updating of the relative dominance based on 1 and 2 in such a way that expected outcomes reinforce the relative dominance only slightly, whereas unexpected outcomes give rise to a relatively large change in the dominance. Thus a damped positive feedback ensues.
4. A criterion of viability: is there enough food made available. This criterion is used to adjust unknown parameters.

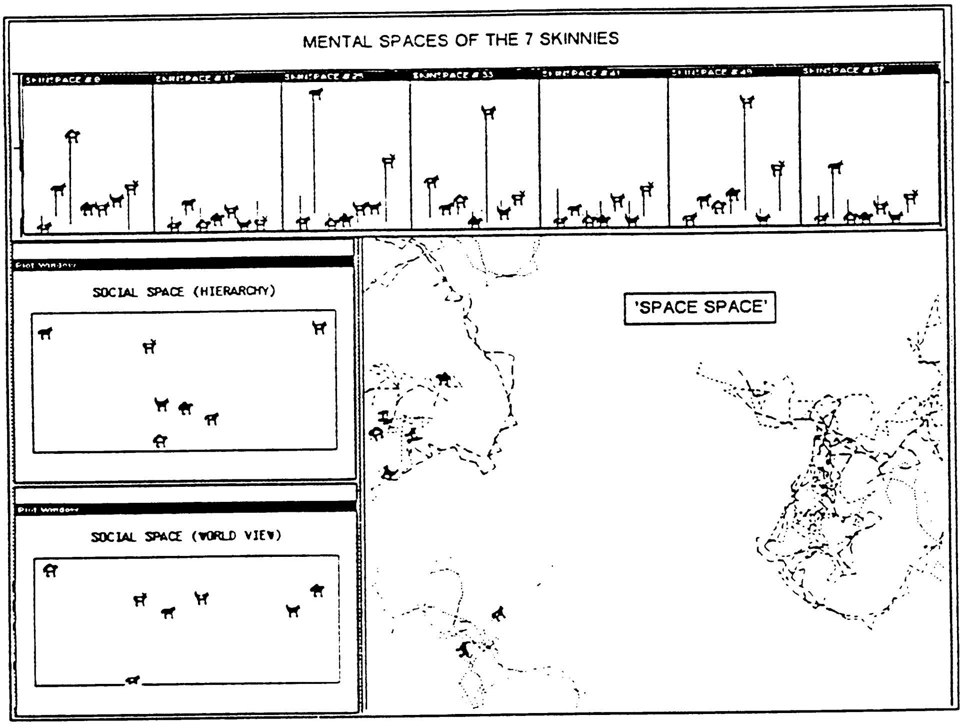


FIGURE 2 Display of spatial and social structure in a MIRROR world inhabited by SKINNIES. SPACE-SPACE(lower right): space in which the SKINNIES move about and interact. MENTAL SPACES (top row): relative dominances of the SKINNIES as perceived by each of them. SOCIAL SPACE (hierarchy) (middle left): Plot of the largest two principal components of the similarities of SKINNIES as measured by the estimate of the others. Corresponds to the hierarchy. SOCIAL SPACE (world view) (lower left): like the hierarchy, but based on their own estimate of the others: represents average spatial structure. The same icon is used in all spaces to represent the same individual.

It turns out that the resulting (simple) structure is sufficient to generate the stable class structure, provided that the nest space is subdivided into a CENTER (where the brood is and all interactions take place) and a PERIPHERY where inactive (common) workers dose part of the time. This, indeed, seems to be the case in live nests. Moreover, it turns out that the model generates a number of other phenomena observed in live bumblebee nests, most notably: (1) the queen was killed at the end of the season; (2) her departure resulted in chaos in the nest; and (3) the time at which the queen was killed was a stable, self-regulated property of the social structure which remained the same for a wide range of growth rates of the nest (a variable parameter in live nests). This time of departure of the queen is crucial for the "fitness" of the colony, as only afterwards generative offspring are reared. Interestingly, and counterintuitively, when this socially regulated clock breaks down because of stagnant growth of the nest, the small number of workers do kill the queen too early—not too late as a simple "force of numbers" argument would suggest. This is observed in live nests as well, and was used as an argument against a social influence of the timing of the "worker rebellion."[1]

Thus, the MIRROR world provided a fairly complete integrated view on the social structure of bumblebee colonies, although its initial goal had been much more limited. Moreover, it leads us to the hypothesis that a variety of self-regulating social structures which seem "adaptive" can be generated by the combination of TODO and DODOM. This implies that a change in the TODO (maintenance requirements, overpopulation, etc.) results automatically in a change in social structure. Thus, social structure is not an independent parameter which is optimized to fit certain circumstances (compare Oster and Wilson[29]), but an (emergent) property of its circumstances. Also intriguing was the important influence of spatial structure on social structure and vice versa.

3.2 AN ARTIFICIAL LIFE MODEL: THE CONCORDANCY BETWEEN SPATIAL AND SOCIAL STRUCTURE

The latter conclusions were further investigated in a MIRROR world which was not designed to represent any particular species but did incorporate the same type of interactions in a continuous spatial environment. The DWELLERs of this MIRROR world we called SKINNIES, to indicate that (1) they are the most meagre implementation of such a spatial TODO/DODOM structure and (2) the individuals know each other personally, i.e., have a representation of each other in their SKINSPACE (mental space).

Figure 2 summarizes the model as well as some model results. SKINNIES live in small groups in SPACE-SPACE (lower right window in Figure 2). If they meet an other SKINNY(or are disturbed by one in their neighborhood), they either initiate an overt DODOM interaction or fly. The choice depends on a DODOM interaction in their own SKINSPACE (which is represented in the upper left of Figure 2) in which their representation of self and interaction partner interact. When such an interaction is lost, they fly in SPACE-SPACE; otherwise, they DODOM in SPACE-SPACE, displaying their dominance estimate according to their SKINSPACE. Notwithstanding these interactions, they are inherently "social," i.e., when no other SKINNY is close enough, they will move towards the nearest (group of) SKINNIES.

This straightforward, rather simple-minded, interaction structure generates an interesting social structure: SKINNIES tend to form faithful pairs or small groups, which meet once in a while. On meeting a struggle ensues, after which it is most likely that the original pairings re-emerge intact. Moreover, it appears that for groups larger than two, the most dominant SKINNY tends to be in the middle and to interact with several submissive SKINNIES, whereas the latter ones only interact with the "boss." This interaction structure was not at all expected or even preconceived, and was only observed with the aid of DWARFs.

The lower left corner of Figure 2 gives two "social spaces" in which the group of SKINNIES live. The upper one is a representation of the hierarchy, the most dominant SKINNIES to the right. There is no absolute measure of hierarchy, but the ordering agrees well with all pairwise dominance relationships. The space is constructed from the estimate that the SKINNIES have of each other. More specifically, it optimally represents the similarity of the SKINNIES as viewed by other SKINNIES. Hierarchies are very common in groups of animals. The lower social space compares the "world view" of the SKINNIES, i.e., it optimally represents the similarity of the SKINNIES with respect to their estimate of the other SKINNIES. Interestingly, this does not correspond to the inverse analysis which represents the well-known social-concepts hierarchy; instead it gives the dominant SKINNIES a central position. In fact, it represents the average spatial structure of the group very well: the more dominant SKINNIES are located in SPACE-SPACE in the middle of larger groups and are together. Although this is not so easily detectable, it appears that such a spatial/social structure exists in many groups: the present research shows that this relationship between social and spatial structure is an intimate one (both are caused by the other and generated by the other). This is indeed reflected in our language: we are "close" to each other or "remote."

We conclude that, by studying an artificial world of SKINNIES, we can dig out relationships, which otherwise are obscured by all other things that are going on, but which nevertheless are an important force in shaping many worlds. We also conclude that one basic interaction structure, like the DODOM mechanism, can lead to a great variety of self-regulating macro structures depending on the "environment" in which it occurs. In particular, we conjecture that the DODOM/TODO mechanisms are (co)responsible for the variety of social structures observed in groups of animals. The more general implications of the occurrence of versatile, regulating, emerging structures in biotic systems are discussed in the following section.

4. SELF-STRUCTURING, ADAPTATION AND EVOLUTION

The above-described MIRROR worlds have non-trivial self-structuring properties: new, unexpected macrostructures emerge. These macrostructures are not only recognized by us, but are also recognized by the model. Recognition by the model can be explicit—OBSERVERs and DWARFs generate a representation of the macrostructure—or implicit—the emergence of the macrostructures is crucial for other properties of the model. For example, the elite group and the common worker group in the bumblebee world are recognized as such by the OBSERVERs, and their formation brings about the switch from worker offspring to generative offspring. Likewise the "friendship pairs" in the SKINNY world are recognized by DWARFs and determine the outcome of interactions and the spatial structure if pairs meet. The emerging structures represent temporary (semi)invariant relations: they remain embedded in the variable structure world and can (and do) evolve and dissolve. It is this dynamic aspect of the emerging structures which leads to "adaptive" behavior in the MIRROR worlds discussed. The behavior is adaptive in the sense that under a change of circumstances, the "shape" of the emerging structure changes so as to be "optimal" or at least "sufficient" in these circumstances to bring about some other property (or emerging structure) in the world.

This is clearly demonstrated in the bumblebee world: when a change of circumstances leads to a higher growth rate of the nest (in this case, the change is an external parameter, but a change of circumstances, of course, can be due also to a change in other emerging structures), the difference between elite workers and common workers becomes less pronounced with respect to the behavior of the individuals and with respect to membership. Conversely when the growth rate of the nest stagnates, a very pronounced small elite group is formed. By this mechanism, the switch between worker offspring and generative offspring are "optimally" timed. For a large range of "normal" growth rates, the timing is set to the end of the season: under the assumption of ergonomic constraints, in this way a maximum number of generative offspring is reared because the exponential growth phase is maintained as long as worker offspring are produced.[29] The earlier switch in the case of stagnant nest growth seems "optimal" as well: if there is no growth of worker force anymore, why wait for the end of the season?

Clearly, the fitness of a bumblebee colony depends crucially on the emerging structure. There is no concept in the model formulation akin to a switch from worker to generative offspring, and even less to a timing of such a switch. In other words, there is nothing like a gene representing this switch (or its timing) which by random mutation and selection is set to a sufficiently correct (or optimal) value. Instead, the combination of the DODOM interaction structure and the maintenance characteristics of a bumblebee nest creates the elite structure which, in combination with some other behavioral parameters (e.g., the fact that the queen tries to prevent workers from rearing new queens and laying (drone) eggs of their own), generates an adequate regulation of the generation of generative offspring. The point is not only that the switch is regulated by multiple "genes" and that each of these "genes" is involved in other (crucial) processes, but also that:

1. It is the regulation of the timing rather than the timing itself which is the interesting feature.
2. An emerging structure (which is likely to be self-regulating) can be easily exploited for a variety of "purposes" (fitness criteria) simultaneously.
3. Emergent structures determine what can be used as crucial fitness criteria (i.e., if no elite structure should arise from DODOM and maintenance, the ergonomic optimality would not be reached by a switch but along a quite different route, e.g., via physiological adaptation to, say, a multiannual colony structure).

In feedback mechanisms, cause and effect, of course, are distinguished with respect to time scale only (if at all). Nevertheless, the DODOM mechanism reverses what we would normally see as cause and effect. An individual does not primarily win because it is dominant but, rather, because it happens to win it becomes dominant (and therefore wins, etc.). Thus, a bumblebee becomes a member of the elite because she happens to interact with elite workers, therefore, happens to win once in a while from elite workers and, therefore, becomes elite and interacts with elite workers. Likewise SKINNIES happen to be near each other in space, therefore come to "know" each other well, and therefore will remain together in space.

A similar reversal is suggested by the above observations for evolutionary processes. A property is not selected because it has a high fitness value, but, because of the set of available properties, a semi-invariant structure emerges and these semi-invariant structures create new fitness dimensions, which can be optimized by adjustment of available properties. This slight change of viewpoint has a number of "nice" side effects; for example:

1. No *generatio spontanae*. Conventional evolutionary theory shuffles the interesting processes behind the curtain of unanalyzed (random) events: properties are created by mutation and only their selection is studied. The shift in viewpoint can open this curtain a little bit through studying conditions for the emergence of (semi)invariant relations and their exploitation.

2. "Arrow of complexity." Such a process may create an "arrow of complexity," instead of the "arrow of efficiency" created by conventional evolutionary concepts. This is because "self-sufficient" properties cannot keep up their self-sufficiency when (part of) their function is fulfilled by an emerging structure which is self-regulating and/or under the protection of some additional selective constraint. Indeed, the molecular record of evolution shows that conserved sites often have at least a dual function (e.g., the conserved sites in tRNA function in transcription as well as in translation).[18]

The great challenge for Artificial Life models is to obtain insight in the formation of self-regulating emergent structures and the landscape they create, rather than to try to create entities which evolve in externally supplied landscapes. If we will meet this challenge, our models may be caught in the arrow of complexity; if so, maybe the time will come when these models become more interesting to observe than life itself, and they might be observing us. If so, we will truly have: MIRROR beyond MIRROR, Puddles of LIFE.

ACKNOWLEDGMENTS

Mirror modeling was and is developed in close collaboration with Dr. B. Hesper.

REFERENCES

1. Blom, J. van der (1986), "Reproductive Dominance within Colonies of Bombus terrestris (L.)," *Behaviour* **97**, 37–49.
2. de Boer, R. J. and P. Hogeweg (1985), "Tumor Escape from Immune Elimination: Simplified Precursor Bound Cytotoxicity Models," *J. Theor Biol* **113**, 719–736.
3. Codd, E. F. (1968), *Cellular Automata* (New York: Academic Press).
4. Farmer, J.D., S. A. Kauffman, and N. Packard (1986), "Autocatalytic Replication of Polymers," *Physica* **22D**, 50–67.
5. Farmer, J. D., N. H. Packard, and A. S. Perelson (1986), "The Immune System, Adaptation & Learning," *Physica* **22D**, 187–204.
6. Hartman, H., and G. Y. Vichniac (1986), *Inhomogeneous Cellular Automata (INCA) in Disordered Systems and Biological Organisation*. Eds. Bienenstock et al. (Berlin: Springer Verlag), 53–57.
7. Hogeweg, P., and B. Hesper (1974), "A Model Study in Biomorphological Description," *Pattern Recognition* **6**, 165–179.
8. Hogeweg, P., and B. Hesper (1979), "Heterarchical Self-Structuring Simulation Systems: Concepts and Application in Biology," *Methodology in Systems Modelling and Simulation*, Eds. B. P. Zeigler et al. (Amsterdam: North Holland), 221–232.
9. Hogeweg, P. (1980), "Locally Synchronised Developmental Systems, Conceptual Advantages of the Discrete Event Formalism," *Int. J. General Systems* **6**, 57–73.
10. Hogeweg, P., and B. Hesper (1981a), "Two Predators and a Prey in a Patchy Environment: An Application of MICMAC Modeling," *J Theor Biol.* **93**, 411–432
11. Hogeweg, P., and B. Hesper (1981b), "On the Role of OBSERVERs in Large-Scale Systems," *UKSC Conference on Computer Simulation* (Harrogate: Westbury House), 420–425.
12. Hogeweg, P., and B. Hesper (1983), "The Ontogeny of the Interaction Structure in Bumble Bee Colonies: A MIRROR Model," *Behav. Ecol Sociobiol.* **12**, 271–283.
13. Hogeweg, P., and B. Hesper (1985a), "Socioinformatic Processes, a MIRROR Modelling Methodology," *J Theor Biol.* **113**, 311–330.
14. Hogeweg, P., and B. Hesper (1985b), "Interesting Events and Distributed Systems," *SCS Multiconference 1985* (La Jolla: Simulation Councils, Inc.), 81–87.
15. Hogeweg, P., and B. Hesper (1986), "Knowledge Seeking in Variable Structure Models," *Simulation in the Artificial Intelligence Era*. Eds. M. S. Elzas, T. I. Oren & B. P. Zeigler (Amsterdam: North Holland). 227–243

16. Hogeweg, P., and B. Hesper (1988a), An Adaptive, Self-Structuring Non-Goal-Oriented Modelling Methodology," *Modelling and Simulation Methodology: Knowledge System Paradigms*, Eds. M. S. Elzas, T. I. Oren and B. P. Zeigler (Amsterdam: Nor^th Holland), in press.

17. Hogeweg, P. (1988b), "Cellular Automata as a Paradigm for Ecological Modelling," *Applied Math. & Computation*, in press.

18. Hogeweg, P., and D. A. M. Konings (1985), "U1 snRNA: The Evolution of its Primary and Secondary Structure," *J. Mol. Evol.* 21, 323–333.

19. van Honk, C., and P. Hogeweg. (1981), "The Ontogeny of the Social Structure in an Captive Bombus Terrestris Colony," *Behav Ecol & Sociobiol.* 9, 111–119.

20. Ingerson, T. E., and R. L. Buvel (1984), Structure in Asynchronous Cellular Automata," *Physica* **10D**, 59–68.

21. Janssen, J. M., and A. Lindenmayer (1987), "Models for the Control of Branch Positions and Flowering Sequences of Capitulain Mycelismuralis (L.) Dumont (Compositae)," *New Phytol.* **105**, 191–220.

22. Kauffman, S. A. (1969), "Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets," *J. Theor Biol.* **22**, 437–467.

23. Kauffman, S. A. (1984), "Emergent Properties in Random Complex Automata," *Physica* **10D**, 145-156.

24. Langton, C. G. (1984), "Self-Reproduction in Cellular Automata," *Physica* **10D**, 135–144.

25. Langton, C. G. (1986), "Studying Artificial Life with Cellular Automata," *Physica* **22D**, 120–149.

26. Lindenmayer, A. (1968), "Mathematical Models for Cellular Interactions in Development I and II," *J Theor Biol.* 18, 280–299 and 300–312.

27. Natamura, K. (1981), "Synchronous to Asynchronous Transformation of Polyautomata," *J. Comput System Sci.* **23**, 22–37

28. Von Neumann, J. (1966), *Theory of Self-Reproducing Automata*, Ed. A.W. Burks (Urbana: University of Illinois Press).

29. Oster, G. F., and E. O. Wilson (1978), *Caste & Ecology in the Social Insects* (Princeton: Princeton Univ Press).

30. Packard, N. H., and S. Wolfram (1985), "Two-Dimensional Cellular Automata," *J. Statistical Physics* 38, 901–946.

31. Park, J. K., K. Steiglitz, and W. P. Thurnston (1986), "Soliton-Like Behaviour in Automata," *Physica-D* 19D, 423–432.

32. Sejnowski, T. J., P. K. Kienker, and G. E. Hinton (1986), "Learning Symmetry Groups with Hidden Units: Beyond the Perceptron," *Physica* 22D, 260–275.

33. Smith III, A. R. (1984), "Plants, Fractals and Formal Languages," *Computer Graphics* 18(3), 1–10.

34. Thomas, R. (1973), "Boolean Formalisation of Genetic Control Circuits," *J. Theor Biol.* 42, 563–585.

35. Thomas, R. (1985), "Kinetic Logic as a Formal Description of Asynchronous Automata and Biological Models," *Dynamical Systems and Cellular Automata*, Eds. J. Demongeot, J. Cole, and M. Tchuente (New York: Aacademic Press), 269–282.

36. Ulam, S. (1970), ***Essays on Cellular Automata***, Ed. A. W. Burks (Urbana: University of Illinois Press).

37. Vichniac, G. Y. (1984), "Simulating Physics with Cellular Automata," *Physica* 10D, 96–116.

38. Wolfram, S. (1984a), "Cellular Automatas as Models for Complexity," *Nature* 311(4), 419–424.