The EcoCyborg Project: A model of an artificial ecosystem.

by
Lael Parrott

Department of Agricultural Engineering
McGill University, Montreal
August, 1995.

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements of the degree of Master of Science.

ISBN 0-612-08040-4

Abstract

Lael Parrott M.Sc. (Agr. Eng.)

A model of an artificial ecosystem has been formulated for use as a tool to investigate the dynamics of autonomous biosystems. The model is part of a composite model of an EcoCyborg which consists of an ecosystem and its control system, both of which are contained inside a cylindrical space station. The objectives of this project were to design a model of the ecosystem, and to develop a method for its creation and implementation within the overall framework of the EcoCyborg Project.

The modeling approach that has been adopted for the ecosystem model is individual-based and object-oriented. This enables the inclusion of a description of the abiotic environment, as well as of the organisms that inhabit it. A total of 1000 species representing a range of taxonomic groups may be modeled. Individuals in each species are described by their behaviours and phenotypic traits.

The ecosystem model will be linked with the other components of the EcoCyborg model in a multi-process simulation under OS/2 Warp. The behaviour of the system will be studied to elucidate prelimary guidelines for the design, maintenance and control of complex systems.

Résumé

Lael Parrott M.Sc. (Génie Agr.)

Un modèle représentant un écosystème artificiel a été conçu en vue d'étudier le comportement dynamique de biosystèmes autonomes. Cet écosystème, muni d'un système d'auto-gestion, se situe à bord d'une station spatiale de forme cylindrique. Ensemble, ces deux éléments constituent le modèle complexe dénommé EcoCyborg. Le présent travail visait à la conception du modèle de l'écosystème artificiel ainsi qu'une méthode de développement respectant la structure du projet EcoCyborg.

Le développement du modèle de l'écosystème suit une approche attitrant à chaque individu un statut d'objet. Ceci permet une description complète de l'aspect physique de l'écosystème ainsi que de sa population vivante. Le modèle pourrait considérer en somme 1000 espèces vivantes représentant la gamme des groupes taxonomiques. Les individus de chaque espèce choisie seront définis en fonction de leurs comportements et de leurs traits phénotypiques.

On prévoit que le modèle de l'écosystème ainsi que les autres composantes du modèle EcoCyborg seront liés à travers des simulations sous OS/2 Warp. La dynamique du système permettra l'extraction de concepts préliminaires traitant au développement, à l'entretien et à la gestion de systèmes complexes.

Acknowledgments

Most certainly first in the list of people who deserve acknowledgment is my supervisor, Dr. Robert Kok. He has committed countless hours to consultation, ideation, and advice about the project. The EcoCyborg is his brainchild; without Dr. Kok, none of us would be here.

Us.

This work is inseparable from the contributions of my EcoCyborg colleagues: Grant Clark and Robert Molenaar, who share computer escapades and fish drama with me daily. Then, of course, there is René Lacroix, who started it all. We are all indebted to him for his continued guidance and his reassuring presence on the other side of the door. And finally, Julie Bâcle deserves warm appreciation for diligently collecting species data all summer long.

My menagerie, Matt, Brontë, and the rabbits, Lucy and Oreo, all needed kisses at just the right times. Abudywa.

Table of Contents

1. Introduction

It is noteworthy that, while the destruction and suppression of other living creatures to further the needs of humans has been a matter of rote, the *creation* of life has been shunned, gathering all the superstitions of a technological taboo. The result has been a veritable void in scientific and engineering research that has been left to the auspices of science fiction writers and Frankensteins. Our inability to create new living systems represents a fundamental lack of knowledge and understanding. Immense resources are deployed by agri-chemical companies and military departments to fabricate highly specialized methods of population decimation, and forestry companies clear-cut vast swaths of virgin forest, yet engineers are unable to design a creature as simple as a dandelion. While the prophesies of Mary Shelley provide a poignant lesson to be heeded, the most natural and immediate applications of artificial life concern, not monsters, but the understanding of presently existing systems. As the Earth's biosphere is increasingly affected by anthropocentric activities, the need for an engineering methodology specifying the design, construction, and maintenance of viable ecosystems, and biosystems in general, has become apparent. Humans as a species have evolved to be dependent on such a narrow range of environmental conditions to survive that, in the interest of both preserving a hospitable climate on Earth, and of creating new human habitats in space, we will soon need to understand how to build new biospheres that mimic the fundamental characteristics of terrestrial ecosystems.

The EcoCyborg Project has, therefore, been undertaken to investigate the engineering aspects of living system design, specifically those systems which are autonomous. Work on this project is a continuation of research conducted by Robert Kok and Rene Lacroix, in which a simulation of an autonomous agro-ecosystem in a hypothetical greenhouse environment was conducted (Kok and Lacroix 1991; Lacroix and Kok 1991; Kok and Lacroix 1993; Lacroix 1994). The particular objective of the EcoCyborg Project is to be able to characterize the system state of *any* type of ecosystem with relatively few key parameters, and, based on these, to learn how to design an ecosystem having a prespecified dynamic behaviour. This entails quantitatively specifying the initial assemblage of components (and their states) which comprises a viable system, as well as setting these in motion along a desired trajectory. The intent is to develop an engineering methodology that is applicable to all types of systems, ranging from controlled greenhouses and extensive agricultures to life-support stations in space, and even the Earth's own ecosphere.

The study of ecosystems is perceived to be a subset of a larger investigation into the design and dynamics of autonomous biosystems. A *biosystem* is any collection of interconnected living entities and their supporting environment. This definition includes, but is not limited to, carbon-based terrestrial ecosystems. It also includes purely virtual systems, as long as their constituent entities are alive and interact. *Autonomy* implies that a system is intentionally self-guiding and controlling to at least a fair degree. This presumes that a system be capable of information processing and decision making, such that it is able to recognize, and choose, from an ensemble of possible trajectories, that which will guarantee its continuity. An autonomous system may veer from a self-preserving path, but it will do so in full awareness of its own impending destruction. Thus, the system must be sensible of itself as existing within a larger environment, which generally requires some degree of consciousness. Consciousness is therefore considered to be a necessary quality of an autonomous system. Autonomy is a fuzzy state in which a system might experience various degrees of membership, i.e., an entity may be self-governing in some ways and controlled in others. Nevertheless, given the above definition, it is clear that even partial autonomy would be an advantageous state for any living system to achieve, greatly augmenting its chances of survival.

By supplying an ecosystem with a sufficiently complex control system so as to render it autonomous, it will become capable of viability at the system level, rather than existing as simply a collectivity of locally interacting components. A substantial part of the EcoCyborg Project is, therefore, geared towards the study of control systems as a maintenance strategy for natural and artificial ecosystems. On Earth, most individual creatures demonstrate some degree of autonomy, whereas larger composite systems, such as ecosystems, do not. For example, natural ecosystems, such as marshlands, exhibit a certain amount of intrinsic control in their maintenance of physical processes such as nutrient recycling, but they lack the capacity to actively direct their course. If an ecosystem were to be endowed with greater autonomy, perhaps through the addition of a more complex control system, its ability to persist under a larger variety of conditions could be enhanced. In particular, the addition of cognition and consciousness as a control strategy would increase the flexibility of the system and enable it to make intelligent decisions based upon the predicted outcome of its actions.

To study the relationships between ecosystem composition and behaviour, and the benefits of cognitive control as a strategy for improved system viability, a composite model is being written. This model is composed of a number of sub-models which are being developed concurrently by different members of the EcoCyborg research group. One of these models is of an enclosed, artificial ecosystem. Models of extrinsic Pavlovian and cognitive controllers are also being written (by Robert Molenaar, Ph.D. candidate). The ultimate aim is to link these models in an interactive simulation in order to investigate the overall behaviour of a cyborged ecosystem.

This master's thesis is the first comprehensive report to come out of the EcoCyborg Project. In it, the layout and definition of the entire project, as well as the design of an overall modeling approach for the ecosystem sub-model (hereafter referred to simply as the ecosystem "model"), are detailed. The objective of this particular part of the EcoCyborg Project was to formulate a model which could be expected to: (1) display complex behaviour which is similar to that of a natural ecosystem; (2) interface during simulation with extrinsic controlling modules; and, (3) provide the degree of flexibility required to investigate the outcome of various control strategies, system configurations, and initial conditions. To compose such a model, a definition and specification procedure using a specialized interface has been developed. Parts of this interface have been written, and the model's creation, based on the approach presented herein, is presently ongoing.

2. Literature Review

By studying models of artificial ecologies, researchers in the overlapping fields of ecology, biospherics and artificial life and complex systems theory, have begun to grapple with questions that address the nature and composition of living systems. There have been various approaches to the development of these models, ranging from the construction of physical prototypes of space-station biospheres, to the creation of "software ecologies," based on the interactions of purely virtual entities. Although a wide variety of modeling techniques have been developed, there is no literature to indicate that anyone has ever coupled the fields of artificial intelligence and biosystems engineering to create autonomous ecosystems.

2.1. The advent of closed-system ecology and biosystems engineering: Physical models

The foundation of biosystems engineering in North America can be attributed to a group of people who studied closed-system ecology during the 1960s and 70s. During this time, several separate experiments were undertaken to ascertain the feasibility of mass-closed systems on a scale smaller than Earth. Folsome, at the University of Hawaii (Folsome and Hanson 1986), established marine microbial systems in sealed flasks, some of which persisted for over 15 years. Similar fresh water ecosystems that existed under mass-closed conditions for four years were developed by Maguire (Hanson 1982) at the University of Texas, and Taub at the University of Washington. Hanson (1982) established materially closed brackish water systems of one and two litres that contained tropical shrimp having lengths up to 14 mm. Unbeknownst to the Americans, Kirensky in the U.S.S.R. was also initiating closed ecosystem studies, using algae to provide the necessary gas and water to support a human (Gitelson et al. 1989). In 1982, these and other projects were presented at the First Invitational Workshop on Closed-System Ecology, at which a number of key research issues were established. Many of these addressed some of the fundamental aspects of ecosystem design, all of which are still relevant today. They are (Hanson 1982):

> -What are the parameters which constitute the best indicators of a closed system state?
>
> - If one materially closes any heretofore untested but logical assemblage of autotrophic and heterotrophic organisms, what is the probability that it will maintain itself?
>
> - What are the minimum gas, liquid, and solid, volumes or masses, per unit biomass that will permit closed ecosystems to persist? These values may be expected to vary significantly as functions of species lists, physical characteristics, chemical composition of the inorganic phases, and energetic environments of the systems.
>
> - What are the minimum sets of species required for viability of closed ecosystems; what are the ecological characteristics of the species which make up these sets; and are there critical patterns of interaction among such species sets?
>
> - What are the earliest, most sensitive, and accurate, indicators of impending instability in closed ecosystems?

These questions are central to the design and maintenance of ecosystems which are both materially open and closed, and have been a continuing focus of research in biosystems engineering and theoretical ecology.

The persistence of the various flask-based ecologies led to the construction of larger models of mass-closed ecosystems, and the potential applications of these to human life support in space became apparent. In 1986, the United States' Capital Commission on Space proclaimed that "to explore and settle the inner Solar System, we must develop biospheres of a smaller size, and learn how to build and maintain them" (Nelson et al. 1992). This directive led to an escalation of research related to the development of mass-closed biological systems. At the Kennedy Space Center in Florida, NASA engineers constructed a Biomass Production Chamber in which food crops were grown in a heated, ventilated chamber, and air and water recycling systems were tested (Fortson et al. 1993; Fortson et al. 1992; Prince and Knott 1989). Additional studies focused on the development of bioregenerative waste management systems, contaminant control, and more efficient biomass production (Averner 1989; MacElroy et al. 1989). Work on the project is presently ongoing. The overall research objective of NASA is to design and construct a fully regenerative controlled ecological life support system (CELSS) that will support humans during extended space flights. CELSS designs will, therefore, always be limited by the small size of the vessels in which they will be contained. Thus, while the biosphere inside a CELSS may include an artificial ecological community with plants, animals and humans, it will not necessarily represent the Earth's biosphere except by functional analogy.

Other biosphere projects have been undertaken to investigate the potential for human life support, inside a space station or lunar base, for example. In Russia, the BIOS-3 project involved the construction of a sealed complex that contained human living quarters connected to three phytotrons used to grow crop plants (Gitelson et al. 1989). The entire compound was 315 m^3. Over the years 1972-1984, two and three person crews lived in the BIOS-3 complex for periods of up to six months, and were fully supported by food, oxygen and water produced in the phytotrons. A considerably more extensive artificial biosphere (Biosphere II) has been constructed in Arizona by Space Biospheres Ventures (Nelson et al. 1992). Biosphere II is a materially closed, informationally and energetically open system designed to support a human crew of eight. It has been developed with the broad aim of human life support, but within an environment that more closely resembles the Earth's ecosystem than other projects. The interior volume of Biosphere II is 204,045 m^3 which is subdivided into seven zones: intensive agriculture, human habitat, rain forest, savannah, ocean, desert and "lungs" (expandable compartments to regulate pressure). Each of the ecological zones contains up to 200 plant and animal species as well as insects, micro flora and fauna, etc. Material recycling within the system is maintained by naturally occurring mass-exchange processes within the biomes. During the early 1990's, eight people were enclosed within Biosphere II for two years. Allegations of scientific invalidation during the experiment have led to the abandonment of further work for the time being.

All of the above biosphere projects have been largely focused on the engineering aspects of physical (wetware and hardware) construction. Since existing engineering theory lacks a methodology for the design of systems composed of living entities, most of the work in this field has been purely exploratory. The development of such large-scale physical models is expensive and time consuming. For this reason, ecologists and others have been trying to establish a theoretical framework with which to describe and categorize ecosystems and their behaviour. Such knowledge would not only help to explain the functioning of the Earth's ecosystems, but would also serve as a basis for the development of biosystems engineering principles. To do so, they have applied a number of modeling techniques, the first of which were analytical in nature.

2.2. The top-down approach: Analytical models

The first analytical models captured the overall behaviour of an ecosystem by describing its macroscopic elements with a few governing equations. This is known as the top-down approach. It has been used to study key concepts in ecology theory, such as the relationships among ecosystem function, species diversity and interconnectance, by modeling the interactions between populations.

Traditional multi-species population dynamics are described by a set of deterministic equations that describe a vector field in *m* dimensional space:

$$\frac{dN_i(t)}{dt} = F_i\left(N_1(t), N_2(t), N_3(t), \ldots, N_m(t)\right) \tag{1}$$

where $N_i(t)$ is a population size (May 1973). The classical Lotka-Volterra model of a one-predator, one-prey system with continuous growth consists of the following two equations:

$$\frac{dH(t)}{dt} = H(t)\left[a - \alpha P(t)\right] \tag{2a}$$

$$\frac{dP(t)}{dt} = P(t)\left[-b + \beta(t)\right]. \tag{2b}$$

Here, *H(t)* and *P(t)* are the prey and predator populations at time t, respectively. The parameters *a* and *b* relate to the birth and death rates of the prey and the predator, and α, β to the interaction between the species. These equations, and variations thereof, have been widely applied to a large class of models (May 1973 and references therein). All of these models are based on the assumption that communities fluctuate about a stable population density. For any of these, behaviour of the system in the neighbourhood of steady state is characterized by the *community matrix*, which is derived through a linearization of the model equations. Each element, a_{ij}, in the community matrix describes the net effect of species *i* on species *j* at steady state. Stability requires that all eigenvalues of the community matrix have negative real parts. If one or more eigenvalues have positive real parts, then the system is assumed to have no stable point.

Thus, most analytical models of this type are based upon the notion of stability, which is an elusive concept when applied to ecological systems. The most generally accepted definition of stability relates to the propensity of all system variables to return exactly to their initial steady state values following perturbation (Pimm 1984). A system is locally stable if this condition applies only to small perturbations. Austin and Cook (1974) observed that in biological systems, such a severe restriction may not be applicable. They found that a species which is perturbed will often return to a limit cycle or point which is close, but not identical, to the original, and that this configuration will be equally persistent as the first. Such behaviour is termed buffered stability. Both of these definitions of stability refer to a system's response to sudden perturbations. Alternatively, if a continuous and gradual change of system parameters leads to a new state which is topologically isomorphic to the original, unperturbed state, then a system is said to be structurally stable (May 1973).

Community matrix analysis has been used to assess stability of all types of ecosystem models. Tregonning and Roberts (1979), for example, used a simple form of the generalized (multi-species) Lotka-Volterra equations to study the chances that a collection of species would be persistent. Starting with random collections of 50 species, the community matrix eigenvalues were computed, and the species with the most negative eigenvalue would be removed, until a stable configuration evolved. They found that, in general, the system would stabilize at half the original number of species. Nisbet and Gurney (1976) performed a similar type of neighbourhood stability analysis to determine the effect of material cycling in a closed system. They concluded that the inclusion of a decomposer component (which is often ignored in models) to the trophic chain had a stabilizing effect on the system.

Intuitive assumptions and early theoretical work (Pimm 1984) suggested that an increase in species diversity would lead to increased global stability, due to the buffering effect of multiple food chains within the trophic levels. Later analytical models, however, suggested exactly the opposite trend (May 1973; Pimm 1984). Many authors have cautioned against the acceptance of this result (Green 1993; May 1973) with the argument that complex systems in nature are not formed randomly, and have arisen through a long process of evolution and natural selection. Thus, since non-viable systems collapse immediately, only the viable configurations have persisted on Earth. May (1973, 4) therefore recommends that "theoretical work should not try to prove any theorem that 'complexity implies stability', but instead should focus on elucidating the very special sorts of complexity, the singular strategies, which may promote such mathematically atypical stability".

A number of hypotheses to explain the necessary characteristics, or "singular strategies" of a complex, yet stable, system have been posed. Some ecologists argue that not all species have an equal importance in the food web of a particular ecosystem. Bond (1994, 238) defines a keystone species as one whose "activity and abundance determine the integrity of the community and its unaltered persistence through time, that is, stability". Thus the removal of a keystone species would cause a cascading effect through a food web. Lawton and Brown (1994) present an alternative viewpoint with the redundant species hypothesis, in which species richness is irrelevant; what matters is that the total biomass at each trophic level is maintained. This hypothesis has been supported by a number of studies in which a common structure in food webs from different habitats has been observed, regardless of the number of species (Lawton and Brown 1994; Pimm et al. 1991). Common features include: proportions of top, intermediate and basal species, ratios of predators to prey, proportions of trophic linkages between different kinds of species, the lengths of food chains (i.e., number of trophic levels) and linkage density. Further studies have focused particularly on linkage density, or connectance (defined as the ratio of actual links to topologically possible links in the food web). Several authors (Green 1993; Green 1994; May 1973; Pimm et al. 1991) have reported a critical level of connectivity beyond which a system becomes suddenly unstable. Green (1993) found the behaviour of systems with near critical connectivity to be essentially chaotic, and suggested that this realm may be the source of variety in biological communities. It seems that, in general, the more species there are in a community, the less connected it should be in order to be stable and, conversely, the more connected a community, the fewer species it should have to be stable (Pimm 1984; Pimm et al. 1991). It should be noted, however, that this result may arise partially from the rather unfortunate definition of connectance in which the value of the ratio's denominator (the number of possible trophic links) is bound to increase at a faster rate than the numerator as the number of species is increased.

An alternative to the above multi-species modeling approach was presented by Austin and Cook (1974) who wrote a population model that incorporated a number of ecological concepts that had not previously been considered. The model was based on difference equations that included a large decomposer food chain and spatial competition between plants, as well as the usual predation and growth components. In addition, predator species had the ability to modify their behaviour in response to varied availability of prey. Instead of predicting the system's behaviour with mathematical analysis, the model was used in a discrete-time simulation. At each time step, the system's present state was compared to its previous state to determine whether it had reached a stable point or limit cycle. Results of simulations with up to 45 species showed the presence of multiple stable points. With increasing species numbers, there was a corresponding increase in the number of attractive stable points; these, however, became less aggregated in state space.

The first evidence for the presence of chaos in ecological systems was presented in a very influential paper by Hassell et al. (1976). They wrote a simple single-species, discrete-generation model capable of producing chaotic data as the value of one parameter was increased. Sets of parameter values for the model were then estimated from real population data for twenty-eight different species. With two exceptions, all of the species exhibited stable, steady state behaviour. The first exception was the Colorado Beetle, which displayed persistent cycles, and the second was Nicholson's blowflies, which exhibited chaotic dynamics. Arguments ensued about whether these were valid findings, and whether chaos was a significant property of ecosystem dynamics. Since then, the presence of chaos has been detected in a number of other biological data sets and a series of new techniques has been developed to estimate properties of the attractors underlying dynamical systems (Godfray and Blythe 1990). The two most common techniques relate to attractor reconstruction and the estimation of correlation dimensions, and many ecological data sets have been successfully subjected to this type of analysis (e.g., Godfray and Blythe 1990; Mees 1990).

The recognition in ecology that not all viable systems necessarily exhibit point or cyclical dynamic states has led to the growth of new approaches in ecological modeling that are not biased toward strict assumptions of stability. In addition, as discussed below, there has been a broad methodological shift in the biological sciences in which biosystems are perceived to be "complex adaptive systems exhibiting behavior that emerges from the interaction of a large number of elements from the levels below" (Taylor and Jefferson 1994, 1). The result of these ideological changes has been the development of new models that emphasize the importance of tiny components in determining ecosystem behaviour.

2.3. The bottom-up approach: Individual-oriented models

In most analytical models, species interaction is modeled at the macroscopic level, with the population acting as an object with a given density and set of predictable behaviours. A population, however, is not a true object; it is merely an artificial construct that has proven to be a mathematically convenient way of lumping the collective behaviour of a large number of individuals (Judson 1994; Kawata and Toquenaga 1994). Criticism of population-level models is based on the argument that "[e]ven when [they] adequately describe population trends and dynamics, they only summarize the integrated actions of individuals; they do not simulate the basic processes responsible for population change" (Plant and Stone 1991, 261). This new emphasis has led to the birth of individual-oriented models, which describe the activities and traits of every single organism in a population. No interactions beyond the scope of the organisms are modeled; these arise as the sum of individual behaviours. The drawback is, of course, that there are a lot of individuals in an ecosystem. New advances in computing technology, however, have nearly eliminated any restrictions of this type. Similarly, large-scale individual models have been applied with considerable success by physicists who describe fluid turbulence in terms of single molecules, and by astronomers who take single stars as the basic unit for modeling the collisions of galaxies (Hogeweg and Hesper 1990). It has been suggested that the impact of individual modeling in ecology may be even greater than that in physics and astronomy since organisms are more diverse and history-dependent than molecules or stars (Hogeweg and Hesper 1990).

An individual-oriented model is not created using the traditional systems of equations approach. It would take tens or hundreds of lines of differential equations to express a simple organism's behaviour as a function of all its influencing variables. In addition, such equations poorly represent non-linear effects such as thresholding or if-then-else conditionals, which arise very frequently when describing animal behaviour (Taylor and Jefferson 1994). Thus, an alternative modeling paradigm has arisen which is based on representing behaviour using many simple calculations. There are numerous approaches to this. For example, a population may be represented as a set of coexisting computer programs, one for each individual; a cellular automaton may be used to model an ecosystem as a set interacting spatial cells; or an object-oriented model may be created such that each individual is a self-contained object that passes messages to other objects.

It is not true that all population models disregard the properties of individuals. As Maley and Caswell (1993) point out, the difference is more subtle. They divide individual-oriented models into two classes, the first of which represents an intermediate approach between the traditional population model and the truly individual-based model. In what they term *distribution models,* the state of a population is described by a set of abundances, or distributions, of individuals into categories such as age or size. This is a common way to overcome the shortcomings inherent in modeling a population as a consistent entity. The population dynamics are then produced by updating the distribution functions repeatedly according to a set of rules, usually defined with differential equations. The underlying assumption of these models is that all individuals experience the same environment. They are, therefore, applicable only to large populations in which individuals are relatively similar and the environment is temporally stable. In a true individual model (in Maley and Caswell's terminology, a *configuration model*), each individual is considered separately. This type of model allows for the more realistic case of a heterogeneous environment which includes varying external influences as well as interactions between spatially proximal individuals. The advantage of traditional population-based approaches, and the intermediate class of distribution models, is that they can often be solved analytically or by applying numerical techniques in order to check the results of a simulation. Conversely, an individual-oriented model *requires* simulation to generate results, and its predictions are nearly impossible to verify analytically.

One of the basic tenets of individual-based modeling is that the combined interactions of simple entities can give rise to interesting macroscopic behaviour. In this manner, the global dynamics of a system can be seen to be an inherent property of its constituent parts. This approach is often referred to as "bottom-up" modeling, since behaviour arises from the bottom rather than being imposed by overriding equations. When such behaviour is unexpected, that is, it is not immediately foreseeable upon inspection of the specification of the system, it is termed emergent. Assad and Packard (1992, 145) have defined emergence on a relative scale from weakly emergent ("behavior is deducible in hindsight from the specification *after* observing the behavior") to strongly emergent ("behavior is deducible in theory, but its elucidation is prohibitively difficult") and maximally emergent ("behaviour is impossible to deduce from the specification"). Strong, or maximal emergence seems to be a typical characteristic of most living systems. Thus any sufficiently complex ecological model must be implemented through computer simulation in order to predict its behaviour. This requirement parallels the notion of "universal computation" in cellular automata theory (Hogeweg 1988), in which a resulting configuration cannot be computed any faster than by the cellular automaton itself.

The majority of individual-based ecosystem models has arisen out of recent artificial life research, whose aim is the synthesis of living forms within computers and other artificial media. Like ecologists, those who work with artificial life are trying to examine and categorize the fundamental characteristics of living systems. In fact, many of the concepts related to individual modeling, such as the emergence of global patterns, originated from artificial life theory, led by scientists at the Sante Fe Institute. The field of artificial life is often divided into two realms: weak and strong (Kawata and Toquenaga 1994). Weak artificial life programs contain models of organisms that mimic real organisms, whereas strong artificial life programs are an attempt to create completely new life forms that may bear no resemblance to terrestrial organisms. The degree to which a virtual artificial ecology represents physical reality is strongly related to the emphasis of its creator's objectives.

EVOLVE III (O'Callaghan and Conrad 1992; Rizki and Conrad 1985), although its first version was written before the term was coined, may be classed as a weak artificial life program. It has been developed to study ecosystem evolution by modeling organisms that each have their own phenotype and genotype. Each organism has 15 phenotypic traits (e.g., temperature optimum, rate of energy intake, age) that are coded by a collection of up to 40 genes which are modeled as sequences of 200 nucleic acid bases. There are six possible types of activities that an organism can perform, such as resource collection, reproduction, migration and death. The organisms interact on a two-dimensional grid of locations, each of which has a unique set of environmental conditions such as light intensity and temperature. Many variants of the EVOLVE III model containing up to 1000 organisms (this requires 2.5 MB of RAM) have been used in simulation. Experimental results show a number of evolved behaviours, such as symbiotic feeding strategies, arms race development and adaptability to changing environmental conditions.

Incorporating a stronger biological orientation, the JABOWA class of models is another example of weak artificial life that has been developed to study forest succession dynamics (Botkin et al. 1972; Huston 1992). In these models, individual trees surrounding the area dominated by one fully grown tree compete vertically for light and growth space. Trees are defined by age and size-specific traits as well as environmental tolerances. The abiotic environment is described with an elevation, soil depth, moisture holding capacity and rockiness, temperature and precipitation rate. JABOWA-like models have been widely used to study succession in very different forests, and show good agreement with physical data.

TIERRA, developed by Ray (1994; 1992), is a highly popularized strong approach to artificial life which merits mention. A TIERRA simulation constructs a virtual computer inside the RAM space of the physical computer on which it is resident, and then creates a single ancestor program. The program's instructions provide it with the capacity to find free memory space in the RAM "soup" of the virtual computer and to produce a copy of itself. During the reproduction stage, there is the possibility for evolution through mutation. The "soup" is quickly filled with the ancestor's offspring, all of which must compete for space (memory) and energy resources (CPU time) to reproduce. Thus, a highly competitive, Darwinian-style battle emerges in which only the fittest survive. The results are best described by Ray (1992, 2):

> From a single ancestral "creature" there have evolved tens of thousands of self-replicating genotypes of hundreds of genome size classes. Parasites evolved, then creatures that were immune to parasites, and then parasites that could circumvent the immunity. Hyper-parasites evolved which subvert parasites to their own reproduction and drive them to extinction. The resulting genetically uniform communities evolve sociality in the sense of creatures that can only reproduce in cooperative aggregations, and these aggregations are then invaded by cheating hyper-hyper-parasites.
>
> Diverse ecological communities have emerged. These digital communities have been used to experimentally study ecological and evolutionary processes: e.g., competitive exclusion and coexistence, symbiosis, host/parasite density dependent population regulation, the effect of parasites in enhancing community diversity, evolutionary arms races, punctuated equilibrium, and the role of chance and historical factors in evolution. It is possible to extract information on any aspect of the system without disturbing

it, from phylogeny or community structure through time to the "genetic makeup" and "metabolic processes" of individuals. Digital life demonstrates the power of the computational approach to science as a complement to the traditional approaches of experiment, and theory based on analysis through calculus and differential equations.

TIERRA is a good example of how the processes of a living system can be successfully reproduced within a simulated environment. Its life-like properties have sparked considerable debate about the distinction between the simulation of life and the *realization* of life (Kawata and Toquenaga 1994).

If spatial interaction is the key factor of importance when modeling an ecological system, then the use of a cellular automata formalism may be appropriate. In this case, the basic units for modeling are grid cells on a terrain instead of organisms. The likeness to an individual-based model is, however, quite strong (Green 1993). A cellular automaton is a large tessellation of finite-state cells each of whose state is updated in discrete time steps according to a deterministic rule that depends on the states of neighbouring cells. Generally the number of states that a cell can have is small (2-4) and the rules for determining them are straightforward. Nonetheless, interesting results emerge from seemingly simple configurations. Wolfram (1984) qualitatively defined four classes of characteristic limiting forms that a cellular automaton may attain and Langton (1990) later embellished on these and provided a quantitative method by which to distinguish them. The four classes are as follows: (1) spatially homogeneous state [point attractor]; (2) sequence of simple stable or periodic structures [periodic attractor]; (3) chaotic aperiodic behaviour [strange attractor]; (4) complicated localized structures, some propagating. The fourth class represents the state which Langton has coined "the edge of chaos", and is the realm in which the dynamics of living systems is believed to fall.

The cellular automata formalism has been used to model a number of spatial phenomena in ecology, particularly vegetation succession. A model by Hogeweg (1988) randomly assigns a species (from 40 possibilities) to all of the cells on a grid and then calculates each cell's next state using a probabilistic function based on the frequency of species in the neighbouring cells. A small probability is reserved for the influx of a species that is not located nearby. Successive iterations showed different degrees of variability in the vegetation map depending upon the scale at which it was studied. Green (Green 1993; Hogeweg 1988) studied the effects of "space-filling" processes (such as seed dispersal or animal migration) in contrast to "space-clearing" processes (fire, storms and other large disturbances). Through the use of a cellular automata model of fire-based succession in Australian forests, he concluded that in the absence of space-clearing effects, clumping behaviour promoted the existence of species that would otherwise be eliminated by superior competitors. Consequences of this include the formation of ecological zones in a forest which maintain high diversity and which are resistant to change. He showed that the introduction of clearing or fire to such a forest community caused sudden catastrophic changes.

If both spatial interactions between species and individual development are important, then an individual-oriented model is the more appropriate formalism. Of the multifarious ways that such a model can be implemented, some of which were presented above, there is one method that is natively suited to the application: the object-oriented approach.

2.4. Object-oriented applications

A good description of the object-oriented modeling technique is given by Rumbaugh et al. (1991). Only a brief overview of the terminology will be presented here.

The building blocks of object-oriented programming (OOP) ar taxonomic groups of related objects which are referred to as classes. Individual occurrences of a class are called instances. A key concept of OOP is encapsulation: both the data and the code associated with an object are incorporated in its description. An object's data structure is defined with a collection of instance variables, or attributes. Each class of objects has a set of specific behaviours, or self-contained subroutines (code), which are called methods. Objects communicate with one another by sending messages which are interpreted by their methods. Since methods are encapsulated in objects, different objects can respond uniquely to the same message. This is known as polymorphism. Classes must be organized into a hierarchy of classes and subclasses which inherit the attributes and methods of their superclasses. The description of a subclass may add specialised methods and attributes, and may also constrain the values that an inherited attribute can assume.

An object-oriented model may, therefore, be based upon an abstract framework whose structure is analogous to that of a real system. Since object-oriented models consist of objects with complex internal dynamics that interact with one another, they are ideally suited to the individual-based approach in ecological modeling. The natural implementation of an individual is as an object with a set of unique attribute values and a collection of behaviours that it shares with other members of its class. Aspects common to a type of organism, such as a mammal, can be encoded in a general class of which more particular types (e.g. species) are descendants. The environment can be represented as a number of spatial patches, similar to a cellular automaton, but with continuous states. For example, each patch might be implemented as an object whose data structure includes its abiotic state as well as a list of the individuals currently in that location. A good example of an ecological model based on the above-described OOP structure is given by Maley and Caswell (1993).

Despite the conceptual strengths of the object-oriented paradigm, few ecosystem models have been developed using OOP techniques. Silvert (1993) has given a number of reasons why this technology has not been readily adopted by ecologists. Among the disadvantages he lists are the inefficiency of currently available compilers of object-oriented languages, which make the development of large models unfeasible, and a lack of support for the use of floating-point numbers in early language versions. In addition, due to the nature of OOP languages, object-oriented models are usually based on discrete event simulation or queuing theory, whereas most ecological modelers prefer to create time-driven simulations. Plant and Stone (1991) created a predator-prey model using a combination of object-oriented programming and rule-based reasoning. They, too, concluded that although the object model was conceptually closer to the natural system than other models, it required massive amounts of memory and ran very slowly. Their model executed one simulated day in 2 to 3 minutes of real time with only 500-1000 individuals (computer type unspecified). Newer language modifications may improve speed. Drogoul et al. (1992) used the ActTalk extension of the Smalltalk language to create a model based on an ant colony. The ants' interactions were used to illustrate self-organization as an example of functional emergence.

RAM (Taylor et al. 1988) is an artificial life modeling system that creates individuals ('progRAMimals') using an object-oriented structure, but outside of an OOP language environment. Its creators have warned that they "do not adhere to any particular discipline, and [they] violate object-oriented philosophy in at least one major respect" (279), however, the flexibility and richness of their modeling system can be attributed to the object-oriented nature of its specification. The abiotic environment of the RAM world is a two-dimensional grid of cells that each contain a number of variables and procedures. Likewise, each animal resides in a cell and has an associated collection of animal variables and procedures. Animals are instances of species classes and each individual member of a species shares the same behavioural code. RAM has been used to simulate fairly simple situations, such as the formation of leks by sage grouse, the control of mosquito populations by insecticides and the traditional predator-prey scenario. The authors have stressed that it has not yet been employed to its full potential.

2.5. Section Summary

In summary, the major approaches to ecosystem modeling can be categorised into three basic groups: physical prototypes, analytical population models, and individual-based models. The construction of physical models has helped in the formulation of key biosystems engineering objectives as well as provided insight into the operation of materially closed ecosystems. Early analytical studies based on population models also attempted to address some of the basic issues related to ecosystem functioning. Increased recognition that traditional analytical models failed to capture the basic processes responsible for population change led to the development of individual-based models. New modeling and simulation approaches based on these have enabled the inclusion of a greater degree of detail in ecological models, particularly with respect to environmental influences and intra-species interactions that occur on a spatial scale. One of these techniques is object-oriented programming. In the development of the ecosystem model for the EcoCyborg Project, the latest tools of ecological modelers are being used to create a virtual rendition of a space biosphere whose design is partially based upon those that have been described above.

3. Context Modeled and Simulated

Described in this section is the context which has been envisioned, for which the EcoCyborg model will serve as a case study. The context is a purely imaginary system that does not exist in physical reality but which is based, however, on foreseeable technology. There is no intention to use the model as an engineering prototype for the construction of a physical system; the modeled context serves solely as a convenient paradigm for the investigation of the design of autonomous biosystems in general. Thus, the system being modeled and simulated is an entirely hypothetical scenario which has been devised to suit the objectives of the research project.

The context consists of an ecosystem and a control system enclosed within an orbiting space station, and is, in this respect, similar to the biospheres designed for human life-support. The space station setting has been chosen for the conveniences that it allows with respect to assumptions regarding energy flux and the ease with which a system boundary can be delineated. The entire system is referred to as an EcoCyborg and is shown schematically in Figure 1. As indicated in the figure, the main parts of the EcoCyborg are the *enclosure,* the *control system,* and the *ecosystem,* which is subdivided for conceptual simplicity into an *encompassment* and a *biological community*. Outside of the EcoCyborg lies its *outer surroundings* which contains a radiant energy source.

3.1. Outer surroundings

The outer surroundings comprises everything that lies outside of the enclosure, which, in the present case, is outer space. It is assumed that a radiant energy source is always available from which light can be captured by the EcoCyborg. The presence of any other bodies is not considered. For example, the control system does not actively import information regarding anticipated collisions with meteorites, dust storms, exploding super novae or the likes, and materials from nearby asteroids or planets cannot be collected. Thus, energy exchange is the only interaction that occurs between the system and its outer surroundings.

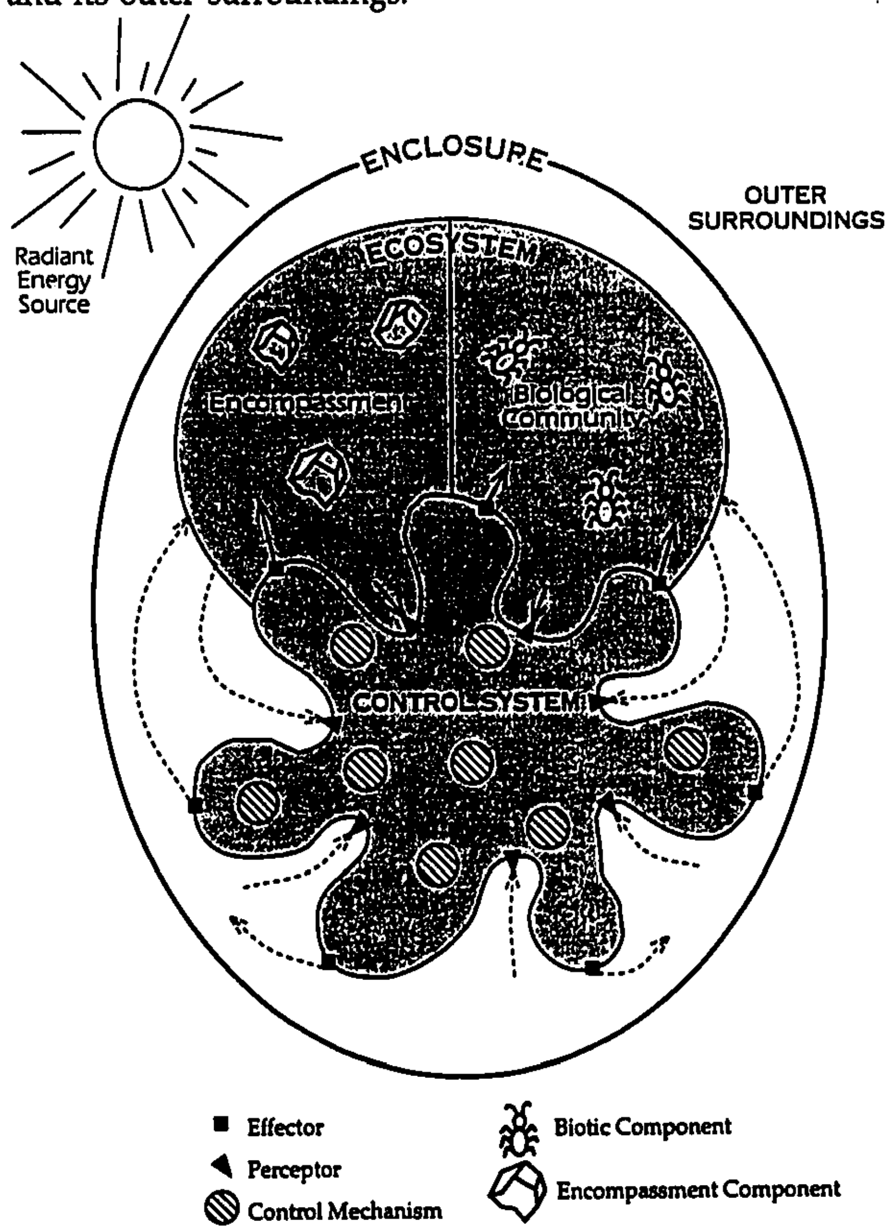


Figure 1: Diagram of the EcoCyborg and its surroundings.

3.2. Enclosure

Although the design details of the physical construct that encloses the ecosystem are beyond the scope of this research project, some conceptualization has been undertaken simply to embellish the context and to clarify particulars.

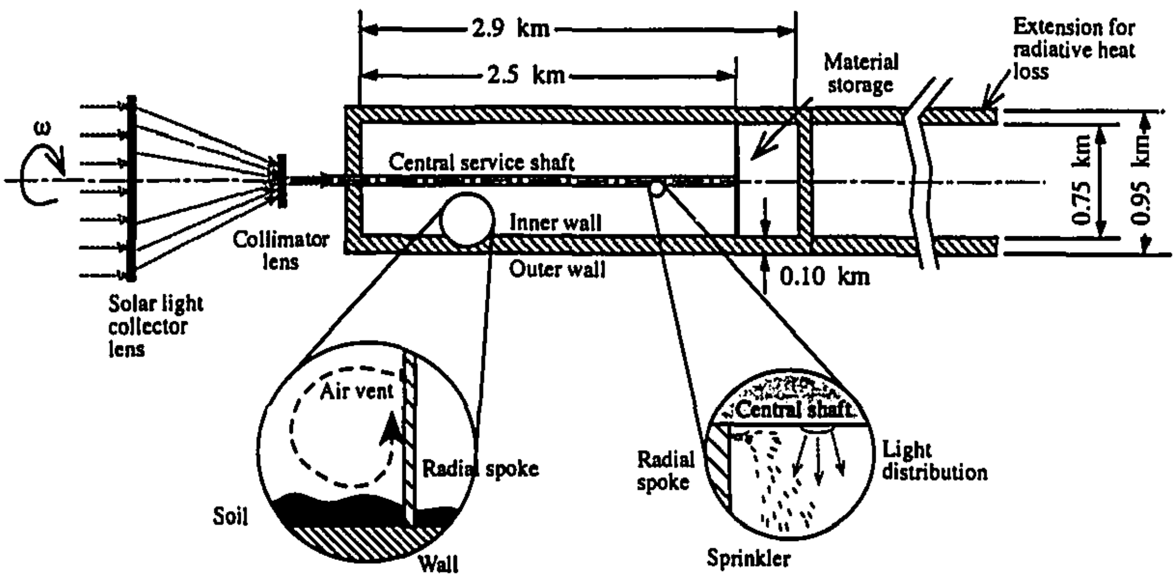


Figure 2: Drawing of the enclosure and related sub-systems.

The ecosystem is contained within a cylindrical enclosure that rotates around its longitudinal axis while maintaining a stable orbit near a radiant energy source (Figure 2). The enclosure is assumed to be mass impermeable but open to energy flow. The main cylinder has an interior length of 2.5 km and diameter of 0.75 km, with a wall thickness of 0.1 km. A cylindrical shaft, 50 m in diameter, runs longitudinally through the center. Thus, the enclosure has an internal volume allocated to the ecosystem of about 1 km^3. A materials storage chamber is appended to the main cylinder at one end. Rotation generates a centripetal force, equivalent in magnitude to the Earth's gravitational field, that enables the organisms in the ecosystem to reside on the 5.8 square kilometre surface of the inner shell. A relatively thin soil layer covers the shell on the inside; this constitutes the terrain for all of the life forms in the ecosystem. The rest of the internal volume is filled with atmospheric gases. The gas phase total pressure is maintained between 96-105 kPa and the partial pressures of the various gases are manipulated to maintain the same composition as air. This is achieved by moving atmospheric components to and from the exterior materials storage chamber and/or by adding/removing gaseous H_2O.

A number of subsystems are employed in the maintenance of an internal environment that is conducive to life support. The center cylindrical shaft contains control equipment and service assemblages such as water purifiers, sprinklers, and light distribution devices. Radial spokes connecting the central shaft to the outer walls are used to exchange materials between the ecosystem and the control subsystems, as well as to provide structural integrity to the enclosure. For example, air may be circulated through heat exchangers in the central shaft and then distributed to the ecosystem out of vents in the spokes. Water may also be redistributed from the spokes as rain, or it may be funneled to a central watercourse.

A large concentrator lens coupled with a collimator lens located at one end of the space station collects radiant energy from a nearby star. A receiver in the central shaft captures sufficient energy to generate light to drive the ecosystem and to operate the EcoCyborg as a whole; light input to the ecosystem is controlled to create radiant intensities that resemble terrestrial patterns. Waste heat is rejected from the system in a supervised manner from the outer enclosure surface, such that the internal temperature is maintained within terrestrial limits and exhibits Earth-like patterns of variation (e.g., 365 day cycle with four seasons). To increase the surface area available for radiative heat loss along the outer wall, a "tail" extension is appended to the space station.

3.3. Ecosystem

The ecosystem consists of all of the organisms in the system together with the habitat that supports them. These are discussed with reference to a conceptual subdivision of the ecosystem into two parts: the encompassment and the biological community.

3.3.1. Encompassment

The term "encompassment" is used synonymously with the ecological term "habitat" to describe the abiotic environment of the ecosystem. The encompassment has four major sections: the soil, the buffer zone, the gas phase (aerial region), and the watercourse.

The buffer zone is the large materials storage chamber located at one end of the enclosure and is not in direct contact with the biological community. Without the vast material buffers of the Earth, biogeochemical cycles in a closed system operate far more rapidly than they do on a large planet. For example, in the Biosphere II project, CO_2 cycled daily compared to Earth's global cycle of 10-12 years (Nelson et al. 1992). Depending upon the degree of control which is adopted, this could result in a poor distribution of materials, causing nutrient deficiencies in the soil or atmospheric imbalances. For this reason, large solid state reserves of primary components (e.g. O_2, CH_x, N_2, H_2O and minerals) are located in exterior storage to use as buffers.

All of the organisms in the biological community reside within the other three sections of the encompassment, interfacing directly with the soil, the watercourse and the gas phase. The soil, which completely covers the inner enclosure shell, has a depth that ranges between 5-10 m. It forms a terrain which is slightly undulating, with a gradual slope towards the watercourse which runs down a grade following the cylinder's circumference. This necessitates an Escherian discontinuity in the landscape, whereby water that reaches the end of the stream ponds and is then pumped, or otherwise transported, up a cliff to resume its flow. During this transfer stage, minerals and any biological contaminants are removed from the water. Some of the minerals leached into the stream water may be reapplied to the soil as fertilizer. The stream is fed by rain, groundwater and storage water (when necessary to augment flow). The water table level is regulated through irrigation (via rain) as well as by managing the stream flow rate. Lastly, a climate is imposed, as described above, by the various enclosure subsystems that work to keep the atmospheric composition, temperature, and radiation intensity within the limits suitable for terrestrial life and fluctuating according to terrestial patterns.

3.3.2. Biological Community

The biotic components are derived from a mature temperate woodland and shrubland biome on Earth. Members of the biological community are representative of almost all taxonomic groups, ranging from viruses and micro-organisms to higher plants, reptiles and mammals, with the exception that there are no aquatic organisms. (The model, however, will not necessarily include all of the species present in the context.) To avoid conflicts with the control system, there are no human inhabitants. The woodland and shrubland setting has been chosen rather arbitrarily, mainly for its aesthetic qualities. This biome type has also been chosen because of its great diversity and number of small plants and animals that can be used to compose an intricate food web. The average dry weight plant biomass for a woodland and shrubland biome on Earth is 6 $kg \cdot m^{-2}$, with an average net primary productivity of 0.7 $kg \cdot m^{-2} \cdot yr^{-1}$. Animal biomass is an average of 4.7×10^{-3} $kg \cdot m^{-2}$ of dry organic matter, and animal production averages 3.5×10^{-3} $kg \cdot m^{-2} \cdot yr^{-1}$. (All numbers are from Whittaker (1975), and are in agreement with those given by Rodin et al. (1972).) The space station ecosystem is similar to its terrestrial counterpart with respect to plant and animal biomass density and productivity and there is no explicitly designated agricultural area. Due to the low productivity of this type of ecosystem (about half that of a temperate evergreen forest), it might, however, be necessary to include one should a community of humans need to be supported.

3.4. Control system

The ecosystem is controlled at a number of implementation levels, ranging from physical to cognitive. The addition of a control system is intended to provide the ecosystem with a level of sophistication in its comportment that is currently not present in natural ecosystems on Earth. The behaviour of existing ecosystems arises from the interactions of thousands of different components, and by no means exhibits any willful direction. Control, however, implies intent. The purpose of the control system is to guide the ecosystem such that its dynamic behaviour best achieves the goals of the controllers (as set by the system developer). There are two types of controllers: intrinsic and extrinsic, which interface with the ecosystem via perceptors and effectors. Components of the control system permeate all areas of the EcoCyborg and, in particular, most of the intrinsic control mechanisms are included in the composition of the ecosystem and its enclosure. This control approach was previously developed by Kok and Lacroix for the control of greenhouse systems (Kok and Lacroix 1991; Kok and Lacroix 1993; Lacroix and Kok 1991; Lacroix 1994). The combination of mechanisms that interact to form a control system for the ecosystem closely resembles the methodology described in the above-mentioned publications.

3.4.1. Control mechanisms

A control mechanism is a device that maintains the value of a particular ("controlled") variable at, or near, a target value. To do this, the mechanism may take into account the values of any number of "considered variables" and then implement a decision by setting new target values for several "manipulated variables". A manipulated variable itself may or may not be a controlled or considered variable. All of the control mechanisms are networked together to form one integrated system. Depending upon the sophistication of the decision process employed by a control mechanism, it may be part of the intrinsic or extrinsic portion of the control system.

3.4.1.1. Intrinsic control

Intrinsic control is the simplest type; it is intended to imitate, as closely as possible, the physical and instinctive reactions of biological creatures. Intrinsic control mechanisms are inherent to the structural components of the ecosystem and its enclosure. They are responsible for maintaining many of the environmental conditions that have been described above in the encompassment and enclosure sections. Physical-level mechanisms are fairly independent and serve to implement the regulatory activity of the ecosystem, such as temperature maintenance. At this level, the distinction between the control system and the controlled system is not clearly delineated. Instinctive-level mechanisms show a more definite decision making process than physical mechanisms, although their response is still purely reflexive in nature. Instinctive mechanisms respond invariantly to potentially dangerous changes, such as lack of oxygen in the atmosphere, by closing (or opening) a valve or setting a switch from ON to OFF etc.

3.4.1.2. Extrinsic control

Extrinsic control mechanisms are not inherent to the objects in the ecosystem, but are essential for the manifestation of its behaviour. The closest approximation to this type of control on our planet is that provided by the global network of human social and political structures. Unlike the intrinsic control system, extrinsic mechanisms are virtual, and reside on external machines such as intelligent agents or computers. There are two levels of extrinsic control: Pavlovian and cognitive. As the name suggests, Pavlovian control mechanisms carry out routine tasks that have been learned. In the most complex cases, this may involve situational behaviour in which a mechanism reacts to a pattern in the values of considered variables. Pavlovian mechanisms are not capable of analyzing the consequences of their actions beforehand. In contrast, cognitive control mechanisms imitate many of the functions of human intelligence, performing such tasks as learning, reasoning, and the analysis and synthesis of information. These mechanisms are also capable of constructing and testing models of the EcoCyborg. The cognitive control network may attain a certain degree of consciousness, depending upon the configuration of its constituent mechanisms.

3.4.2. Effectors and perceptors

The control system interfaces with the various components of the ecosystem via perceptors and effectors which serve as information retrieval and actuator devices. Perceptors include all of those elements that function as sensors, providing the control mechanisms with quantities that describe the state of the encompassment and the biological community. These might include thermocouples, gauges or tactile sensors and artificial vision on robots, for example. Effectors carry out control decisions within the ecosystem. They include such things as valve actuators, mechanical equipment drivers and robotic operators. Each time a control mechanism establishes a control strategy, that strategy will ultimately be implemented as a series of directives to various effectors. Effectors and perceptors reside inside the ecosystem and the enclosure; they form the boundary between the control system and that which is controlled.

4. Technique to Model and Simulate Context

A composite model is being written to investigate the design and behaviour of the context described above. This model will be composed of three parts: an ecosystem model, a Pavlovian controller model and a cognitive controller model. Each of the parts of this model will be implemented by separate simulation programs, the executable versions of which are called modules. An additional module will be used to impose a climate on the ecosystem, based on a radiation and temperature model. Several other modules are also being developed to facilitate dynamic data access and the overall execution of the simulation. All of these modules will interact during simulation as a means of enacting the dynamics of the EcoCyborg.

Any model that is intended for use in simulation must be designed to suit the programming environment in which it will be implemented. Development of the various parts of the EcoCyborg model has, therefore, been largely influenced by the equipment and tools available, and the modeling, simulation, and implementation approaches that have been adopted for the application as a whole. Thus, the overall layout of the modeling and simulation approach for the EcoCyborg Project will be presented here before a more detailed description of the ecosystem model is given. This will help to place the design of the ecosystem model in the context for which it has been developed, with regards to the anticipated limitations and requirements of the simulation environment.

4.1. Equipment and tools

A variety of hardware and software tools have been, and will be, used to model and later simulate, the EcoCyborg. Present hardware includes several IBM-PC compatibles equipped with standard Intel 486 microprocessors and VGA quality graphics. So far, these have been used to compose sections of the three sub-models. Large-scale simulations may require a faster CPU and will use at least 64 MB of memory. If necessary, some coprocessor intensive activities may be downloaded to a mainframe or work station. A separate computer may be employed solely to run a graphics display routine. The simulation will be run under OS/2 Warp in order to exploit the multitasking capabilities of that operating system (see discussion of the simulation approach, below).

Most of the programming code for the simulation modules will be written in IBM C for OS/2, although a wide range of other software tools will be employed where appropriate. For example, rule-based expert systems (e.g., Guru) and artificial neural networks (e.g., NeuralWare) will be used to write parts of the cognitive controller module. Programs to implement the temperature and radiation models will be written in Matlab 386 by MathWorks. The ecosystem module will be written entirely in C. Microsoft QuickBasic has been used to compose a number of interface applications that aid in the specification of the ecosystem model prior to simulation.

4.2. Modeling approach

Modularity has been a key focus for the development of the EcoCyborg model. As mentioned above, it is being composed as a collection of (sub)models, each of which will be used in simulation by a separately compiled program (see below) residing in an executable module. The modules are named for their respective contents: Ecosystem, Pavlovian Controller and Cognitive Controller. Different approaches have been employed to create models to represent these three parts of the EcoCyborg. Since the ecosystem model is the focus of this paper, the techniques adopted for the creation of the other models will not be presented here. A fourth module, the Weather Generator, is used to impose a climate on the ecosystem; the temperature component of this is described in detail in the Appendix.

The modeling approach that has been adopted to describe the ecosystem is individual-based and object-oriented. The structure of the model is patterned after the Smalltalk/V conventions (with some liberal indiscretions), and includes a hierarchy of objects that describes the attributes and methods of everything in the ecosystem. This class structure has not, however, been implemented in an object-oriented language. Instead, the hierarchy is maintained in data matrices which store all of the information about each object. These include pointers to C subroutines that function as object methods. The matrices contain the entire state of the ecosystem at any time and, in fact, represent the structural form of the model itself. The decision not to use an object-oriented language was deliberate, based upon the documented experience of other ecological modelers (e.g., Silvert 1993) as well as the author's own observations. A trial ecosystem model (based on a simple 2 species predator-prey scenario) was written in Smalltalk/V during the summer of 1993, and when test simulations were run that included more than 1000 individuals, execution speed became unacceptably slow. It is expected that considerable speed will be gained by using a language such as C, for which a highly optimized compiler is available. Also, by storing instance variables in matrix format, access to the values should be very fast. In addition to the slow run time speed, the OOPS vision of reusable objects is not very versatile. For example, since every object in Smalltalk is partially defined according to its position in the inheritance tree, it becomes difficult to add or remove components from a system if that involves a restructuring of the class hierarchies. The holistic approach to object creation in OOPS also makes it difficult to differentiate between the model specification and the simulation structure. Thus, with these limitations in mind, the ecosystem model has been composed using a hybrid of procedural and object-oriented techniques.

4.3. Simulation approach

The simulation approach that will be adopted has been described in detail by Lacroix et al. (1994). In general, the EcoCyborg model will be implemented through a multi-process, multi-level simulation that employs the multitasking capabilities of OS/2. OS/2 can dynamically prioritize tasks and events in order to effectively allocate resources such as memory, access to input and output devices, and CPU time slices. In doing so, the operating system differentiates between "threads" (execution paths), "processes" and "screen groups" (sessions). This enables a user to keep a number of executable modules all concurrently memory resident. Thus, the EcoCyborg simulation will be implemented as a number of separate processes, which may be run concurrently in one session. Control and allocation of CPU resources will be maintained via semaphores that will be called from each process using OS/2 application program interface (API) calls. Interprocess communication (for example, value passing) will be achieved through the use of shared memory segments that will also be addressed with API calls. This approach will enable considerable flexibility during model development and execution for a number of reasons. First, it allows for a division of the model into separate modules, enabling independent development of each as a separate research project. In addition, placing different components of the EcoCyborg model in separate simulation modules eliminates any restriction of using only one programming language, enabling specific software packages to be used for different modules if they are considered more appropriate to a task.

4.4. Implementation approach

In total, the simulation will be composed of seven modules, each of which will reside in a separate OS/2 process. Three of these will represent the constituent parts of the EcoCyborg model (Ecosystem, Pavlovian Controller and Cognitive Controller). There will also be a Weather Generation module, based on an ancillary model. All of these must be specified (setting of initial conditions) and composed (creation of objects, control mechanisms, etc.) before a simulation is run. Once this has been done, a fifth module, the Simulation Manager, will initialize the system. The Simulation Manager will also be responsible for governing time management and memory allocation during the entire simulation procedure. Two other processes, the Recorder and the Display Interface modules, will collect and display data dynamically.

5. Structure and Composition of the Ecosystem Module

The Ecosystem module will simulate the behaviour of the artificial ecosystem, using a program which implements the model described below. The model is a representation of all components that *reside* in the ecosystem (although some may be *part of* the control system). They are: the biological community, the encompassment, effectors, perceptors, and intrinsic control mechanisms. The following section contains a description of the model that has been designed to describe these components, followed by an explanation of how this model will be structured and composed for use in simulation.

5.1. Nature of the model

Before detailing the specifics of the model's design and creation, a description of its general nature will be given. The scope of the model is largely a reflection of the overall objectives that influenced its creation. For example, the desire to capture the complexity of a natural ecosystem led to the arbitrary decision that a "toolbox" of about 1000 species would be required in order to adequately represent all of the functional groups in an ecosystem. In addition, the design of the encompassment component of the model was based on the objective to include spatial effects when computing environmental influences or individual exchanges. Thus, the following is an explanation of the degree to which the model represents the hypothetical ecosystem presented in Section 3.

The model focuses upon species interaction, through the exchanges between discrete individuals. Each individual will be described by about 100 variables (attributes) and will be able to perform about 100 different actions (methods). The attributes and methods will mimic the traits and activities of real, physical organisms. In this respect, the model bears a close resemblance to many weak artificial life models. There may be up to 100,000 individuals, representing 1000 different species of plants, arthropods, mammals, reptiles, etc. (Although the structure of the model can theoretically accommodate an unlimited number of species, 1000 has been set as an upper bound for this project, due to considerations of simulation time restraints.) Viruses are not included in the model, and micro-organisms are not explicitly defined, i.e., they are not represented as species, and their activities will be assumed to be properties of the environment in which they reside. For example, the soil is assumed to have an innate ability to decompose organic matter and fix nitrogen. Thus, through the implicit inclusion of such species as soil micro flora and fauna, the real number of modeled species will, in effect, exceed 1000.

Some "individuals" are actually conglomerates. Although each population is modeled as a collection of individuals wherever possible, for some species (e.g. insect or grass species) this approach is beyond the realm of practicality. In these cases, the populations consist of groups of organisms that function as single individuals. The decision to model a species on an individual rather than on a grouped basis is made when the survival of an individual begins to have some consequence within the social order of its community. For example, the death of a single worker ant is of no consequence to an ant colony; ants have impact only when they form complex social units; hence ants are modeled on a lumped basis. Most mammals, on the other hand, form social units (however briefly) in which each member plays an integral and generally irreplaceable role, e.g., as a parent, hunter or scout. Thus, these species are always modeled on a truly individual basis. Each animate object (individual or lumped mass) has attributes and methods which define it. Individual members within a species differ in the values of their attributes but do not vary with respect to the kinds of actions that they may take (methods). All of the attributes are superficial; none of the species is described by a genotype. The model does not accommodate the evolution of species, nor does it include true sexual reproduction. Animals, for example, must find a member of the opposite sex in order to reproduce, but the offspring will not inherit the traits of their parents. New individuals are always created using randomly selected attribute values (this is discussed in detail below).

The model includes the effects of both spatial and environmental factors on individual behaviour. The ground surface is modeled as a 100 x 100 unit grid that covers a rectangular area equal in dimension to that of the unrolled cylindrical shell in which the ecosystem is enclosed. A three dimensional terrain is mapped to the grid as a set of elevations. For each grid cell, the total soil mass, nutrient composition, organic matter content and water table level is monitored. Dissolved nutrients carried in the groundwater flow are also tracked. Each animate object is located on the grid, which allows for the consideration of spatial proximity to determine the influences of predators or competitors, and plant productivity is a function of the environment in the grid cell in which it grows. Climatic variables are considered in the determination of functions such as plant growth, respiration, photosynthesis and evapotranspiration.

Since the model is of a materially closed ecosystem, mass and energy transfer within, and across (in the case of energy), the system boundary needs to be considered. The model is structured so that mass accounting can be performed by means of maintaining mass balances for some of the basic life-supporting compounds: N_2, NO_x, CO_2, CH_x, H_2O, O_2, CO_2, P, and K. The total mass in the system is equal to the mass in the encompassment plus the mass in the biological community. Some energy accounting is also performed. In particular, the transfer of arbitrary energy units is computed within the biological community food web to model the vitality of individuals. It is assumed that energy input and rejection through the walls of the enclosure is a non-limiting process.

5.2. The object hierarchy

A hierarchy has been established to model the identity and classification of each object in the ecosystem, the major classes of which are shown in Figure 3. All classes inherit from the parent class, Simulation Object. Subclasses of Simulation Object are: Physical Object, Virtual Object, Input Object, Control Object and Management Object. The hierarchy has been constructed to conform to the object-oriented paradigm, and generalizations to subclasses have been made on the basis of commonality of methods. For example, the Heterotroph class is generalized on the basis of eating habit, rather than morphology. All classes inherit from their ancestors, and some exhibit cases of multiple inheritance, in which they share features with more than one kind of object. For example, Groundwater is both a Water Component and a Ground Component, and members of the class inherit attributes and methods from both superclasses.

- **Simulation Object**
    - *Physical Object*
        - Biotic Component
            - Autotroph
                - Chemosynthesizing Organism
                - Photosynthesizing Organism
            - Heterotroph
                - Carnivore
                - Herbivore
                - Omnivore
                - Detrivore
                - Parasite
        - Encompassment Component
            - Ground Component
                - Soil Mineral
                - Soil Organic Matter
                - Groundwater (ref. Water Component)
                - Dissolved Mineral
                - (Surface Water)
            - Atmosphere Component
                - Atmospheric Gas
                - Atmospheric Water Droplets (ref. Water Component)
            - Water Component
                - (Atmospheric Water Droplets)
                - (Atmospheric Gas -- H_2O)
                - (Groundwater)
                - Surface Water
                - Water in Biotic Components
            - Solid Storage Component
    - *Virtual Object*
        - (Intrinsic Control Mechanism)
    - *Input Object*
        - Temperature Input
        - Radiation Input
    - *Control Object*
        - Effector
        - Perceptor
        - Controller Mechanism
            - Intrinsic Control Mechanism (ref. Virtual Object)
    - *Management Object*
        - Grid Object

Figure 3: Object hierarchy of the major classes in the ecosystem model.

Encompassment Component objects (Ground Components, Atmospheric Components, etc.) have mass as their most significant attribute. The Grid Object class is a special type of object that simply internalizes and acts upon Ground Component objects. Each instance of Grid Object represents one grid cell of the terrain, and contains, as an attribute, a collection of instances of each subclass of Ground Component. This form of object linking implies that, each time a Grid Object instance is updated during simulation, the masses of its Soil Mineral objects, Soil Organic Matter objects etc., will be changed to reflect the material flux in that locality which has occurred over the elapsed time period.

- **Autotroph**
  - *Chemosynthesizing organism*
  - *Photosynthesizing organism*
    - Tree
      - Needle-leafed
      - Broad-leafed evergreen
      - Evergreen-sclerophyll
      - Broad-leafed deciduous
      - Thorny
      - Rosette tree
      - Bamboo
    - Climbing plant
      - Woody vine
      - Non-woody vine
    - Shrub
      - Needle-leafed
      - Broad-leafed evergreen
      - Evergreen-sclerophyll
      - Broad-leafed deciduous
      - Thorny
      - Rosette shrub
      - Stem succulent
      - Semishrub
      - Dwarf shrub
    - Epiphyte
    - Herb
      - Fern
      - Grass
      - Sedge
      - Other graminoid
      - Forb
    - Thallophyte
      - Lichen
      - Moss
      - Liverwort

Figure 4: Autotroph class hierarchy (species classes omitted). Key: **Level 1**; *Level 2*; Level 3; Level 4

The Biotic Component class contains most of the objects in the model. Subclasses of its two major branches, the Autotroph class and the Heterotroph class, are given in Figures 4 and 5. Fabricating an inclusive, yet simple, class structure presented all of the difficulties that have plagued biological taxonomists over this century. The division of the Autotroph class is based upon the scheme given in Whittaker (1975, 62), in which he classifies land plants according to growth-form. Whittaker's groupings were perceived to be utilitarian from a number of viewpoints. The classes combine plants that influence their surroundings in similar ways, whether it be due to their size, nutrient requirements, or palatability to herbivorous organisms. The Heterotroph class has been subdivided in such a way so as to maximize method inheritance, based on a purely arbitrary judgment. Depending on the granularity of behaviours modeled, or on the emphasis placed by the modeler on form, a different hierarchy might be more appropriate. In this particular hierarchy, the basis for generalization is not consistent at each level. For example, the Mammal class is generalized based on housing type (which implicitly specifies a range of movement methods), whereas the Bird class is generalized according to eating habits.

Consecutive levels in the Autotroph and Heterotroph class hierarchies are numbered 1 through 5, with the top classes at Level 1. All of the classes in Levels 1-4 are abstract classes, that is, they serve to define an object type, but have no instances of their own. Level 5, at the base of the object tree, represents species classes. Due to the large number of species classes (1000), they are not included in the figures. Every organism modeled in the ecosystem is an instance of one of the Level 5 classes (and is, therefore, a disjoint instance of that species' superclasses). All of the instances of a particular species class constitute a population. Each species class is defined by its position in the object hierarchy, and by a list of attributes and methods, many of which are inherited from its Levels 1-4 ancestor classes. Attributes or methods which are added to the class definition at Level 5 are generally very

Heterotroph

- *Carnivore*
  - <u>Mammal</u>
    - Ranging mammal
    - Burrower
    - Tree inhabitant
    - House builder
  - <u>Reptile</u>
    - Turtle
    - Lizard
    - Snake
    - Crocodile
  - <u>Amphibian</u>
    - Frog
    - Toad
    - Newt
    - Salamander
  - <u>Bird</u>
    - Raptor
    - Insectivorous bird
  - <u>Arachnid</u>
    - Spider
    - Harvestmen
    - Tick
    - Mite
    - Scorpion
  - <u>Insect</u>
    - Flying insect
    - Crawling insect
- *Herbivore*
  - <u>Mammal</u>
    - Ranging mammal
    - Burrower
    - Tree inhabitant
    - House builder
  - <u>Reptile</u>
    - Turtle
    - Lizard
    - Snake
    - Crocodile
  - <u>Amphibian</u>
    - Frog
    - Toad
    - Newt
    - Salamander
  - <u>Bird</u>
  - <u>Arachnid</u>
    - Spider
  - <u>Mollusc</u>
    - Snail
    - Slug
  - <u>Insect</u>
    - Flying insect
    - Crawling insect
- *Omnivore*
  - <u>Mammal</u>
    - Ranging mammal
    - Burrower
    - Tree inhabitant
    - House builder
  - <u>Reptile</u>
    - Turtle
    - Lizard
    - Snake
    - Crocodile
  - <u>Amphibian</u>
    - Frog
    - Toad
    - Newt
    - Salamander
  - <u>Bird</u>
  - <u>Insect</u>
    - Flying insect
    - Crawling insect
- *Detrivore*
  - <u>Insect</u>
    - Flying insect
    - Crawling insect
  - <u>Worm</u>
    - Earthworn
  - <u>Fungi</u>
    - Mushroom
    - Mould
    - Rust
- *Parasite*
  - <u>Fungi</u>
    - Mushroom
    - Mould
    - Rust

Figure 5: Heterotroph class hierarchy (species classes omitted)
Key: **Level 1**; *Level 2*; <u>Level 3</u>; Level 4.

DESCRIPTIVE ATTRIBUTES	LOGICAL STATE ATTRIBUTES
ID number	alive
age	asleep
age of maturity	in tree
maximum age	in house
gender	pregnant
x-coordinate	in heat
y-coordinate	has mate
x-coordinate of house	
y-coordinate of house	**MULTIPLE VALUE STATE ATTRIBUTES (0 TO 1 RANGE)**
litter size	healthy
gestation period	hungry
birthmass	thirsty
maxmass	crowded (plant)
mass	afraid
height	cold
volume	oxygen deficient
total leaf area	nitrogen deficient
metabolic rate	
photosynthetic rate	

Figure 6: Sample attributes for organisms.

specific characteristics particular to the species. Listings of sample attributes and methods are shown in Figures 6 and 7, respectively, to provide examples of the types of states and interactions which are modeled by the Biotic Component class. Each species class is described with approximately 100 different methods and 100 attributes, most of which are inherited from superclasses. The majority of the methods are very simple operations which act on the values of an individual's attributes.

5.3. Definition of classes

Designing an object model such as the one given above, with its classes and their descriptions, is the first step in any object-oriented development. Once this has been done, the model must be implemented within a programming environment. This involves entering class definitions so that they can be used later, in the model specification phase, to create instances which contain the class characteristics. The procedure used to define the various classes in the object tree is described below. Since the ecosystem model is not being implemented with an object-oriented language, the approach used to define class structures is quite different from that which is employed by the Class Browsers of Smalltalk/V or C++, for example. In the class definition phase of the model, an interface (written in QuickBasic) is

SPECIAL	**REPRODUCTION RELATED**	**CONSUMING FOOD**
daily attribute update	look for mate	eat plant
check self existence	find mate	eat part of plant
check weather	lose mate	eat seed
check soil properties	lay egg	eat insect
act	fertilize egg	eat animal
	get pregnant	drink
MOVEMENT	give birth	be completely eaten
move towards	be born	be partly eaten
move away from	flower	increase hunger level
fly away	be pollinated	decrease hunger level
land in tree	make seeds	increase thirst
run up tree	disperse seeds	decrease thirst
leave tree	germinate	photosynthesize
carry something		uptake water
be carried	**GROWTH**	transpire water
enter house	grow	respire
leave house	age one day	metabolize
sleep	increase mass	excrete mass
wake up	decrease mass	
hibernate	extend roots	**COLLECTING FOOD**
	grow taller	search for food
DEATH	drop leaves	store food
wilt		
be chopped down		**HOUSING RELATED**
starve		dig burrow
freeze		expand burrow
die of old age		make nest
die 1 (leaving corpse)		defend territory
die 2 (no corpse)		

Figure 7: Sample methods for organisms

used to collect data describing each class in the Biotic Component and Control Object branches of the Simulation Object tree. Encompassment Component classes are purely conceptual in their implementation, and, because of their simple structure, are not explicitly defined. All of the class structure information is stored in matrices (or is implicit, as is the case for Encompassment Objects) for use later in the model specification phase. The complete class definition procedure is presented in the flowchart given in Figure 8 (back cover insert).

The interface that is used to input class definitions has been designed to be robust and versatile, so that the same library of data input subroutines can be used for all of the programs that are part of the model definition and specification phases. Each interface program shown in Figure 8 has five associated "support files". The PAR file is opened first; it contains a list of the program's operating parameters, including the names of the other four support files. The ALF file contains the text of all of the comments and instructions that the program requires to communicate with the user. Most of these are standard messages that are used by all interface programs. The VAL file contains the default, minimum and maximum values of all the numeric and string variables that are input by the user. To facilitate data input, a MEM file contains a record of the values that were input during previous runs of the program; the user may chose to load these as default values for the present run. The SCR file can be used to record all of the necessary keyboard inputs, so that the program may be self-operating. In addition to the standard support files, most programs have associated data files that are used to store all of the information that is collected from the user, whether it be a class description or a list of climate parameter values.

Most of these files can be added to, or edited. Many of them are affiliated with a "symbiont file". Symbiont files serve a purpose similar to that of a file header; they contain a short description of the length and contents of the file to which they are affiliated. Further details of the operations and contents of the specific programs and files shown in Figure 8 are given below in separate sections.

5.3.1. Biotic component classes

Definition of the five levels of biotic component sub-classes (Figure 8) is a three stage process that results in the creation of a single matrix containing complete descriptions of all the species classes (Level 5) in the ecosystem model. In a fourth stage, the food web matrix is generated. Lastly, all of the information collected in the previous stages is assembled into a number of matrices to facilitate species class information access during simulation.

The first stage is the method and attribute listing step. Prior to their class assignments, all the methods and attributes must be defined and created. It is anticipated that, maximally, there will be about 1000 of each. With the Method Definition Program, each method is assigned a string variable name and an integer identification number. For reference purposes only, each method definition also includes a descriptive tag explaining the method's intended usage. The *method file*, which is produced by the Method Definition Program, contains the list of method names and descriptions, by order of identification number.

Definition of attributes (Attribute Definition Program) is similar to that of methods, with the addition of an extra step. To allow for individual variation within populations, values of attributes are assigned randomly to new class instances according to pre-defined distributions. Thus, it is necessary to specify a distribution type as part of an attribute description. There are 18 distribution types covering a range of possibilities from binary to Gaussian, for each variable type (i.e., logical, integer or real). For example, integer valued attributes (such as age in years) may be assigned a value according to a uniform distribution between set minimum and maximum limits, or the value may be chosen from a binomial distribution having a known mean and standard deviation. In addition, many attributes may have different distribution parameter values, depending upon whether a mature or immature individual is being initialized. Since the creation of mature individuals (as opposed to immature ones) is a special case that will only occur during system initialization (Section 5.4), all attributes require "infant creation" values. Thus, for each attribute, a distribution type is assigned, and is designated by a three digit code, one digit of which is used to specify whether or not separate "adult" creation values are required. The attribute names, descriptive tags, and distribution types are stored in the *attribute file* and passed, along with the *method file,* to the Class Definition Program.

The Class Definition Program is used to create classes at Levels 1 through 4. To define a class, the following items must be specified: class level, class name, identification numbers of all applicable attributes and methods, and the values of the parameters required to specify the distribution of each attribute for both mature (if required) and immature individuals. Each attribute distribution type requires a unique set of parameters to describe it (e.g., values of the mean, standard deviation and minimum and maximum values, where applicable). The values of these parameters will vary in their magnitude depending upon the attribute being specified, and the class to which it applies. Methods and attributes which will be inherited from superclasses need not be specified, with the exception of attributes whose values are to be redefined at that level. The following might be a description of the Carnivore class at Level 2:

Level: 2
Class name: *Carnivore*
Methods: 4 56 7 89 216 753 .... (method identification numbers)
Attributes: 78, 0.265 0.3 (attribute #78, followed by distribution values)
112, 40 60
.
.
.
19, 3.2 5.64 0.23 9.2

This information is input for each class in the hierarchy and is written in the manner shown above to the corresponding *level file* .

The four level files are used as input to the Species Class Definition Program. At this stage, Level 5 classes are defined and inheritance is computed. Each class is assigned a common name as well as a composite name that indicates its ancestry. To define a Level 5 class, it is sufficient to simply specify its lineage through the object tree by providing the names of the classes from which it inherits attributes and methods. The class definition is then composed by parsing and combining the definitions of the four ancestor classes. If desired, additional attributes and methods may be added to the class definition at this level. In particular, it may be expedient to redefine creation values for some attributes, or to add specialised methods that distinguish the species. The model can accommodate the creation of up to 1000 Level 5 (species) classes. The complete species class definitions, including all inherited traits, are stored in the *level 5 description file*. Using this, a food web definition in the form of a n x n (n = # of species) matrix designating relative "eat preferences" for each species is compiled by the Eat Preferences Designation Program. The results of the entire Biotic Component class definition phase are then compiled by several intermediate programs to compose six master matrices that contain all of the information required to create species class instances. The contents of the files containing these matrices are described in Table I.

5.3.2. Effectors and perceptors

The Perceptor and Effector Definition Programs (Figure 8) are used to define the various types (classes) of effectors and perceptors, resulting in *master effector* and *master perceptor* files. Effectors are differentiated according to the kind of action they take; there are a number of possibilities such as 'open valve', 'move object', 'kill individual', etc. Built into each effector is a conflict resolver to handle multiple, and likely contrary instructions originating from different control mechanisms. In each resolver, the degree of influence of each control mechanism must be specified; in most cases, instinctive control messages will take precedence over cognitive directions. Since the role of a perceptor is simply to deliver a variable value, these are of only one type. Attributes of effectors and perceptors are the minimum, maximum and gain (conversion factor) values of the variable they address or perceive.

5.3.3. Intrinsic control mechanisms

In the Intrinsic Control Mechanism Definition stage (Figure 8), two types (classes) of mechanisms are defined: Physical Control Mechanisms and Instinctive Control Mechanisms. Attributes of these mechanisms include

NAME	CONTENTS OF FILE
Attribute File	Master list of attribute names, ID numbers, descriptions and distribution types
Attribute Type Matrix	Number of parameters to read for each attribute based on its distribution type
Attributes Matrix	Logical matrix of attribute applicability for each species: attribute ID # versus species #
Attributes Vector	Number of attributes for each species
Creation Matrix for Species	Parameter values for the creation of immature individuals of selected species
Eat Preferences Matrix	Square matrix of species eat preferences, designated by a value between 0 and 1
Effector File	List of selected effectors, variable addressed by each, and operating parameter values
Elevation File	Elevation of each grid point on the terrain
Enclosure Parameter File	Interior dimensions of the cylindrical enclosure
Encompassment Matrix	Mass of each tracked substance in different parts of the encompassment (e.g. soil, atmosphere,...)
Intrinsic Control File	List of selected intrinsic control mechanisms, their set points, and effectors and perceptors addressed
Level 1-4 Description Files	Class name, description, number of methods and attributes, list of applicable methods and attributes by ID #, default parameter values for the creation of immature and mature individuals
Level 5 Description File	Species class name, lineage, description, number of methods and attributes, list of applicable methods and attributes by ID #, parameter values for the creation of mature and immature individuals (inherited attributes and methods are included)
Master Creation Matrix for Species	Parameter values for the creation of immature and mature individuals of all species
Master Effector File	List of all effector types
Master Intrinsic Control File	List of all intrinsic control mechanism types
Master Perceptor File	List of all perceptor types
Method File	Master list of method names, ID numbers and descriptions
Methods Matrix	Logical matrix of method applicability for each species: attribute ID # versus species #
Methods Vector	Number of methods for each species
Perceptor File	List of selected perceptors, variable perceived by each, and operating parameter values
Population Matrix	One line per individual; species name and attribute values
Preliminary Population Matrix	One line per individual; species name and attribute values minus location attributes
Soil Matrix	Listing of material composition of soil blocks by grid cell
Sub-master Creation Matrix for Species	Parameter values for the creation of immature and mature individuals of selected species
Weather Parameter File	Characteristics of the climate to be simulated

Table I: Description of files used in the ecosystem model definition and specification phases.

lists of the effectors from which they receive information, the perceptors that they address, and the identities of the variables that are considered in their decision making process. The definition of intrinsic control mechanism types is stored in the *master intrinsic control file.*

5.4. Specification of initial conditions

A procedure has been devised to enable the specification of different ecosystem compositions and initial conditions. The steps in this procedure are described below and shown in the flowchart in Figure 9 (back cover insert). In general, this is done by selecting object types whose characteristics have been previously defined, and then creating instances of these. There are three main regions that need to be initialized in order to configure an ecosystem for simulation: the encompassment and enclosure, the biological community, and the intrinsic control system. Whereas the class structure is a relatively fixed characteristic of the model, the possibilities for specifying different configurations are practically limitless. The model specification phase is structured such that any number of different experiments can be undertaken by altering the scenarios for any of these three regions (as well as for the Pavlovian and cognitive controllers, although not dealt with here). In this manner, it will be possible to vary species diversity, terrain type, atmospheric composition, etc., for different simulations.

5.4.1. Encompassment and enclosure

The boundary of the ecosystem and the nature of the terrain on which the biological community interacts is largely determined by the enclosure. Specification of the enclosure dimensions is a necessary prerequisite to the generation of any encompassment components, which, it turn, is a prerequisite to the creation of biotic components. In general, specifying the enclosure involves inputting a suitable diameter and length for the mass-impermeable cylindrical shell to the Enclosure Definition Program. Once this is completed, a terrain is generated by the Terrain Definition Program, which assigns elevations to a rectangular grid, the dimensions of which equal that of the unrolled cylinder. Normally, the assigned terrain will represent that which was described above in Section 3.3.1, with land sloping towards a watercourse that runs down a grade following the cylinder's circumference.

Once the enclosure size and terrain have been defined, the overall material composition of the abiotic system can be specified. In the Encompassment Definition Program, major components are initialized according to their situation in one of four zones: the soil, the gas phase, the water course or buffer storage. The masses of the life supporting substances resident in each zone are recorded in the *encompassment matrix*. In the soil zone, the mass is further subdivided and placed into grid cells. For each grid cell, the amount of nutrients, as well as the water content is specified. The elevation and mass composition of each cell is recorded in the *soil matrix*. These two matrices are input to the Ecosystem module, and serve to define the material state of the entire encompassment. These will be resident in memory during a simulation, and, for purposes of mass accounting, will be updated with each time increment.

5.4.2 Biological community

The specification of a biological community implies the creation of instances of the various species classes. Minimally, one founding population must be initialized and dispersed about the encompassment in order to compose at least a rudimentary ecosystem for simulation.

The first step in specifying a biological community is the selection of species to include in the ecosystem via the Species Selection Program. The biological community may be composed from instances of all, or any combination of, the 1000 species described by the Level 5 classes. Selection of a species list results in a *sub-master creation matrix for species* which retains only the data for the chosen species. Extraneous information pertaining to unused species is then stripped from the other six Biotic Component class description files (File Stripping Program) to avoid filling large amounts of memory with irrelevant data during simulation. Next, for each selected species, a population is generated by creating the desired number of instances. Individual differentiation within a population is achieved by assigning each instance unique attribute values according to the distribution parameter values given in the *sub-master creation matrix for species.* Initial populations contain a range of ages, from newborn to very old, thus attribute values for both adult and immature individuals are required at this stage. Since instances created during the course of a simulation will all be infants, adult creation values are stripped from the *sub-master creation matrix for species* before it is made available as input to the Ecosystem module. The specific description of each individual, namely, its species type and its attribute values, is stored in the *preliminary population matrix.* The final step in the specification of biotic components involves the placement of each individual on the terrain grid according to directions given by the Dispersal Program. Dispersal is actuated on a population basis, and may be entirely random, or clumped according to environmental or spatial parameters. The grid space location (x and y coordinates) is maintained for each organism as a set of two attributes. The complete *population matrix* will serve as the collection of organisms that is simulated by the Ecosystem program. This, along with the other matrices defining the biological community, will reside in memory during a simulation.

5.4.3. *Intrinsic control*

Specification of an intrinsic control system to accompany a composed ecosystem is a fairly straightforward process. First, the desired number and types of effector and perceptor instances are specified and composed from their class definitions using the respective programs shown in Figure 9. Next, during the Intrinsic Control Mechanism Selection and Composition phase, intrinsic controller instances are made and linked to the appropriate perceptors and effectors. The various operating parameter values and set points for each of the mechanisms are also designated at this stage. Files describing the effectors, perceptors, and intrinsic control mechanisms that have been specified are inputs to the Ecosystem module and become integrated into the main program's code.

5.5. Formation of executable code

The Ecosystem module is the executable program that is used to animate the ecosystem model after its components have been specified in the manner described above. Assembly of the executable code for the module is shown in Figure 10. The process involves linking a large number of subroutines with a main control loop. For example, each of the 1000 methods called by members of the biological community is enacted by a subroutine. Similarly, the sequences of operations that are carried out by each control

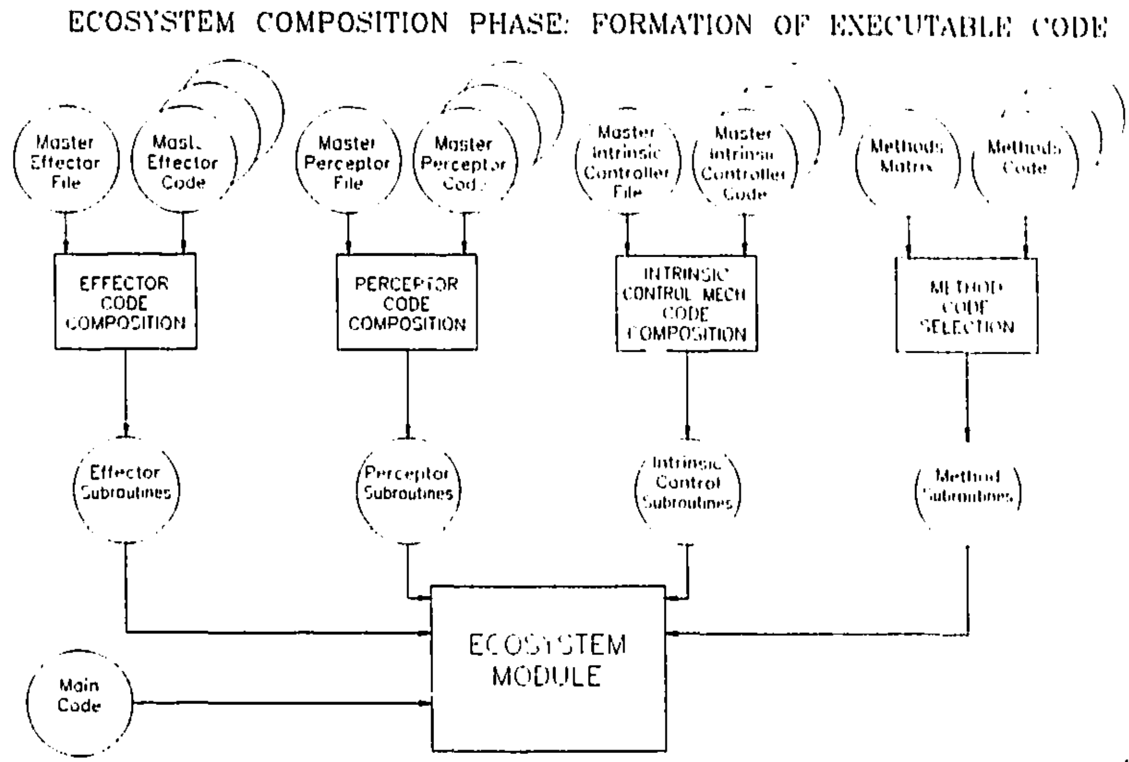


Figure 10: Formation of the program code for the executable ecosystem module.

mechanism, effector, and perceptor are also compartmentalized in separate subroutines. All of these subroutines are compiled and linked to the main ecosystem code to form an executable program. This module operates by placing all of the model's matrices into memory and then updates the ecosystem state in discrete time steps, which are allocated by the Simulation Manager module, while exchanging information with the other simulation processes. Updating the ecosystem involves activating the intrinsic control system, animating the biotic components, and calculating a new state for the encompassment. Each organism has a general method called "act" which contains its behavioural routine, and which calls various methods according to the organism's activity priorities. The main control loop updates the state of the biological community by sending the message, "act", to each object. In a similar manner, each of the encompassment components and control mechanisms is activated, and all of the effectors and perceptors are called to respond to any messages from intrinsic or extrinsic controller objects.

6. Accomplishments to Date and Future Work

Formulation of a model of this size requires rigorous foresight and attention to detail. It is for this reason that the design of an overall layout for the development of the model has been very important. At this stage, the structure and composition of the ecosystem model has been completely formulated and preliminary programming of the interface has begun. There remain, however, numerous details to consider before any final simulation code is written. Most of the details pertain to how, and what, variables will be passed between the ecosystem and other modules. Also, the specific operations of many of the organism's methods have yet to be confirmed. For example, equations for metabolic, respiratory, or growth functions have not been fixed.

At present, work is being done to define and parameterize the Biotic Component classes. Programs to input methods (Method Definition Program), attributes (Attribute Definition Program) and Levels 1-4 class definitions (Class Creation Program) have been written, and the Species Class Definition program is presently being composed. Preliminary methods and attributes lists are being made, and a list of approximately 1000 woodland and shrubland species has been assembled. Attribute values describing those traits (e.g. birthmass, mature height, litter size, etc....) that have been documented in the ecological literature are presently being collected by Julie Bâcle (Undergraduate student, Dept. of Agricultural Engineering) for each of these species. Notes about their behavioural tendencies and life cycles are also being recorded. This work is almost complete, and has resulted in a large database of biological information that will be used to parameterize the model.

Many of the programs for the model specification phase still need to be written. The majority of these will be standard data input units, so that their development is expected to be fairly quick. The Terrain Definition Program has already been written. This was based on a terrain model created in the early 1980's which was updated by Grant Clark (Clark and Kok 1995). The bulk of future work to be done pertains to the writing of actual simulation code: the 1000 methods and the main Ecosystem module program. This represents approximately two years of coding time as well as another year of debugging and experimentation. Thus, this thesis serves, largely, as a definition of the framework for future work to be done within the EcoCyborg Project.

7. Other Modules

There are six other modules that are being developed concurrently with the Ecosystem module. Several of these are the basis of other graduate research projects.

The climate is being modeled separately from the ecosystem. Radiant intensity, temperature values, and rainfall rates will be synthesized by the Weather Generator Module and be imposed upon the ecosystem. These will follow terrestrial patterns. A model to generate temperature values, based on a Fourier transform approach, has been written; this is described in the Appendix. Radiation and rainfall models are presently being developed. As a result of the imposed climate and the closed mass cycling through the ecosystem, a number of conditions may arise, indications of which will be seen in the values of a number of other variables, for example: relative humidity, vapour pressure, total pressure, soil water table level and soil available water. Since temperature and radiation values are externally imposed, and cannot be adjusted by the control system, environmental conditions will be maintained within ranges comfortable for terrestrial life by manipulating other encompassment variables.

The extrinsic controller modules, both Pavlovian and cognitive, are being composed by Robert Molenaar, using a similar specification approach to that of the intrinsic control system. While models of the Pavlovian controller mechanisms are being written in C, the cognitive controller is being modeled with a number of inter-linked expert systems and neural networks. The possibilities of using a neural network capable of learning as the simulation proceeds are being investigated. Both extrinsic controllers will rely on the perceptors in the ecosystem to provide information about its current state. Thus, their knowledge acquisition from the system will be limited by the number and type of variables which are perceived and reported. Any variables that are spatially dispersed (such as species population locations or soil water content) will be reported at one-tenth of the granularity (i.e., on a 10 x 10 grid) of that considered within the ecosystem model. This lack of resolution has been adopted in order to keep the amount of data that is transported between processes to a reasonable level of flow. In this manner, each controller will guide the system in an informed, but not omniscient, state.

The Recorder and Display Interface modules will be developed by Grant Clark, whose aim is to study data arising from the simulations. The Recorder module will output periodic "snapshots" of the system, capturing data which will then be analyzed off-line for interesting patterns, etc. Each snapshot may contain up to 70MB of data if it includes the entire system state. The Display Interface will provide a dynamic, visual summary of the system's state and may reside on a separate computer from that which runs the simulations.

8. Discussion

In the formulation of the ecosystem model, many of the benefits of individual-based and object-oriented modeling discussed earlier in the literature review have been accrued. There are, however, some practical disadvantages of the particular approach that has been adopted for this model. Some of the conceptual simplicity of the OOP paradigm has been sacrificed, particularly in the class definition stage. Objects in the ecosystem model will not be *truly* encapsulated in the object-oriented sense, and polymorphism will be difficult to implement fully. If this were to be done (and a final decision has not yet been made), some sort of secondary addressing scheme would have to be used, whereby an object told to "eat" would then look in a table to determine which particular version of the "eat" method to enact. This would decrease overall execution speed during simulation, which may not be worth the ideational gain. Due to the overwhelming number of objects in the model, most decisions to override the conceptual simplicity of OOP have been justified on the basis of improving computational speed.

One of the most common bottlenecks in individual-oriented modeling is that of parameterizing the model. Not only are there literally thousands of parameter values to enter, but when the model is ecological, the values are often difficult to find, if they have been recorded at all. Values such as the mass of an insect larvae, the number of spores produced in a year by a moss, or the amount of seeds eaten by a field mouse, have to be collected for every species in the model. In keeping with a typical theme in biology, mammals have been studied far more extensively than other creatures, thus mammal species have been far easier to parameterize. For many species, selected attribute values have been derived from educated guesses. Fortunately, individual-oriented models tend to be very robust and do not necessarily require extreme precision in order to faithfully represent macroscopic patterns (Hogeweg and Hesper 1990). For example, models of turbulence in fluid flow formulated in terms of bumping molecules are not affected by changing the manner in which the molecules collide. Thus, rough approximations of organism traits are not likely to greatly modify the overall system behaviour. Since the objective is to capture the global patterns of ecosystem dynamics rather than to represent a specific ecosystem, small changes in the system's internal state will not be of significance to this project.

The model can be considered, in many respects, to be representative of weak artificial life which emulates the characteristics of a living system through the combined interactions of simple entities. Although the individuals themselves do not have the capacity to evolve, nor to reproduce themselves (two commonly cited criteria for artificial life), the ecosystem is expected to display life-like behaviour at the system level. Evolution should occur at the population scale. For example, emergent behavioural phenomena such as scramble and contest competition, resource monopolization, the formation of subgroups, and migration around the terrain, may arise. Examples of positive and negative interactions between populations will be widespread, and parasitic or symbiotic relationships might also emerge.

The ecosystem model is primarily strategic in nature and may be claimed to be, at most, a caricature of a physical ecosystem. Nevertheless, it should exhibit enough of the general behaviours of a natural ecosystem to serve as a rough prototype for the examination of a number of ecological questions. Many of these relate directly to the development of an engineering methodology for the design and control of ecosystems. Criteria for selection of complementary species to compose the biological community will have to be ascertained. It is expected that as many as half of these species will become extinct during the initial transitory period as the ecosystem is established. The degree to which a control system can offset some of these critical changes, and the desirability of such interference, particularly during the transition phase, is unclear. In addition, the relationships among biological system variables such as resistance, resilience or stability, and parameters such as species richness, biomass levels or species connectivity, have not been quantitatively defined. Ecologists have yet to agree upon which of these variables can be used to give indicative measurements of impending instability. Observation of the described ecosystem model should address some of these issues, leading to initial suppositions about the design and control of ecosystems as well as contributing to our theoretical understanding of the behaviour of dynamical biosystems in general.

The application of cognitive decision making as a control technique will require some experimentation to determine in what manner and to what degree a cognitive controller should intervene with regular ecosystem functions. This will depend, to a large extent, on the purpose and intent of the extrinsic control system. It should not, in general, set a prerogative agenda that favours any particular species or sanctimonious objective; in most instances it will be desirable to create cognitive mechanisms which are entirely system-oriented and wholly geared towards sustaining the viability of the ecosystem over the long-term. The controllers will have to decide which parameters are indicative of impending instability or system failure, and set limiting values for these. Based upon the ecological considerations mentioned above, these parameters might include such things as energy flux, chemical and biological patterns of mass transfer, species list persistence, population changes, etc.

The tendency of the ecosystem to perpetuate itself within limits set by the control system will be highly dependent upon the initial conditions. It is hoped that by experimentation with different starting states, a healthy ecosystem will emerge that is robust and exhibits resilience to external and internal perturbations. The EcoCyborg is expected to behave as a continuous, dynamical system which functions according to underlying, deterministic equations rather than generating random noise. It is highly likely that the system will exhibit chaotic attractors whose states may be divided by fractal basin boundaries. This type of behaviour will be actively sought in the time series of state variables. Since it is the general consensus that in dynamical systems in nature, chaos is typical rather than the exception, the observation of chaotic behaviour in the EcoCyborg would be considered a strong indication of success.

9. Contributions to Knowledge

The following were original contributions to knowledge and to the EcoCyborg Project research group:

I. Layout and definition of the entire EcoCyborg Project.

II. Development of the framework for an ecosystem model which:
- contains many individuals interacting in a spatially heterogeneous environment
- can be interfaced with an extrinsic control system

III. Design of an individual-based and object-oriented approach to the modeling of complex systems.

IV. The development of a method for composing and simulating an object-oriented model in a procedural language, including the writing of an interface to accomplish this.

IV. Contribution to the formulation and layout of future work to be done.

10. Summary and Conclusions

In summary, an artificial ecosystem model has been written as a tool to investigate the design and dynamics of autonomous biosystems. The model, through the use of an individual-oriented approach, represents a physical ecosystem by considering its fundamental characteristics and behaviours to arise as the result of interactions between multiple levels of complex components. It will be used to study the effects of various control strategies on different ecosystem configurations. The use of modeling and simulation techniques as research tools provides a degree of lenience and flexibility that could not be attained by constructing a physical model. By working in the virtual realm, hundreds of controlled experiments can be run to single out specific design considerations, necessary building materials, control procedures, etc. This should lead to the development of an overall engineering methodology that can be applied to a diverse range of autonomous biosystems, from intensive greenhouse systems to space biospheres and, especially, terrestrial ecosystems.

References

Assad, A. M. and N. H. Packard. 1992. Emergent colonization in an artificial ecology. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life,* eds. F. J. Varela and P. Bourgine, 143-152. Boston, MA: MIT Press.

Austin, M. P. and B. G. Cook. 1974. Ecosystem stability: a result from abstract simulation. *Journal of Theoretical Biology* 45:435-458.

Averner, M. M. 1989. Controlled ecological life support system. In *Lunar Base Agriculture: Soils for Plant Growth,* 145-153. Madison, WI: ASA-CSSA-SSSA.

Bond, W. J. 1994. Keystone species. In *Biodiversity and Ecosystem Function,* eds. E. D. Schulze and H. A. Mooney, 237-253. Berlin: Springer-Verlag.

Bourgine, P. and F. Varela. 1992. Introduction: towards a practice of autonomous systems. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life,* eds. F. J. Varela and P. Bourgine, 143-152. Boston, MA: MIT Press.

Botkin, D. B., J. F. Janak and J. R. Wallis. 1972. Some ecological consequences of a computer model of forest growth. *Journal of Ecology* 60: 849-872.

Clark, G. and R. Kok. 1995. Generation of a virtual terrain. Paper presented at the Northeast Agricultural and Biological Engineering Conference, July 30-Aug. 2, Bar Harbor, Maine. pp. 15.

Drogoul, A., J. Ferber, B. Corbara and D. Fresneau. 1992. A behavioral simulation model for the study of emergent social structures. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life,* eds. F. J. Varela and P. Bourgine, 143-152. Boston, MA: MIT Press.

Folsome, C. E. and J. A. Hanson. 1986. The emergence of materially-closed system ecology. In *Ecosystem Theory and Application,* ed. N. Polunin, 269-288. New York: John Wiley & Sons Ltd.

Fortson, R. E., J. C. Sager, J. O. Bledsoe, R. M. Wheeler and W. M. Knott. 1992. Current performance of the NASA Biomass Production Chamber. ASAE Paper No. 92-4001. St. Joseph, Mich.: ASAE. pp. 16.

Fortson, R. E., G.W. Stutte, J. C. Sager and R. M. Wheeler. 1993. CELSS reliability and plant response to environmental stress. ASAE Paper No. 93-4006. St. Joseph, Mich.: ASAE. pp. 10.

Gitelson, I. I., I. A. Terskov, B. G. Kovrov et al. 1989. Long-term experiments on man's stay in biological life-support system. *Advanced Space Research* 9(8): 65-71.

Godfray, H. C. and S. P. Blythe. 1990. Complex dynamics in multispecies communities. *Philosophical Transactions of the Royal Society of London* B 330: 221-233.

Green, D. G. 1993. Emergent behaviour in biological systems. In *Complex Systems - From Biology to Computation,* ed. D. G. Green and T. J. Bossomaier, 24-35. Amsterdam: IOS Press.

Green, D. G. 1994. Connectivity and complexity in landscapes and ecosystems. *Pacific Conservation Biology,* in press.

Hanson, J. 1982. Workshop on closed-system ecology, held at the California Institute of Technology, Pasadena, California, USA, during 18-22 January 1982. *Environmental Conservation* 8(4): 357-359.

Hassell, M. P., J. H. Lawton and R. M. May. 1976. Patterns of dynamical behaviour in single species populations. *Journal of Animal Ecology* 45: 471-486.

Hogeweg, P. 1988. Cellular automata as a paradigm for ecological modeling. *Applied Mathematics and Computing* 27:81-100.

Hogeweg, P. and B. Hesper. 1990. Individual-oriented modelling in ecology. *Mathematical and Computational Modelling* 13(6): 83-90.

Huston, M. 1992. Individual-based forest succession models and the theory of plant competition. In *Individual-Based Models and Approaches in Ecology,* eds. D. DeAngelis and L. Gross, 408-420. New York: Routledge, Chapman and Hall, Inc.

Judson, O. 1994. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution* 9(1): 9-14.

Kawata, M. and Y. Toquenaga. 1994. From artificial individuals to global patterns. *Trends in Ecology and Evolution* 9(11): 417-421.

Kok, R. and R. Lacroix. 1991. An analytical framework for autonomous and enclosed agro-ecosystems. A case study: the greenhouse. 2º Convegno Nazionale: Il Vivaismo Orticolo, Foggia, Italy, May 1991. pp. 14.

Kok, R. and R. Lacroix. 1993. An analytical framework for the design of autonomous, enclosed agro-ecosystems. *Agricultural Systems* 43: 235-260.

Lacroix, R. 1994. A framework for the design of simulation-based greenhouse control. Ph.D. Thesis, McGill University, Montreal.

Lacroix, R. and R. Kok. 1991. A cognitive controller with recourse to simulation. Presented at the Automated Agriculture for the 21st Century ASAE Symposium, Chicago, USA, December 1991.

Lacroix, R., R. Kok and G. Clark. 1994. OS/2 as a setting for the simulation of an enclosed agro-ecosytem under cognitive control. Submitted for publication to Canadian Agricultural Engineering, April 1995.

Langton, C. G. 1990. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* 42: 12-37.

Lawton, J. H. and V. K. Brown. 1994. Redundancy in ecosystems. In *Biodiversity and Ecosystem Function*, eds. E. D. Schulze and H. A. Mooney, 255-270. Berlin: Springer-Verlag.

MacElroy, R. D., J. Tremor, D. L. Bubenheim and J. Gale. 1989. The CELSS research program: a brief review of recent activities. In *Lunar Base Agriculture: Soils for Plant Growth*, 145-153. Madison, WI: ASA-CSSA-SSSA.

Maley, C. C. and H. Caswell. 1993. Implementing i-state configuration models for population dynamics: an object-oriented programming approach. *Ecological Modelling* 68: 75-89.

May, R. M. 1973. *Stability and Complexity in Model Ecosystems.* New Jersey: Princeton University Press. pp. 235.

Mees, A. I. 1990. Modelling complex systems. In *Dynamics of Complex Interconnected Biological Systems*, eds. T. Vincent, A. Mees and L. Jennings, 105-123. Boston: Birkhäuser.

Nelson, M, P. Hawes and M. Augustine. 1992. Life systems for a lunar base. In *The Second Conference of Lunar Bases and Space Activities of the 21st Century*, 513-518. NASA Publication 3166, Vol. 2. Houston: NASA.

Nisbet, R. M. and W. S. C. Gurney. 1976. Model of material cycling in a closed ecosytem. *Nature* 264: 633-634.

O'Callaghan, J. and M. Conrad. 1992. Symbiotic interactions in the EVOLVE III ecosystem model. *Biosystems* 26:199-209.

Pimm, S. L. 1984. The complexity and stability of ecosystems. *Nature* 307: 321-326.

Pimm, S. L., J. H. Lawton and J. E. Cohen. 1991. Food web patterns and their consequences. *Nature* 350: 669-674.

Plant, R. and N. Stone. 1991. Object-oriented models. In *Knowledge-based Systems in Agriculture*, 259-285. New York: McGraw Hill.

Prince, R. P. and W. M. Knott. 1989. CELSS Breadboard Project at the Kennedy Space Center. In *Lunar Base Agriculture: Soils for Plant Growth*, 155-163. Madison, WI: ASA-CSSA-SSSA.

Ray, T. S. 1992. Evolution, ecology and optimization of digital organisms. Santa Fe Institute working paper 92-08-042.

Ray, T. S. 1994. TIERRA.DOC: Documentation for the Tierra Simulator V4.1. Located at: ftp://tierra.slhs.udel.edu/

Rizki, M. M. and M. Conrad. 1985. EVOLVE III: a discrete events model of an evolutionary ecosystem. *Biosystems* 18: 121-133.

Rodin, L.E., N.I. Basilevich and N.N. Rozov. 1972. Productivity of the world's main ecosystems. In *Productivity of World Ecosystems, Proceedings of a Symposium Presented at the V General Assembly of the Special Committee for the IBP*, 13-26. Washington, D.C.: National Academy of Sciences.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen. 1991. *Object-Oriented Modeling and Design.* New Jersey: Prentice Hall.

Silvert, W. 1993. Object-oriented ecosystem modelling. *Ecological Modelling* 68: 91-118.

Taylor, C., D. Jefferson, S. Turner and S. Goldman. 1988. RAM: Artificial life for the exploration of complex biological systems. In *Artificial Life*, ed. C. G. Langton, 276-295. Reading, Mass.: Addison-Wesley Publ. Co.

Taylor, C. and D. Jefferson. 1994. Artificial life as a tool for biological inquiry. *Artificial Life* 1(1): 1-13.

Tregonning, K and A. Roberts. 1979. Complex systems which evolve towards homeostasis. *Nature* 281: 563-564.

Whittaker, R.H. 1975. *Communities and Ecosystems*, 2nd Ed. New York: Macmillan Publishing Co., Inc. pp. 385.

Wolfram, S. 1984. Cellular automata as models of complexity. *Nature* 311: 419-426.

Appendix

DAILY AVERAGE TEMPERATURES: MODELING AND GENERATION WITH A FOURIER TRANSFORM APPROACH

336-608A
Special Topics in Agricultural Engineering
Term Paper

Submitted by

Lael Parrott
December 1994

INTRODUCTION

To study the behaviour of natural and artificial ecosystems by means of simulation, we couple a number of input and output components to a dynamic ecosystem model and then simulate the behavior of the overall system with the composite model. Components may include weather simulators, control mechanisms or data collection modules. Temperature is one of the most important variables that "drive" an ecosystem and we have therefore written an input module to generate it. Before writing the component we considered our needs in order to identify performance criteria for the model's functioning in a simulation.

The nature of the ecosystem model itself has largely defined the desired features of the temperature module. Our overall requirement is that it be able to produce a fairly realistic, yet granular, yearly time series of temperature values corresponding to any type of climate. For example, when modeling an ecosystem we sometimes need to imitate physically real weather at a specific geographical location, and at other times we want to create a purely hypothetical climate. Thus, the component should allow both. It should therefore be versatile enough to apply to different types of climates and be easily adjustable so as to permit the generation of appropriate values corresponding to them. Although, ultimately, the time spacing of output temperature values should be variable, it was considered sufficient to operate initially at a single, rather modest granularity, i.e., one value per (simulated) day. Because we need the ability to run simulations of any arbitrary duration, the component should be able to generate unlimited weather data. We don't, however, need the entire temporal variability of temperature to be taken into account (i.e., long term climatic shifts); only variations within a period of less than one year are of interest. Because at this stage in our work the spatial variability of weather within an ecosystem is not yet considered, it is sufficient for the module to generate values for only one site.

In accordance with the above requirements, the component is based on and generates Daily Average Temperature (DAT) values. It generates them a year at a time, the years being independent of one another, but belonging to the same "family". The particular family is defined with a number of parameters whose values may be derived from physical weather data, or chosen arbitrarily to generate a hypothetical climate. Thus, our objectives have been to: a) detect and describe the patterns in real DAT data, b) to compose a mathematical model based on that description, c) to determine how well the model could be tuned to several data sets and d) to explore the model's ability to generate hypothetical DAT values.

Our approach has been as follows: initially we studied a number of real DAT data sets and we wrote a mathematical DAT model. We have based our model on the spectral description (Fourier transform) method, because of its simplicity and universality. This is a simple way to generate fractal patterns (Saupe, 1988). Next, we tuned the model to several real DAT data sets and then compared the outputs from the tuned models with the real values. This required the development of suitable criteria and a method of comparison. Finally, we used the model to generate some hypothetical DAT values.

LITERATURE REVIEW

Many natural systems of interest in agro-environmental studies are driven by meteorological variables such as solar radiation, rainfall and dry-bulb temperature. Although physical data, either historical or arriving in real time, may be used as input for simulating the behavior of a natural system, it is generally more convenient to use synthetic data. One reason for this is that the amount and quality of physical data available for a particular location is invariably quite limited so that only a narrow range of system response can be studied. Consequently, much effort has been expended on the creation of models, and the programs incorporating them, with which natural variables can be mimicked. In physical reality these variables are continuous but they can only be recorded and dealt with in a discrete manner; usually values are spaced at finite, equal intervals of time, the size of the interval depending on the application. Therefore, weather data (as well as values for similar natural variables such as hydrologic flow rates etc.) are almost always available as discrete, equally-spaced time series and the modeling-simulation problem for such variables consists of generating similar sets. For a hypothetical climate the synthesized values may be spaced at any interval but for an imitated climate these must be spaced at the same or a larger increment than the physical data.

For the modeling of natural variables a number of related approaches have been developed, most of them based on some statistical interpretation of physical data, the sophistication of any given approach depending on the specific needs of the author, the resources available and the intended user group. Generally, data are analyzed according to some prior notion or impression about the variable and parameters are then extracted which are subsequently used in conjunction with a model whose structure is based on the same notion; such models constitute explicit descriptions of the variable. It is, however, also possible to create implicit descriptions with for example, neural networks (Cook and Wolfe, 1991; Kok *et al.*, 1994). In the neural approach no prior notion is required but one of its limitations is that hypothetical climates cannot be readily created. In the explicit description approach the explanation of weather variables and related phenomena is strongly influenced by the prior notions of movement of the earth around the sun, the moon around the earth and the earth about its own axis. As a result, most authors endorse the supposition that the behavior of such variables is due to two elements: predictable and erratic (Amato *et al.*, 1986; Boileau, 1983; Julian, 1967; Kline *et al.*, 1985; Matalas, 1967; Phillips, 1984; Richardson, 1981). The first element describes the calculable variation of the variable, repeated every day, every 365.24 days etc. and in some cases this is all that is considered in constructing a model. This can be done on a strictly theoretical or on an empirical basis and the model can simply consist of a fairly short polynomial or Fourier series, intended to describe the element as exactly as possible, within the allowed complexity of the approach and over the frequency range required, e.g., daily, monthly or annual values might be generated. Phillips (1984) demonstrated that the predictable part of hourly solar flux or temperature can be adequately described with 34 Fourier coefficients. For simulations, however, the intent is usually to have the same variability in the simulated as in the physical data. To accomplish this the erratic (mostly short-term and small magnitude) variations in the variables are customarily interpreted as being random noise which is then modeled with the aid of stochastic methods and added to the model of the predictable element. In practice it doesn't matter that the erratic element is probably chaotic rather than random in nature; as long as the stochastic methods generate noise patterns which are similar to the real ones, the model will be able to mimic the physical variable adequately.

Although the erratic element is usually regarded as being random, this does not imply that the events of which the series is composed are considered to be independently distributed in time. On the contrary, the noise in natural variables is often analyzed with autocorrelation techniques, including the autoregressive moving average (ARMA) method and, accordingly, generating processes used in the models are frequently based on these and related procedures such as the first-order Markov method (Amato *et al.*, 1986; Julian, 1967). However, because the autocorrelation function and the power spectrum are a Fourier transform pair (Julian, 1967; Phillips, 1984) harmonic analysis can be used analogously to autocorrelation to characterize a noise. In this way a "noise spectrum" can be described with relatively few parameters and, consequently, by setting only a few parameters, the essential aspects of such an entire spectrum can be specified in a model so that an imitation of the erratic component can be generated fairly readily. This approach is simple and can also be computationally advantageous. In a spectral description both predictable and erratic variations can be treated similarly, so that there is no need to create a dissimilarity between them. Of course, in theory, although this is also possible in the autocorrelation approach, harmonic analysis (i.e., based on Fourier transformation) has the advantage that the contributions of the various component frequencies to the total signal are so readily identifiable and quantifiable. In this vein, Phillips (1984) pointed out that the first few coefficients of an FFT of climatic data will be much larger than the rest and represent the predictable, long-term weather pattern whereas the other coefficients represent the unpredictable, erratic variations. West and Schlesinger (1990) have indicated that the noise spectrum will probably exhibit inverse power law behavior because weather is the result of a very complex process composed of many fractal mechanisms.

With the harmonic approach it is quite convenient and fairly simple to set up a generating process to imitate a variable's behavior, or to custom design it. The method readily allows the correspondence between real observations and model parameters to be calculated so that tuning for a specific geographical location is straightforward. Also, the relationship between parameter modification and the meaning of that in terms of its physical representation is immediately evident.

MODEL DEVELOPMENT

The overall intent in developing the model was that it be able to consistently generate output which belonged to the same family as natural weather data. Judgment of the model was therefore based upon its ability to effectively imitate the characteristics of physical weather patterns. A number of criteria were established which served as guidelines for the model's creation as well as providing a suitable framework by which to judge its performance. These will be briefly reviewed at this point so that the aim of the subsequent data analysis may become clear. The delineation of these criteria respected the modeling approach of Casti (1992): "[T]he concept of a model of a natural system N is a generalization of the concept of a subsystem of N, and...the essential feature of the modeling relation is the exploration of the idea that there is a set of circumstances under which the model describes the original system to a prescribed degree of accuracy." To this end, the criteria were used to determine the degree to which the model described physical reality at the granularity of one day. Two types of comparisons were made between the physical and simulated climates. For the first type, individually generated output signals were assessed to determine whether or not each was a member of the family of curves for the climate being imitated. For the second type, a group of output signals was evaluated and compared to a group of signals derived from physical weather data to determine whether or not the two groups represented the same family. For each type of comparison, it was deemed sufficient that the simulation output be similar to the physical climate with regards to the following characteristics:

1. *The DAT vectors*

2. *The WAT (Weekly Average Temperature) and MAT (Monthly Average Temperature) vectors*

3. *The first derivative of the DAT vectors taken over a time interval of one day, i.e., the DDAT (Difference in Daily Average Temperature) vectors*

4. *The histograms (showing relative abundance versus temperature) of the DAT, WAT, MAT and DDAT values*

5. *The magnitude versus frequency curves of the DAT values in the frequency domain, as obtained with the Fast Fourier Transform (FFT)*

6. *The phase angle versus frequency curves of the DAT values, also obtained with the Fourier Transform.*

In addition to meeting the above 6 sufficiency conditions, the model must not:

1. *Produce impossible weather*

2. *Produce values that exceed the minimum or maximum temperatures that could conceivably occur in the modeled climate.*

In order to develop a model that met the conditions above, weather data from three Canadian climates was collected and analyzed to determine its characteristics. Analysis involved decomposition of the data into basic components.

Data Analysis

Raw weather data for three sites: Vancouver, Winnipeg and Montreal, was provided by the Canadian Atmospheric and Environmental Service (CAES). For each day over a fifteen year span (1970-1984), hourly dry bulb temperature recordings were obtained. Daily average temperatures were then calculated by summing each day's temperature readings and dividing by 24. This resulted in 15 DAT vectors for each site (365 days/yr., leap days omitted). The WAT, MAT and DDAT vectors were then computed from the DATs. The WAT values were calculated by taking the 7 day weekly averages of the DATs for each of 52 weeks in a year. Similarly, MAT values constituted 30 day monthly averages of the DATs, approximately covering the 12 calendar months of the year. The DDAT values were calculated for each year by taking the absolute differences between successive DATs. The amount of data contained in the various vectors is given in Table I. All of the data analysis and the creation of the model itself was done using the software package MATLAB 386 (Version 3.5m, The Mathworks, Inc., South Natick, MA, USA).

Table I: Amount of physical temperature data obtained.

CAES DATA	24 values/day x 365 days/yr = 8766 values/yr
DAT VECTORS	1 value/day x 365 days/yr = 365 values/yr
WAT VECTORS	1 value/7 day week x 52 weeks/yr = 52 values/yr
MAT VECTORS	1 value/30 day month x 12 months/yr = 12 values/yr
DDAT VECTORS	1 value per difference between days = 364 values/yr

For each climate, averages and standard deviations for the 15 year period were calculated. These are shown in Figure 1. As another measure of the characteristics of the climates, histograms for each of the variables of interest (i.e., DAT, WAT, MAT and DDAT values) were generated. These are shown in Figure 2. Following this, a more detailed analysis was performed to decompose the weather data into constituent components. The analysis, as described below, was applied to each of the 45 DAT vectors (15 from each site, 365 DAT values per vector) independently and separate parameters were calculated for each. Due to the quantity of ensuing data, results will be presented in statistical form, based on 15 year averages. In the description of data processing below, each vector of 365 DAT values is referred to as a "signal". The intermediate results obtained in decomposing a sample signal into its components are shown in Figure 3.

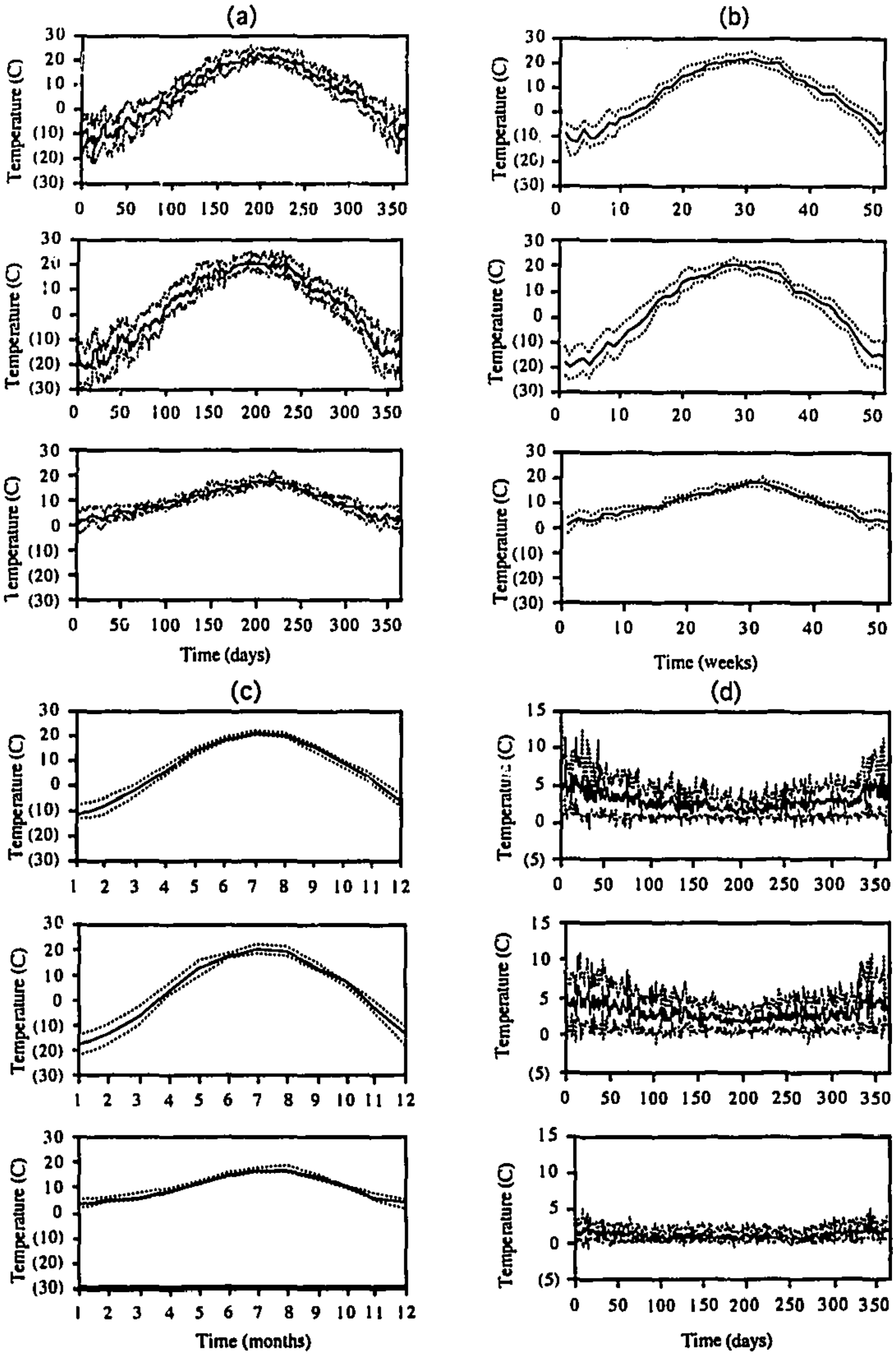


Figure 1: Characteristics of weather data from three Canadian sites. Daily means and standard deviations for each climate over a 15 year period are shown. Top, Montreal; Middle, Winnipeg; Bottom, Vancouver.
(a) DATs; (b) WATs; (c) MATs, and (d) DDATs

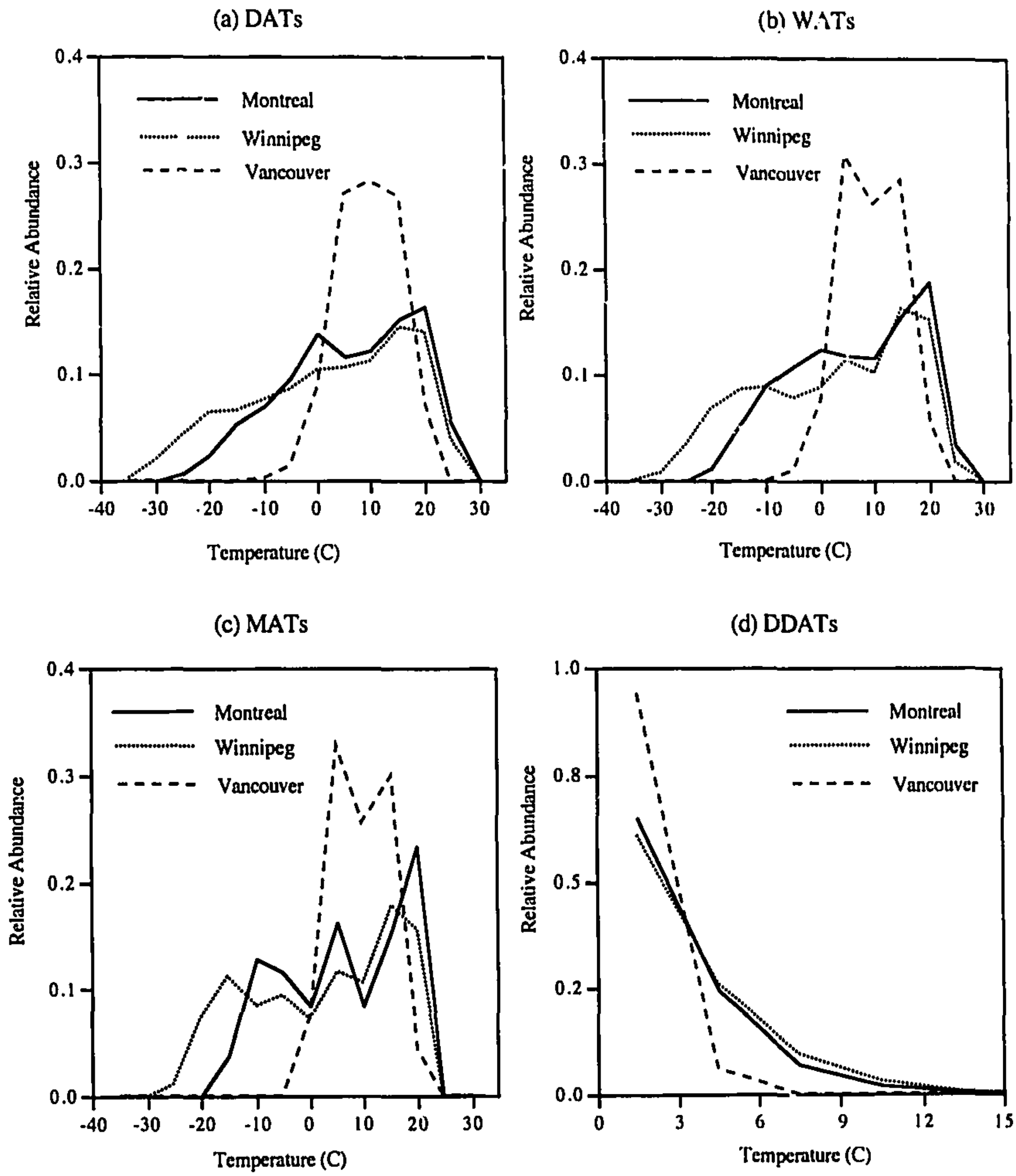


Figure 2: Histograms showing the relative abundance of temperatures for the three climates. Results are based on 15 years of data.

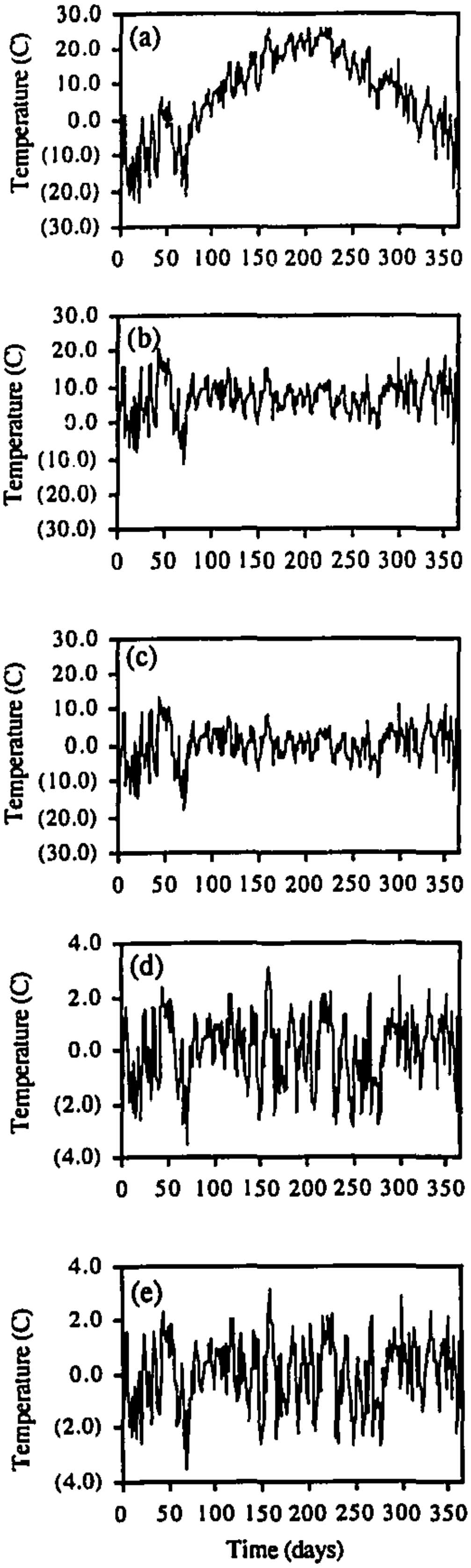


Figure 3: Graphs depicting the intermediate results in data analysis for a sample signal (Montreal, 1984) (a) original signal; (b) signal minus the primary sinusoid; (c) baseline removed; (d) beat frequency sinusoid removed; (e) secondary sinusoid removed: final noise signal

As can be seen on the plots of the DATs, WATs and MATs (Figure 1a,b,c), each climate exhibits a distinctive underlying sinusoidal character that reflects the overall yearly temperature fluctuation. This was the first component identified in the DAT signals. It was described with a least square sinusoid of frequency 1 $yr.^{-1}$ which was subtracted from the signal. The original and resulting signals are shown respectively in Figures 3a and 3b. Next, the mean of the remaining signal (the baseline sinusoid at frequency 0 $yr.^{-1}$) was determined and subtracted (Figure 3c). From this, it is clear that the resulting signal exhibited sinusoidal amplitude modulation at the frequency 1 $yr.^{-1}$. This is also discernible from the shape of the DDAT plots (Figure 1d) in which the daily variations clearly decrease in amplitude over the summer months for each climate.

Thus, the third component identified in the signal was another least square sinusoid of frequency 1 $yr.^{-1}$ with a non-zero baseline. Unlike the previous sinusoids, this one was calculated from the absolute value of the signal which was then divided by this modulating "beat" sinusoid (Figure 3d). Anticipating that the final signal would be processed using FFT techniques, a third least square sinusoid and baseline were then subtracted from the signal in order to obtain a zero magnitude for frequencies 0 and 1. The final signal (Figure 3e) represents that portion of the temperature data which is usually considered to be random noise. Since the large signal components at frequencies 0 and 1 have been removed, the noise signal covers a frequency range from 2 to 182 $yr.^{-1}$.

Thus far, the model has been developed as the sum of the various components described above, as represented by the equation:

$$F(t) = \{a_0 + a_1 \sin(\tau + a_2)\} + \{b_0 + b_1 \sin(\tau + b_2)\}$$

PRIMARY SINUSOID + BEAT SINUSOID

$$* \left\{ c_0 + c_1 \sin(\tau + c_2) + \sum_{i=2}^{182} d_{1i} \sin(\tau * i + d_{2i}) \right\} \tag{1}$$

* SECONDARY SINUSOID + NOISE

The means and standard deviations of the parameters $a_{0,1,2}$, $b_{0,1,2}$ and $c_{0,1,2}$ are presented in Table II for the three climates of interest. As is implicit in equation (1), the noise was described as the sum of 181 sinusoids. In order to determine values for the parameters (d_{1i} and d_{2i}) for each constituent sinusoid, a frequency analysis was performed.

Table II: Means and standard deviations of parameter values for 3 sites. (Model results are shown in parentheses.

Parameter		Sample Size	Montreal	Winnipeg	Vancouver
a_0	mean	15	6.2327 (6.2094)	2.5381 (2.5110)	9.6858 (9.6692)
	std	15	0.6919 (0.6664)	1.0245 (0.9397)	0.4004 (0.3746)
a_1	mean	15	15.613 (15.4412)	19.036 (18.6334)	7.0607 (7.0442)
	std	15	0.7355 (0.7418)	1.1938 (1.1675)	0.4986 (0.4970)
a_2	mean	15	-1.9020 (-1.8980)	-1.8438 (-1.8448)	-1.8759 (-1.8773)
	std	15	0.0540 (0.0549)	0.0694 (0.0706)	0.0806 (0.0776)
b_0	mean	15	3.6473 (3.6187)	4.4927 (4.2458)	1.9398 (1.9358)
	std	15	0.2589 (0.3108)	0.2834 (0.3520)	0.1680 (0.1970)
b_1	mean	15	1.5928 (1.5508)	1.8204 (1.7266)	0.5905 (0.6103)
	std	15	0.3328 (0.4444)	0.4058 (0.5343)	0.2361 (0.2734)
b_2	mean	15	1.3026 (1.2886)	1.3102 (1.3046)	1.5784 (1.4713)
	std	15	0.1929 (0.2867)	0.2179 (0.3397)	0.2245 (0.7101)
c_0	mean	15	-0.0108 (0.0003)	-0.0205 (0.0002)	-0.0041 (-0.0006)
	std	15	0.0106 (0.0160)	0.0134 (0.0166)	0.0123 (0.0114)
c_1	mean	15	0.0913 (0.0734)	0.1295 (0.0800)	0.0895 (0.0605)
	std	15	0.0337 (0.0381)	0.0431 (0.0402)	0.0407 (0.0388)
c_2	mean	15	1.0270 (-0.0218)	1.3067 (0.0024)	1.3147 (-0.0098)
	std	15	1.0486 (1.7330)	0.7613 (1.7631)	2.5510 (1.7712)

Each noise signal was Fast Fourier transformed, and the resulting magnitudes (d_{1i}) and phase angles (d_{2i}) were analyzed further. Sample curves of these are shown in Figure 4. For each site, the FFT analysis resulted in 15 vectors of 181 magnitudes and 15 vectors of 181 phase angles. The resulting magnitudes and phase angles covered a frequency range from 2 to 182 yr.$^{-1}$(the range 183 to 363 being a mirror image of the first half). Shown in

Figures 5a and 5b are the 15 year mean magnitudes and phase angles versus frequency for the Montreal data. The standard deviations of these are also shown.

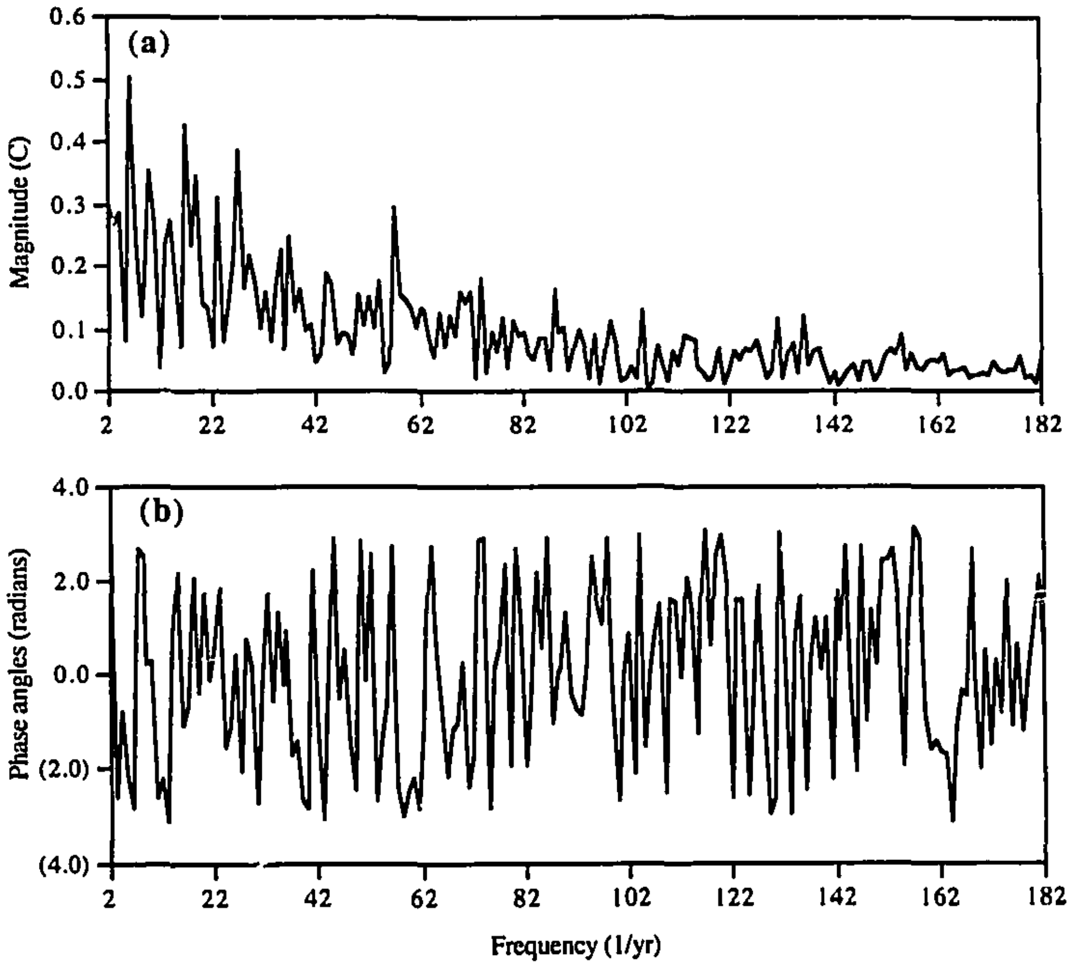


Figure 4: Results of the Fast Fourier Transform on a sample noise signal (Montreal, 1984). (a) magnitudes; (b) phase angles

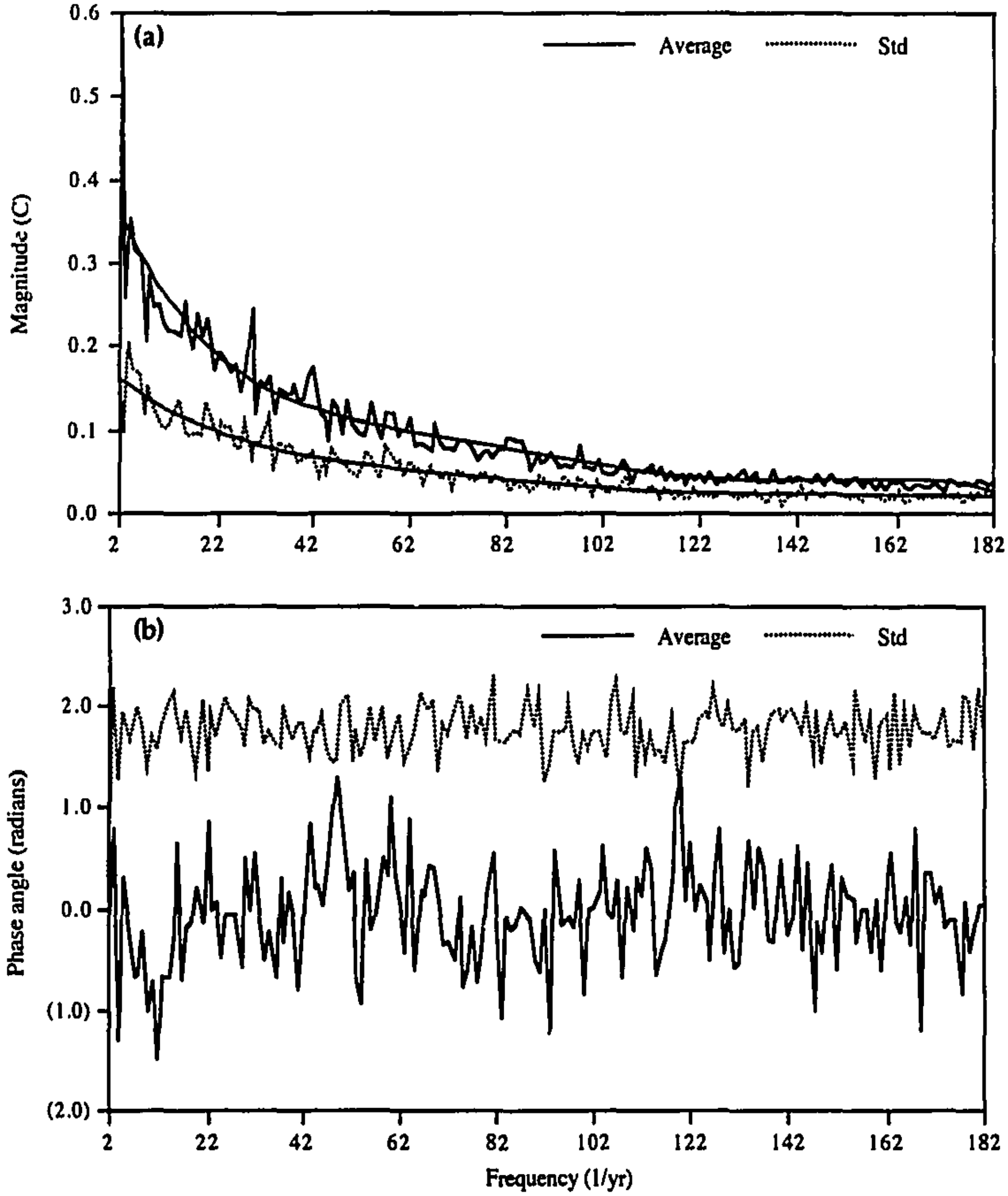


Figure 5: Results of the Fast Fourier Transform for 15 years of Montreal data. (a) Average and standard deviation of the magnitude at each frequency, with least-square polynomial curves; (b) Average and standard deviation of the phase angles

Data Statistics

A number of statistical tests were performed on the parameters obtained above to determine their distributions and interdependence. The first nine parameter values (i.e., those shown in Table II) which describe the behaviour of the signal at frequencies 0 and 1 were found to be normally distributed over the 15 years (α=0.1). A correlation analysis was performed on these parameters to determine the degree to which they were related. The resulting triangular matrix of coefficients is shown in Table III. From this, it is apparent that most of the parameters are not significantly correlated. There appears to be, however, a high correlation between the amplitude of the primary sinusoid with the baselines of the primary and beat sinusoids. There may also be some relationship between these two baselines.

Table III: Matrix of r-values for the nine model parameters.

Parameter	a_1	a_2	a_0	b_1	b_2	b_0	c_1	c_2	c_0
a_1	**1**	0.1	**-0.95**	0.79	-0.48	**0.96**	0.3	-0.04	-0.43
a_2	0.1	**1**	-0.14	0.16	-0.24	0.21	0.41	0.07	-0.24
a_0	-0.95	-0.14	**1**	-0.73	0.43	**-0.91**	-0.34	0.06	0.44
b_1	0.79	0.16	-0.73	**1**	-0.55	0.87	0.47	0.01	-0.61
b_2	-0.48	-0.24	0.43	-0.55	**1**	-0.51	-0.14	0.23	0.19
b_0	0.96	0.21	-0.91	0.87	-0.51	**1**	0.4	-0.01	-0.51
c_1	0.3	0.41	-0.34	0.47	-0.14	0.4	**1**	0.1	-0.81
c_2	-0.04	0.07	0.06	0.01	0.23	-0.01	0.1	**1**	-0.24
c_0	-0.43	-0.24	0.44	-0.61	0.19	-0.51	-0.81	-0.24	**1**

With regards to the noise parameters (d_{1i} and d_{2i}), the magnitudes and phase angles were analyzed to determine their statistical characteristics. For each frequency, the distribution of the magnitudes over 15 years was investigated. The majority (84%) were found to be normally distributed (α=0.1). To describe the general relationship between magnitude and frequency, least square, fifth order polynomials were fit to the 15 year means and standard deviations of the magnitudes for each of the three sites (sample polynomial curves are shown for Montreal data in Figure 5a). Next, a relationship between the magnitudes and the phase angles was sought. A linear correlation analysis was performed on the magnitudes and the phase angles. No relationship whatsoever was found. An autocorrelation analysis was then performed on the phase angles; this also failed to bring to light any significant relation. Since the phase angles showed no discernible pattern, they were therefore assumed to be randomly and uniformly distributed between $-\pi$ and π. This assumption is further supported by the fact that the mean (0.018) and standard deviation (1.830) of the phase angles (based on 15 year of Montreal data) are very similar to the theoretical mean and standard deviation (0 and 1.814) for a continuous, uniformly distributed random variable ranging from $-\pi$ to π.

In summary, the total number of parameters derived from the data analysis is 30; 18 to describe the principal sinusoid parameters and 6 polynomial constants for each of the curves fit through the noise magnitude means and standard deviations.

MODEL COMPOSITION

The model generates one yearly DAT vector at a time by performing the inverse operations of those done in the data analysis. To do this, it is necessary to first calculate a number of parameter values that can be used to generate signal components that are similar to those of the imitated climate. Each of these components is generated in turn and the output DAT vector, $F(t)$, is calculated according to equation (1). For any climate of interest, the parameters used in the model must be generated within the ranges determined through data analysis. The complete procedure is described below.

First, the noise portion of the signal is generated. For each frequency, a magnitude is calculated from the polynomials fitted through the means and standard deviations of the magnitudes for that climate. This is done as follows:

$$MAG_f = MEAN_f + STD_f * RAND_n \qquad f \ \varepsilon \ 2,3,......182 \qquad (2)$$

where $RAND_n$ is a normally distributed random variable having a mean of 0 and a standard deviation of 1. In order to meet the condition that the model not produce impossible weather, $RAND_n$ is, however, restricted to a range of -2.5 to 2.5. In this way, during the successive generation of DAT values for the same climate, the magnitude at any given frequency will essentially be normally distributed with a mean of $MEAN_f$ and a standard deviation of STD_f. The phase angles are generated according to a random uniform distribution ranging from $-\pi$ to π. The values of the 181 time-domain sine waves described by these magnitudes and phase angles are then calculated and summed (Eqn. 1) to produce a simulated noise signal.

After the noise signal has been generated, the three principal sine waves (frequencies 0 and 1) are calculated. Constants describing the baselines, amplitudes and phase shifts of these sinusoids are calculated in the manner described by equation (2) whereby $MEAN_f$ and STD_f ($f \ \varepsilon \ 0 \ or \ 1$) correspond to the means and standard deviations calculated for each respective parameter during data analysis (Table II). The secondary sinusoid is then calculated for c_0, c_1 and c_2 and added to the noise signal. Next, the beat sinusoid is calculated (using the values of b_0, b_1 and b_2) and multiplied with the signal. The primary sinusoid is then also calculated and added to the signal, producing a final DAT vector (t = 1 to 365). Since the original temperature values obtained from the weather stations were recorded with an accuracy of one-tenth °C, values in the final simulated DAT vector were rounded to keep the same precision.

MODEL TESTING AND RESULTS

The model was tested via simulation of both real and hypothetical climates. Three different groups of simulations were used to judge the ability of the model to consistently imitate real weather patterns on a year by year basis. A fourth group of simulations was based on testing the capacity of the model to generate hypothetical weather patterns. First, a number of yearly simulations were run for each of the three climates and the resultant DAT vectors were plotted. These were visually inspected to ascertain the degree to which they resembled real DAT curves. Sample results of one run for the Vancouver climate are plotted in Figure 6 together with a real DAT vector.

The second group of simulations was devised to compare a set of 15 simulated vectors with a set of 15 real vectors. This was done, according to the procedure described below, for each of the three Canadian climates. To begin, 15 consecutive runs were completed for the climate of interest, resulting in 15 DAT vectors, comprising 5475 (15 years x 365 days) simulated DAT values. WAT, MAT and DDAT vectors were then calculated from these DATs. For each of them, 15 year average vectors were calculated and plotted together with the corresponding 15 year average vector from the climate being modeled. They are shown for the Winnipeg climate in Figure 7. Next,

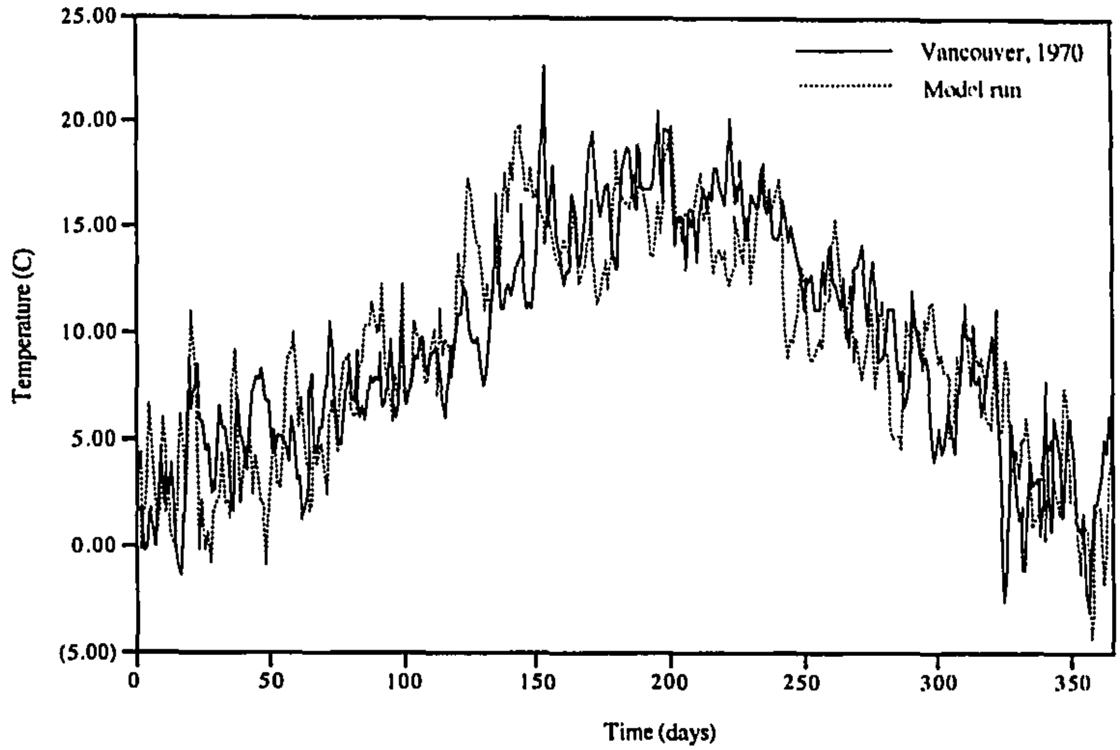


Figure 6: Sample simulated DAT vector for the Vancouver climate, plotted together with a real DAT vector.

the distributions of the DATs, WATs, MATs and DDATs were found. Histograms showing samples of these for the three different climates are shown in Figure 8. As part of the model testing procedure, each simulated DAT vector was also analyzed to determine its sinusoidal components using the same method that had been applied to the original weather data. Thus, from each of the 15 simulated DAT vectors a noise signal was extracted and the magnitudes and phase angles of these were determined. The average magnitudes from 15 runs were then compared to the 15 year average magnitudes of the modeled climates. Plots of these are shown in Figure 9 for all three climates.

Thirdly, a larger group of simulations was run to determine the overall ability of the model to produce sets of DAT vectors that exhibited the same

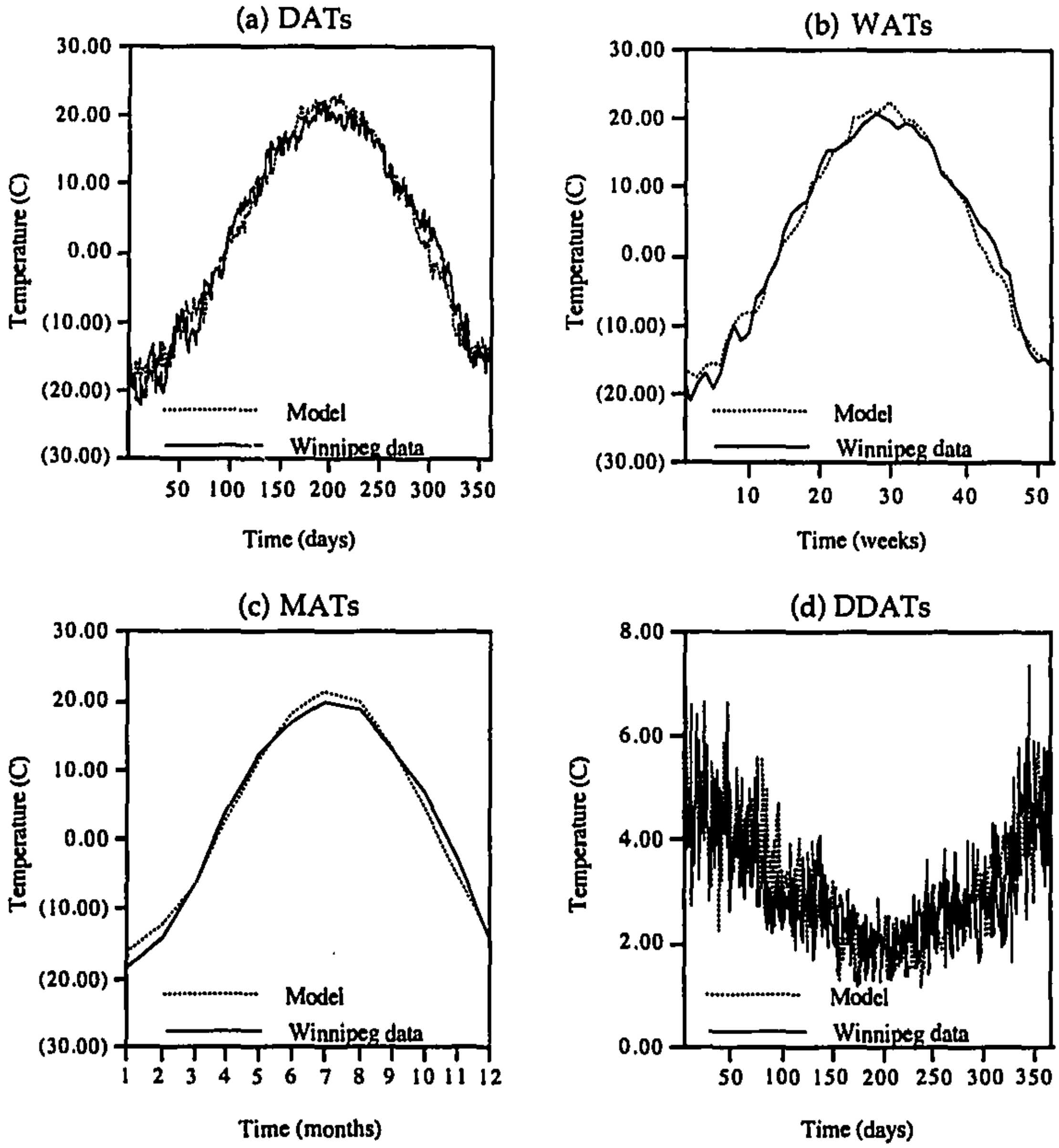


Figure 7: Average results of 15, one year runs of the model shown together with data from the Winnipeg site.

characteristics as the family of vectors being imitated. To this end, 100 sets of 15 DAT vectors were generated for each climate and the average behaviour of these was compared to the behaviour of the set of 15 vectors from the climate being imitated. Each of the 1500 simulated vectors for each climate was decomposed in the same manner as outlined above, and the parameters describing their constituent sinusoids were calculated. Average values (in brackets) are shown in Table II. For example, in Vancouver, the mean baseline temperature (a_0) that one would expect to find for a fifteen year period is 9.6858°C, whereas the model simulates Vancouver-like weather with a mean baseline of 9.6692°C. Next, the average error between the model output values and the data from the physical climates was determined

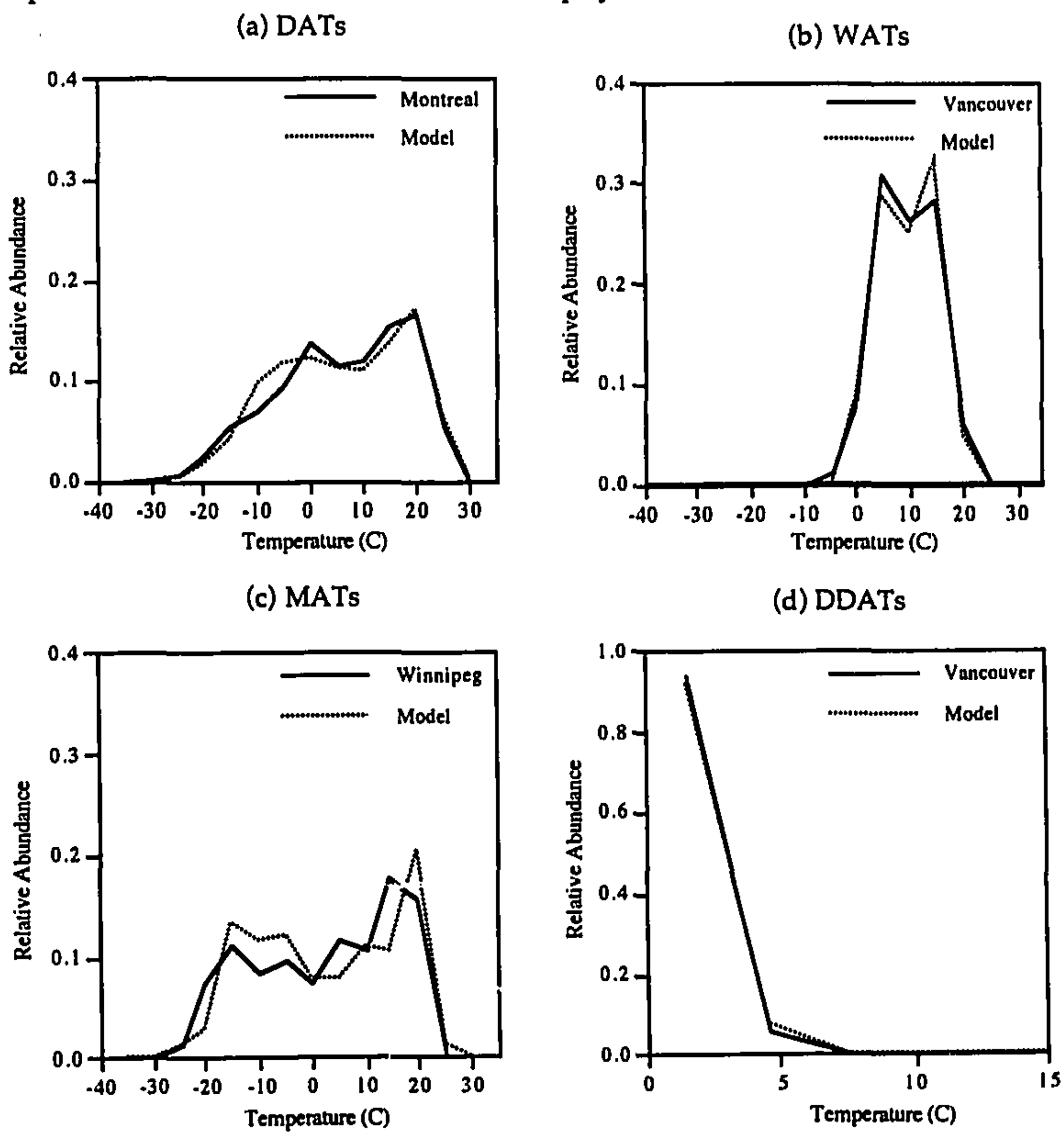


Figure 8: Sample histograms showing the relative abundance of temperatures from the model output together with temperature data from the three climates. Results are based on 15 years of simulated and real data.

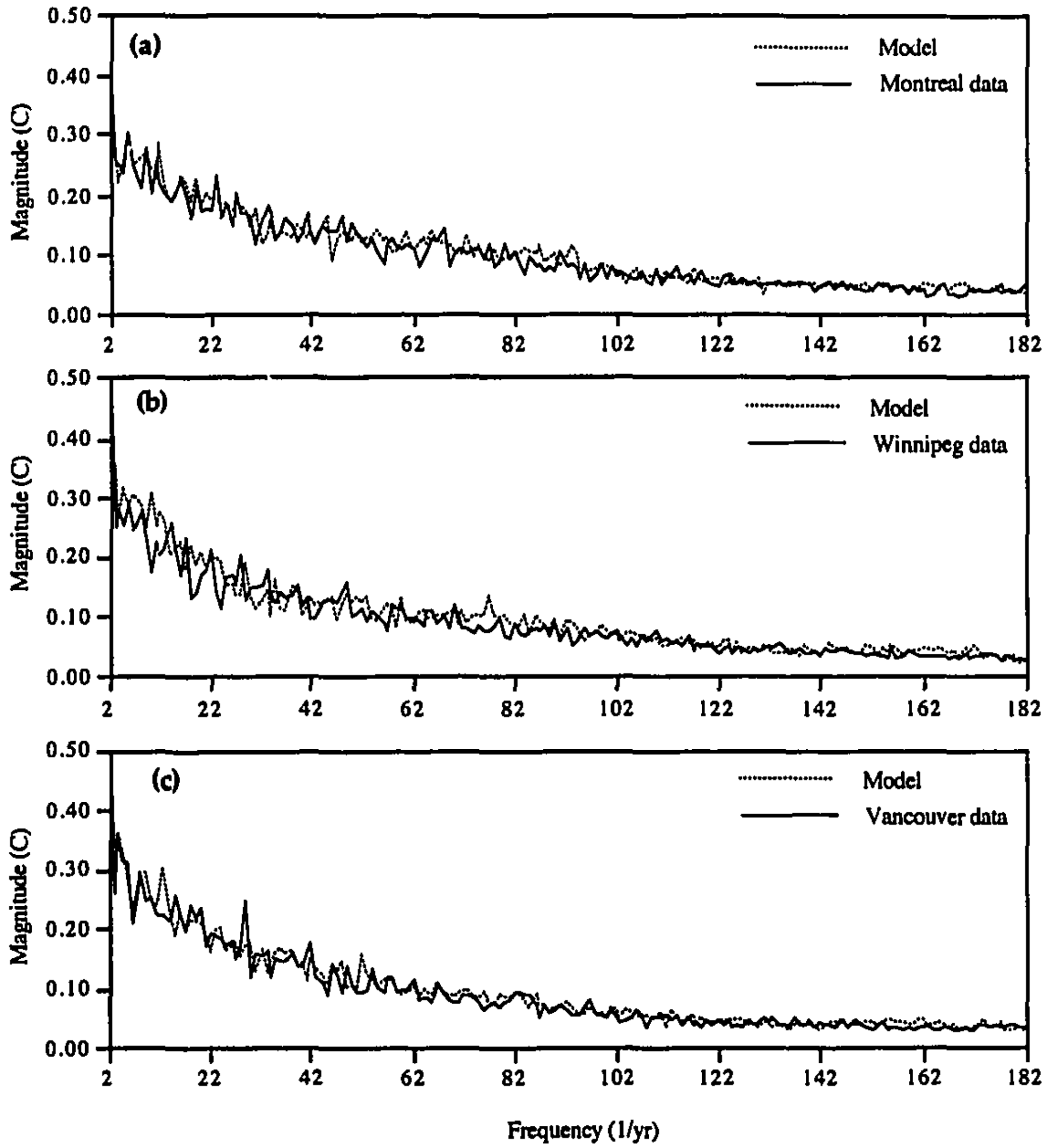


Figure 9: Comparison of magnitude versus frequency results for both the physical and modeled climates. (a) Montreal; (b) Winnipeg; (c) Vancouver

in the following manner: For each of the 100 sets of 15 DAT vectors, average DAT, WAT, MAT and DDAT vectors were calculated. The absolute differences ($|\Delta T_{DAT}|$, $|\Delta T_{WAT}|$, $|\Delta T_{MAT}|$ and $|\Delta T_{DDAT}|$) between the simulated values and the physical values were then found and for each of these, the mean, the median, and the maximum values (mean$|\Delta T_{DAT}|$, median $|\Delta T_{DAT}|$, max $|\Delta T_{DAT}|$; mean $|\Delta T_{WAT}|$, etc.) were calculated. Thus these 12 measures were obtained for each of the 100, 15 year runs for each climate. Lastly, the 100-run averages of these values were then calculated. They are shown in Table IV. Based on this comparison, the model simulates Montreal DAT values, which, for any given 15 year period, will have an average median error of 1.2°C per day when compared with 15 year average values for the real climate.

Table IV: Differences between physical and simulated data vectors. (Error values in °C.)

		Montreal	Winnipeg	Vancouver
DAT	Mean error	1.5	2.1	1.0
	Median error	1.2	1.7	0.9
	Max error	7.6	8.5	3.7
WAT	Mean error	1.2	1.8	0.9
	Median error	0.9	1.4	0.8
	Max error	4.8	5.8	2.7
MAT	Mean error	0.8	1.4	0.7
	Median error	0.6	1.2	0.6
	Max error	2.4	3.4	1.5
DDAT	Mean error	0.7	0.8	0.3
	Median error	0.6	0.6	0.3
	Max error	3.5	3.7	1.4

The fourth phase of testing involved the use of the model to produce hypothetical DAT values that might be required in, for example, the simulation of an artificial ecosystem. In this case, three sets of synthetic parameter values were input to the model to produce three completely different climates. Sample yearly runs for each of these are shown in Figure 10. The generation of these required the input of means and standard deviations for each of the nine principal parameter values (Table V) as well as coefficients describing the polynomial curves used to generate magnitudes for the noise signal (Table V).

Table V: Parameters used to create three hypothetical climates.

Parameter		Climate A	Climate B	Climate C
a_0	mean	24.5	-18.0	19.0
	std	2.0	0.6	2.0
a_1	mean	6.0	16.0	6.0
	std	0.5	0.3	0.9
a_2	mean	0.8	-1.9	-1.8
	std	0.1	0.04	0.1
b_0	mean	0.8	0.9	1.2
	std	0.3	0.3	0.1
b_1	mean	0.4	0.5	0.4
	std	0.2	0.2	0.2
b_2	mean	-2.2	1.3	1.6
	std	0.1	0.1	0.2
c_0	mean	0.000	0.003	-0.002
	std	0.100	0.200	0.010
c_1	mean	0.020	0.030	0.070
	std	0.004	0.014	0.030
c_2	mean	1.2	1.3	1.0
	std	1.0	0.6	2.6

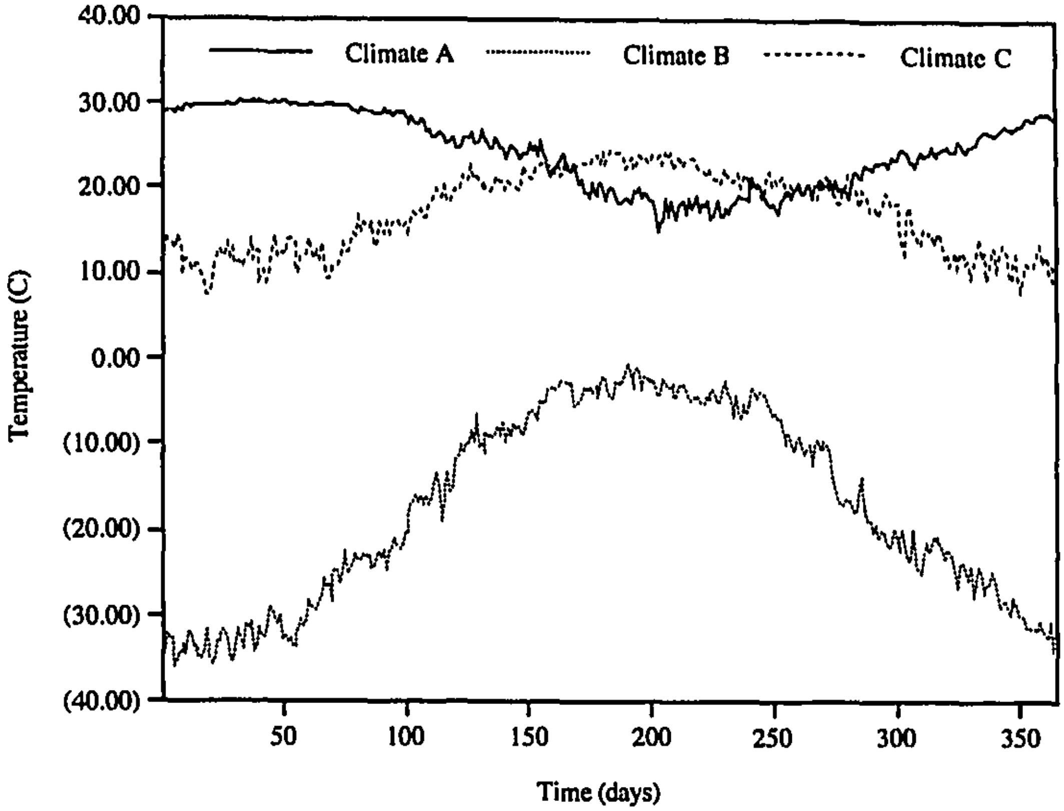


Figure 10: Sample yearly runs for three hypothetical climates.

DISCUSSION AND CONCLUSIONS

With regards to the overall goal stated at the outset, which was to produce weather that was representative of a given climate, the model appears to perform fairly well. The average DAT, WAT, MAT and DDAT vectors produced by the component are similar to those of the imitated climate within reasonable limits. In addition, the histograms of these values are also similar to those for the climates being modeled. Analysis of the simulated DAT vectors shows that they are composed of principal sinusoids that have magnitudes, baselines, and phase angles similar to those of the physical climates (Table II). For all three locations, the simulated weather has a slightly warmer growing season than that of the real climate. This is most marked for the Winnipeg climate, for which the simulation produces weather which is, on average, 1°C higher than the physical data for each day over the summer months. This discrepancy could be due to the assumptions made in the model regarding the distribution of parameters. In general, however, the model output is acceptably close to the real weather data.

To improve the model, there a few things that might be addressed. The first concerns the generation of principal sine wave parameter values, which are presently chosen independently. As shown in the correlation analysis (Table III), there are some significant relationships between these that could be included in the model. In addition, we suspect that there is an underlying relation within the phase angles of the noise signal which we were unable to discern. The tendency of the model to produce slightly warmer, less irregular, winter weather for the continental climates is likely attributable to a lack of correlation between the simulated phase angles.

The Fast Fourier Transform approach proved to be an appropriate, and conceptually simple technique to employ in the modeling process. It enabled an effective description of the sinusoidal patterns found in physical temperature data, providing a flexible method for the simulation of any climate. The model was easy to use to produce both real and hypothetical weather, and will serve as an input component to the ecosystem models that we are developing.

REFERENCES

Amato, U., A. Andretta, B. Bartoli, B. Coluzzi, V. Cuomo, F. Fontana and C. Serio. 1986. Markov processes and Fourier analysis as a tool to describe and simulate daily solar irradiance. *Solar Energy* 37(3):179-194.

Boileau, E. 1983. Use of some simple statistical models in solar meteorology. *Solar Energy* 30(4):333-3

Casti, John L. 1992. Reality Rules: Picturing the World in Mathematics (I). New York, N. Y.: John Wiley & Sons. pp. 388.

Cook, D.F. and M.L. Wolfe. 1991. A back-propagation neural network to predict average air temperatures. *AI Applications* 5(1):40-46.

Julian, P.R. 1967. Variance spectrum analysis. *Water Resources Research* 3(3):831-845.

Kline, D.E., F.E. Woeste, V.O. Shanholtz and J.F. Reid. 1985. Co-generation of hourly dry-bulb and wet-bulb temperatures. *Transactions of the ASAE* 28(3):875-880.

Kok, R., R. Lacroix, G. Clark and E. Taillefer. 1994. Imitation of a Procedural Greenhouse Model with an Artificial Neural Network. *Canadian Agricultural Engineering* 36(2): 117-126.

Matalas, N.C. 1967. Time series analysis. *Water Resources Research* 3(3):817-829.

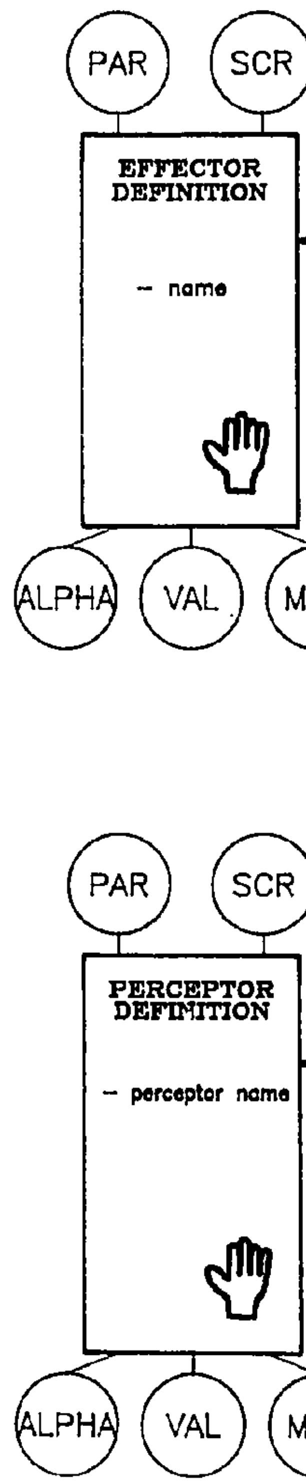
Phillips, W.F. 1984. Harmonic analysis of climatic data. *Solar Energy* 32(3):319-328.

Richardson, C.W. 1981. Stochastic simulation of daily precipitation, temperature and solar radiation. *Water Res. Res.* 17(1):182-190.

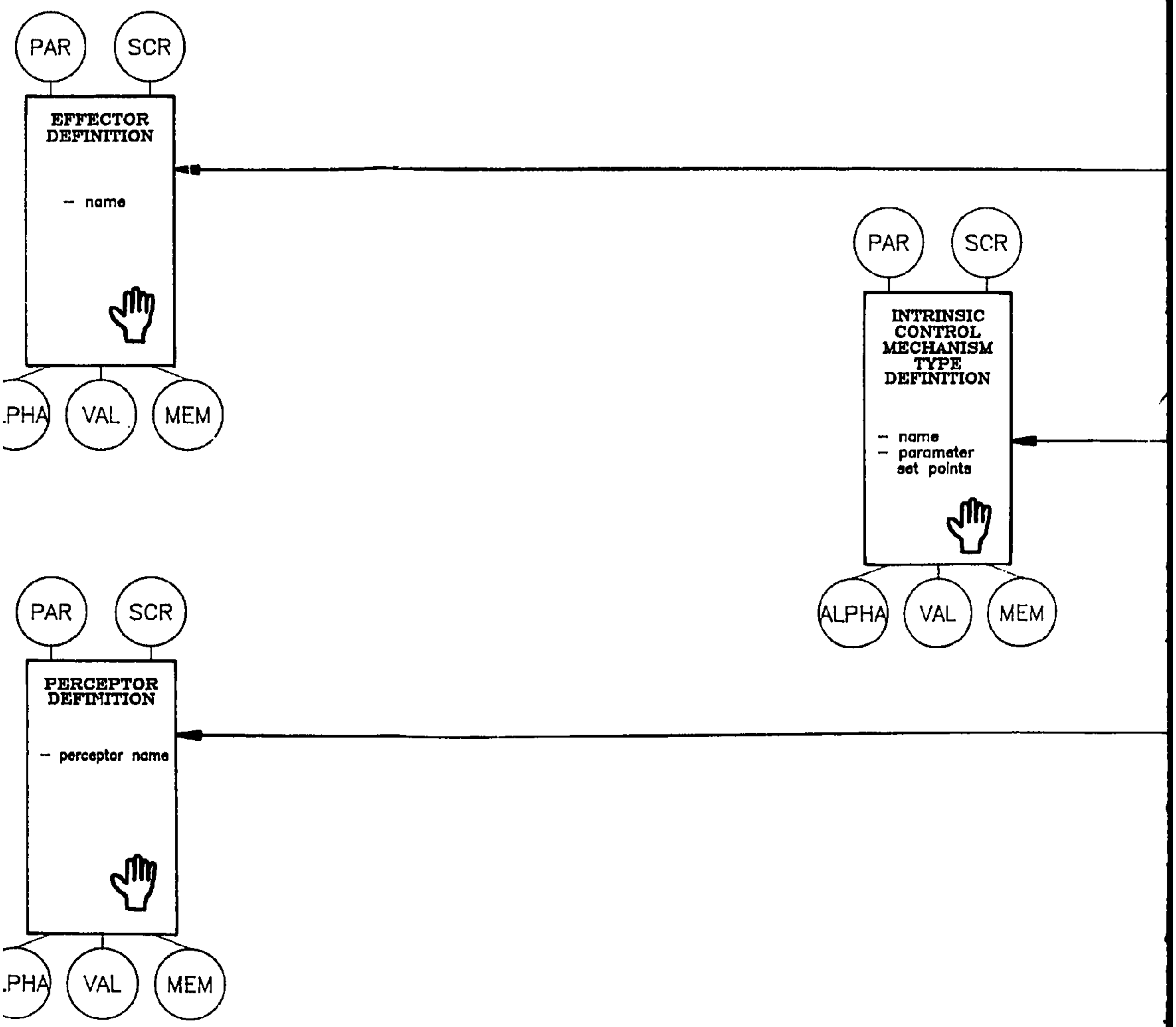
Saupe, D. 1988. Algorithms for random fractals. In *The Science of Fractal Images*, ed. H. Peitgen and D. Saupe. New York, N.Y.: Springer Verlag.

West, B.J. and M. Schlesinger. 1990. The noise in natural phenomena. *American Scientist* 78(1):40-45.

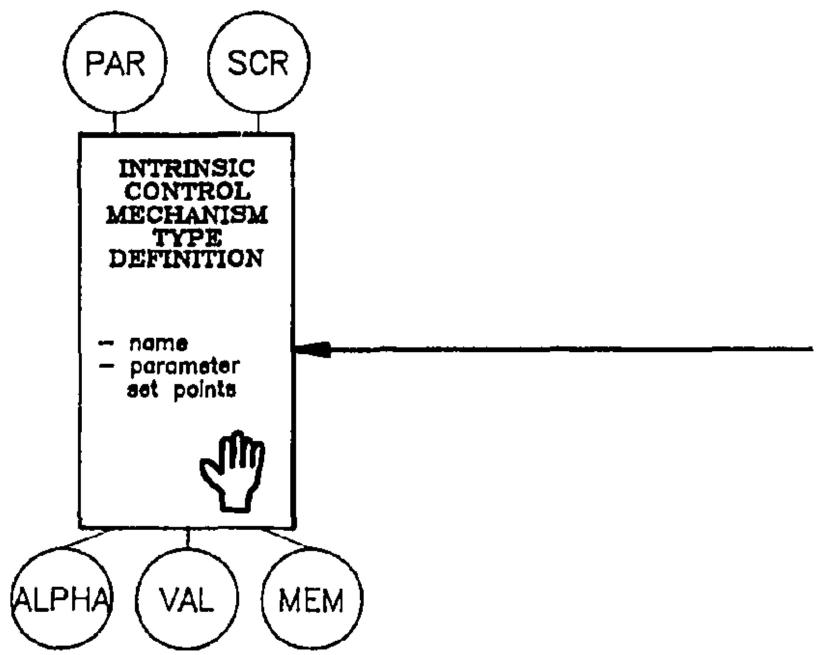
DEFINITION OF E



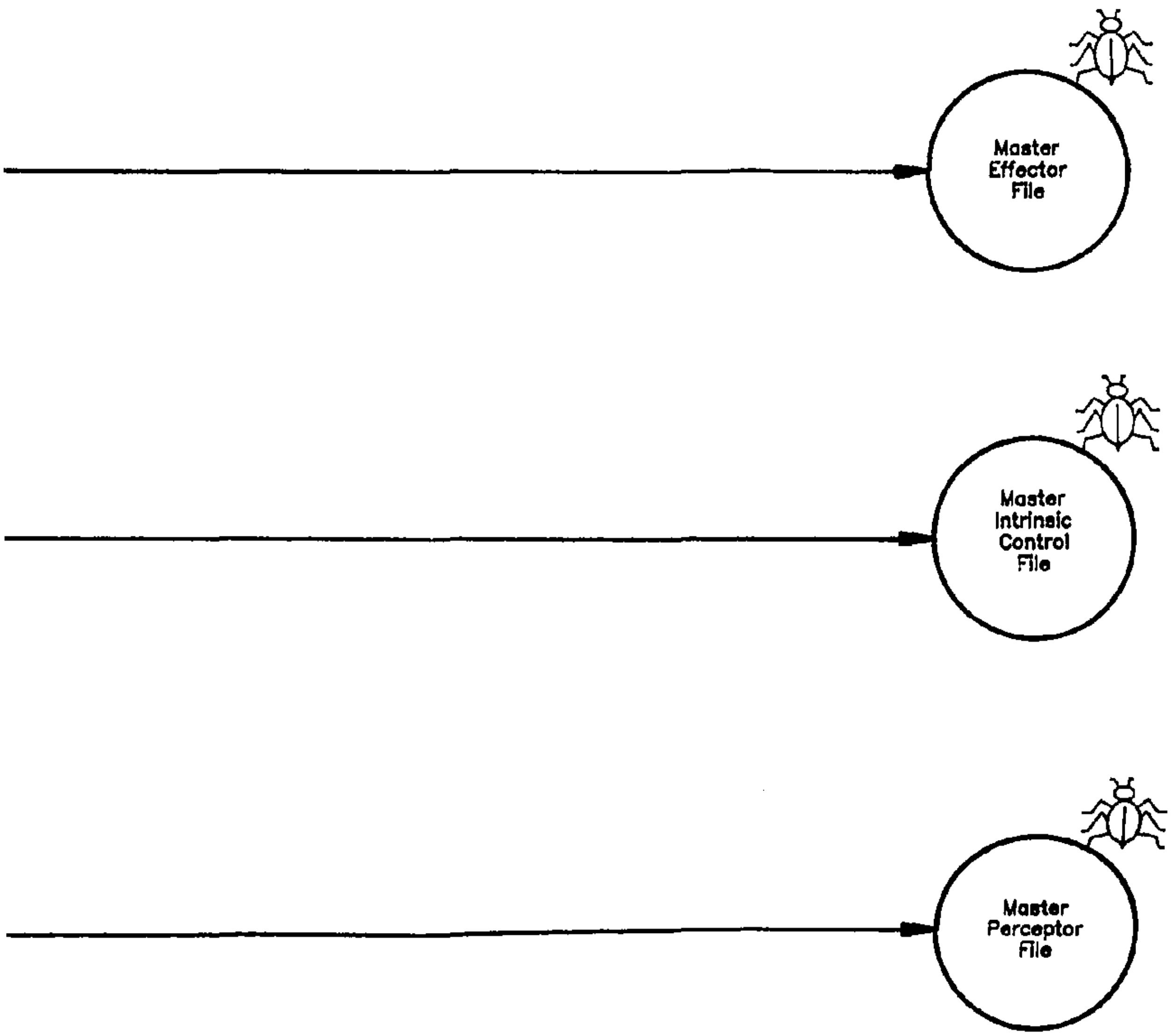
I OF EFFECTOR, PERCEPTOR AND INTRIN



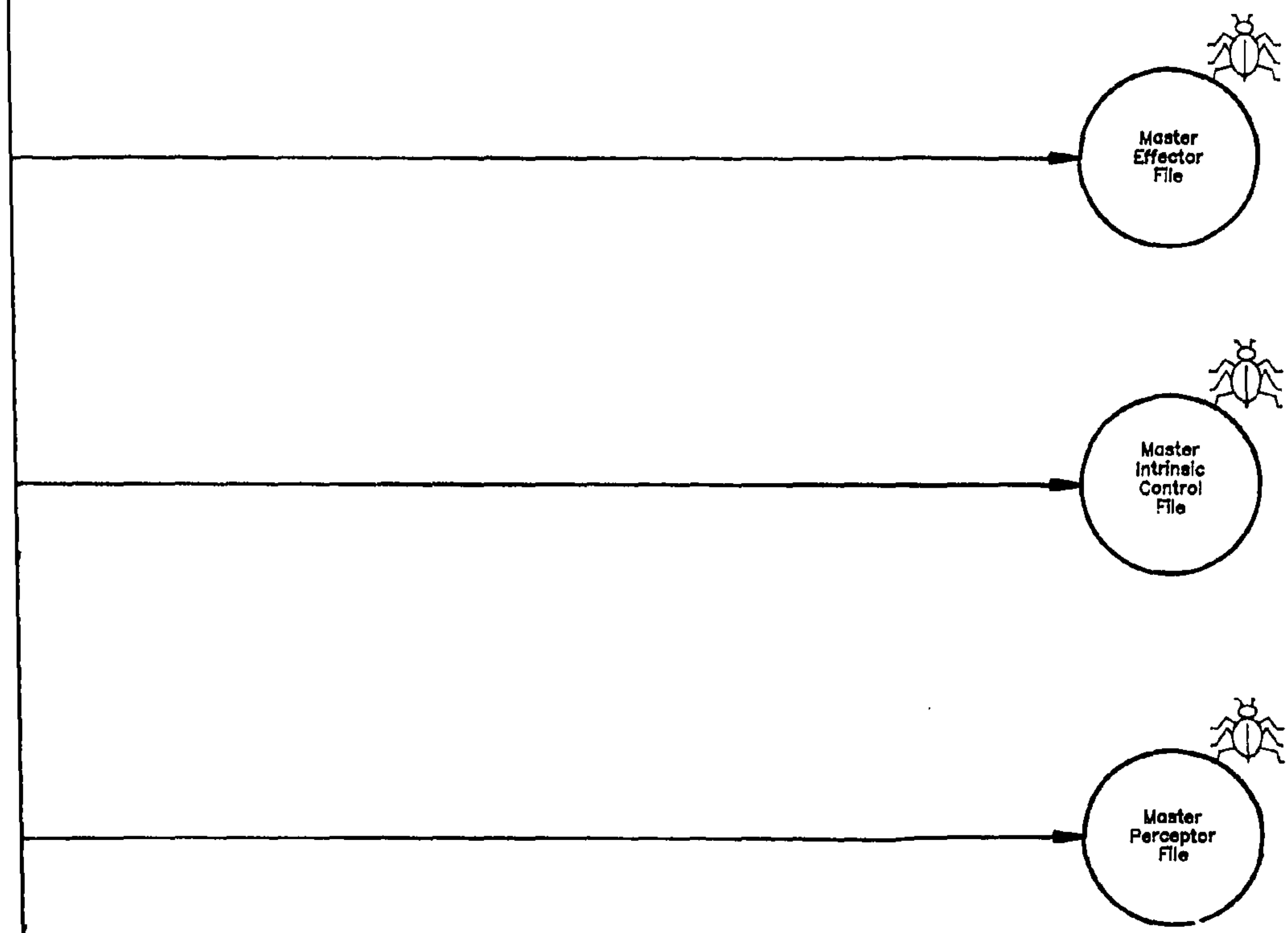
TOR, PERCEPTOR AND INTRINSIC COI

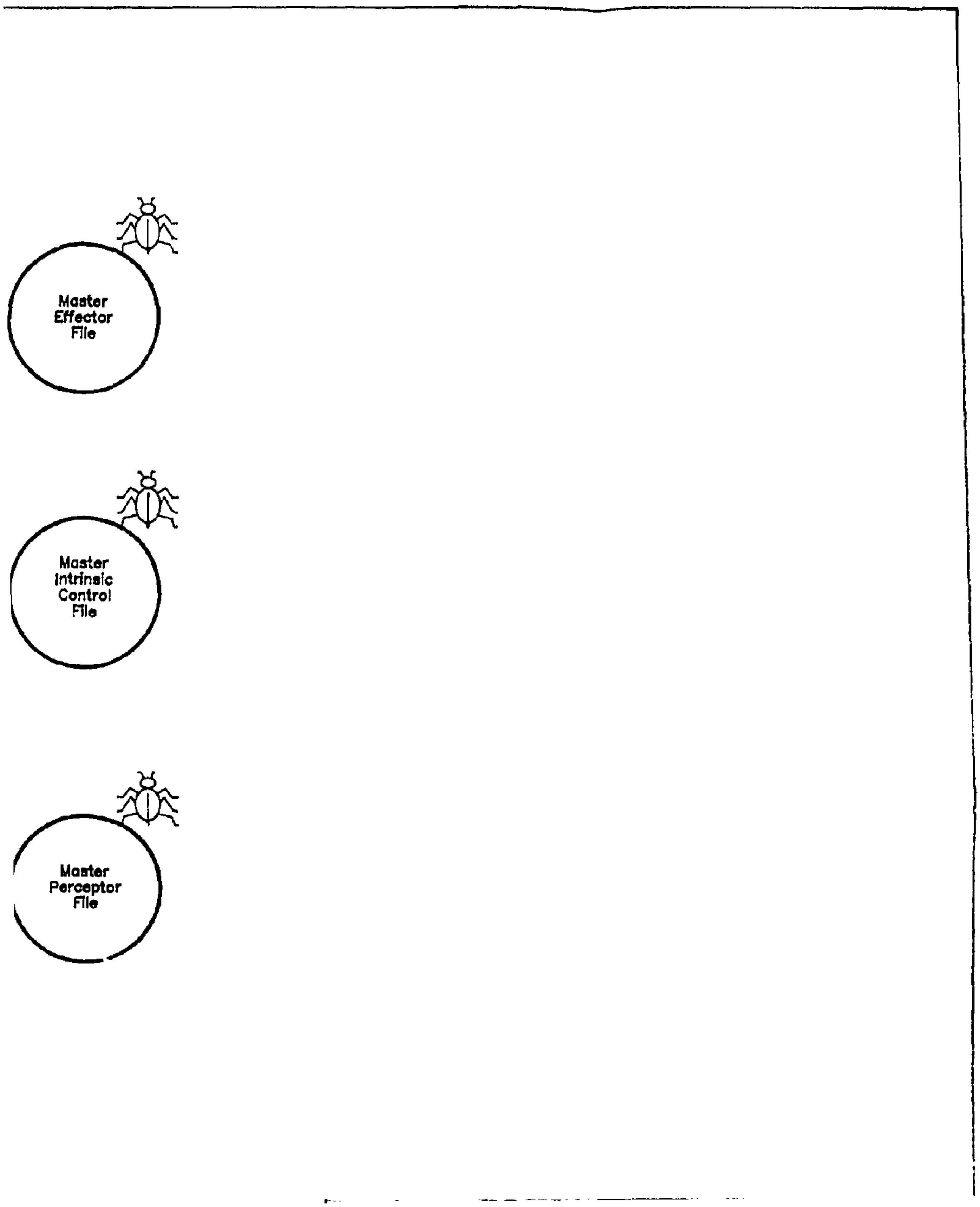


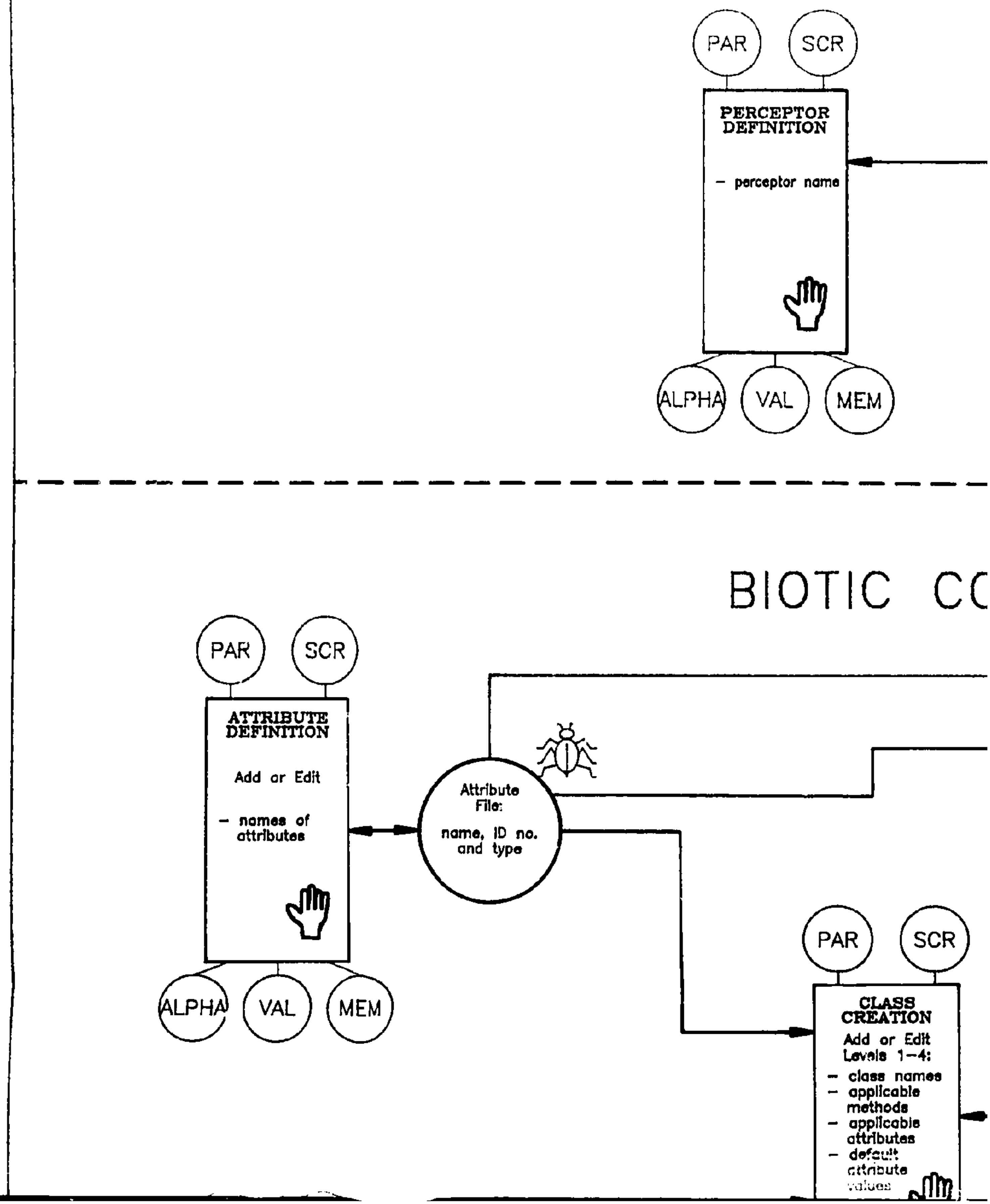
CONTROL MECHANISMS



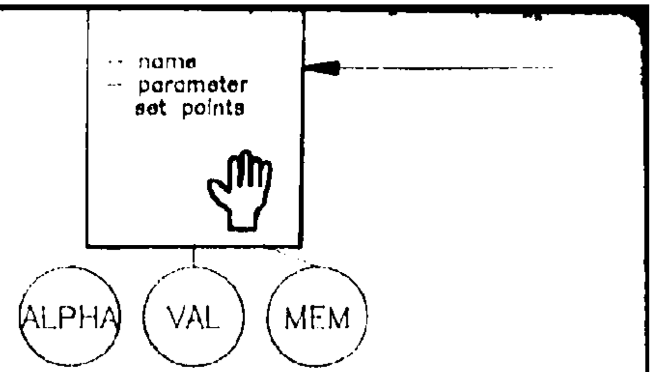
RINSIC CONTROL MECHANISMS



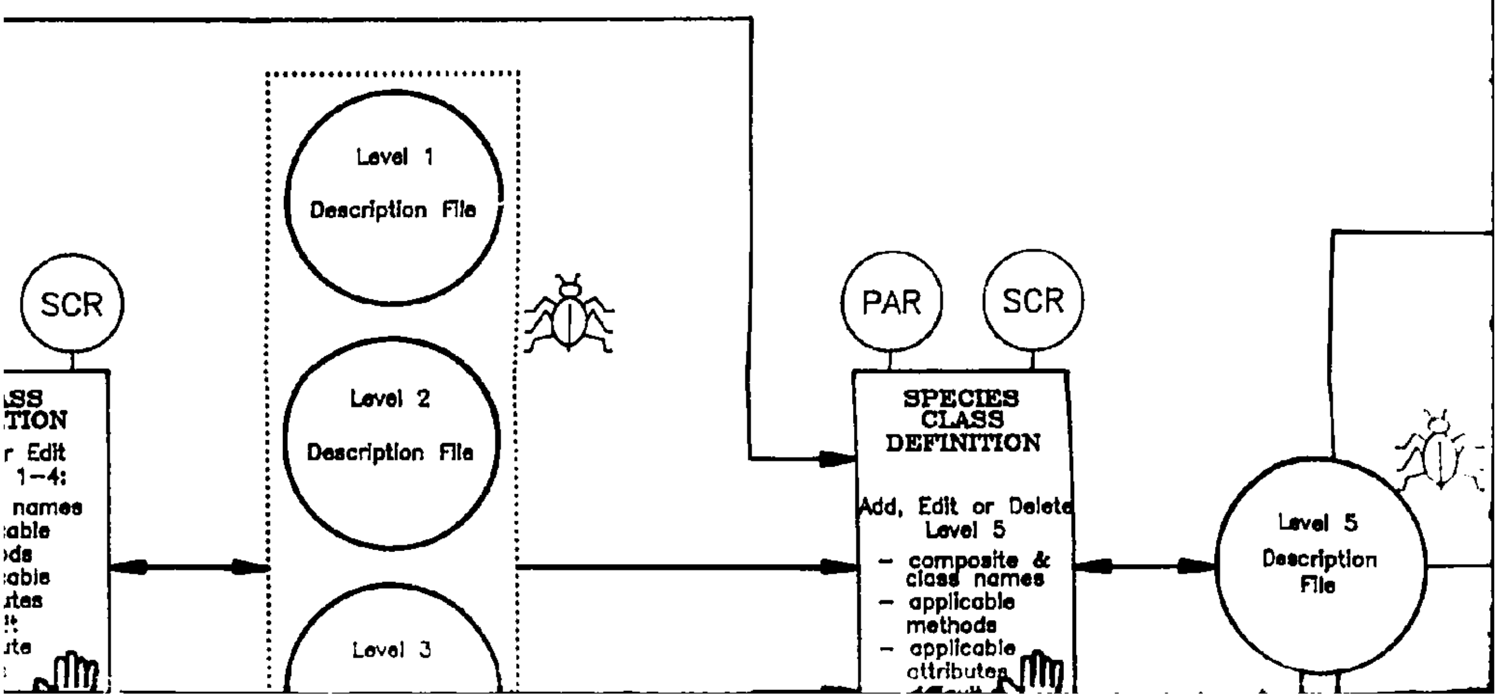

Master Effector File
Master Intrinsic Control File
Master Perceptor File

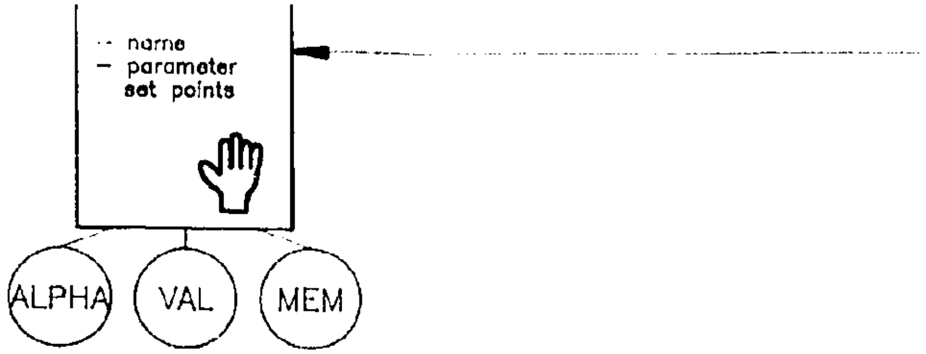


PAR
SCR
PERCEPTOR DEFINITION
– perceptor name
ALPHA
VAL
MEM
BIOTIC CC
PAR
SCR
ATTRIBUTE DEFINITION
Add or Edit
– names of attributes
ALPHA
VAL
MEM
Attribute File:
name, ID no. and type
PAR
SCR
CLASS CREATION
Add or Edit Levels 1–4:
– class names
– applicable methods
– applicable attributes
– default attribute values

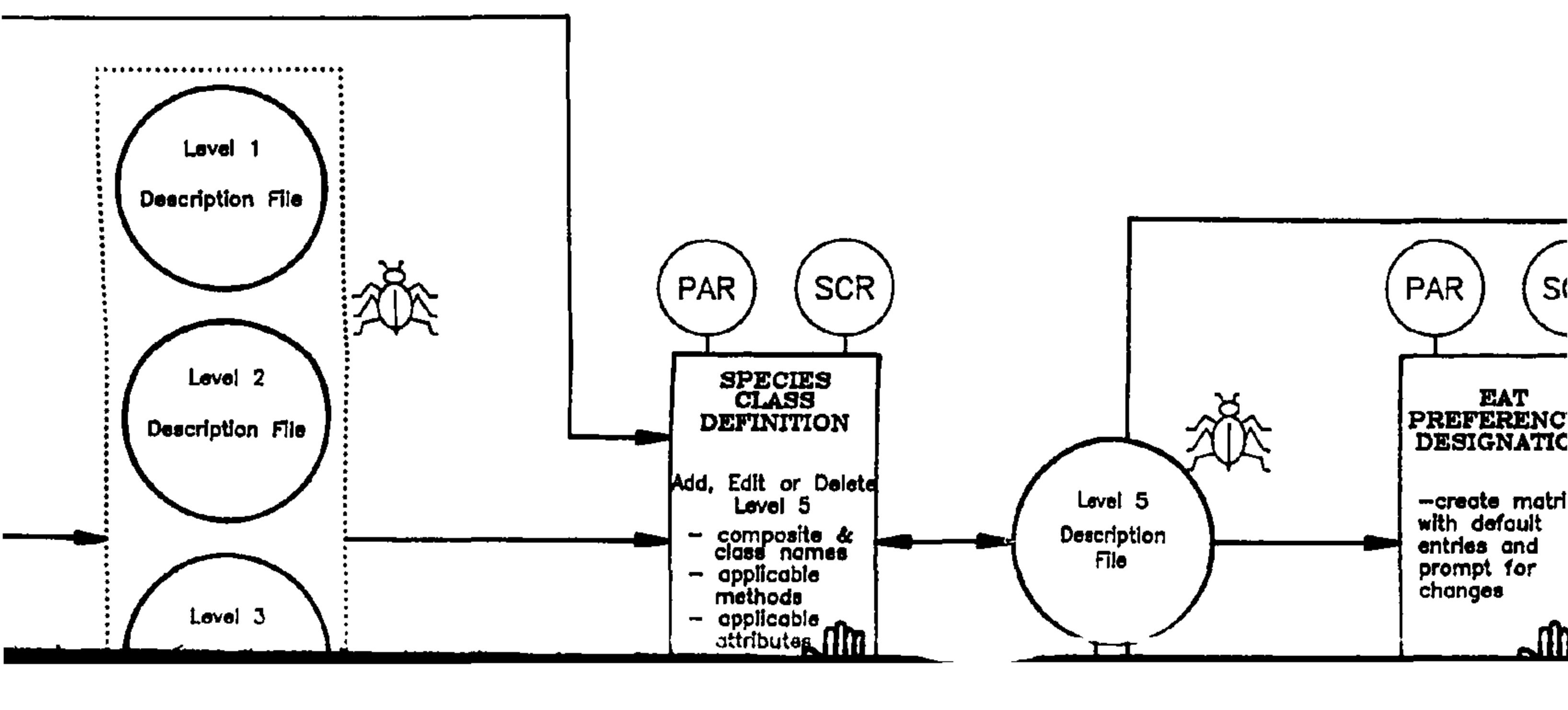


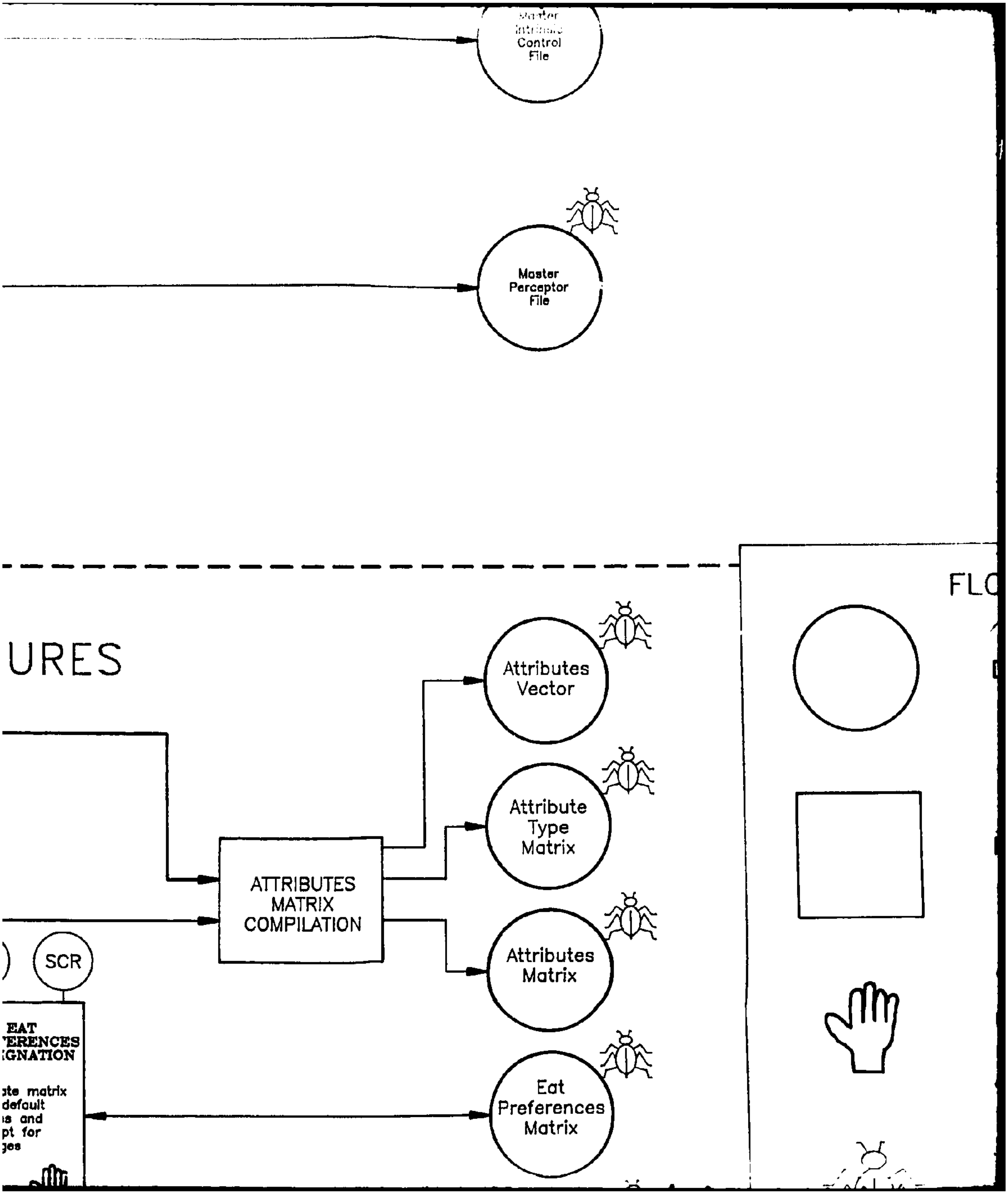
COMPONENT CLASS DEFINITION PR

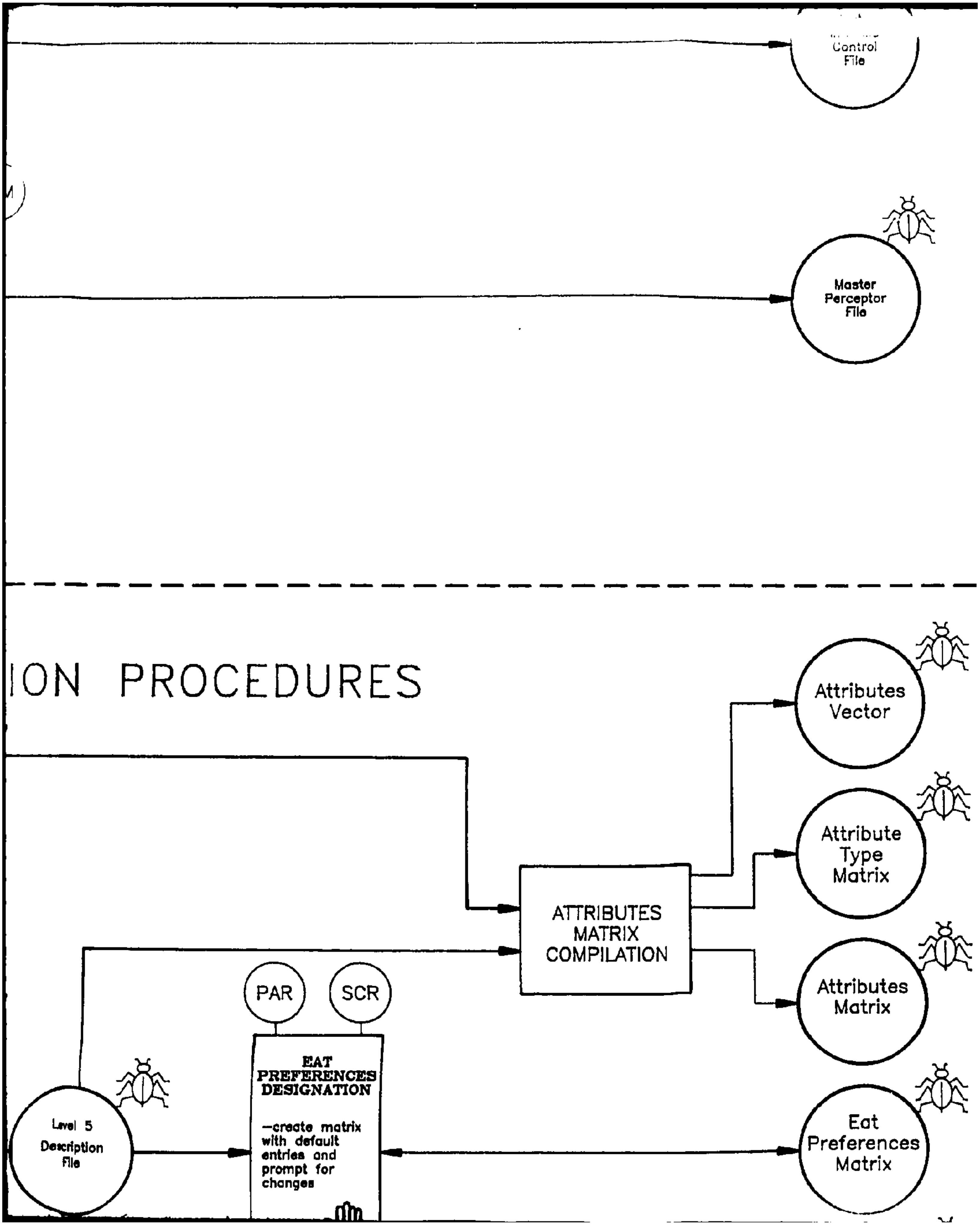


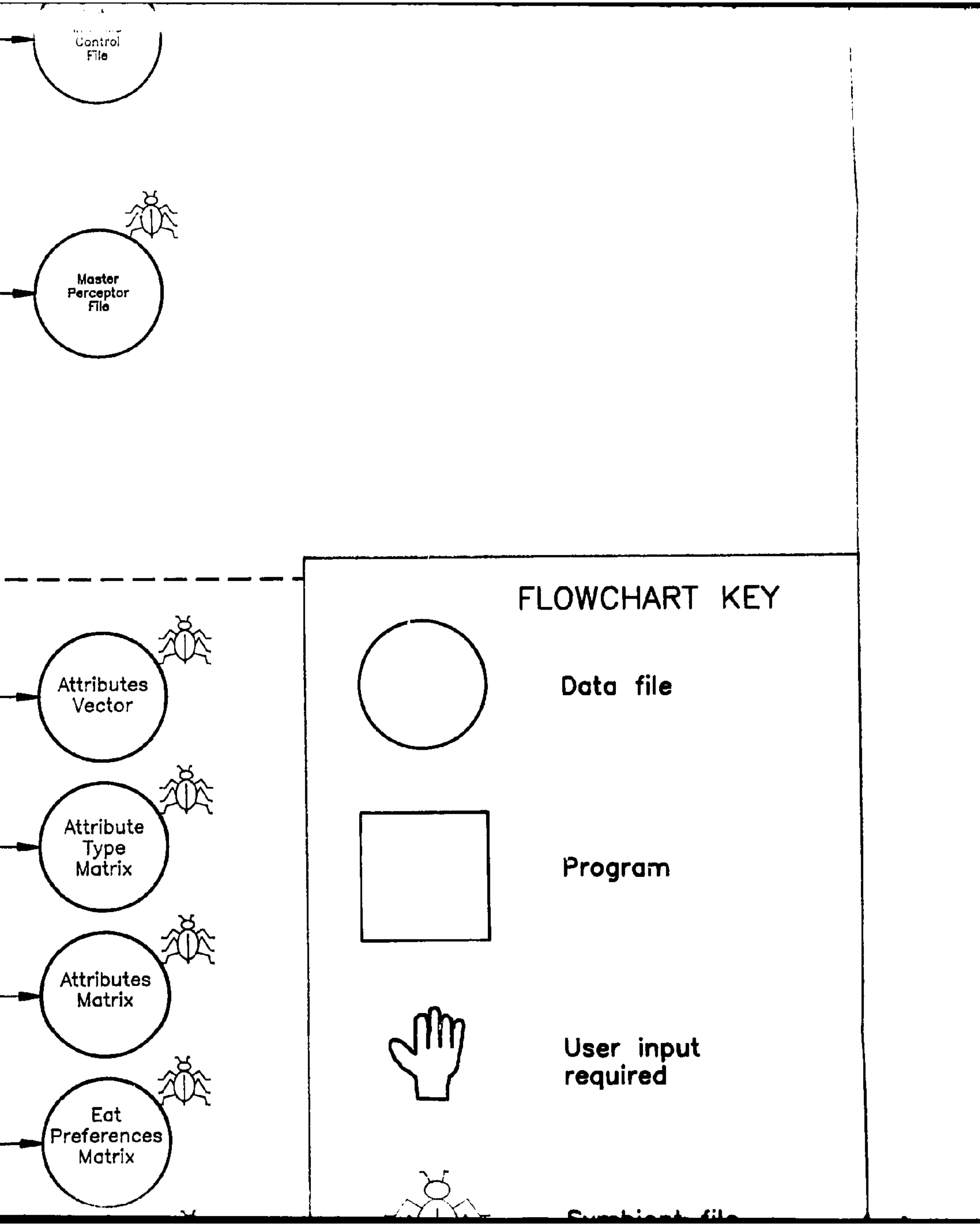


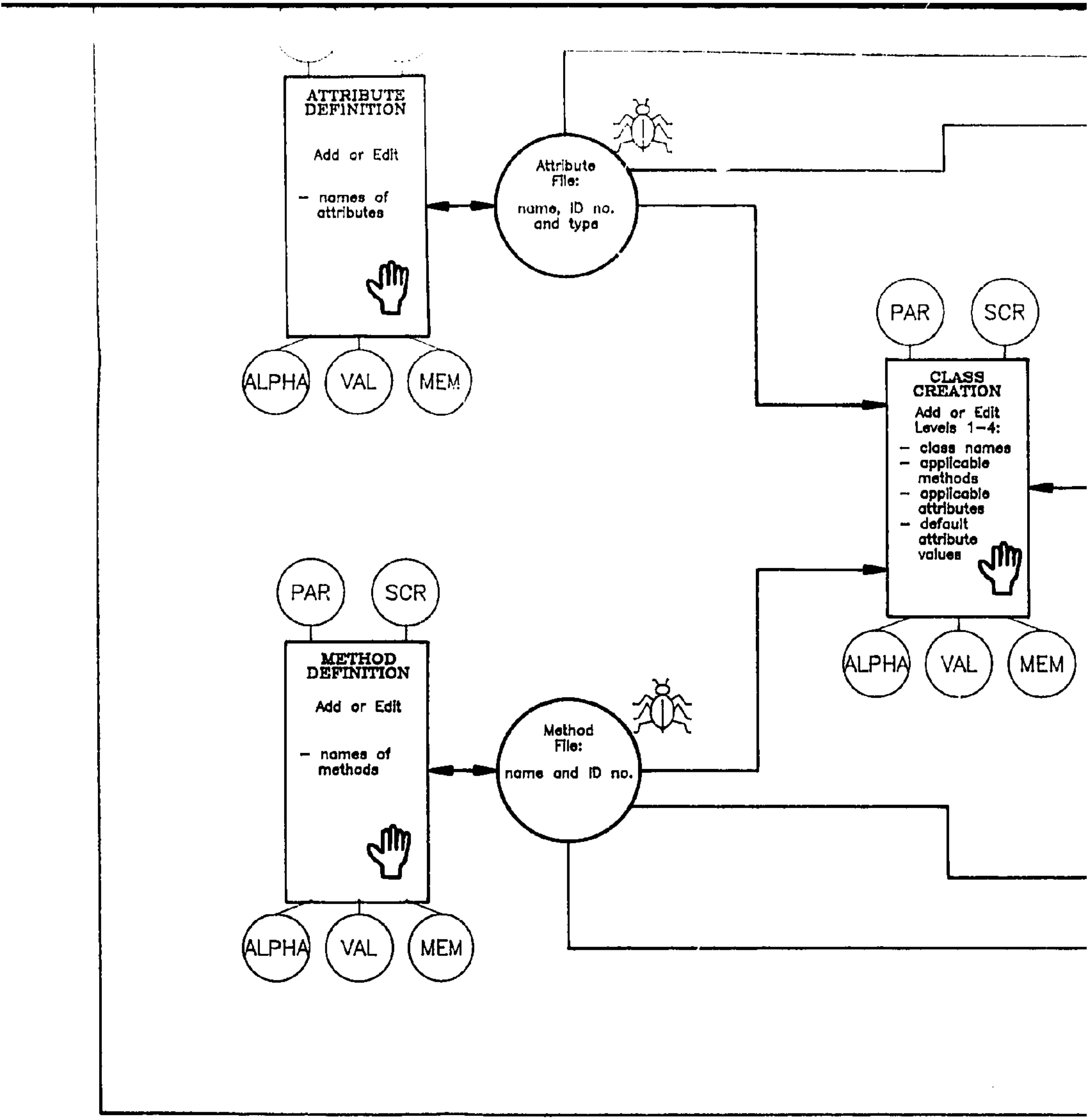
MPONENT CLASS DEFINITION PROCEDUR

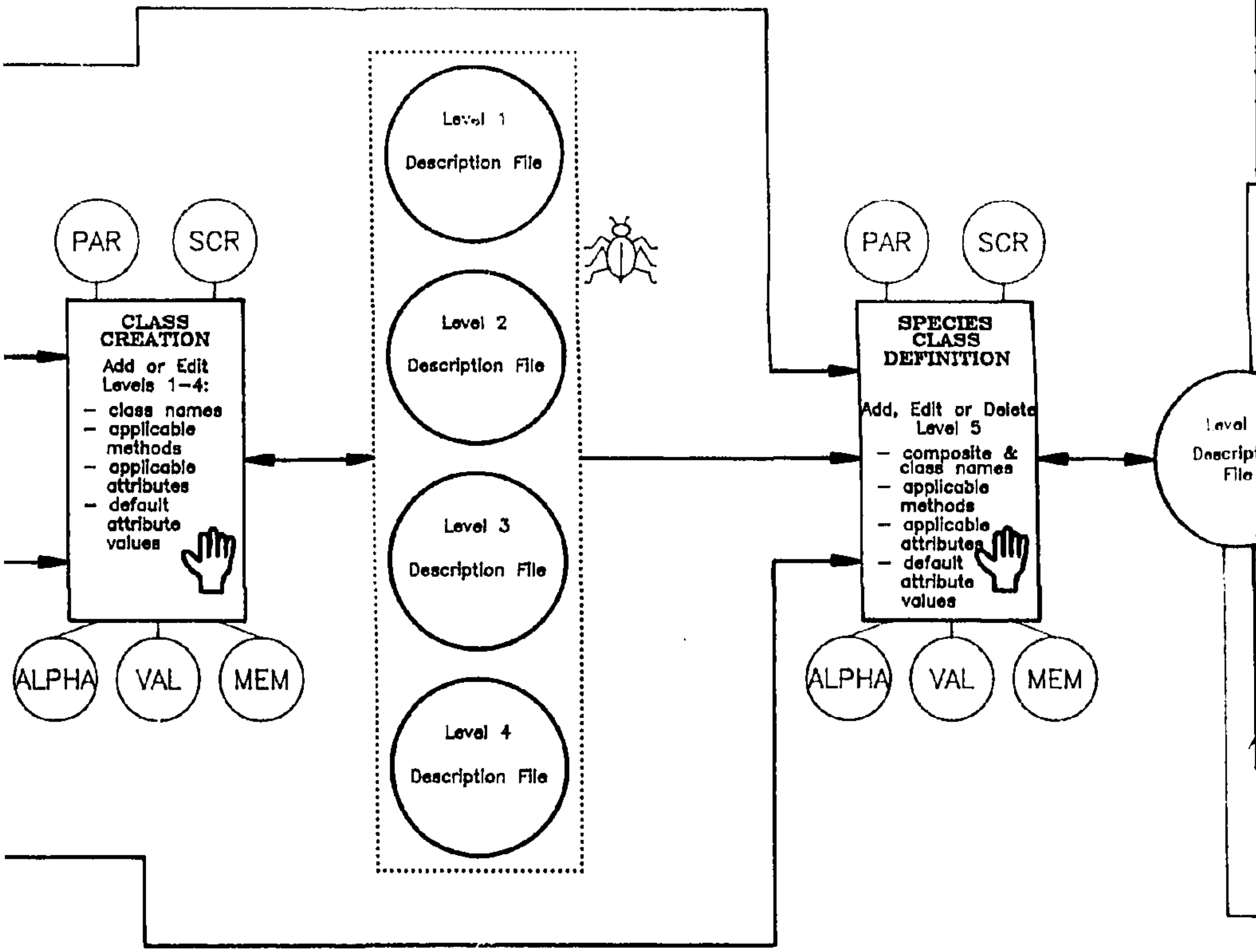



Control
File
Master
Perceptor
File
URES
FL
Attributes
Vector
Attribute
Type
Matrix
ATTRIBUTES
MATRIX
COMPILATION
Attributes
Matrix
SCR
EAT
ERENCES
GNATION
te matrix
default
s and
pt for
ges
Eat
Preferences
Matrix

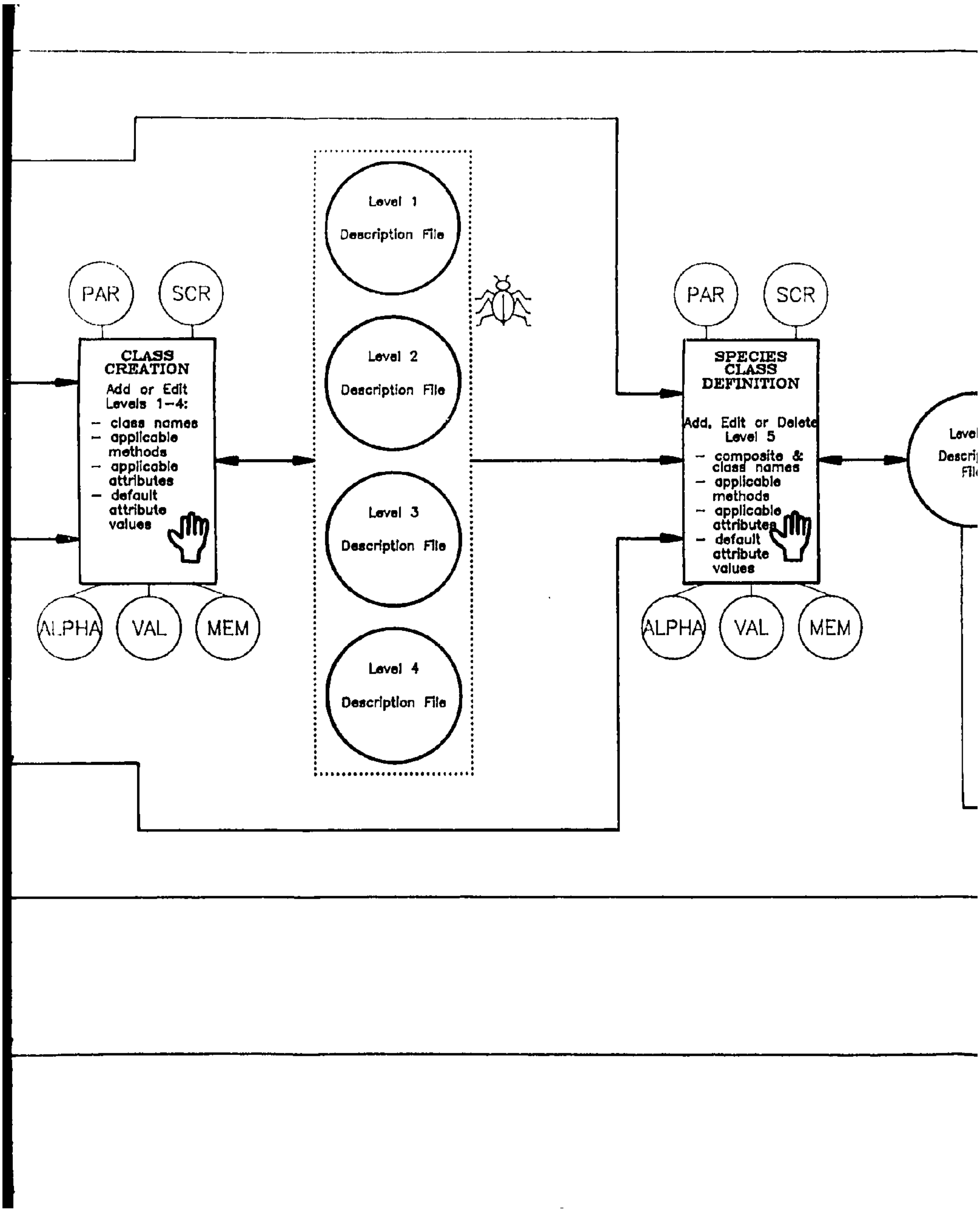

Control
File
Master
Perceptor
File
ION PROCEDURES
Attributes
Vector
Attribute
Type
Matrix
ATTRIBUTES
MATRIX
COMPILATION
Attributes
Matrix
PAR
SCR
EAT
PREFERENCES
DESIGNATION
—create matrix
with default
entries and
prompt for
changes
Level 5
Description
File
Eat
Preferences
Matrix

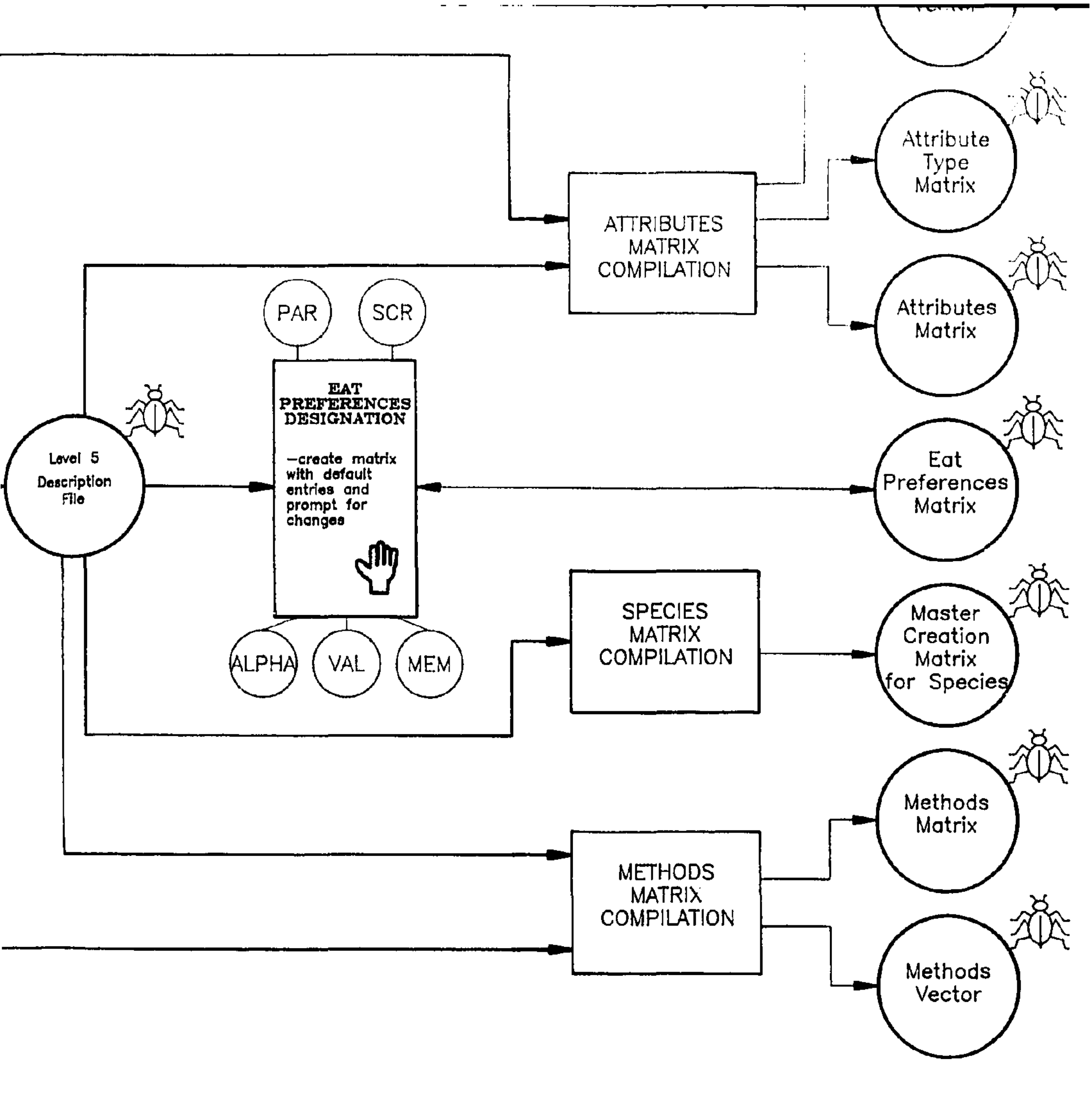

Control File
Master Perceptor File
Attributes Vector
Attribute Type Matrix
Attributes Matrix
Eat Preferences Matrix
FLOWCHART KEY
Data file
Program
User input required


ATTRIBUTE DEFINITION
Add or Edit
- names of attributes
ALPHA
VAL
MEM
Attribute File:
name, ID no. and type
PAR
SCR
CLASS CREATION
Add or Edit Levels 1-4:
- class names
- applicable methods
- applicable attributes
- default attribute values
ALPHA
VAL
MEM
PAR
SCR
METHOD DEFINITION
Add or Edit
- names of methods
Method File:
name and ID no.
ALPHA
VAL
MEM

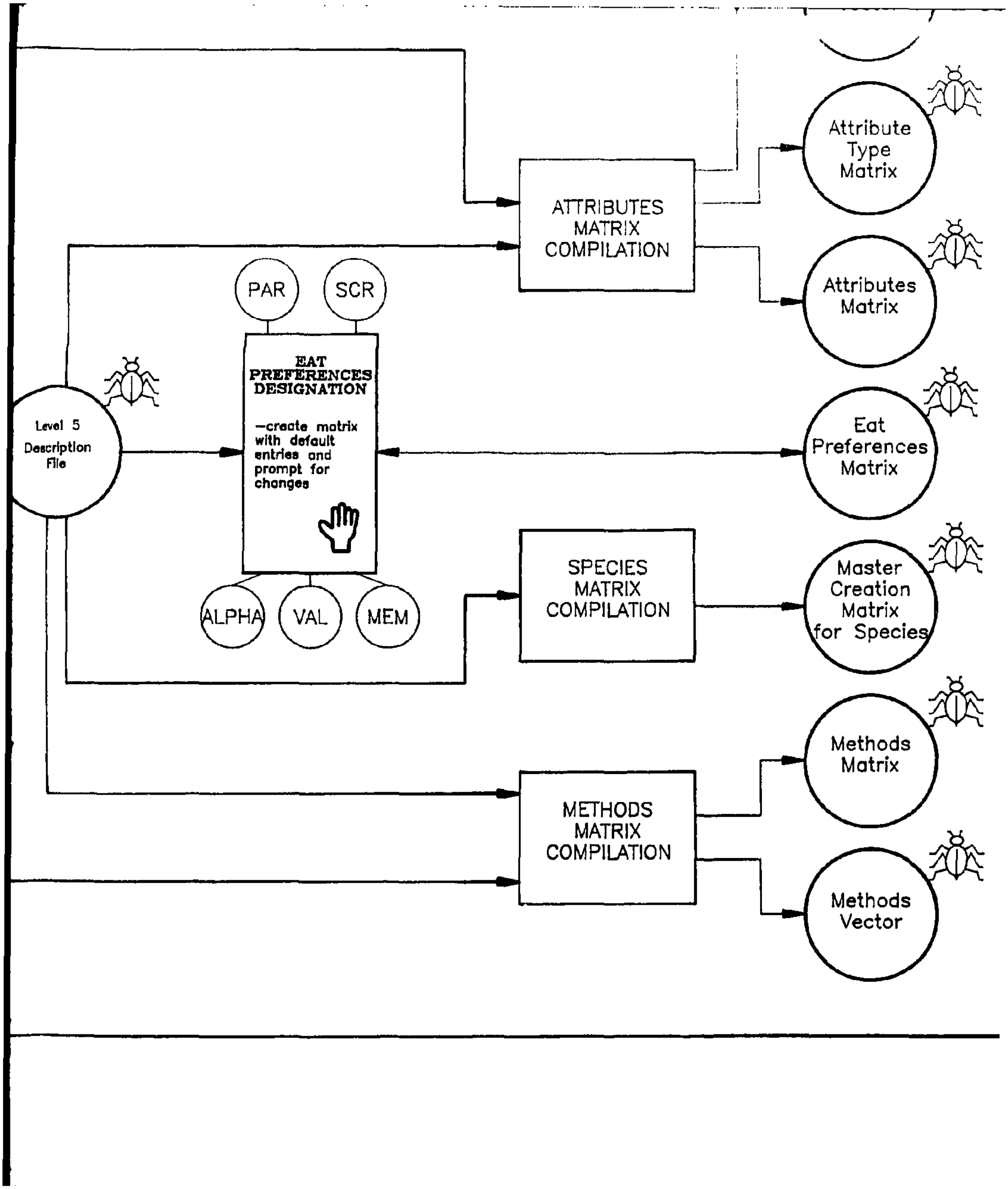


PAR
SCR
CLASS CREATION
Add or Edit Levels 1-4:
- class names
- applicable methods
- applicable attributes
- default attribute values
ALPHA
VAL
MEM
Level 1 Description File
Level 2 Description File
Level 3 Description File
Level 4 Description File
PAR
SCR
SPECIES CLASS DEFINITION
Add, Edit or Delete Level 5
- composite & class names
- applicable methods
- applicable attributes
- default attribute values
ALPHA
VAL
MEM


PAR
SCR
CLASS CREATION
Add or Edit Levels 1-4:
- class names
- applicable methods
- applicable attributes
- default attribute values
ALPHA
VAL
MEM
Level 1 Description File
Level 2 Description File
Level 3 Description File
Level 4 Description File
PAR
SCR
SPECIES CLASS DEFINITION
Add, Edit or Delete Level 5
- composite & class names
- applicable methods
- applicable attributes
- default attribute values
ALPHA
VAL
MEM



Attribute Type Matrix
ATTRIBUTES MATRIX COMPILATION
Attributes Matrix
PAR
SCR
EAT PREFERENCES DESIGNATION
—create matrix with default entries and prompt for changes
Level 5 Description File
Eat Preferences Matrix
SPECIES MATRIX COMPILATION
Master Creation Matrix for Species
ALPHA
VAL
MEM
Methods Matrix
METHODS MATRIX COMPILATION
Methods Vector



ATTRIBUTES MATRIX COMPILATION
Attribute Type Matrix
Attributes Matrix
PAR
SCR
EAT PREFERENCES DESIGNATION
—create matrix with default entries and prompt for changes
Level 5 Description File
Eat Preferences Matrix
ALPHA
VAL
MEM
SPECIES MATRIX COMPILATION
Master Creation Matrix for Species
Methods Matrix
METHODS MATRIX COMPILATION
Methods Vector

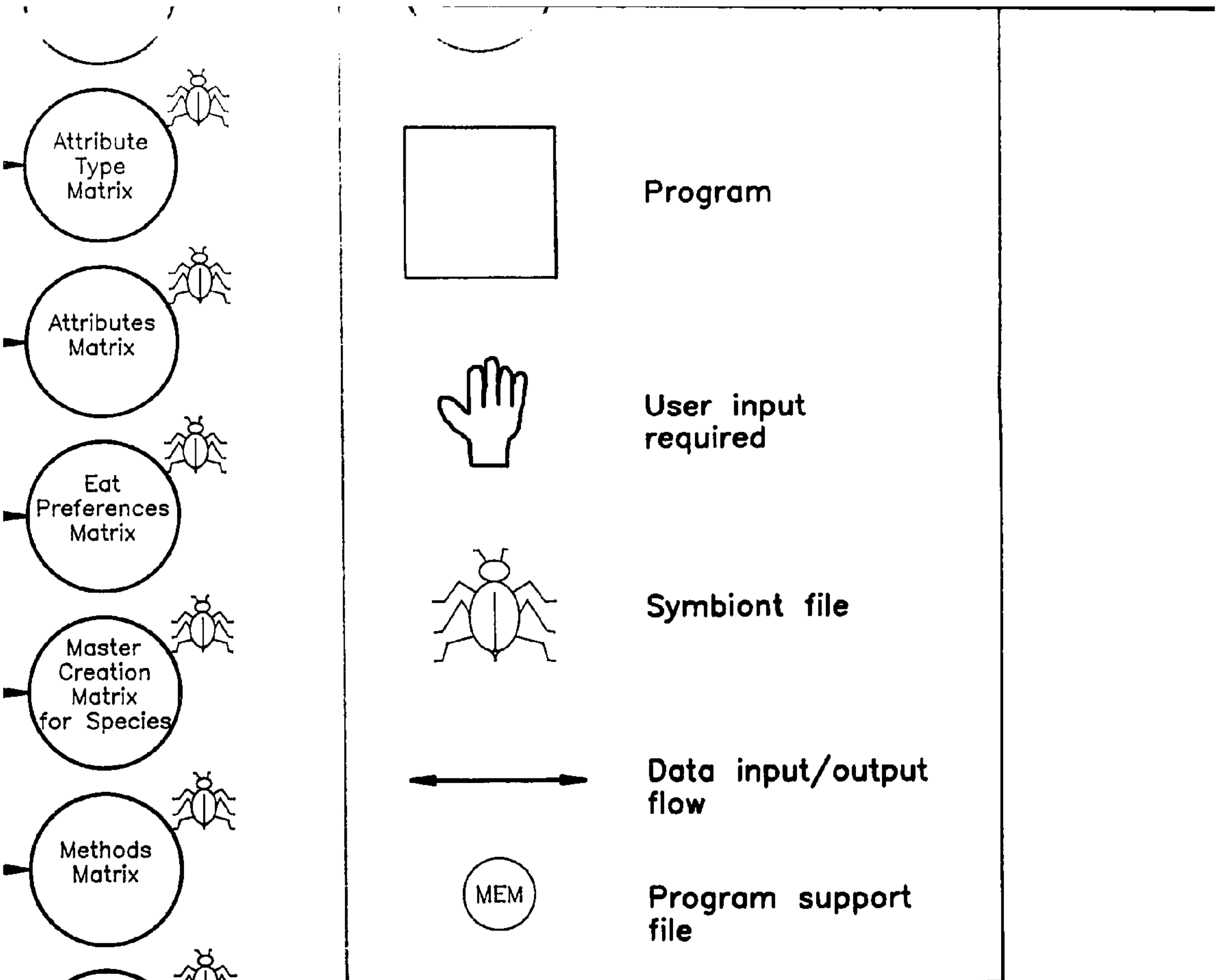
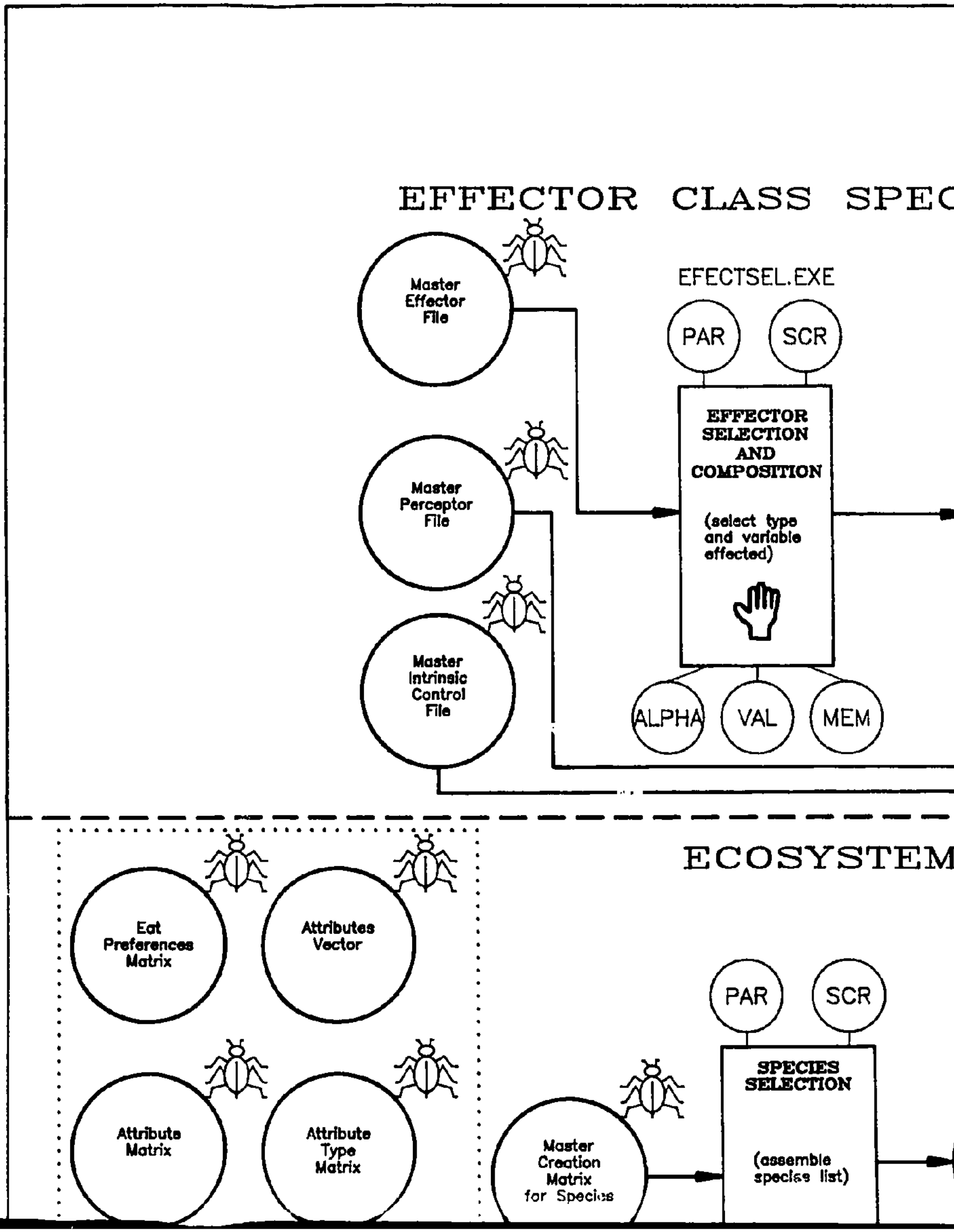
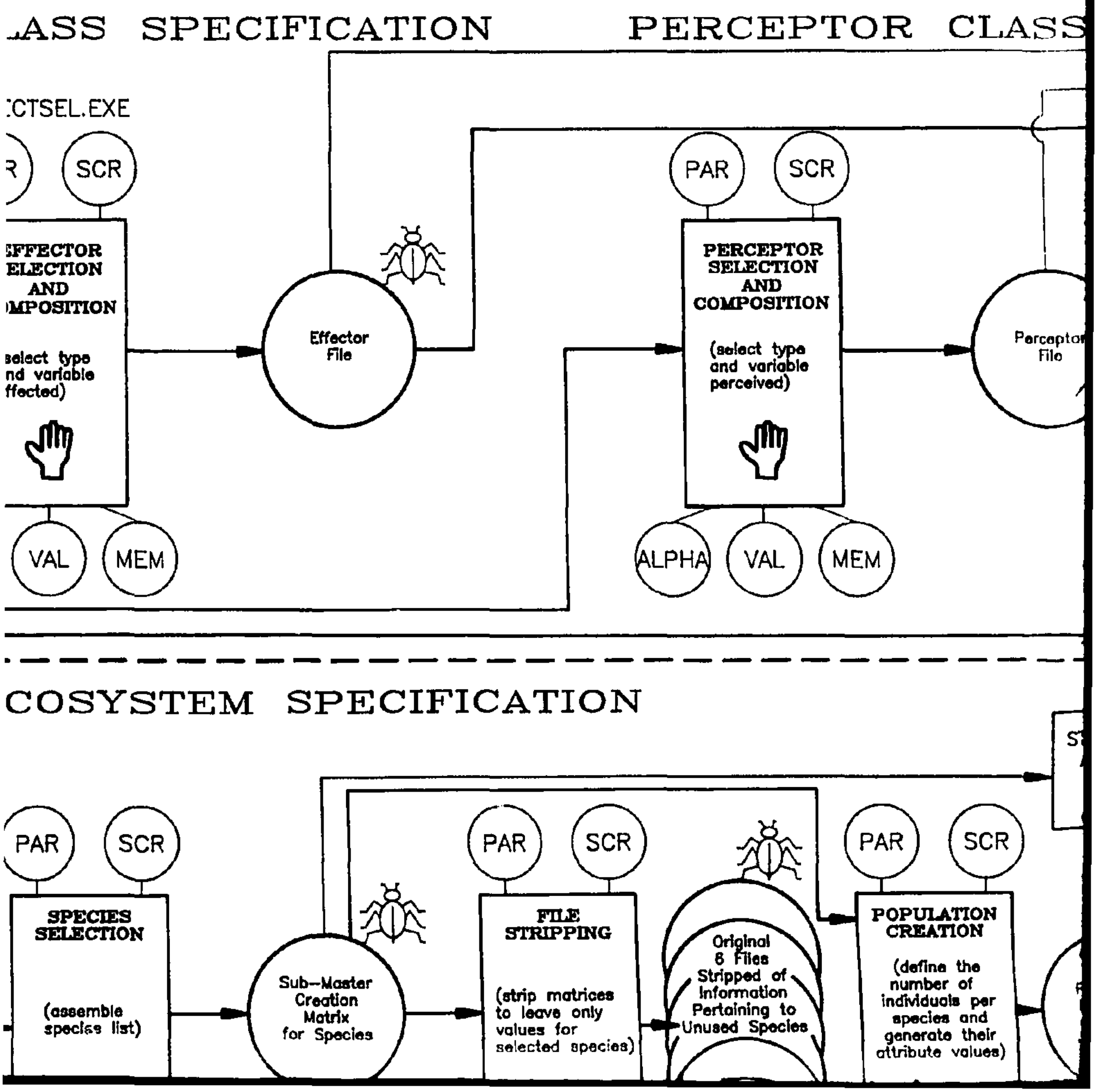


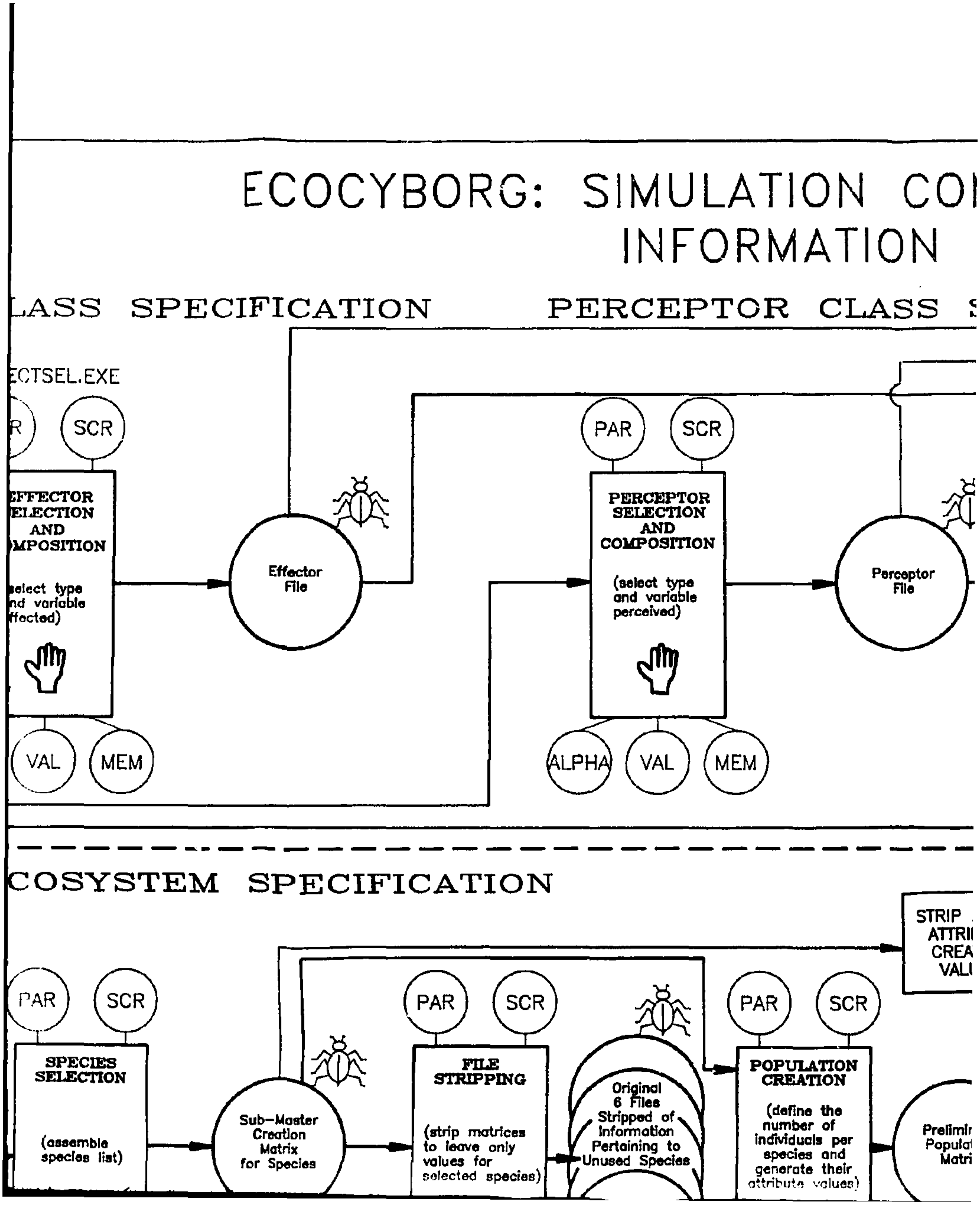
FIGURE 8: Class definition phase of the ecosystem model.



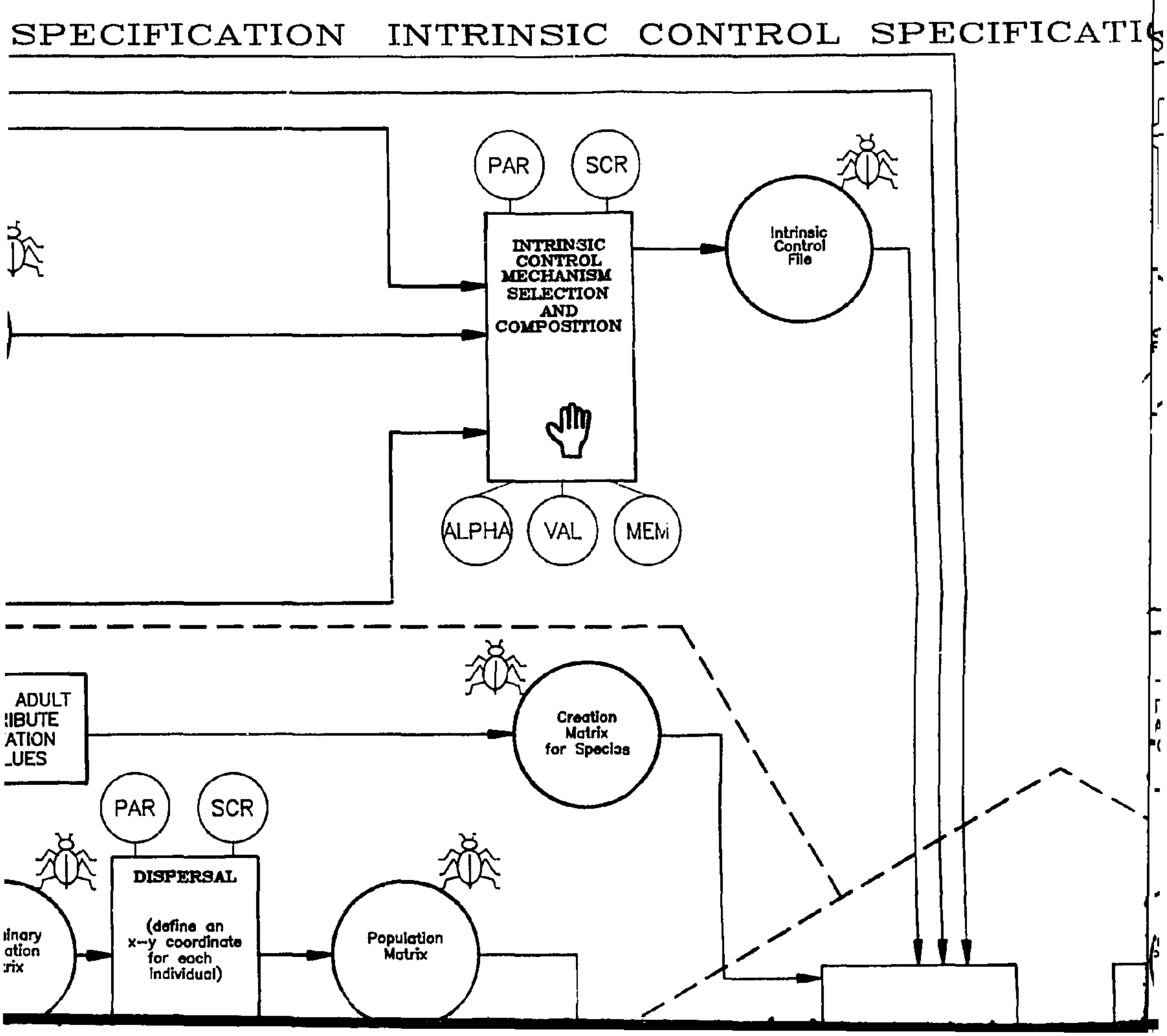
EFFECTOR CLASS SPEC
Master Effector File
Master Perceptor File
Master Intrinsic Control File
EFECTSEL.EXE
PAR
SCR
EFFECTOR SELECTION AND COMPOSITION
(select type and variable effected)
ALPHA
VAL
MEM
ECOSYSTEM
Eat Preferences Matrix
Attributes Vector
Attribute Matrix
Attribute Type Matrix
Master Creation Matrix for Species
PAR
SCR
SPECIES SELECTION
(assemble species list)

ECOCYBORG: SIMULATION C
INFORMATION



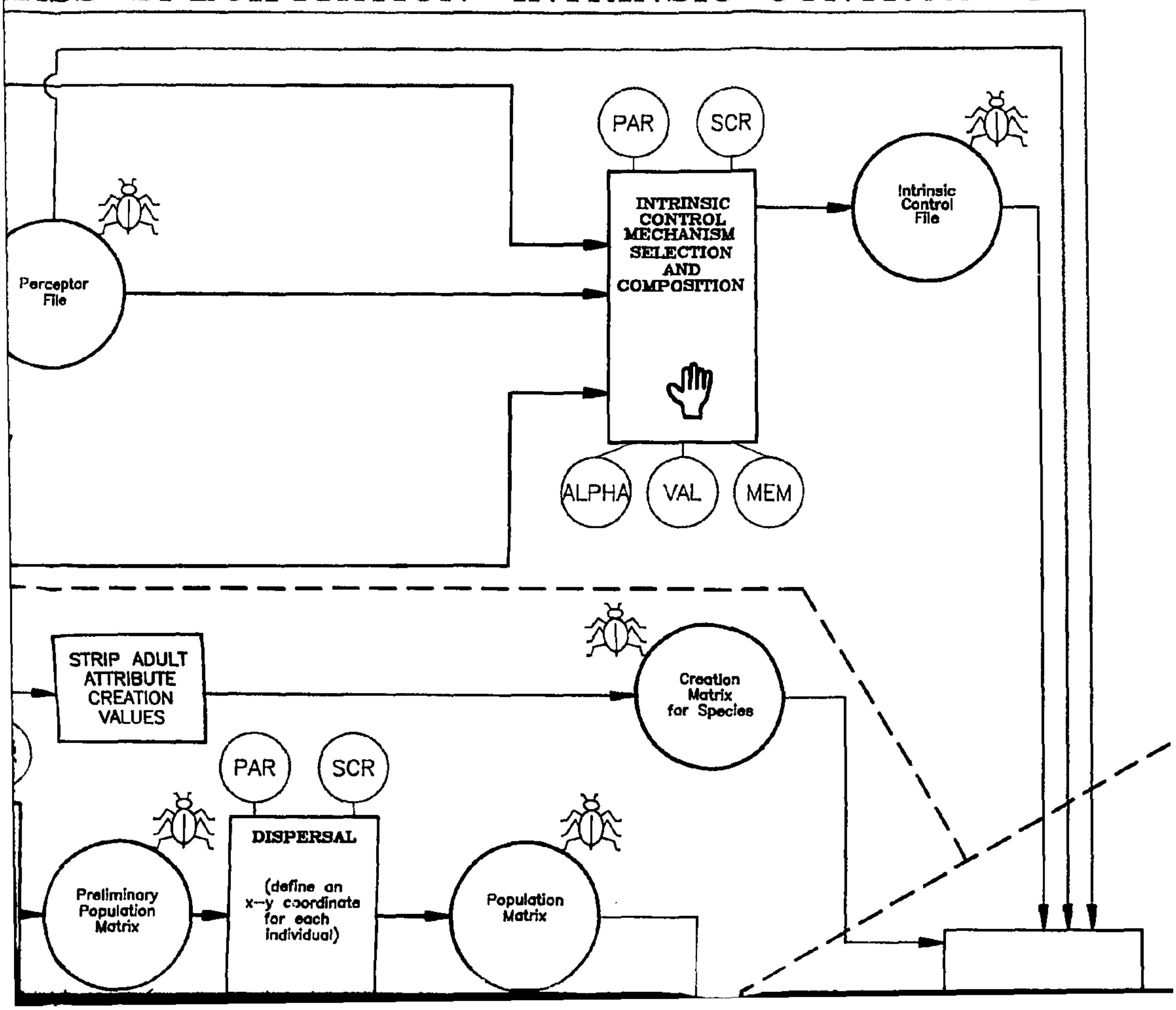

ECOCYBORG: SIMULATION CO
INFORMATION
LASS SPECIFICATION
PERCEPTOR CLASS
ECTSEL.EXE
R
SCR
EFFECTOR
ELECTION
AND
MPOSITION
elect type
nd variable
ffected)
VAL
MEM
Effector
File
PAR
SCR
PERCEPTOR
SELECTION
AND
COMPOSITION
(select type
and variable
perceived)
ALPHA
VAL
MEM
Perceptor
File
COSYSTEM SPECIFICATION
STRIP
ATTRI
CREA
VALU
PAR
SCR
SPECIES
SELECTION
(assemble
species list)
Sub-Master
Creation
Matrix
for Species
PAR
SCR
FILE
STRIPPING
(strip matrices
to leave only
values for
selected species)
Original
6 Files
Stripped of
Information
Pertaining to
Unused Species
PAR
SCR
POPULATION
CREATION
(define the
number of
individuals per
species and
generate their
attribute values)
Prelimi
Popula
Matri

MPOSITION PHASE
FLOW

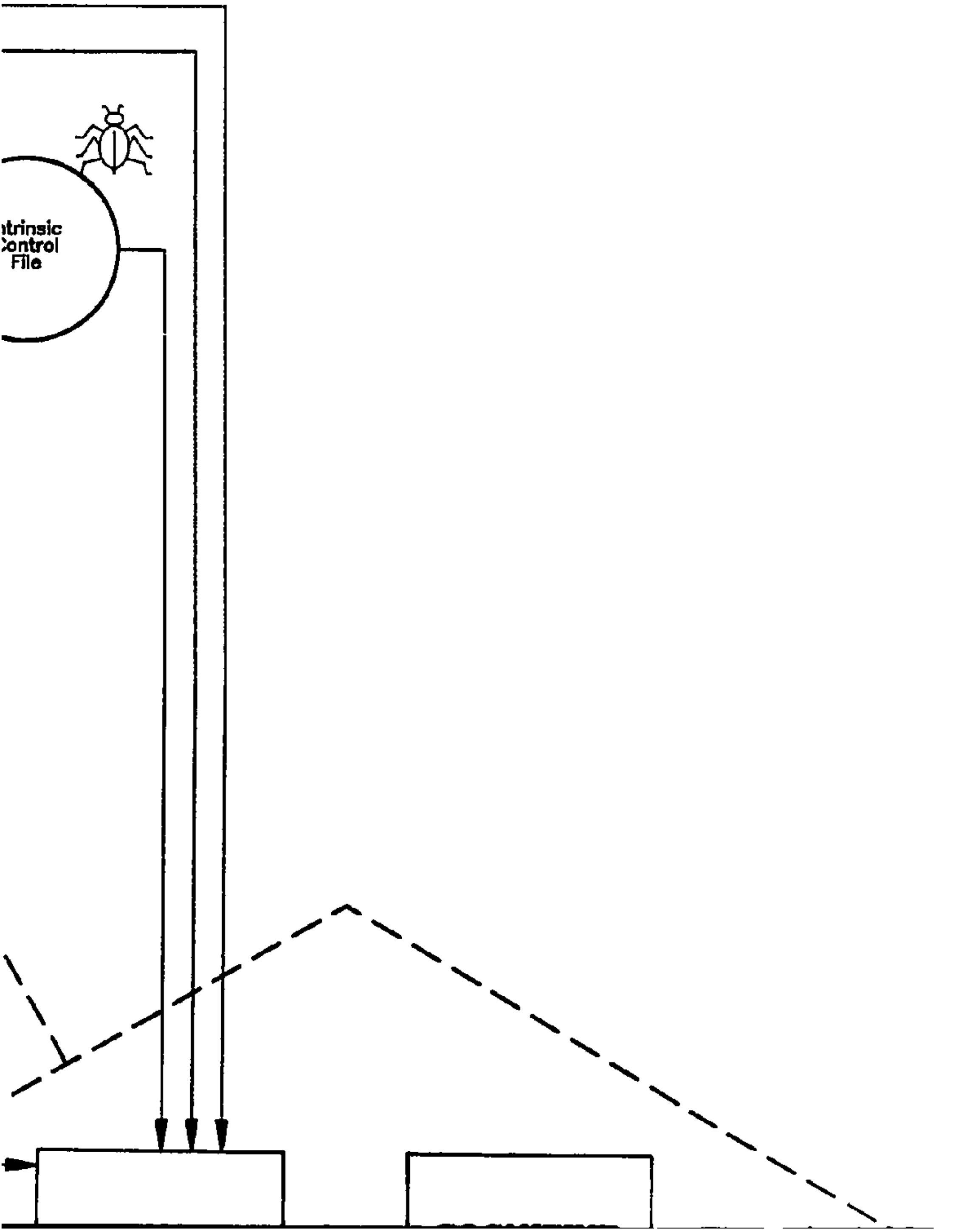


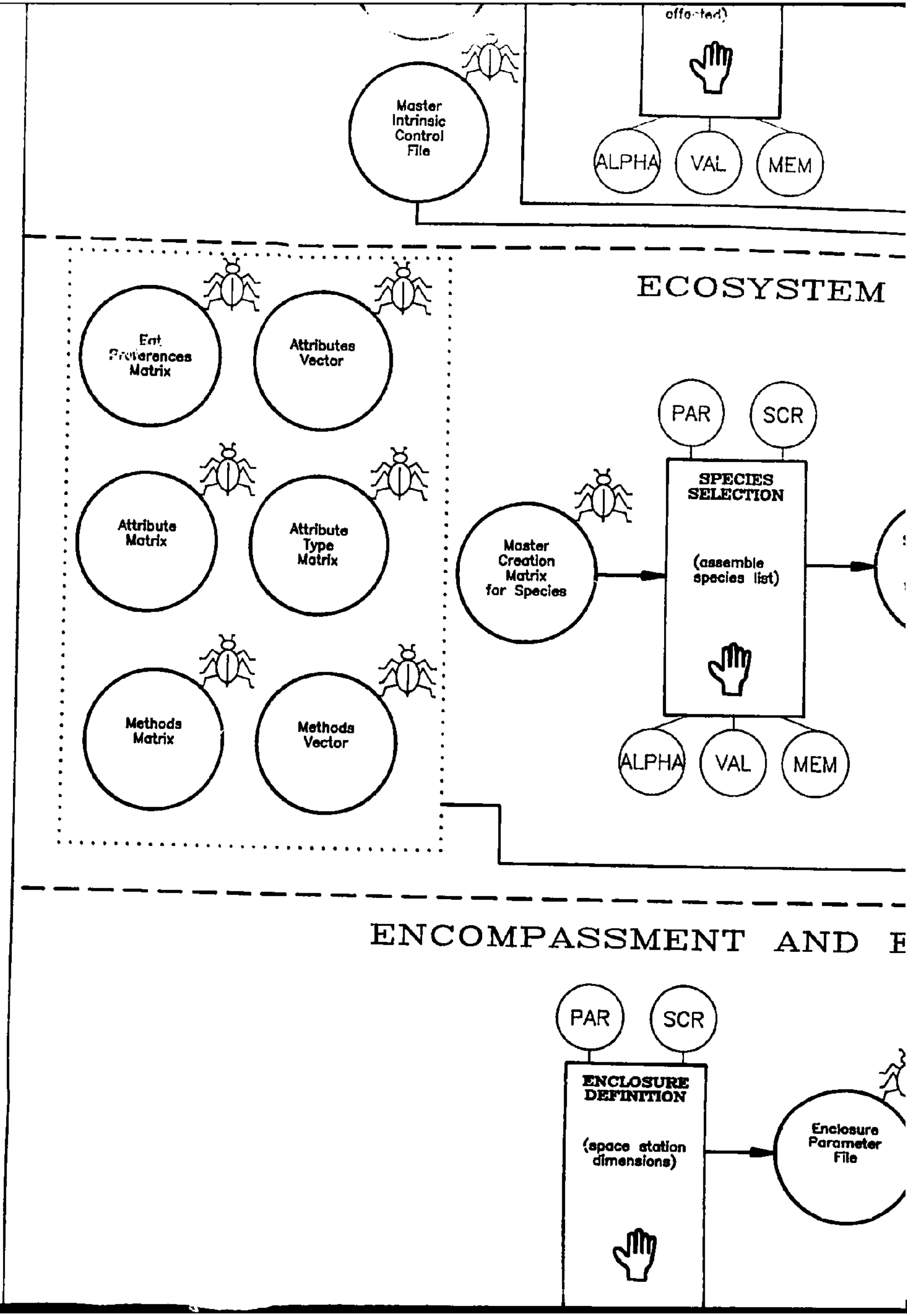
N COMPOSITION PHASE
TION FLOW

LASS SPECIFICATION INTRINSIC CONTROL SPECIF

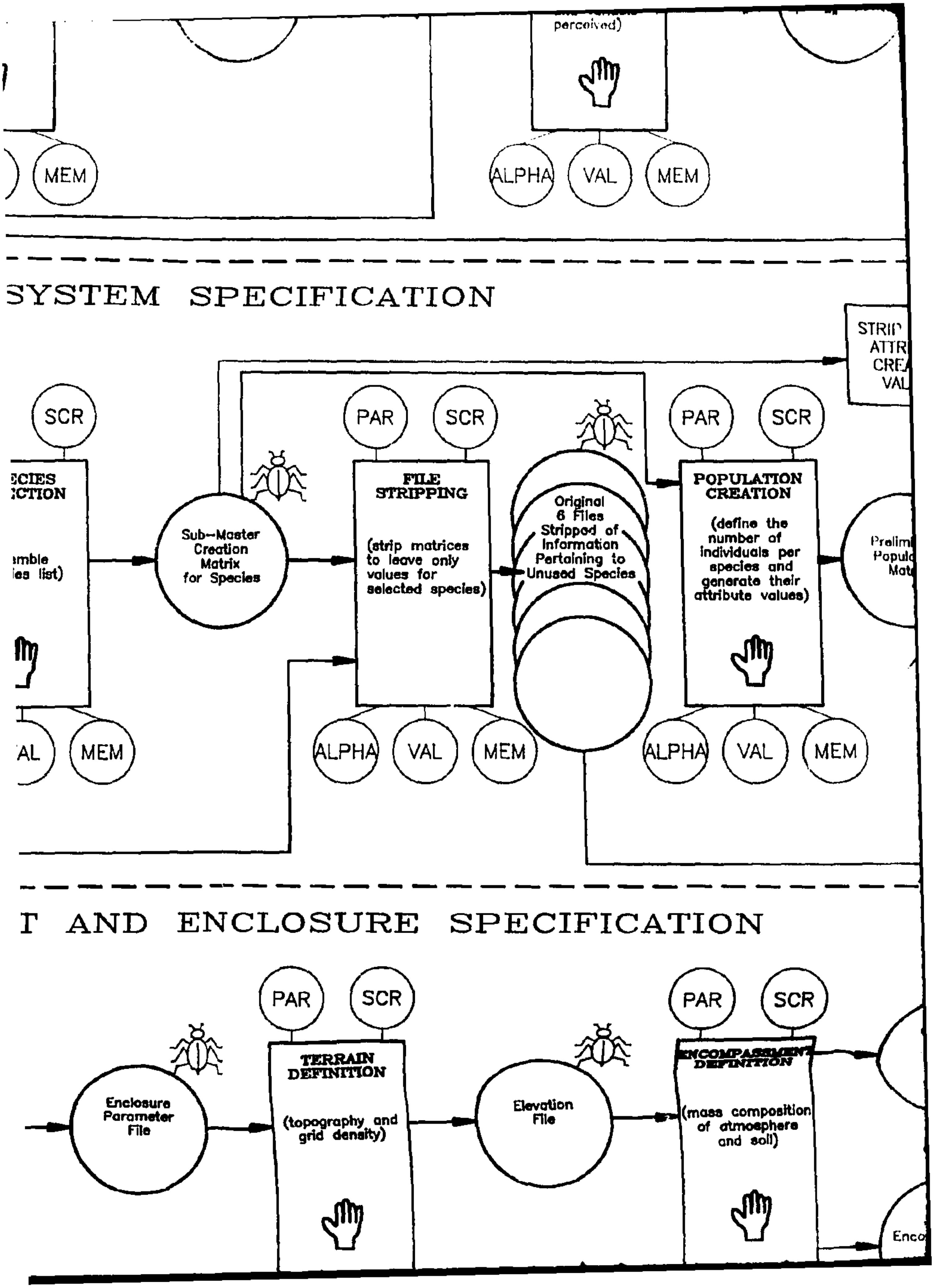


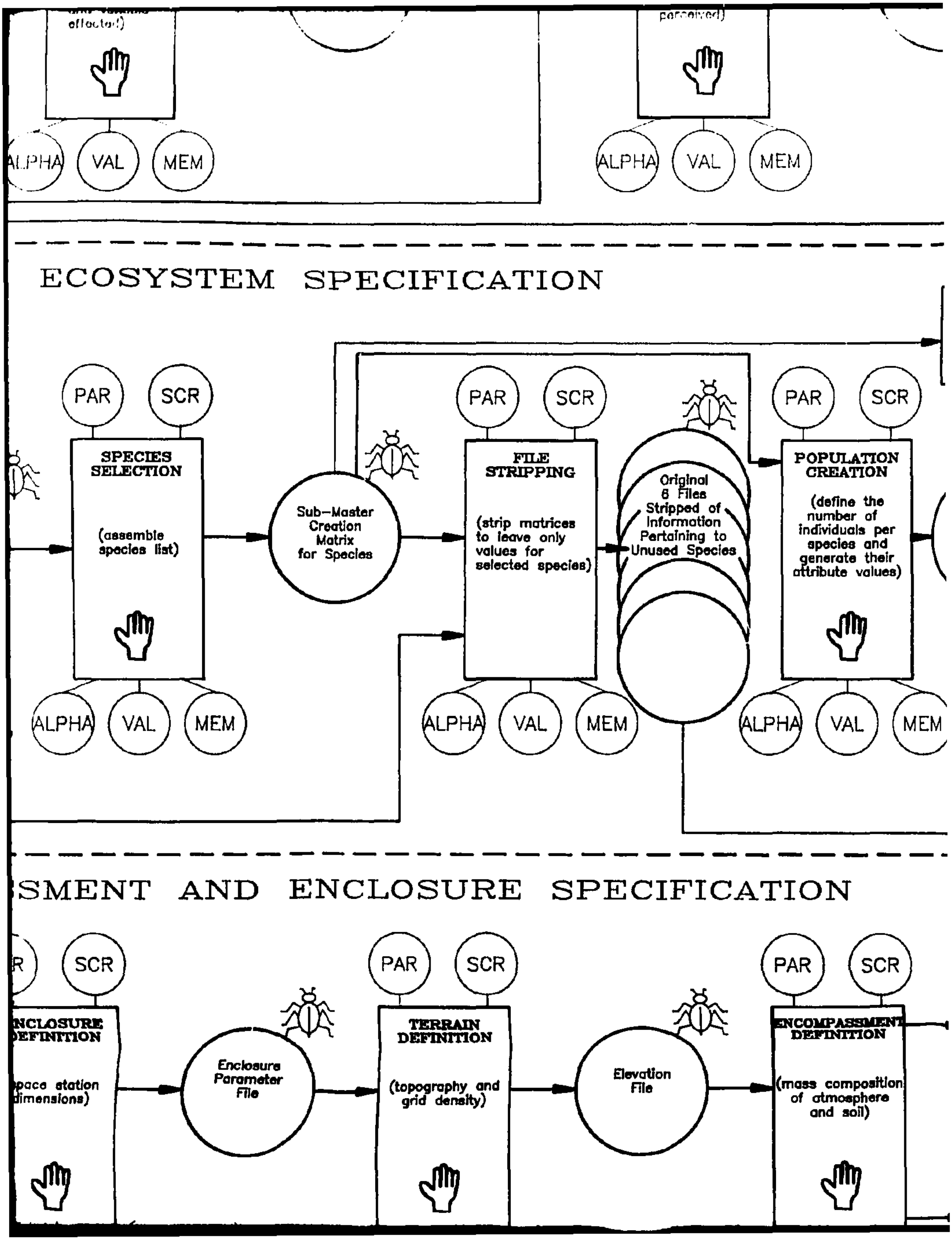
ROL SPECIFICATION

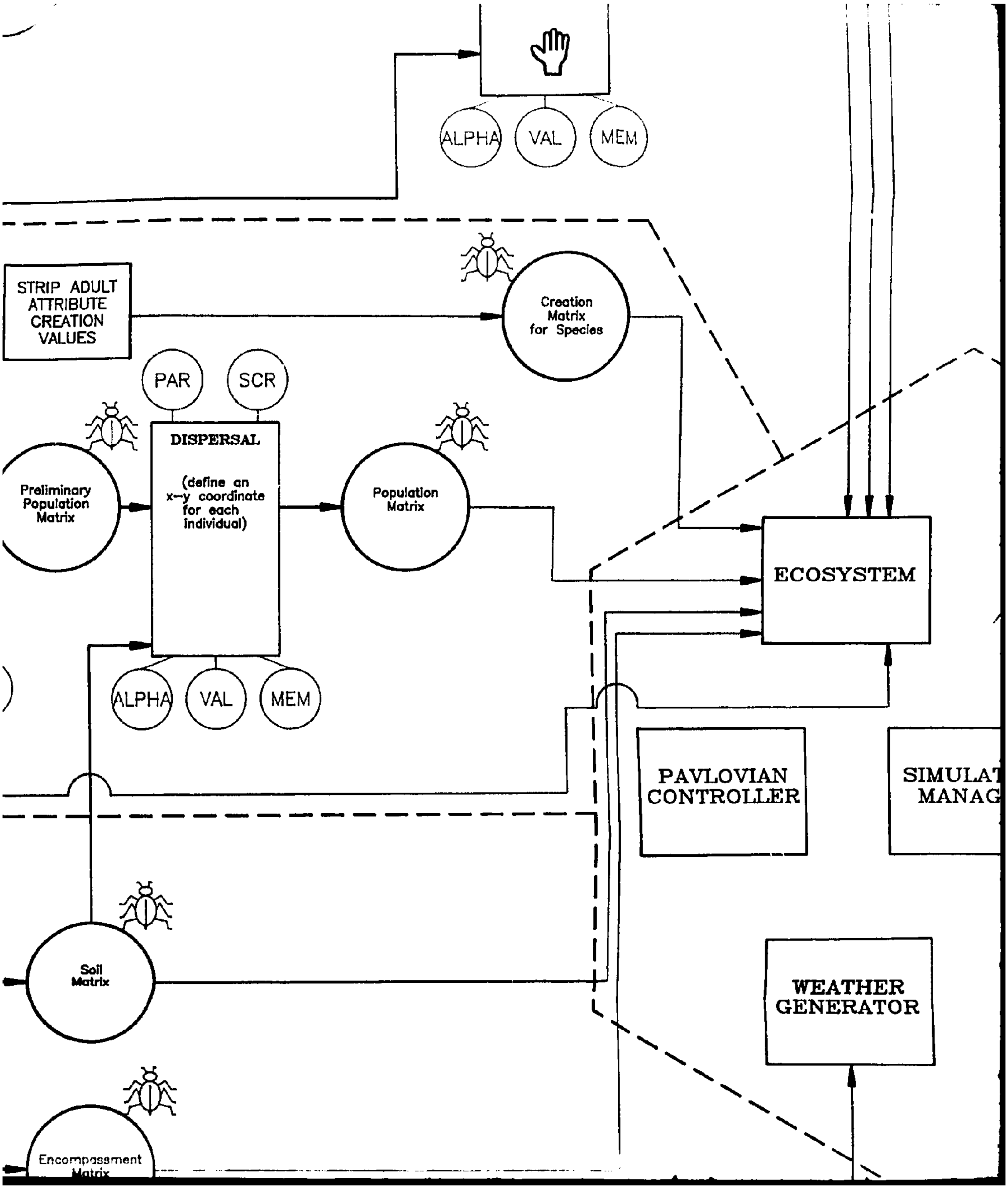




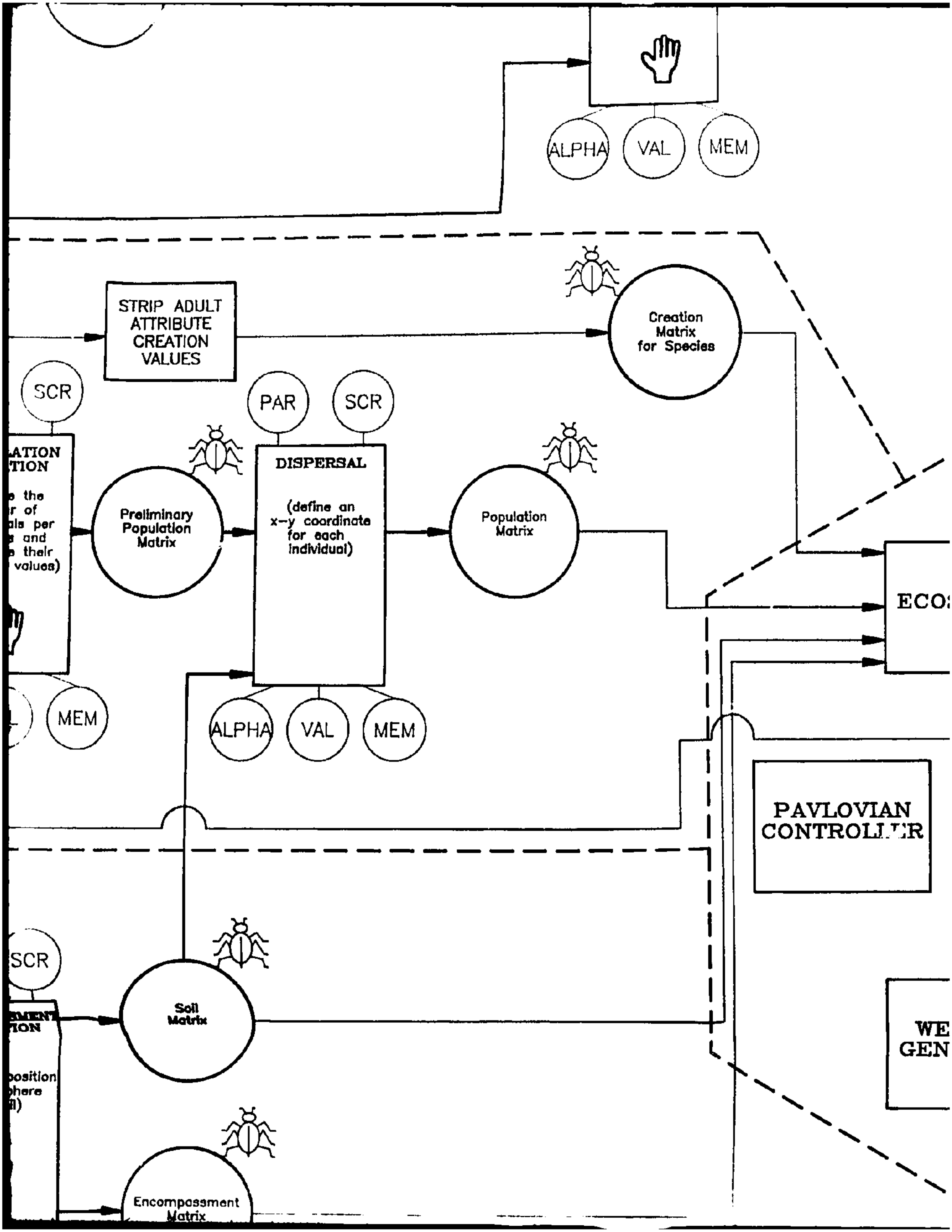
Master Intrinsic Control File
ALPHA
VAL
MEM
ECOSYSTEM
Eat Preferences Matrix
Attributes Vector
Attribute Matrix
Attribute Type Matrix
Methods Matrix
Methods Vector
Master Creation Matrix for Species
PAR
SCR
SPECIES SELECTION
(assemble species list)
ALPHA
VAL
MEM
ENCOMPASSMENT AND E
PAR
SCR
ENCLOSURE DEFINITION
(space station dimensions)
Enclosure Parameter File


perceived)
MEM
ALPHA
VAL
MEM
SYSTEM SPECIFICATION
STRIP
ATTR
CRE
VAL
SCR
ECIES
CTION
mble
ies list)
Sub-Master Creation Matrix for Species
PAR
SCR
FILE STRIPPING
(strip matrices to leave only values for selected species)
Original 6 Files Stripped of Information Pertaining to Unused Species
PAR
SCR
POPULATION CREATION
(define the number of individuals per species and generate their attribute values)
Prelim
Popula
Mat
AL
MEM
ALPHA
VAL
MEM
ALPHA
VAL
MEM
T AND ENCLOSURE SPECIFICATION
PAR
SCR
Enclosure Parameter File
TERRAIN DEFINITION
(topography and grid density)
Elevation File
PAR
SCR
ENCOMPASSMENT DEFINITION
(mass composition of atmosphere and soil)
Enco

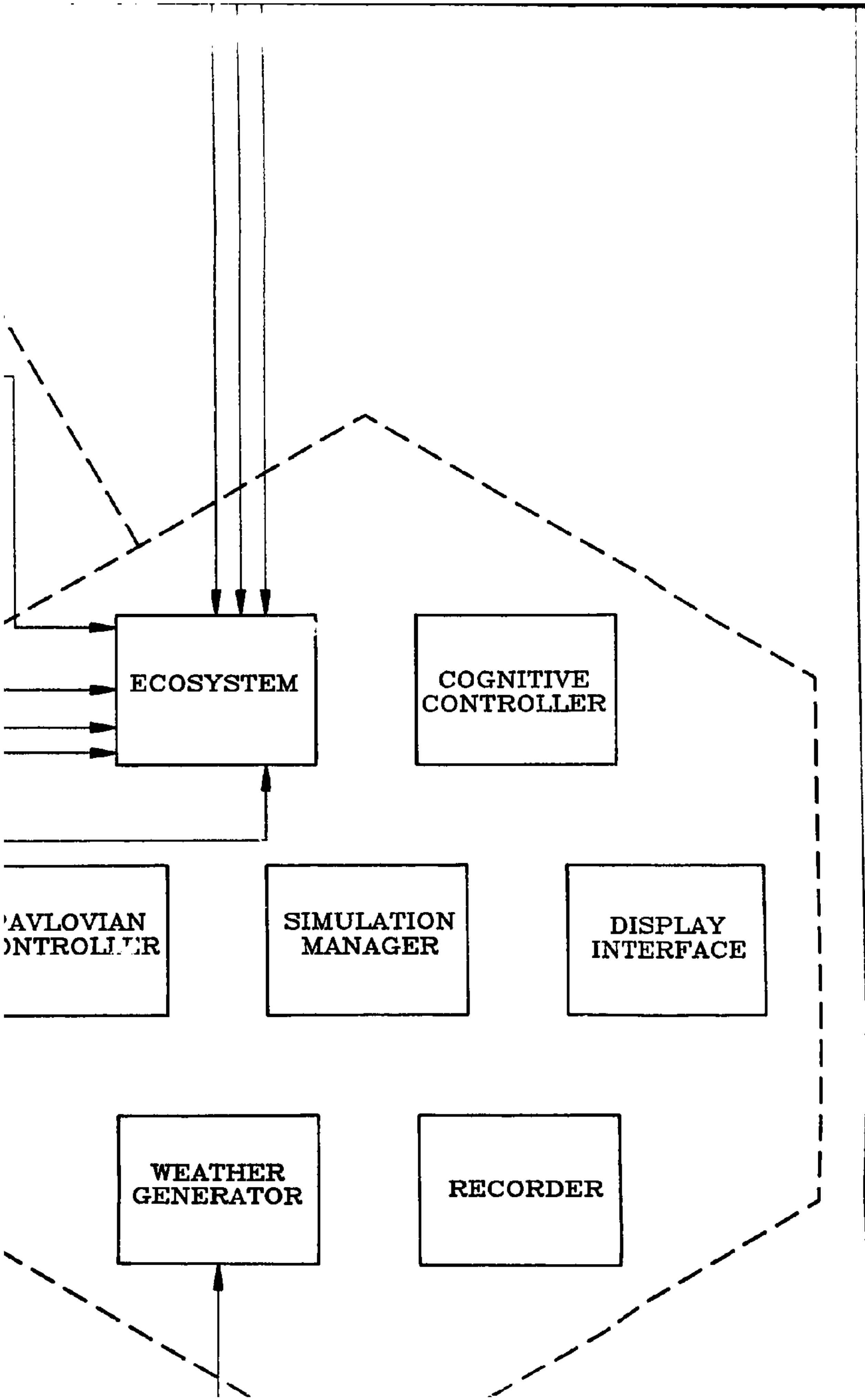

effector)
perceived)
ALPHA
VAL
MEM
ALPHA
VAL
MEM
ECOSYSTEM SPECIFICATION
PAR
SCR
SPECIES SELECTION
(assemble species list)
ALPHA
VAL
MEM
Sub-Master Creation Matrix for Species
PAR
SCR
FILE STRIPPING
(strip matrices to leave only values for selected species)
ALPHA
VAL
MEM
Original 6 Files Stripped of Information Pertaining to Unused Species
PAR
SCR
POPULATION CREATION
(define the number of individuals per species and generate their attribute values)
ALPHA
VAL
MEM
SMENT AND ENCLOSURE SPECIFICATION
SCR
NCLOSURE DEFINITION
pace station dimensions)
Enclosure Parameter File
PAR
SCR
TERRAIN DEFINITION
(topography and grid density)
Elevation File
PAR
SCR
ENCOMPASSMENT DEFINITION
(mass composition of atmosphere and soil)



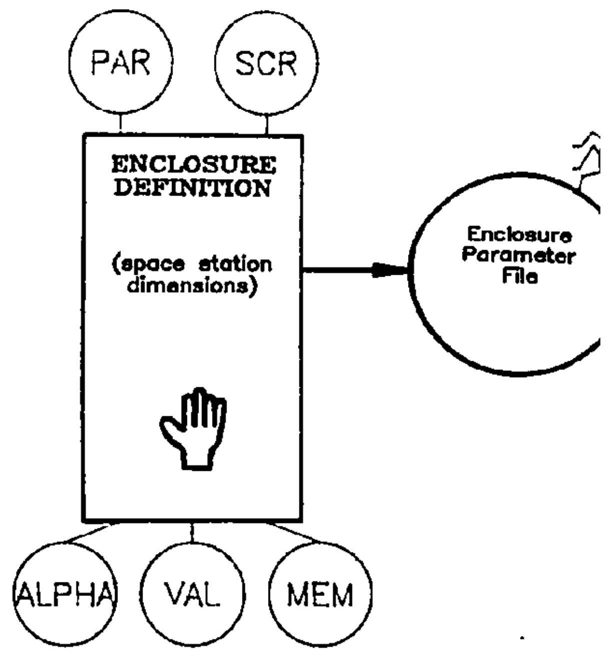
ALPHA
VAL
MEM
STRIP ADULT ATTRIBUTE CREATION VALUES
Creation Matrix for Species
PAR
SCR
DISPERSAL
(define an x-y coordinate for each individual)
Preliminary Population Matrix
Population Matrix
ECOSYSTEM
ALPHA
VAL
MEM
PAVLOVIAN CONTROLLER
SIMULAT
MANAG
Soil Matrix
WEATHER GENERATOR
Encompassment Matrix



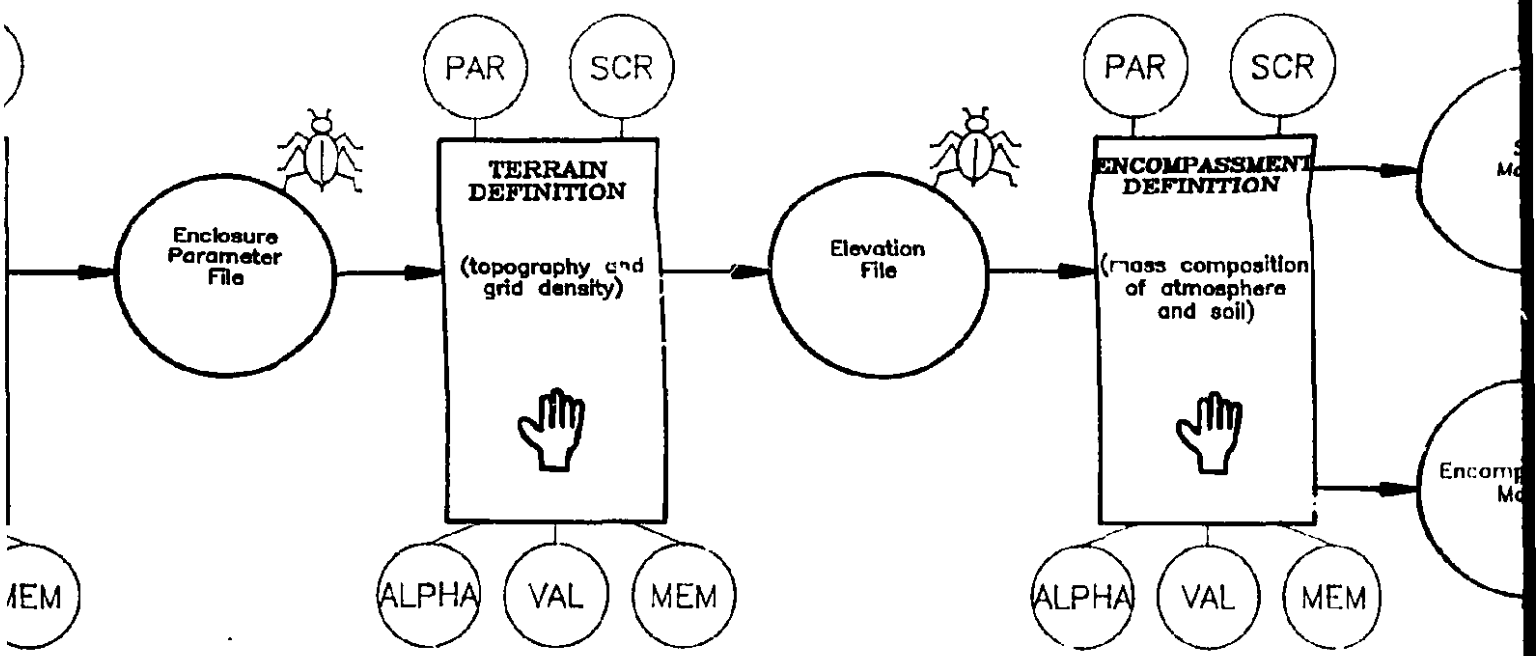
ALPHA
VAL
MEM
STRIP ADULT ATTRIBUTE CREATION VALUES
Creation Matrix for Species
SCR
PAR
SCR
DISPERSAL
(define an x-y coordinate for each individual)
Preliminary Population Matrix
Population Matrix
ALPHA
VAL
MEM
MEM
PAVLOVIAN CONTROLLER
SCR
Soil Matrix
Encompassment Matrix


ECOSYSTEM
COGNITIVE CONTROLLER
PAVLOVIAN CONTROLLER
SIMULATION MANAGER
DISPLAY INTERFACE
WEATHER GENERATOR
RECORDER

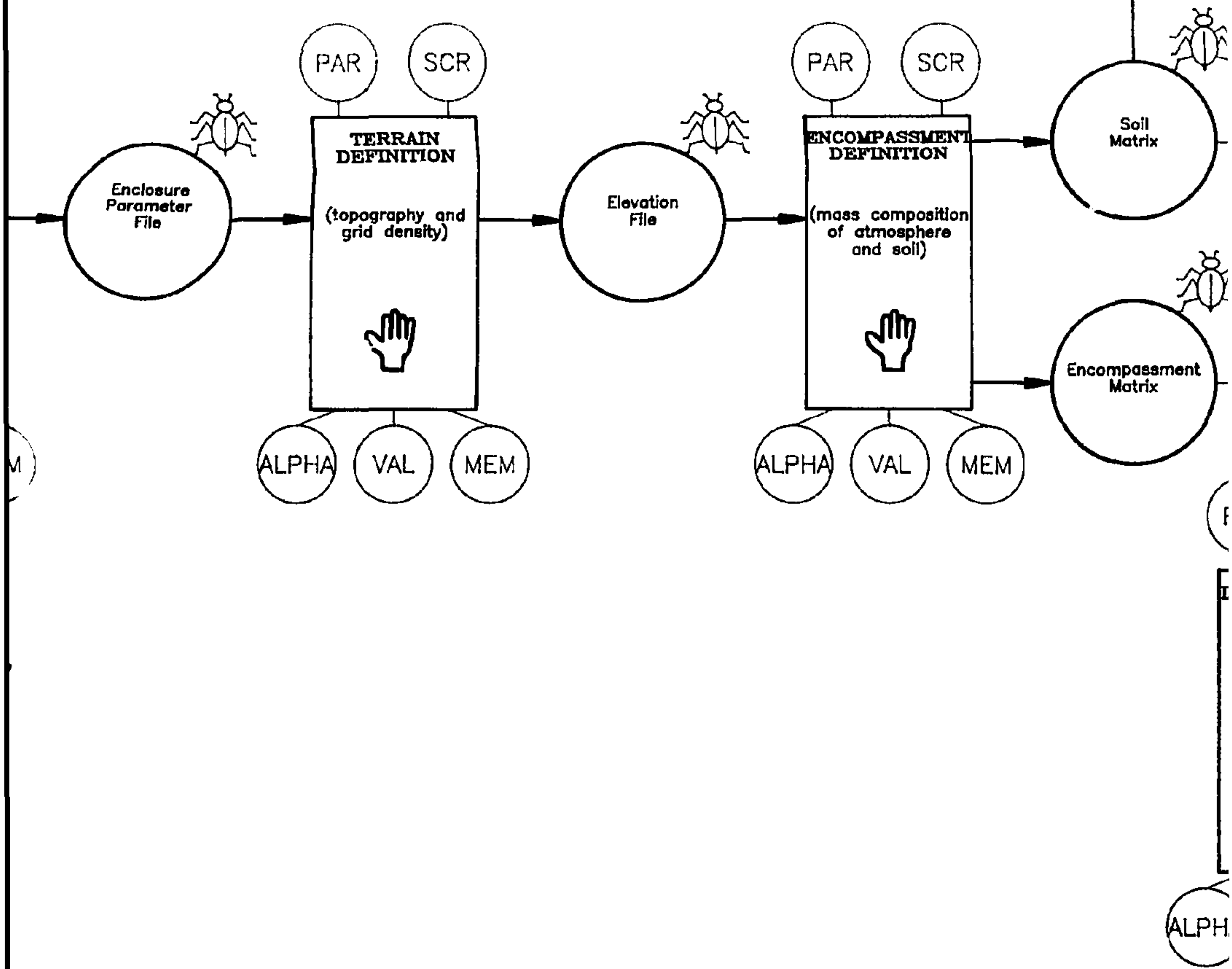
ENCOMPASSMENT AND I



NT AND ENCLOSURE SPECIFICATION



T AND ENCLOSURE SPECIFICATION



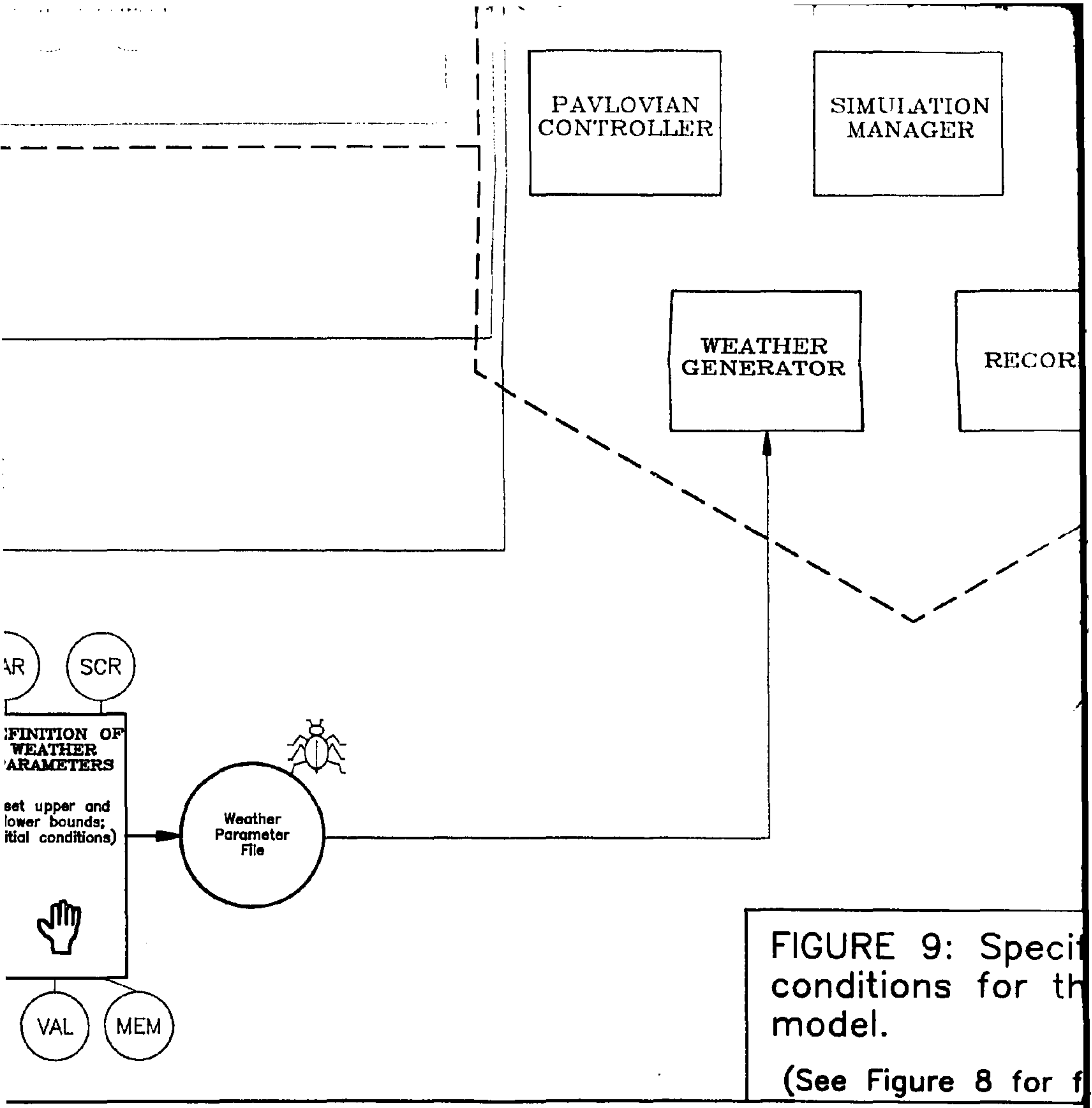
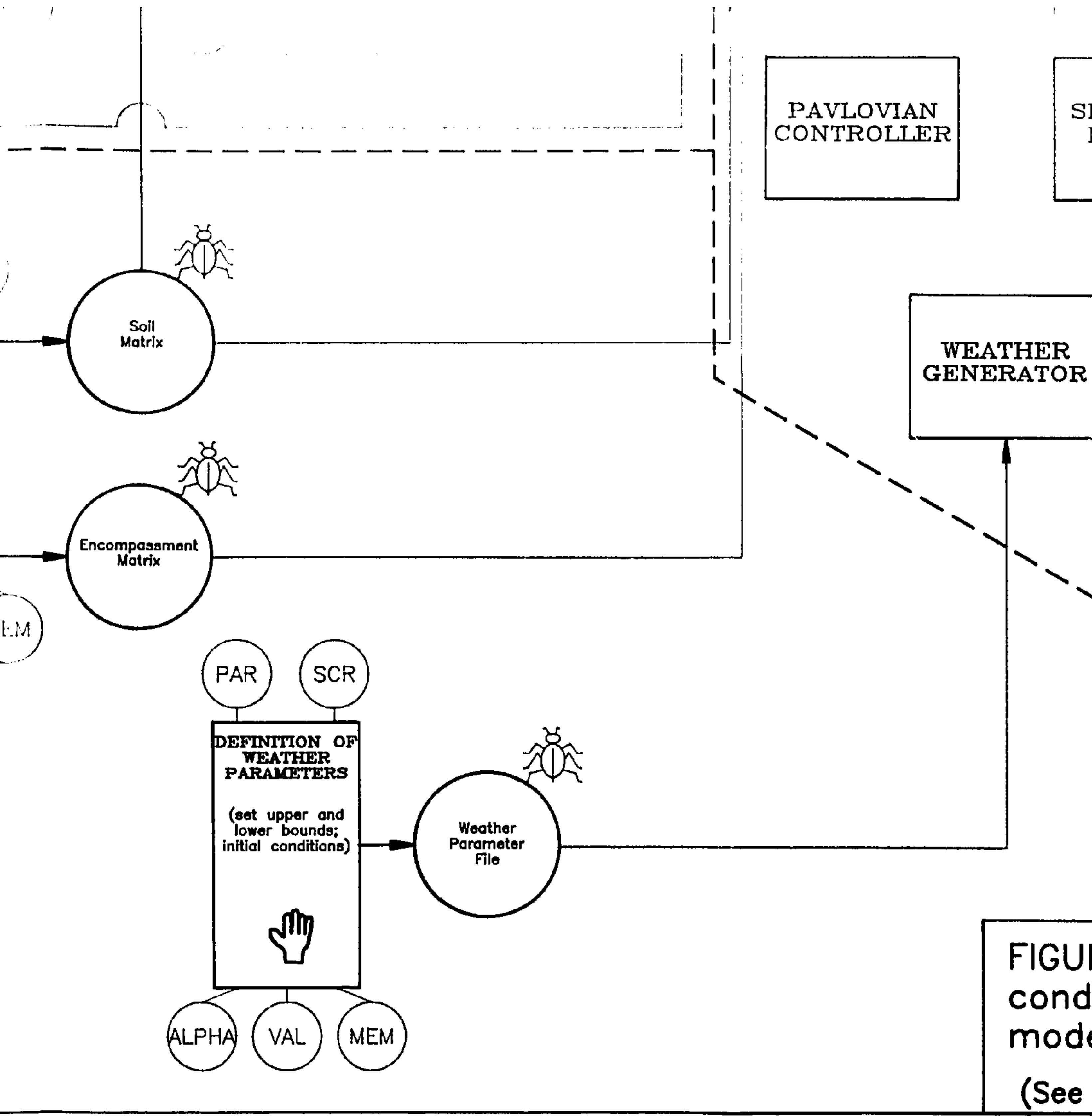


FIGURE 9: Speci
conditions for th
model.

(See Figure 8 for f



PAVLOVIAN CONTROLLER
Soil Matrix
WEATHER GENERATOR
Encompassment Matrix
PAR
SCR
DEFINITION OF WEATHER PARAMETERS
(set upper and lower bounds; initial conditions)
Weather Parameter File
ALPHA
VAL
MEM

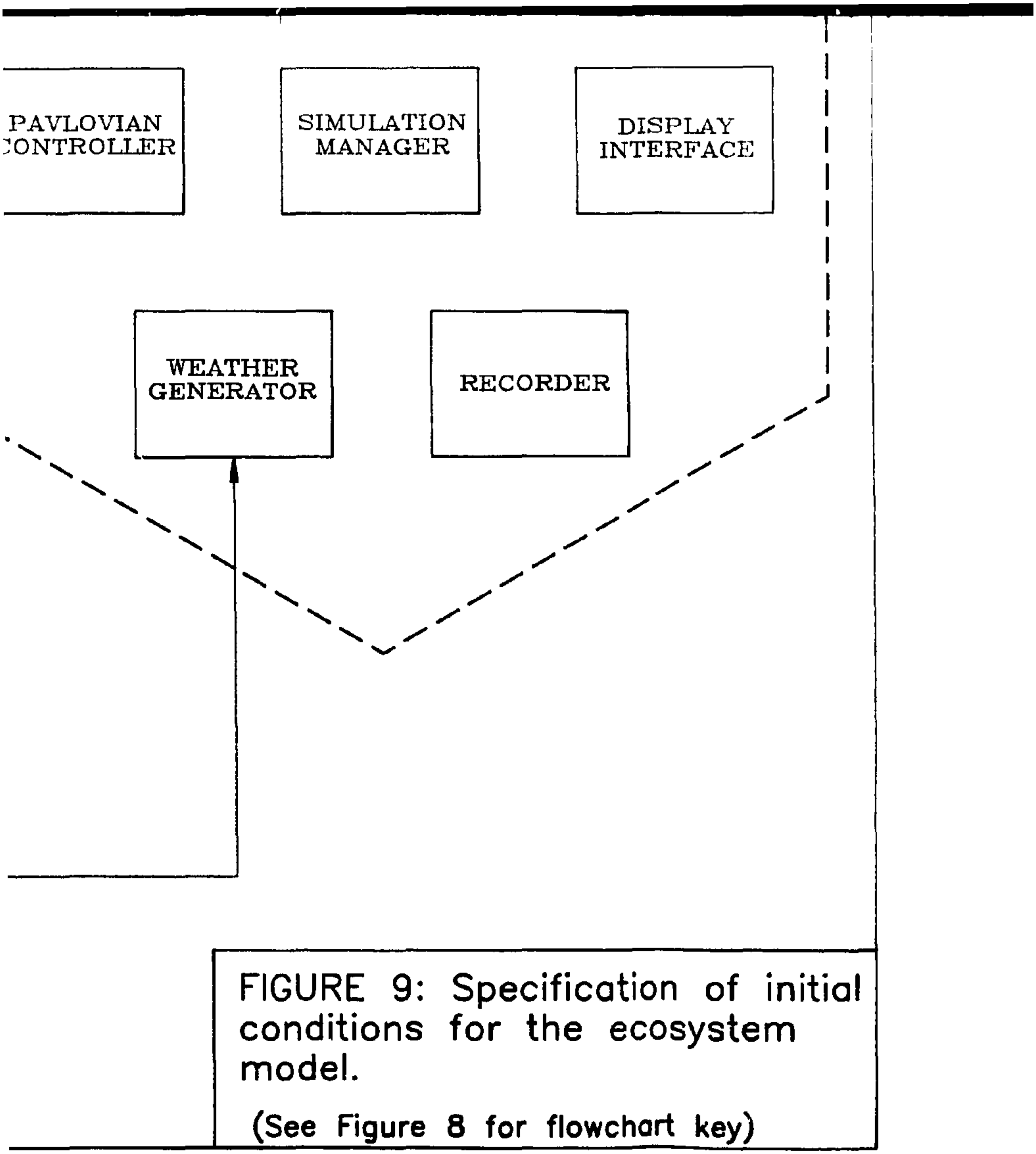


FIGURE 9: Specification of initial conditions for the ecosystem model.

(See Figure 8 for flowchart key)