Ecological Modelling, 46 (1989) 57–72
Elsevier Science Publishers B.V., Amsterdam – Printed in The Netherlands

AI MODELLING OF ANIMAL MOVEMENTS IN A HETEROGENEOUS HABITAT

L. JOSEPH FOLSE, JANE M. PACKARD and WILLIAM E. GRANT

Department of Wildlife and Fisheries Sciences, Texas A&M University, College Station, TX 77843 (U.S.A.)

ABSTRACT

Folse, L.J., Packard, J.M. and Grant, W.E., 1989. AI modelling of animal movements in a heterogeneous habitat. *Ecol. Modelling*, 46: 57–72.

We demonstrate use of object-oriented programming, dynamic linkages, rule-based decision procedures, and several other concepts from the field of artificial intelligence (AI) for modelling animal movements in a heterogeneous habitat. An object-oriented model of a deer that learns about habitat structure, plans movements, and accommodates to changes in a patchy brushland habitat is described and used to simulate effects of patch size on deer movements. Innovative features of this model include: (a) representation of habitat as a network of heterogeneous patches, (b) representation of an individual's knowledge of the environment (memory network) as different from but related to the habitat (habitat network), (c) individual's use of knowledge of the environment to plan paths to goals, and (d) ability for an individual to change its knowledge base when it encounters changes in the environment. Decision rules in the model are hypothetical, but the current application suggests that object-oriented modelling provides a concise yet detailed technology for modelling animal movements.

INTRODUCTION

The ability to model animal movements in a heterogeneous habitat is central to dealing with critical resource management problems such as the impact of habitat fragmentation on native faunas (Buechner, 1987) and is important in understanding spatial dynamics of such fundamental ecological processes as competition, predation (Hogeweg and Hesper, 1985), and secondary production. Modelling tools developed for diffusion processes (Okuba, 1980; Berg, 1983) have been used successfully over the past two decades to model systems with relatively random or probabilistic move-

Contribution No. TA 23195 of the Texas Agricultural Experiment Station.

0304-3800/89/$03.50

ments. However, models based on random walks (Berg, 1983) or probability distribution of turn angles and distances (Siniff and Jessen, 1969) are not appropriate to represent animal movements that involve decision rules contingent on a spatially heterogeneous, temporally dynamic habitat.

Habitat manipulation through brush removal to improve grazing conditions for domestic livestock is a common practice on rangelands throughout much of the semi-arid southwestern United States, and provides a specific example of a resource management problem involving assessment of animal movements through a heterogeneous habitat. The effect of this type of habitat modification on native ungulates such as deer remains controversial. A major question has been whether the area of brush treatment should be divided into many small patches or a few large patches. Predicting deer response to various brush control schemes has been difficult due to our inability to analyze deer movements relative to spatial configurations of brushy and open patches of various sizes.

Recent thinking about the use of artificial intelligence (AI) in ecology and natural resource management suggests that AI techniques may provide new ways to model a variety of ecological systems, including problems concerning animal behavior and animal-habitat interactions (Coulson et al., 1987). Such AI techniques as object-oriented programming and rule-based decision procedures seem particularly well-suited to modelling animal movements in spatially heterogeneous habitats (Graham, 1986; Saarenmaa et al., 1988).

In this paper we further demonstrate the use of several AI concepts for modelling animal movements in a spatially heterogeneous habitat. More specifically, we (a) discuss events and dynamic-linkages as concepts with great potential for ecological modelling; (b) describe an object-oriented model of a deer that learns about habitat structure, plans movements, and accomodates to dynamic habitat changes in a patchy brushland habitat in south Texas; (c) simulate the effect of habitat modifications on movements of the model deer; and (d) evaluate the general utility of an AI approach for modelling animal movements in a spatially and temporally heterogeneous environment.

ECOLOGICAL SCENARIO

AI modelling of animal movements involves concepts that are quite familiar to programmers using modern programming languages, but have only recently been applied to ecological simulation. These concepts include: objects; inheritance, classes, and instances; events, messages, and dynamic linkages. Before explaining these concepts in more detail in the next section, we want to introduce the terms in the context of the following ecological scenario of deer movements.

Objects

Deer move within a relatively bounded area often referred to as a home range, with activiy centers that may shift seasonally or at different stages of a lifetime (Inglis et al., 1987). Each deer in a population may be modelled as an object with a set of attribute variables and a set of rules that change the values of the attributes and pass information to other objects. Location can thus be an attribute of a deer-object, and movement can be monitored as changes in the value of the location attribute.

The geographical area of a deer population includes discontinuities that appear to the human eye to define patches such as brush and grassland vegetation types. Each vegetation patch within the range of a deer can also be modelled as an object. Point sources of water or linear features such as rivers and fences can be modelled as objects as well. Changes in attribute value of habitat objects can be monitored as changes in the ecological system.

Inheritance, Classes, and Instances

Each individual deer shares characteristics in common with other deer, but may have special features associated with its age, sex, and social class. For example, all deer have common physiological requirements that influence their choice of patches. However, young females tend to stay within the natal range longer than young males. Adult male deer tend to range more widely than females during the reproductive season. The characteristics shared by all deer may be reflected in a computer program by inheritance of the rules and attributes encoded in a 'deer class'. Characteristics shared by females may be encoded in a 'female class', and characteristics shared by adults may be encoded in an 'adult class'. A particular adult female is then an instance of the adult-female-deer class and inherits characteristics from each of the component classes. From a programming standpoint, the class concept with inheritance means that computer code can be packaged in appropriate classes and used at lower levels in the model hierarchy without repetition for each individual instance.

Events, Messages, and Dynamic Linkages

An event is an interaction between two objects; thus, a deer eating a plant in a habitat patch is an event. Messages are used to model events. For example, the interaction between a deer and its habitat may be modelled by the deer sending a message to a habitat patch inquiring about the value of its resource attributes. Since the patch is also an object, it can be modelled to respond by 'sending' another message back to the deer about how much of a particular plant is present. Each object is self-contained and may respond to specific messages that it may receive from other objects. The passing of messages may thus be dependent on internal or external changes rather than a fixed sequence of time steps.

When a deer remains within a home range, it appears to have no knowledge of patches outside its area of experience. This condition appears as if paths between adjacent patches do not exist in the memory of the deer. However, as a young deer matures or as resource attributes change seasonally, the deer may enter new patches and abandon old patches. The deer's 'memory' can be considered as a network of 'memory patches' related directly to the network of habitat patches. Neighboring patches can be represented as attributes of each patch, forming the linkages in the network of patches. As the information in the deer's memory changes, neighbors can be added and dropped in a pattern of dynamic linkages. The concept of dynamic linkages in a network of patches provides much more realism than modeling animal movements relative to an arbitrary grid as has been done in the past.

BASIC AI CONCEPTS USEFUL FOR MODELLING

Objects and messages in the ecological scenario described above seem natural in terms of deer and habitat patches; however, the object-oriented programming languages (Smalltalk, Actors, and versions of Lisp, Pascal, and C) carry this abstraction to a very deep and detailed level. With this scenario in mind, we turn to a more formal description of these concepts.

Many biological processes, especially those at the organismal level, depend upon the occurrence of certain events to trigger changes of state. These events do not occur predictably or at constant rates, but they are factors that drive ecological systems. In contrast, traditional approaches to modelling utilize tools that are based on rates or fixed time intervals (for example differential and difference equations; Okuba, 1980). Although these tools are important, they do not handle well some of the discrete events that drive many ecological systems.

Our approach to discrete events is different than that usually taken with discrete-event simulation. Our focus is on simulation of behaviors of individual components of the system, but components whose interactions with one another are discrete events. Traditional approaches to discrete-event simulation are based on queuing theory, which focusses on relationships between statistical distributions, and not on direct modelling of component behavior (for example see Banks and Carson, 1984).

The object-oriented paradigm from software engineering provides a means to simulate event-oriented system behavior on the basis of direct modelling of component behaviors. Object-oriented programming (Bobrow and Stefik, 1986) is a style of programming that involves the following features: (1) Both data structures (or state variables) and procedures for manipulating them (methods) are encapsulated in modules called objects. An object itself selects appropriate internal procedures to manipulate its internal state in response to messages from other objects in the system. Objects have no direct access to another object's internal workings. (2) A simulation model is set up in terms of actors (or objects) and their interactions (or messages) rather than in terms of data structures and procedures as in conventional programming. (3) The concept of class (somewhat analogous to type in conventional programming languages) is used to define generic properties of objects or actors. Actual objects are instances of a class; several objects of the same class may differ by value or state, but not by design or structure. (4) The class concept may be generalized, so that a hierarchy of classes may exist, such that lower level members of the hierarchy inherit properties from higher levels. Thus an object, or instance of a class, inherits properties from all of the classes in its hierarchy. Code or properties that are common to several types of objects in a simulation may be placed in one super-class and inherited through several different sub-classes that define the simulation objects. Features of the object-oriented paradigm important for ecological modelling include creating multiple objects in a simulation through 'cloning' multiple instances of a class and letting them interact, and incorporating heterogeneity among actors by letting each object develop its individual characteristics (state) in response to its unique history during simulation.

Dynamic-linkages among system components is another concept difficult to implement with traditional modelling approaches. Systems models and network models usually establish the network of connections among components of a model at the design stage, and these linkages are integral parts of the model structure throughout the simulation. Linkages among components of biological systems, on the other hand, are created and destroyed dynamically, as functions of component states through time. Linkages are not an a priori part of the system. Individuals in an organismal model, for example, just happen to be in the right (or wrong!) place at a point in time for a particular interaction to occur. Linkages are thus dynamic.

Software engineering, and especially AI, has developed many programming techniques for creating, manipulating, and searching dynamic lists, trees, and networks. This technology, implemented in modern computer languages, is readily available for building models of biological processes with dynamic linkages. We use dynamic networks and search techniques to implement the linkages in our models. Our models were developed in Franz

Extended Common Lisp with Flavors (an object-oriented extension) on a small Sun Microsystems workstation (Sun 3/52 with 4 Mbyte RAM).

MODEL DESCRIPTION

Viewed very simply, our deer-habitat model consists of habitat objects and animal objects. Each type of object possesses both attributes and

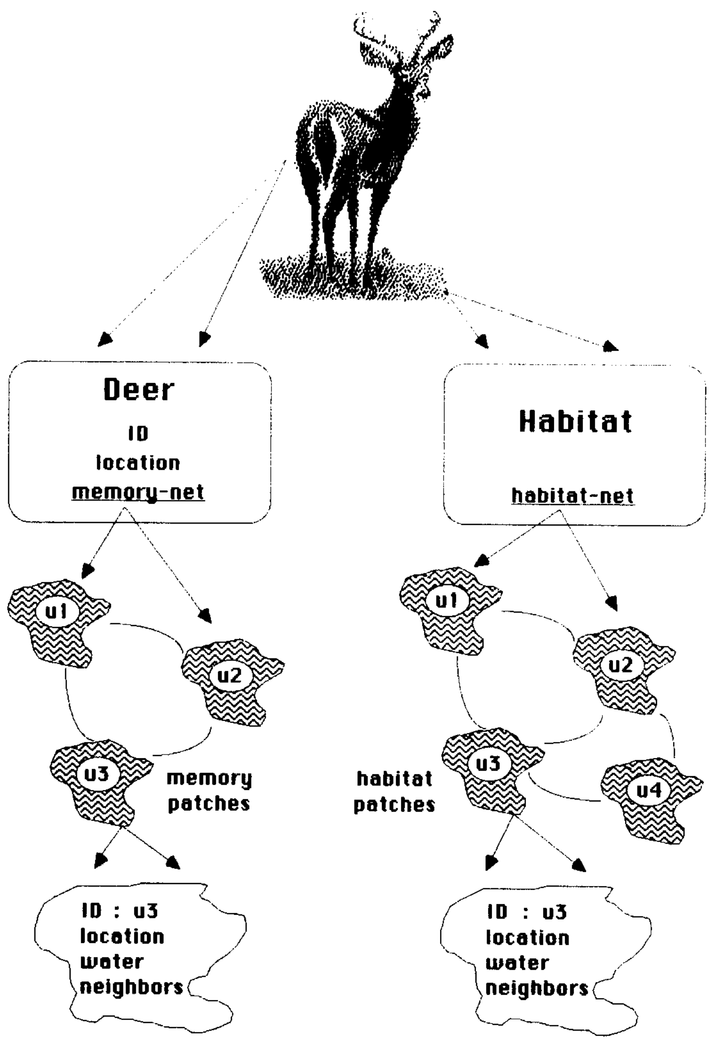

Fig. 1. Schematic representation of the model illustrating relationships among the model deer, memory-net, memory-patches, habitat, habitat-net, and habitat-patches. The memory-net is derived from the habitat-net by the model deer, but may be only a partial representation of the habitat.

Habitat-patch Attributes

ID	u240
location	(2.51 2.97)
water	nil
neighbors	(u250 0.1315 u241 0.0905 u239 0.2915)

Fig. 2. Example of attribute values for a particular habitat patch in the model's representation of a spatially heterogeneous environment. Sample data indicate that patch u240 is 2.51 km east and 2.97 km north of the origin of the study area. It has no water and 3 neighbors: u250, u241, and u239 at distances of 0.13, 0.09, and 0.29 km respectively.

processes, and is capable of communicating with other habitat and animal objects (Fig. 1). There are many types of objects in the model, but specific objects include the habitat (a network), individual habitat patches, the model deer, the model deer's memory (a network), and individual memory patches. Values of an object's attributes define its state or condition at a given point in time. Processes or methods define the procedures or functions that an object can perform. These methods include the ability to pass messages to other objects concerning present attributes.

A particular habitat patch object represents a 'patch' of ground on the study site with actual values for attributes of: (a) location, (b) presence of water, and (c) identities and distances of neighboring patches (Fig. 2). Each patch object can provide information upon request concerning its attributes, can change values of its attributes, and can link or unlink with other patches to form a dynamic patch network.

An animal object represents a white-tailed deer that is characterized by a location in space and time, and a memory of the habitat. Each deer object has methods to (a) move about or explore the habitat, (b) update its memory, (c) plan a path to reach a specific location (e.g. a patch with water), and (d) move toward a specific location. The interactions among the objects are event-driven by messages from other objects, and the linkages or relationships among components are dynamically established as functions of the objects' individual states during the course of interaction.

Although this simple overview connotes neither the richness of the object-oriented modelling paradigm nor the complexity of the current model, it does summarize the general features of the model and provides a useful framework for the more detailed description that follows.

Spatial heterogeneity: Patch attributes

The model habitat represents vegetation patches determined from photo-interpretation and ground-truthing of aerial photographs of a study site in

Uvalde county in the brushland biome of south Texas (Tchamba, 1988). The study site was a pasture treated in blocks to improve production of cattle forage. Vegetation types were untreated brush, herbicide treated brush, and cleared fields (Fig. 3a). Habitat-patch objects ($n = 264$) were defined from the vegetation map with consideration for disjunct features such as roads and fences (Fig. 3b). In the model, linkages between patch centers defined

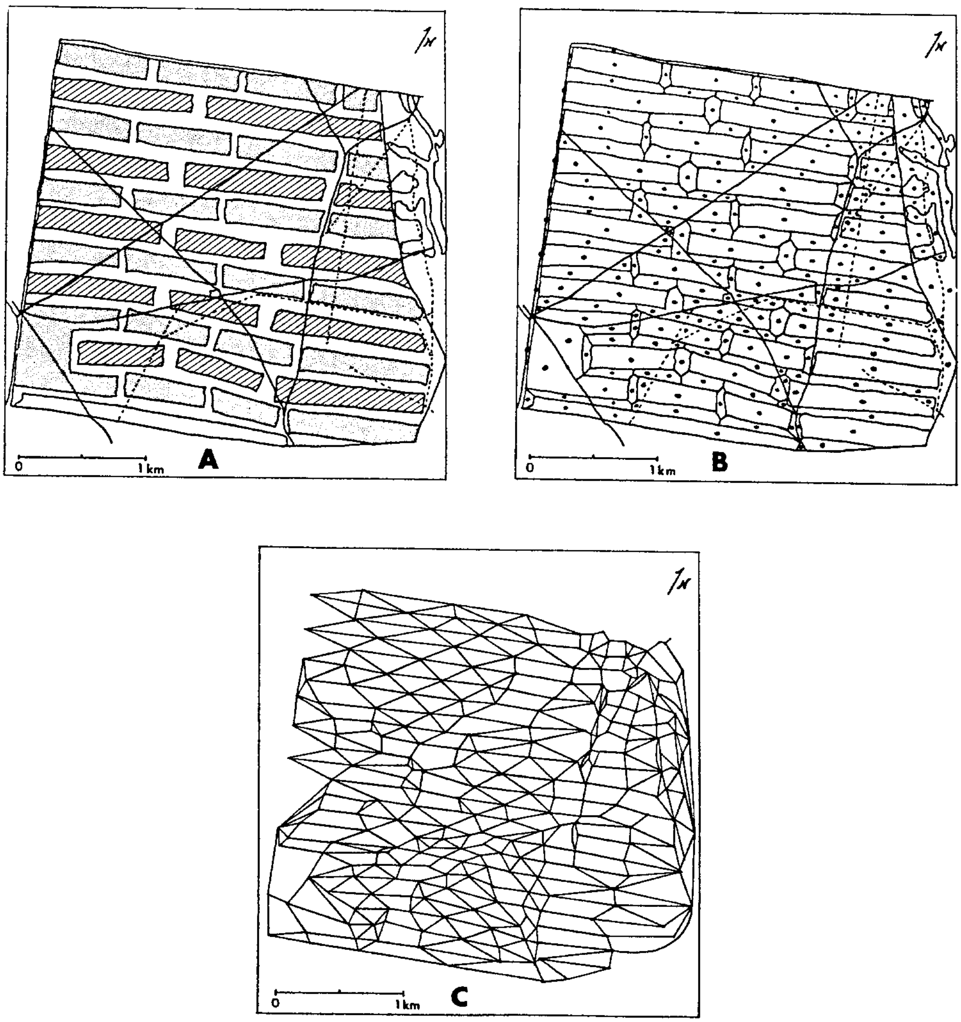

Fig. 3. Steps involved in representation of a heterogeneous environment: (A) Vegetation patches delineated from aerial photographs. Shaded areas are cleared of brush, hatched areas are treated with herbicides, and remainder is in native brush. (B) Definition of corresponding habitat-patch objects with patch centers indicated by dots. (C) Network illustrating linkages among patch centers for patches defined in (B).

the habitat network and limited potential movements available to model deer objects during simulations (Fig. 3c).

The linkages of patches in the habitat network were altered in sequential simulations to correspond to alternative ranch management practices. The

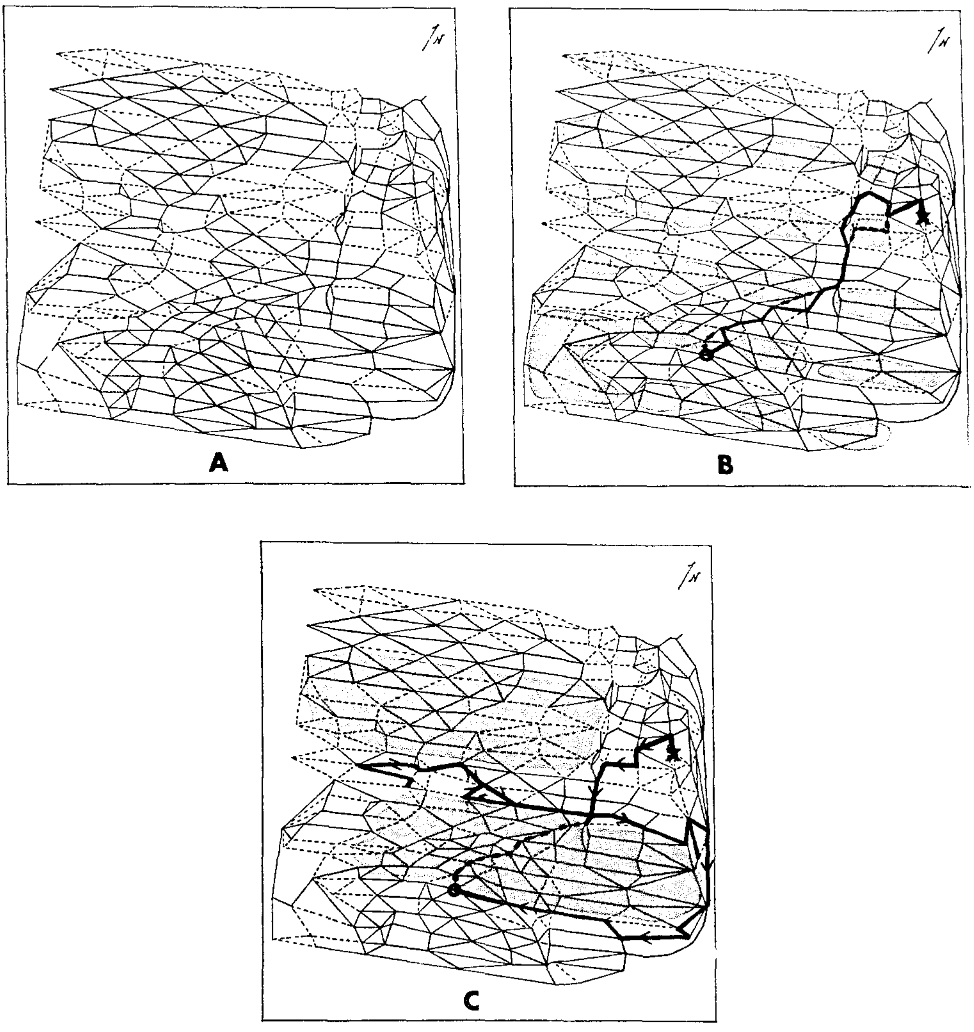

Fig. 4. Habitat linkages and deer memory linkages superimposed. Network of solid lines is the memory-map of model deer after initial exploration of the habitat. Dotted lines represent additional linkages in the habitat that the model deer failed to sample. Shaded areas represent patches made unavailable to the model deer by brush removal. The memory-map was used by the model deer for planning routes through the habitat. (A) Habitat and initial memory networks superimposed. (B) Small patch treatment. (C) Large patch treatment. In (B) and (C), heavy dashed lines represent paths planned and heavy solid lines represent paths actually traversed by the model deer from patch "X" to patch "O".

model deer had complete freedom of movement in the null simulation, in which all patches, regardless of actual treatment, were linked together for simulations and designated as 'untreated pasture' (Fig. 3c, 4a). The 'small patch treatment' (Fig. 4b) designated the existing configuration of vegetation, in which treated patches were unlinked from the habitat network prior to simulation, thus restricting potential movement of the model deer. A different linkage structure was used to represent the 'large patch treatment' (Fig. 4c). Simulations were run with each of these three habitat representations.

For simulation purposes, a simple rule was used to define linkages among habitat-patch objects. Patches of cleared field vegetation were considered unfavorable for deer movement, hence these patches were not included in the linking messages to the habitat that defined the network structure. All other patches were linked to their suitable neighbors. The actual linking (and unlinking) of patches in the habitat network was accomplished by loading messages into the simulation from external files on command from the computer console.

Memory map of model leer

The model deer's memory was represented as a dynamic linked-network of individual memory-patches that corresponded to habitat-patches. This memory map was developed by the model deer during exploration of the model-habitat. In each simulation, the model deer was directed to explore the 'untreated brush' habitat (by messages from the computer console). It moved by randomly choosing neighboring patches until it satisfied a stopping rule (specified number of patches or a specific goal patch). The model deer created a new instance of a memory-patch each time it entered a new habitat-patch. Attributes from the habitat-patch were stored in the memory-patch except for the list of neighboring patches. Linkages between memory-patches were stored only when the model deer moved between the corresponding habitat-patches. Thus both the memory-patches and their linkages (solid lines in Fig. 4a) were a subset of those present in the habitat (solid plus dotted lines in Fig. 4a). If the model deer reentered a known habitat patch, it queried the patch and updated memory-patch attributes and also linkages if it arrived by a new path. This process is illustrated in Fig. 5.

Once the model deer constructed a memory map of the habitat, it was directed (from the computer console) to find a path to a habitat-patch containing water. Either a random search or a search based on the model deer's memory map was used to find the goal patch, depending on the message received from the computer console. For the random-model simula-

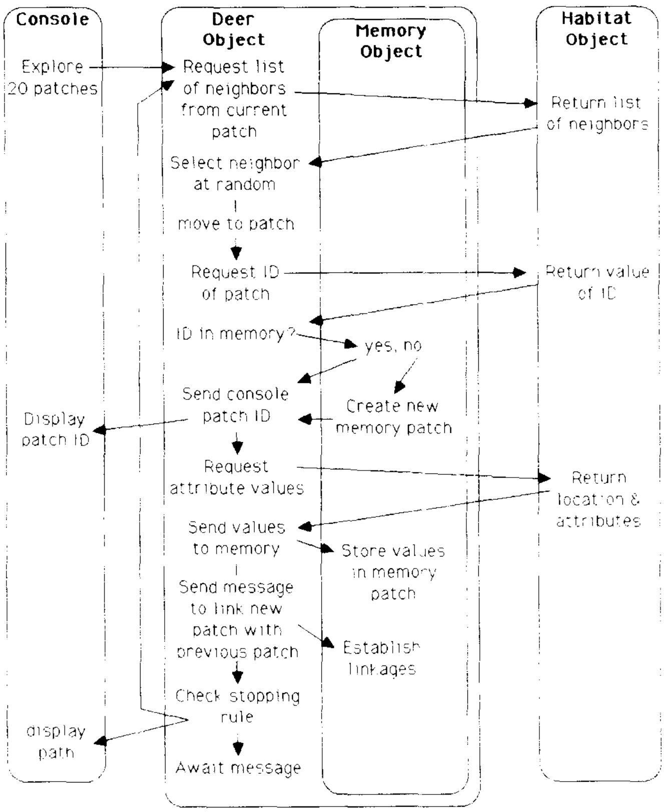

Fig. 5. Illustration of message-sending in the object-oriented paradigm. The objects shown are the console (computer sceen and keyboard), the model deer object, the memory object, and the habitat object. Not shown are finer details, such as individual habitat or memory patches. The figure represents a sequence (from top to bottom) of messages interchanged among objects during the exploration activity of the model deer.

tions, the model deer used the same decision rules as in the exploration mode to move until the goal patch was located. For the memory-model simulations, the model deer used the A* heuristic search algorithm (an object-oriented implementation in Lisp Flavors based on Nilsson, 1980, and Tanimoto, 1987) to plan a path through its network of memory-patches to the goal patch. This search algorithm is neither a 'depth-first' nor a 'breadth-first' search of a network, but a modified 'best-first' search in which additional state information is used to evaluate the 'best' adjacent node of the network to explore next. The heuristic criterion used to optimize behavior of the search process was the model deer's estimate of straight-line distance to the goal. Once a path was planned, the model deer moved through the model-habitat following the plan. However, the actual path traversed differed from the planned path if linkages in the model-habitat had changed due to habitat treatment. When a planned movement was blocked, the model deer updated its memory map, replanned, and continued. This process was continued until the goal was reached or until the model deer discovered that the goal-patch was no longer connected to its present patch via the memory map. All paths planned and traversed were output to the computer console.

SIMULATIONS

The simulations were designed to compare performance of two alternative models of hypothetical decision rules controlling deer movements. The first model was based on a random search strategy using local search with no memory. The second model included decision rules for storing and modifying information in a memory map as described above. The performance of each model was compared in simulations of deer searching for water in each of 3 treatments (no brush treatment, cleared brush treatment in small patches, cleared brush treatment in large patches). The small patch treatment (Fig. 4b) represented the experimental configuration of brush removal on the study area. The homogeneous habitat (Fig. 4a) represented the study area prior to brush removal, and the large patch treatment was a hypothetical representation of the study area in which the small brush removal patches were aggregated into fewer, larger patches (Fig. 4c).

Behavior of the two models (memory versus random) was compared by simulating movements from a single starting point ("X" in Fig. 4b, c) to a specified goal of water ("O" in Fig. 4b, c). The starting memory map resulting from exploration (solid lines in Fig. 4) was identical for each model deer in each simulation (by design using a repeatable pseudo-random number generator). The memory models were not stochastic, since finding an optimal path between two points in a fixed network is not a random process. Thus the memory model was run once for each habitat treatment. The random search model was repeated 10 times for each habitat treatment. We recorded the specific path and total distance travelled by the model deer during each simulation.

RESULTS

Performance of the random model and the memory-map model of movements differed substantially (Table 1). The random model was relatively insensitive to changes in the habitat mosaic, yielding mean path lengths that ranged from 203 km (small patch treatment) to 302 km (large patch treatment). Coefficients of variation were high for the random model.

TABLE 1

Path distances for model deer movements using memory maps and random movements

Habitat type	Path lengths	
	Memory map	Random (10 trials)
Unmodified brush	2.57 km [a]	272 km (54% CV)
Clearing in small patches	3.13 km [a]	203 km (56% CV)
Clearing in large patches	11.46 km [a] (3.84 km) [b]	302 km (93% CV)

Distances under random trials are means of ten replications (with coefficeint of variation). [a] Memory map distances significantly shorter than mean of random paths in all cases ($P < 0.001$, t-test). [b] New path used after model deer learned of habitat modifications.

The memory-map model yielded path lengths two orders of magnitude shorter than the random model (Table 1). Performance of the memory model was intuitively satisfying. In the small patch treatment, the model deer traversed a path that differed little from the no-treatment environment (Fig. 4b), but was 3 times shorter than the path initially traversed in the large patch environment (Fig. 4c).

In the large patch treatment, the model deer moved west along the boundary of the patch when its planned path was blocked, backtracked (cutting a corner on the return path), then found its way around the eastern edge of the patch. Several turns on the path which were longer than the straightline distance because certain linkages (dotted links in Fig. 4) were not in the model deer's memory map. The model deer changed its memory map to correspond to changes encountered due to habitat treatment. When requested to find the same goal a second time, its path length was substantially reduced (Table 1).

DISCUSSION

We have described a simplistic model of deer movements for the purpose of illustrating how some programming techniques from AI and software engineering can be applied to develop models of biological processes such as animal movement in a heterogeneous habitat. The object-oriented modelling approach can represent features of behavior that violate assumptions of more traditional models based on random diffusion processes, or optimization via linear equations. Specifically, the innovative features incorporated in our movement model were: (a) the representation of habitat heterogeneity as a network of patch-objects, (b) the ability to represent the difference between an individual's knowledge of the environment (memory-patch network) and the real world (habitat-patch network), (c) the planning of a path to a goal based on prior knowledge of the environment (A* heuristic search routine), and (d) the ability to change the individual's knowledge base when it encounters a change in the environment (creation of new memory-patch instances, change in attributes of existing patches).

The modelling approach described in this paper is one additional step toward developing more realistic models of animal/habitat interaction based on the premise that properties emergent at the population or community level can be modelled from decision rules shaped by natural selection at the individual level (May and Seger, 1986). The object-oriented programming approach is a substantial departure from the use of random models of animal movements, which assume that the action at any point in time is independent of the previous time unit. Even though the assumption of randomness is likely to be invalid for species that exhibit goal-directed behavior in heterogeneous environments, it is accepted as the basis for widely used analyses of animal home ranges (Dixon and Chapman, 1980; Anderson, 1982; Bekoff and Mech, 1984). Movements of our model deer demonstrated that a random search strategy can yield substantially different predictions about use of space than the memory-map search strategy.

Such differences in average path length can be very important in generating and testing hypotheses about the effects of habitat changes on energetics and water balance of free-ranging deer. A better understanding of such processes is essential for generating predictions about productivity at the population level. At this stage, we do not claim that our model generated expectations about how deer actually move relative to habitat patches of different sizes. However, we feel we have demonstrated that the object oriented-modelling approach provides a concise and detailed technology for generating alternative models that can be tested in the real world.

The object-oriented programming approach is flexible in application to a variety of subject areas in ecology. Hogeweg and Hesper (1985) demonstrated its use in modelling social networks where linkages are dynamic. The operation of parallel processes within separate objects has been used to model predator–prey interactions in habitat patches (Hogeweg and Hesper, 1981), and social units within a population of horses (Graham, 1986). The use of a hierarchical structure of classes to define properties of objects was useful in the structured population model of Graham (1986) and in our representation of habitat objects.

A model developed by Saaremaa et al., (1988) also represented spatial heterogeneity as patch-objects that could be queried by an animal object. That model illustrated concepts that could be used in representing foraging behavior and effects of browsing on the vegetation, as appropriate for modelling moose impacts on forest stands. Our model of deer movements adds one further level of realism by providing the animal-object with procedures for exploring and generating its own mental map that contains a subset of the habitat patches in the environment. Incomplete knowledge of the environment is a constraint that may limit the efficiency of resource use by animals.

Prior to application of object-oriented programming, there was no structured way of representing spatial knowledge in models of animal movements, although some vertebrates behave as if they have a mental map (Olton, 1977; Shettleworth, 1983). Studies of free-ranging mammals have been limited to addressing the cognitive map model in a largely anecdotal manner (e.g. Peters, 1979).

The object-oriented programming approach provides the potential for specifying alternative models about how an animal stores and updates spatial information. For example an information-decay process analogous to forgetting could be included. Social relations (e.g. between mother and offspring) could be included as an influence in modelling initial exploration and storage of a memory map. The utility of this approach is yet to be tested. Its success will lie in the ability to generate testable predictions of how movements would differ given alternative underlying models.

We chose to represent an actual study site in our model environment because we envision further developing the current movement model by successive comparisons of model predictions and radio-telemetry data. Incorporation of spatial information into animal movement models is of pressing importance in management of natural resources. We addressed a question of the effects of the mosaic of plant communities within an area of suitable habitat in this application regarding pasture treatments. Questions about the size and shape of habitat blocks protected in reserves are equally important. Initial attempts to model animal dispersal from refuges have been limited to use of probabilistic movements and grid-square representations of home range (e.g. Buechner, 1987). Application of object-oriented programming to such questions could generate more specific predictions utilizing knowledge of the configuration of existing reserves.

ACKNOWLEDGMENTS

We wish to thank Drs. Jack Inglis, Douglas Slack, and Ken Risenhoover for discussion and review of the manuscript.

REFERENCES

Anderson, D.J., 1982. The home range: a new nonparametric estimation technique. Ecology, 63: 103–112.

Banks, J. and Carson, J.S., II, 1984. Discrete-event System Simulation. Prentice-Hall, Englewood Cliffs, NJ, 514 pp.

Bekoff, M. and Mech, L.D., 1984. Simulation analyses of space use: home range estimates, variability and sample size. Behav. Res. Methods Instrum. Comput., 16: 32–37.

Berg, H.C., 1983. Random Walks in Biology. Princeton University Press, Princeton, NJ, 142 pp.

Bobrow, D.G. and Stefik, M.J., 1986. Perspectives on artificial intelligence programming. Science, 231: 951–957.

Buechner, M., 1987. Conservation in insular parks: simulation models of factors affecting the movements of animals across park boundaries. Biol. Conserv., 41: 57–76.

Coulson, R.N., Folse, L.J. and Loh, D.K., 1987. Artificial intelligence and natural resource management. Science, 237: 262–267.

Dixon, K.R. and Chapman, J.A., 1980. Harmonic mean measure of animal activity areas. Ecology, 61: 1040–1044.

Graham, L., 1986. HAREMS: A generalized data base manager and simulator for barrier island feral horse populations. CPSU Tech. Rep. 32, National Park Service Cooperative Unit, University of Georgia, Athens, GA, 49 pp.

Hogeweg, P. and Hesper, B., 1981. Two predators and one prey in a patchy environment: an application of MICMAC modelling. J. Theor. Biol., 93: 411–432.

Hogeweg, P. and Hesper, B., 1985. Socioinformatic processes: MIRROR modelling methodology. J. Theor. Biol., 113: 311–330.

Inglis, J.M., Brown, B.A., McMahan, G.A. and Hood, R.E., 1986. Deer–brush relationships on the Rio Grande Plain, Texas. Texas Agricultural Experiment Station, College Station, TX, 80 pp.

May, R.M. and Seger, J., 1986. Ideas in ecology. Am. Sci., 74: 256–267

Nilsson, N.J., 1980. Principles of Artificial Intelligence. Morgan Kaufmann, Los Altos, CA, 476 pp.

Okuba, A., 1980. Diffusion and Ecological Problems: Mathematical Models. Springer, New York, 254 pp.

Olton, D.S., 1977. Spatial memory. Sci. Am., 242: 82–98.

Peters, R.P., 1979. Mental maps in wolf territoriality. In: E. Klinghammer (Editor), the Behavior and Ecology of Wolves. Garland, New York pp. 119–152.

Saarenmaa, H., Stone, N.D., Folse, L.J., Packard, J.M., Grant, W.E., Makela, M.E. and Coulson, R.N., 1988. An artificial intelligence modelling approach to simulating animal/habitat interactions. Ecol. Modelling, 44: 125–141.

Shettleworth, S.J., 1983. Memory in food-hoarding birds. Sci. Am., 248: 102–110.

Siniff, D.B. and Jessen, C.R., 1969. A simulation model of animal movement patterns. Adv. Ecol. Res., 6: 185–219.

Tanimoto, S.L., 1987. The Elements of Artificial Intelligence. Computer Science Press, Rockville, MD, 527 pp.

Tchamba, M.N., 1988. Vegetation mosaic and deer distribution in South Texas. MS thesis, Texas A&M University, College Station, TX, 77 pp.