Agent-based Models for Animal Cognition: A Proposal and Prototype

Elske van der Vaart[1,2]
[1]Theoretical Biology
University of Groningen
Kerklaan 30, 9751 NN Haren
The Netherlands
elskevdv@ai.rug.nl

Rineke Verbrugge[2]
[2]Artificial Intelligence
University of Groningen
P.O. Box 407, 9700 AK Groningen
The Netherlands
rineke@ai.rug.nl

ABSTRACT

Animal ecologists have successfully applied agent-based models to many different problems. Often, these focus on issues concerning collective behaviors, environmental interactions, or the evolution of traits. In these cases, patterns of interest can usually be investigated by constructing the appropriate multi-agent system, and then varying or evolving model parameters. In recent years, however, the study of animal behavior has increasingly expanded to include the study of animal cognition. In this field, the question is not just how or why a particular behavior is performed, but also what its 'mental underpinnings' are. In this paper, we argue that agent-based models are uniquely suited to explore questions concerning animal cognition, as the experimenter has direct access to agents' internal representations, control over their evolutionary history, and a perfect record of their previous learning experience. To make this possible, a new modeling paradigm must be developed, where agents' reasoning processes are explicitly simulated, and can evolve over time. We propose that this be done in the form of "if-then" rules, where only the form is specified, not the content. This should allow qualitatively different reasoning processes to emerge, which may be more or less "cognitive" in nature. In this paper, we illustrate the potential of such an approach with a prototype model. Agents must evolve explicit rule sets to forage for food, and to escape predators. It is shown that even in this relatively simple setup, different strategies emerge, as well as unexpected outcomes.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming – *program modifications*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *representations (procedural and rule-based)*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multi-agent systems*; J.3 [**Life and Medical Sciences**]: *biology and genetics*.

General Terms

Algorithms, Experimentation, Theory

Keywords

agent-based models, genetic algorithms, evolution, animal cognition, theory of mind

Cite as: Agent-based Models for Animal Cognition: A Proposal and Prototype, E. van der Vaart and R. Verbrugge, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

1. INTRODUCTION

Agent-based models enjoy quite some popularity in the field of animal ecology [8]. As they are uniquely suited to simulate individual decisions, virtual habitats, and many generations, they allow biologists to investigate questions concerning collective behaviors, environmental factors, and evolutionary scenarios. Verbal accounts of how a particular animal behavior arises can be implemented, then tested, to see whether a theory's assumptions can reproduce empirical observations. Generally speaking, the theories subjected to such 'simulation-based evaluation' can be captured by models involving a fixed repertoire of agent behaviors, where patterns of interest emerge from the interactions between individuals, different parameter settings, or the evolution thereof. Such simulations have provided insights into many different biological systems. In recent years, however, the study of animal *behavior* has increasingly expanded to include the study of animal *cognition*. As David Premack [13] puts it, '...virtually every month another cognitive ability, thought to be unique to humans, is reported in an animal'. In these studies, the question is not just how or why a particular behavior is evoked, but also what its 'mental underpinnings' are. Can animals reason? Plan for the future? Infer mental states? These are fascinating questions, which are, at their core, about what makes humans different from animals – or not so different after all. The problem is that it is very difficult to conclusively establish the presence of any of these mental abilities in subjects that cannot talk. No matter how 'cognitively impressive' a particular behavior *looks*, there are always two alternative explanations for a successful performance: The necessary actions may be *innate*, or they may have been learned previously, without any understanding of why they solve the problem. Take the experiment by Hare *et al.* [9], where the question is whether or not chimpanzees have any concept of *visual perspective*, that is, whether they can reason about what others can and cannot see. In this experiment, two chimpanzees are put in competition over two pieces of food, placed in a central compartment. Both chimpanzees are familiar to each other, with an established dominance relationship. In two separate side compartments, the subjects wait to be granted access, with no view of the baiting. Once the compartment doors are opened, the subordinate can see both pieces of food, while the dominant's view of one of the pieces is obstructed by a barrier (Figure 1.1).

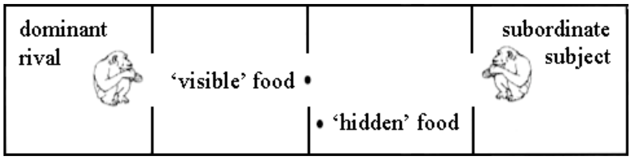


Figure 1.1. Hare *et al.'s* [9] experiment; image after [10].

If the subordinate chimpanzee rushes for the food that is also visible to her dominant competitor, she is unlikely to be allowed to keep it for herself; if, in contrast, she chooses to head for the food that only she can see, she can probably escape with it before the dominant chimpanzee catches her.

What Hare *et al*. find, is that the subordinate chimpanzee generally seems to head towards the hidden food. Therefore, the authors conclude, chimpanzees can reason about who sees what, implying that they have some insight into mental states, a ‘theory of mind’, once thought to be mankind’s prerogative. However, this is not the only way to interpret this result. Others [10] argue that, perhaps, chimpanzees have an innate preference for eating behind barriers. Or maybe, in the course of their daily lives, subjects had simply learned that they were more likely to feed successfully with an obstacle between themselves and their competitors - without understanding why. Although increasingly controlled experiments can help clarify the plausibility of various explanations, this debate continues until today [5], and shows how difficult the issue of animal cognition is.

It seems, however, that the agent-based approach might be exactly what’s required to shed light on the issue. In a model, it *is* possible to exactly control previous learning experience, as well as evolutionary history. In fact, with perfect access to every aspect of the simulation, the whole difficulty of linking observable behaviors to invisible mental states disappears. One can simply examine agents’ internal representations, and check. Of course, this requires that agents *have* internal representations. Current animal ecology models, with their emphasis on behavior, naturally lack such complicating details, and tend to focus on a single on / off variable, such as performing an action versus refraining from it [12]. Elegant as this type of multi-agent model is, its structure cannot be adapted for questions concerning animal cognition, such as “why might ‘theory of mind’ evolve”? The ability to reason about mental states simply cannot be captured by changes in a single variable. One would actually have to evolve agents’ *reasoning processes*. This is, of course, rather difficult, but it would also be terribly useful. The evolutionary origins of ‘theory of mind’ are one of the most hotly-debated subjects in animal cognition today. One prominent school of thought holds that *deception* [6] prompted selection for the understanding of mental states; the other the converse, namely that *cooperation* [11] was the driving factor. In an agent-based simulation, one can simply try both scenarios: Create agents who must either hide food from one another, or cooperate to obtain it. Which population evolves more ‘theory of mind-like’ representations? This is an overly simplified account, of course, but it illustrates the possibilities.

That leaves just one problem – how to evolve agents’ representations. One possibility would be to use neural networks as agents’ internal decision mechanisms, and to create artificial environments where ‘virtual natural selection’ could modify their composition and connection strengths, as is done in [1]. This would certainly allow different internal representations to evolve. Unfortunately, they would also be rather difficult to analyze, as neural networks do not explicitly represent the strategies that account for their success. Despite a large body of work on the subject [2], extracting rules from neural networks is still far from straightforward. In a model where the main purpose is to discover the internal representations that underlie behavior, this is a serious liability. Neural networks, then, are unsuitable for simulations of animal cognition. A second option is to draw inspiration from classifier systems [3], where agents must discover useful new behaviors by evolving their repertoire of decision rules. These consist of simple, pattern-matching classifiers, represented as strings of bits, which specify what actions agents should perform in response to different inputs from the environment, also represented as strings of bits. Although this technique allows natural selection to shape agents’ internal representations, it focuses on rules that compete and evolve *within* agents, rather than on populations of agents who must evolve rules to compete with each *other*. Also, the very low-level representation of agents’ internal decision mechanisms seems to preclude the simulation of moderately complex reasoning processes, which is what animal cognition is all about – especially in a multi-agent context.

Instead, we propose to evolve “if-then” rules to guide agents’ behavior. By specifying the form of such rules, and not the content, qualitatively different reasoning processes can emerge, which may be more or less “cognitive” in nature. For instance, some “if-then” rules might map directly from observations to actions, such as ‘*if* you see an agent, *then* hide your food’, while others might make use of intervening variables, such as ‘*if* an agent is oriented towards you, *then* it sees you’. With sufficiently flexible mechanisms for evolving and applying different “if-then” rules, cognitively complex representations can emerge, which may produce behaviors similar to those of real animals, exposed to different experimental paradigms. In our prototype model, agents must evolve explicit rule sets of a simpler nature, which allow them to forage for food and escape predators. Although the structure of our rule sets does not yet allow “more cognitive” strategies to emerge, even this rather simple setup leads to the evolution of different strategies and unexpected outcomes, illustrating the potential of our proposed approach. The rest of this paper is organized as follows. Our prototype model structure will be described in Section 2, and we will report on our experiments in Section 3. A discussion of our results is offered in Section 4, where we also compare our model to related approaches. General conclusions and future plans will be presented in Section 5.

2. THE PROTOTYPE

Our prototype model is intended as a first exploration of the possibility of evolving agents’ internal representations in a way that is suitable for simulations of animal cognition. These internal representations are implemented as “if-then” rule sets, whose form is fixed, but whose content and composition may be altered by mutation and selection. At present, we aim to demonstrate only proof of concept, and our research questions are modest. Can evolution of “if-then” rule sets produce agents that are reasonably well adapted to their environments? If so, do interesting strategies emerge? And how does the system react to changes in environmental variables? These issues are explored in a prototype model, which consists of four basic components: A Scape, Food, Grazers, and Predators[1]. Grazers are the agents of interest, who must evolve efficient rule sets to survive on the Scape: They must learn to find Food and escape Predators. On the Scape, time advances in cycles; each cycle, all Grazers and Predators can perform one action, and Food is redistributed across the Scape. A full simulation experiment consists of a tournament of 50 rounds of 800 cycles each, and is referred to as a ‘run’ of the model. Grazer rule sets are randomly initialized at the beginning of each tournament, and at the end of each round, the most successful

---

[1] In this paper, the names of model components are capitalized, to differentiate them from the concepts they represent.

agents are selected for reproduction, so that their descendants will populate the next round. Each of the basic model components, as well as the evolutionary algorithm, will be more extensively described below.

2.1 Scape and Food

The Scape is a continuous, torus-shaped world, on which Grazers, Predators and Food exist. It is represented by a rectangle of *xSize* by *ySize*, currently set to 1250 by 650 pixels. Scape inhabitants of all three types - Grazers, Predators and Food – have X and Y coordinates which determine their positions on the Scape. Food is the simplest of these, and it is represented by a stationary square of 3 x 3 pixels, its *foodSize*. It is what Grazers eat to obtain energy, at 50 energy (*startEnergy*) per Food. At the start of each round, 200 Food (*startFood*) are randomly distributed across the Scape. Food does not grow back, but is replenished up to *startFood* at the end of each cycle.

2.2 Grazers and Predators

Grazers are the focus of the prototype model. At the start of each run, 400 are released onto the Scape (*startGrazers)*. They are represented by a circle, 8 pixels in diameter, *grazerSize*. They have X and Y coordinates, as well as an orientation, O. Each cycle, Grazers can move a maximum of 5 pixels straight ahead, *grazerMoveDistanc*e, or change their orientation by a maximum of 5 degrees, *grazerTurnAngle.* Each Grazer has a field of view, a portion of the Scape that it has visual access to. This field of view is 90 pixels deep and 120 degrees wide; these are *grazerViewDistance* and *grazerViewAngle*, respectively. Grazers have five possible actions: Eat, move, turn, search and hide. 'Eat' is automatically performed whenever the Grazer intersects a Food. 'Move' affects the Grazer's X and Y coordinates as one would expect; 'turn' does the same for the Grazer's orientation O. 'Search' is a random combination of 'move' and 'turn'. 'Hide', finally, is a 'freeze' behavior, which makes Grazers invisible to other Grazers and Predators. Except for 'eat', the execution of these actions is guided by Grazers' rule sets, which will be separately discussed in Section 2.3. Each cycle, Grazers must burn 1 energy to survive – their *energyCost.* When their energy level drops below a *minEnergy* of 10, they die. Every rule costs an additional 0.1 energy to maintain (*ruleCost*), and every clause of those rules costs 0.2 energy to evaluate (*clauseCost*). Grazers do not reproduce, but instead are evolved in a tournament structure, which is explained in Section 2.4.

Predators are not unlike Grazers, in that they are represented by circles, have X and Y positions, an orientation called O and fields of view. But they are larger, faster and can see farther: their *predatorSize* is 10 pixels, *predatorMoveDistance* is 10 pixels, and *predatorViewDistance* is 140 pixels. *PredatorTurnAngle* and *predatorViewAngle*, by contrast, are equal to the corresponding values for Grazers, at 120 and 5 degrees, respectively. Predators exist solely to provide a risk of predation for Grazers, and as such do not need to be biologically plausible. They are randomly released onto the Scape at intermittent intervals. At any one time, the maximum number of released Predators is 0.1 * the current number of Grazers, the *predatorRatio*. Each cycle, if the maximum number of Predators is not yet present on the Scape, there is a chance of 0.1 percent, or *predatorOdds,* that an additional one will be released. Like Grazers, Predators can eat, move, turn and search. Unlike Grazers, they do not have evolving rule sets, and their behaviors are pre-programmed, as they are not the focus of attention. Each time step, every Predator moves towards the closest Grazer in its field of view. If it does not see any Grazers, it performs a 'search' action. If it intersects a Grazer, the Grazer dies. Grazers have only one defence against Predators: their 'hide' behavior. Predators cannot see Grazers that are hidden, and will pass over them unharmed. Predators do not have energy levels and cannot die of starvation – they simply disappear when they reach *maxAge*, which is currently set to 100 cycles.

2.3 Grazer Rule Sets

This prototype differs from other agent-based simulations primarily because of this feature: Grazer rule sets. Rather than specifying Grazer behaviors in advance, they are evolved, in the form of explicit "if-then" rules. Each rule consists of at least two, and a maximum of three, clause types, described in Sections 2.3.1 – 2.3.3. At the start of a run, all Grazers are initialized with three randomly configured rules (*initRules*), consisting of only VisibilityClauses and ActionClauses. Each cycle, each Grazer evaluates each of its rules until it finds one that is applicable to its current situation. As soon as it does, it executes the action that the rule specifies. Each rule is evaluated in a fixed sequence, so that the order of rules is relevant.

Table 2.1. Options for rule slots.

[object]	[object]*	[relation]	[action]
Grazer	Grazer(x)	angle-to	move(-to)
Food	Food(x)	distance-to	turn(-to)
Predator	Predator(x)		hide
	Me		search

2.3.1 VisibilityClauses

A rule's requirements are determined by its VisibilityClauses. Each rule has at least one, but may have several. Grazers are initialized with 1 to 3 per rule, *initVis*. A VisibilityClause stipulates what the Grazer must see for its rule to apply. Box 2.1 shows the 'recipe' for such a clause. Options for its [object] slot are displayed in Table 2.1. A VisibilityClause thus specifies a condition like 'if there is a Food visible', or 'if there isn't a Predator in view'. If a VisibilityClause dictates that the Grazer must see a particular type of object, the Grazer's rule evaluation mechanism attempts to match it to the closest visible object of the right type. This matching procedure allows other types of clauses in the rule to refer back to specific objects on the Scape.

Box 2.1. VisibilityClause 'recipe'.

IF [object] [==/!=] VISIBLE

2.3.2 ConditionClauses

ConditionClauses, for instance, must refer back to specific objects. ConditionClauses are optional; rules are initialized without them (*initCon* is set to 0), but mutation can introduce them later. This will be explained further in Section 2.4. ConditionClauses place further restraints on the objects matched by VisibilityClauses. Box 2.2 shows the 'recipe' for such clauses. Options for the slots [object]* and [relation] are displayed in Table 2.1. The * after [object]* indicates that it must be an object already matched by a VisibilityClause: ConditionClauses define relationships between *specific* objects. Thus, a rule with a ConditionClause might start with an expression like 'if there is a Grazer visible, and there is a Food visible, and the distance of the Grazer to the Food is less than 10'. For the [relation] slot, the *angle* between an object A and another object B is defined as the smallest number of degrees A would have to turn to be facing B. Food is considered to have an orientation of 90 degrees for this purpose. The *distance* between two objects is simply the length of the shortest straight line between them.

Box 2.2. ConditionClause 'recipe'.

AND [object]* [relation] [object]* [> / <] [*value*]

If a ConditionClause is not met, the Grazer will first try to match the rule's VisibilityClauses to different objects. If that doesn't help, the rule cannot be applied. A rule can have any number of ConditionClauses. If all specified conditions are met, a rule's requirements have been satisfied, and the Grazer will execute the action corresponding to the rule's ActionClause.

2.3.3 ActionClauses

Each rule has exactly one ActionClause, which specifies the action associated with the rule. ActionClauses come in three different varieties, shown in Box 2.3, with slot options displayed in Table 2.1. If an ActionClause's [action] is 'hide' or 'search', it requires no further specification. If it is 'move' or 'turn', then the clause must define how many pixels to move or how many degrees to turn – with a maximum of *grazerMoveDistance* and *grazerTurnAngle*, respectively. 'Move' is always straight ahead, while 'turn' can be either positive or negative, i.e. clockwise or anti-clockwise. Grazers can also move or turn *towards* a specific object, provided it is one of the objects matched by the rule's VisibilityClauses. These 'move-to' and 'turn-to' varieties of the [action] slot require a previously matched object to complete them. Of course, a Grazer executing this type of ActionClause will still only move *moveDistance* or turn *turnAngle* in the direction of the intended object. Once an ActionClause is executed, the Grazer stops evaluating its rules: Grazers can only use one rule per cycle.

Box 2.3. ActionClause 'recipes'.

THEN ∈ {[action], [action] [*value*], [action] [object]*}

2.4 Grazer Evolution

After each round of a tournament, the most successful Grazers are selected for reproduction. Specifically, the model keeps track of how much Food was collected, and then allows each Grazer to reproduce according to its contribution. All its descendants will have its exact rule set, or slight variations thereof, produced by mutation. The fitness measure thus cares nothing for how long Grazers manage to live, or if they are successful at avoiding Predators. However, the more cycles a Grazer survives on the Scape, the more opportunities it has for collecting Food, and the better it should do in the eventual ranking.

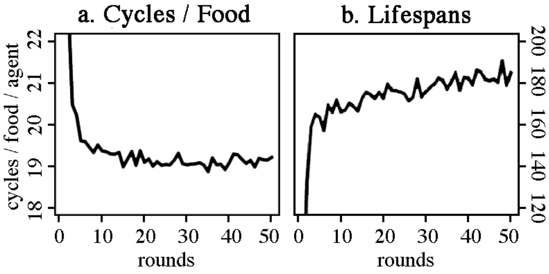
Mutation of Grazer rule sets occurs as follows. Rules may be duplicated, deleted or swapped, clauses may be duplicated or deleted, [object] slots may switch to any other legal value, [*value*] slots may be adjusted by up to 20% (or *mutateValue*) in either direction, and ConditionClauses may be spontaneously created. All of these events have a 1% (or *mutationOdds*) chance of occurring, although they have different effects. Mutations to VisibilityClauses are particularly potent. If the [object] slot of a VisibilityClause changes, all subsequent references to that [object]* are adjusted to match. This allows a rule like 'if a Food is visible, then move to the Food' to become 'if a Predator is visible, then move to the Predator' with one change in the VisibilityClause's [object]. Similarly, if a mutation causes an ConditionClause to refer to an unmatched object, it is deleted; if it causes an ActionClause to do so, a new ActionClause is generated. Thus, 'illegal' rules are repaired. In contrast, 'impossible' rules, such as those starting with 'if Food is visible and Food isn't visible', are left as they are, giving the evolutionary algorithm something to work with.

3. EXPERIMENTS

The behavior of the prototype model described in Section 2 is investigated in three separate experiments. In Experiment 1, the simulation is configured exactly as specified; in Experiment 2, the number of Predators is first increased, then decreased, and in Experiment 3, the amount of Food is varied in the same fashion. In each experiment, we conduct 20 runs per configuration. As stated previously, three exploratory research questions are of primary interest. Does mutation and selection of "if-then" rule sets allow well-adapted agents to evolve? If so, does this produce interesting strategies? And finally, what are the effects of changes to environmental variables? Examining Grazer evolution in Experiments 1, 2 and 3 should provide a first set of answers. Sections 3.1 - 3.3 present an overview of our results, and we discuss their implications in Section 4.

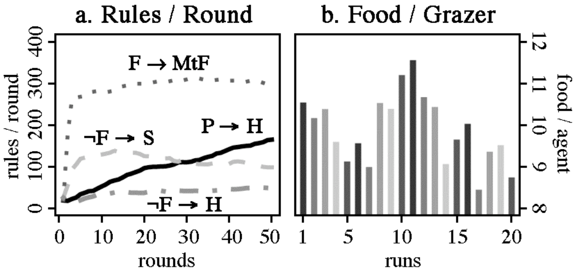
3.1 Experiment 1

In Experiment 1, Grazers definitely seem to adapt to their environment over time. As Figure 3.1 shows, Grazers become more efficient at finding Food (3.1a) and avoiding death (3.1b) as rounds progress. On average, Grazers obtain 1 Food every 24.7 Cycles during rounds 1-5, and improve their performance to 1 Food every 19.1 cycles during rounds 45-50. Likewise, their average lifespan increases from 136.4 to 182.3 cycles over the course of a run. But what rule sets explain their improvement? Generally, for a Grazer to survive on the Scape, three different behaviors are involved: going to Food if it is visible, looking for Food if it isn't, and evading Predators if possible.



Figures 3.1a and 3.1b. Grazer evolution in Experiment 1.

There are three straightforward rules that encode these behaviors – 'if Food is visible, then move to Food', 'if Food isn't visible, then search', and 'if Predator is visible, then hide'. The fates of these rules across rounds are plotted in Figure 3.2a, with an overview of the abbreviations used in Table 3.1. What is immediately apparent from Figure 3.2a, is that the rules for going to and searching for Food, F → MtF and ¬F → S, respectively, are heavily selected for at first, but then reach a plateau relatively quickly, while the rule about hiding from Predators, P → H, spreads slowly, but keeps increasing in prevalence. The same effect seems to be visible in Figures 3.1a and 3.1b. While foraging efficiency stagnates by round 15 or so, lifespans continue to improve. This suggests that Grazers solve the 'food finding problem' relatively easily, while the 'predator problem' keeps challenging them. However, none of the three 'basic rules', F → MtF, ¬F → S and P → H, reach full saturation. In fact, only in runs 1, 11 and 13 do a significant percentage of Grazers evolve rule sets consisting of the exact three basic rules, all in the order P → H, F → MtF, ¬F → S. However, not all runs evolve straightforward variations of the basic rule set, and some of these feature equally well-adapted Grazers.



Figures 3.2a and 3.2b. Further statistics of Experiment 1.

Figure 3.2b shows the average Food obtained per Grazer per round over rounds 45-50, which is essentially what the evolutionary algorithm selects for. It can be seen that the Grazers in runs 1, 11 and 13 do quite well, but that their performance is by no means exceptional. In Sections 3.1.1 - 3.1.3, we will discuss a selection of successful alternatives to the basic rule set, and provide a somewhat more detailed analysis of the runs in which they evolved. First, however, it must be noted that rules with ConditionClauses play no significant role in any of the runs, and will not be considered further until Section 4. As Grazers are initialized with rules that do not include ConditionClauses, they need to be introduced by mutation. Although this occurs, and a few Grazers with ConditionClauses occasionally manage to reproduce, there appears to be no specific selection for them.

Table 3.1. Rule abbreviations.

rule	abbreviation
'if Food is visible, then move to Food'	F → MtF
'if Food isn't visible, then search'	¬F → S
'if Predator is visible, then hide'	P → H
'if Food isn't visible, then hide'	¬F → H

3.1.1 Alternatives to the rule F → MtF

In Figure 3.2a, it looks like the 'if Food is visible, then move to Food' rule spreads to over three-fourths of the Grazer population by round 10, after which its prevalence stabilizes. The real situation is somewhat different: in fifteen out of twenty runs, the F → MtF rule reaches near perfect saturation, while an alternative rule does the same in the remaining five (runs 2, 3, 7, 8 and 18). This rule states that 'if Food is visible, and Predator isn't visible, then go to Food'. This is actually a rather interesting variation. On the one hand, it is a longer rule consisting of three clauses, as opposed to two, so it costs more energy to maintain and evaluate. On the other hand, it may be more energy efficient to use. Other Grazers, with the standard issue F → MtF rule, need to keep their P → H rule, or some variation thereof, as rule #1, to avoid being eaten. This means it always needs to be evaluated, although it is only rarely applicable, costing unnecessary energy. In contrast, 'if Food is visible, and Predator isn't visible, then go to Food' can be rule #1 without inhibiting the effectiveness of a later P → H rule.

3.1.2 *Alternatives to the rule ¬F → S*

Although Figure 3.2a shows that the standard ¬F → S spreads to only 25% of Grazers by round 50, the percentage of Grazers with an equivalent rule is much higher. Provided rule #1 and rule #2 are equivalent to P → H and F → MtF, rule #3 really only needs to express 'if the above two rules don't apply, then search'. ¬F → S is one way of doing this, but so is 'if Predator isn't visible, then search' or even 'if Grazer isn't visible, then search', although that last version is less effective. On average, during rounds 45-50, Grazers spend approximately 20% of their cycles performing 'search' actions. Run 10 is the exception, where the 'move' action is performed in 30% of cycles, as a result of the rule 'if Food isn't visible, then move 5'. In terms of Food per Grazer, this is a very successful run, as can be seen in Figure 3.2b. Although they are hardly more efficient at finding Food than the Grazers of other runs (obtaining one Food every 18.8 cycles during rounds 45-50, as opposed to the average of one Food every 19.2 cycles; one-sample t-test, true mean is not equal to that of run 10, $p = 0.05$), they live for a rather long time (206.7 cycles for the final Grazers of run 10, versus 182.8 cycles otherwise; one-sample t-test, true mean is not equal to that of run 10, $p < 0.0001$). The most straightforward explanation for this fact is that they are relatively successful at avoiding Predators. This could be due to their 'search speed'. 'Move 5' puts the maximum amount of distance between a Grazer and its previous position, while 'search' is a random wiggle. Predators are much faster than Grazers, but the faster a Grazer moves, the more cycles it takes to catch, and the greater the chance that another Grazer will come by to distract a chasing Predator.

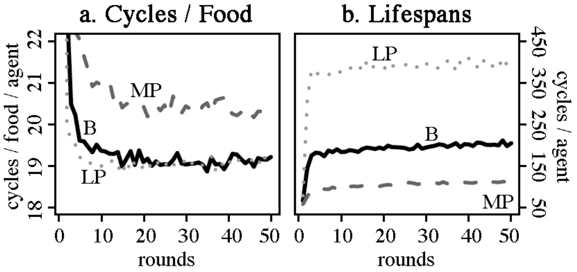
3.1.3 Alternatives to the rule P → H

The most notable alternative to the rule 'if Predator is visible, then hide' is actually 'if Food isn't visible, then hide'. This is a better rule than one might at first expect, as Food appears randomly across the Scape. Essentially, Grazers with this rule are following the strategy 'don't show yourself unless you have a reason to'. In two runs (4 and 6), ¬F → H functions as the main 'predator avoidance rule', as it is present in over 75% of rule sets by the end of the simulation. In another three runs (14, 16 and 17), it represents a significant alternate strategy, as it is possessed by 20 – 25% of Grazers during rounds 45-50. The main problem with the ¬F → H rule is that it occupies a Grazer's 'if Food isn't visible' condition, so that 'if Food isn't visible, then search' can never be successfully evaluated. As a result, most Grazers of runs 4 and 6 have F → MtF as rule #1, ¬F → H as rule #2, and 'if Predator isn't visible, then search' as rule #3, which fulfills the same function. Of the runs where 'if Food isn't visible, then hide' reaches moderate prevalence, run 16 is a rather interesting case. Here, by rounds 45-50, 26.2% of Grazers feature ¬F → H in their rule sets, but often this is *in addition to* its P → H alternative, which is present in 93.3% of the population. This run, therefore, evolves a breed of 'ultra-cautious' Grazers, who live relatively long as compared to the Grazers of other runs (198.9 cycles each, versus 182.8 cycles on average; one-sample t-test, true mean is not equal to that of run 16, $p < 0.0001$) and are relatively inefficient at finding Food (taking 20.5 cycles per Food, versus 19.2 cycles on average; one-sample t-test, true mean is not equal to that of run 16, $p < 0.0001$). As can be seen in Figure 3.2b, this is not a bad strategy, with run 16 performing moderately well as measured by Food per Grazer. Although they take a long time to find Food, they live long enough to compensate.

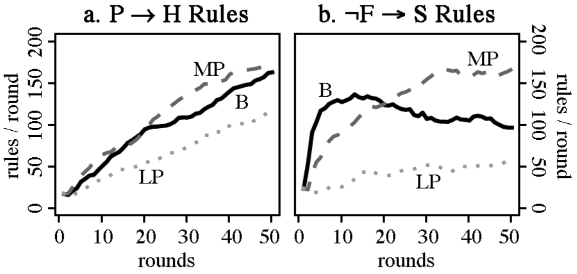
3.2 Experiment 2

In Experiment 1, Grazers clearly adapt as rounds progress, at least as measured by the increase in their lifespans and foraging efficiency. But are they really adapting to a specific environment, or are they simply evolving general purpose rule sets? In this experiment, we investigate that question by studying the effects of varying the number of Predators present on the Scape. First, in the More Predators (*MP*) condition, we raise both *predatorRatio* and *predatorPercent* to 0.3. By contrast, in the Less Predators (*LP*) condition, we lower both parameters to 0.025. Twenty runs of the model were done for each. This should replicate that Grazer statistics improve over time, but ideally, the rule sets evolved should also reflect the different environmental challenges posed.

As can be seen in Figures 3.3a and 3.3b, when compared to those of the Basic (*B*) setup of Experiment 1, Grazers in the MP condition take rather longer to find Food, and enjoy considerably shorter lifespans. To a degree, this is to be expected, as *MP* Grazers face three times as many Predators. What is surprising, however, is that the P → H rule is barely more popular in this condition than in the Basic setup, whereas the ¬F → S rule is clearly more heavily selected for, as is shown in Figures 3.4a and 3.4b. One would expect Grazers to respond to increased Predator pressure by evolving more Predator avoidance techniques, but instead, it seems they adapt with an increased focus on finding Food. With hindsight, this is understandable – if getting caught is inevitable, finding Food should be a Grazer's highest priority – but it is not what we would have predicted beforehand.



Figures 3.3a and 3.3b. Agent evolution in Experiment 2.

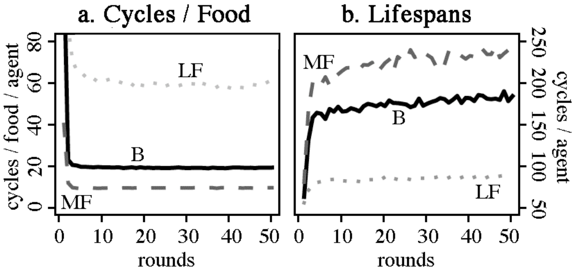


Figures 3.4a and 3.4b. Two rules in Experiment 2.

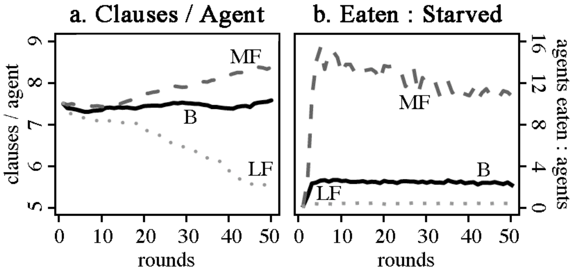
The situation in the *LP* condition appears to be the opposite. From Figures 3.3a and 3.3b, it is clear that Grazers confronted with less Predators live longer than their Experiment 1 counterparts, but that they are no more efficient at finding Food. In fact, as can be seen in Figure 3.4b, it seems they evolve *less* 'search for Food' rules, as opposed to more. Finding Food is Grazers' only significant challenge in this setup, and yet, they do not adapt to it in any obvious fashion. It seems that with greatly reduced predation pressure, Grazers can afford to wait until Food appears in their fields of view, rather than needing to actively look for it. In fact, many Grazers appear to have F → MtF as their only truly useful behavior, with their other rules producing seemingly random effects, such as 'if Grazer visible, then turn 2'. Taken together, this replicates an observation made during Experiment 1: That the 'food finding problem' is too easy in this model, and that Grazers require a significant predation risk to drive their evolution.

3.3 Experiment 3

In Experiment 2, Grazers respond to variations in Predator density by evolving qualitatively different rule sets. Interestingly enough, the changes are in unexpected, though understandable directions. In Experiment 3, we investigate the effects of varying the prototype's other significant environmental variable, which is *startFood*. In Experiment 1, the Scape's Food supply is continuously replenished up to 200 items; we now limit *startFood* to 50 in our Less Food (*LF*) condition, and increase it to 600 in our More Food (*MF*) condition. We perform 20 runs of the model for each new setup.



Figures 3.5a and 3.5b. Agent evolution in Experiment 3.



Figures 3.6a and 3.6b. Notable features of Experiment 3.

As Figure 3.5 shows, Grazers who must evolve in the harsh environments of the *LF* condition are slower to find Food (3.5a) and quicker to die (3.5b) than those in the Basic setup of Experiment 1. Nevertheless, if we look at the rule sets Grazers evolve in this condition, they seem to be perfectly adapted to their surroundings. ¬F → S is twice as popular as in the basic setup, while P → H is only present a tenth as often. In fact, what happens is that Grazer rule sets are increasingly pared down to just two rules, F → MtF and ¬F → SfF. This is illustrated in Figure 3.6a, where the number of clauses per Grazer drops to less than six. With a minimum of two clauses per rule, this implies that Grazers must have less than three rules on average. Grazers in this setup simply cannot afford to pay the energy costs associated with three rules, or to spend time hidden instead of searching for Food.

By contrast, in the *MF* condition, with three times as much Food on the Scape, Grazers live significantly longer and find Food considerably quicker than in Experiment 1. Again, though, their rule sets vary in expected directions. During rounds 45-50, ¬F → S is only a quarter as prevalent as in the Basic condition, while P → H is equally popular. Likewise, the ¬F → H rule makes it to over 75% of Grazers in five runs of the More Food setup, which is true of only two runs in Experiment 1. As can be seen in Figure 3.6b, the *MF* condition is the only one where the ratio of Grazers eaten relative to Grazers starved keeps decreasing over 50 rounds, indicating that they are actually learning to escape Predators without increasing their chances of starvation. Presumably, that is because these Grazers can afford to spend a lot of time hidden, a benefit which they evolve to exploit in later rounds.

4. DISCUSSION & RELATED WORK

The results presented in Section 3 indicate that, indeed, the evolution of "if-then" rule sets allows agents to become better adapted to their environments. In all experiments conducted, Grazers improve their foraging efficiency as rounds progress, and survive longer every successive generation. In that respect, the technique works. Evolved Grazer rule sets also vary sensibly under different environmental conditions: more Food decreases the popularity of 'search for Food' rules, while less Food selects for shorter, more compact rule sets. This is an interesting result, as it indicates that "trying different evolutionary scenarios" with this type of model structure is feasible.

To return to the example mentioned in the Introduction, if this type of simulation were to be applied to the question of 'why theory of mind evolved', this prototype version suggests that, at the very least, the model should produce qualitatively different rule sets in response to competitive and cooperative challenges. If the 'rule recipes' available to the evolutionary algorithm were of sufficiently rich form, different degrees of "cognitiveness" might be ascribable to the rule sets evolved. But would those rule sets be interesting? Could they truly provide genuinely new insights? Our prototype suggests they might. Even in our very simple setup, different Grazer strategies emerge. In Experiment 1, some runs evolve "ultra-cautious" Grazers, while others produce rasher inhabitants. Likewise, a more advanced version of this type of model might produce more 'theory of mind'-like and less 'theory of mind'-like strategies for social survival, even for agents evolved in the same scenario. One might imagine that one of these would best fit chimpanzees, while the other would be characteristic of macaques, or capuchins. On the other hand, an unexpected simulation result may also lead to the formulation of completely new theories, provided that the result is explainable once found. This type of outcome is certainly possible, as demonstrated by Experiment 2, where Grazers seem to react to increased Predator pressure by becoming *less* focused on avoiding them.

These are all promising results. However, the prototype model also exhibits various structural shortcomings, of which we will discuss the most important: in our runs, rules with Condition-Clauses were never selected for. Intuitively, however, there seem to be many useful examples, such as 'if a Predator is visible, and the Predator's angle to me is less than 70 degrees, then hide'. This rule would allow Grazers to hide only when the *Predator* sees *them*, rather than whenever *they* see the *Predator*. Given that the 'hide' behavior is immediately effective, this seems to be a risk-free strategy. Such a rule, however, is never discovered. Presumably, the problem lies with the low number of ConditionClauses available for selection, as they must arise by mutation. The alternative, of allowing the initial rule population to include ConditionClauses, is also not attractive, as it lowers the chance of any first-generation Grazers having even somewhat useful rule sets. It is possible that choosing a higher value for *mutationOdds* would do the trick, but a more extensive overhaul might be more fruitful. One important limitation of our prototype is the rigidity of the rule evaluation mechanism, and improving this aspect of the model may also promote the success of ConditionClauses. In the current implementation, Grazers test their rules' applicability one by one, and execute the first that fits. As this occurs in a fixed order, a Grazer can only benefit from one rule per matching "if" condition. This seriously limits the possibility of gradually evolving ConditionClauses. Addition of a random ConditionClause is likely to make a rule worse, not better, and within the current setup, it is highly unlikely that a 'backup' rule will be available. As an example, take the rule 'if Food visible, then go to Food'. Once this becomes corrupted by an inconvenient ConditionClause, such as 'and the Food is more than 25 pixels away', its Grazer is unlikely to have any other rule which allows it to approach Food. As a result, many Grazers with freshly mutated ConditionClauses are unlikely to reproduce, providing the evolutionary algorithm with very little opportunity to improve them over time. If, by contrast, Grazers could flexibly evaluate their rule sets, and maintain a number of alternatives for any situation, introduction of an unsuitable ConditionClause would be far less lethal. This kind of flexibility could also be desirable for other reasons, as complex reasoning cannot be captured by a sequential decision tree. Ideally, Grazers should be able to prioritize different rules in different circumstances, or to execute several rules in sequence, so that chains of reasoning may be evolved.

One model which implements such flexible rule evaluation is J.J. Grefenstette's SAMUEL [7], which is an agent-based system for learning sequential decision tasks. In [7], it is described how an airplane must learn to escape an approaching missile, by outmaneuvering it until it loses speed. This is done by creating a population of rule sets, which must compete with each other to save the plane as often as possible. Each rule set is tested on a selection of different starting configurations; successful strategies are reproduced, and crossover can exchange rules between sets. Within each set, different rules compete with each other to be executed. This occurs on the basis of past utility, in a manner somewhat similar to classifier systems [3]. What makes SAMUEL so interesting with respect to our model, is that it features comparable high-level rule representations, of which an example is presented in Box 5.1.

Box 5.1. Example SAMUEL rule, from [7].

IF time = [5, 15] AND fuel [low, medium]
AND speed = [600, 800] AND range = [400, 800]
THEN SET turn = right [0.9]

Like a typical classifier system, SAMUEL evolves its rule sets within a single agent, and its control structure is probably too complicated to be directly transferrable to a large scale multi-agent system, which is what an evolutionary simulation of animal cognition would require. Nevertheless, it features several mechanisms which could significantly improve our prototype model. First, a variation on the way in which SAMUEL's rules compete for execution within a single rule set might provide our agents with the cognitive flexibility they require, and create the opportunity for ConditionClauses to evolve. Second, SAMUEL's rule creation mechanism is significantly more sophisticated. It can not only mutate, delete, and alter rules, it can also merge, specialize, and generalize them, based on the circumstances in which they are executed and how useful they prove to be. As our model is not intended to simulate *how* evolution of a particular cognitive skill occurs, but under which circumstances it is likely to do so, this might be a useful addition to our prototype as well. Finally, Grefenstette *et al*. make a point which applies equally well to their design as to ours: Given the high-level rule representations involved, it is not only possible to analyze what comes *out of* evolutionary simulations, but it is also possible to specify what goes *into* them. In their case, they can initialize airplanes with the escape scenarios favored by human pilots, and see if simulation can improve on them; in our case, we can outfit agents with basic survival mechanisms, then see if artificial evolution can produce cognitively more advanced strategies. This seems like a worthwhile avenue to explore in future - in our models, but also in complex multi-agent systems generally.

As for other related work, although we are unaware of any other attempts to study animal cognition using evolutionary agent-based simulations, in [4], Bosse *et al*. construct a formal BDI-based agent model of animal 'theory of mind'. Our two approaches are very different, however, as [4] takes the ability of non-human primates to use 'theory of mind' as a given, and then attempts to formalize it, while our aim is to build simulations that can play a role in the discussion of whether they possess aspects of 'theory of mind' at all, and if so, which ones. The evolutionary component of our work also sets it apart.

5. CONCLUSIONS

In this paper, we have argued that agent-based models can offer a unique perspective on many questions concerning animal cognition, provided that new techniques are developed. The central problem facing empirical researchers in this field is that it is impossible to directly infer the mental representations that underlie observable behavior, so that it is very difficult to conclusively establish how "cognitively impressive" any particular performance really is. In a simulation, however, this problem does not exist, as the experimenter has direct access to agents' internal representations. This suggests that a modeling approach to animal cognition might be extremely fruitful. One issue that is currently receiving a great deal of scientific attention is the degree to which animals may have a 'theory of mind', and specifically, what ancestral conditions may have provoked such an ability to evolve. We proposed a modeling technique that could, in theory, simulate this process. By creating agents with "if-then" rule sets as their internal decision mechanisms, we can allow "virtual natural selection" to modify their content and composition, and investigate which evolutionary scenarios promote the emergence of agents with more "theory of mind like" rule sets. We demonstrated the potential of this technique in a prototype model, where agents were exposed to two selective challenges: avoiding predators, and finding food. Even in our very simple setup, different strategies emerged, and agent rule sets varied suitably under different environmental conditions. With sufficient extensions to the model, it should be possible to simulate real evolutionary scenarios proposed by animal cognition researchers, and investigate their plausibility as explanations for the evolution of 'theory of mind'. A first step towards realizing this goal would be to outfit the prototype's agents with more flexible rule evaluation mechanisms, allowing them to prioritize different rules in different situations.

6. ACKNOWLEDGEMENTS

Elske van der Vaart would like to thank Tim Dorscheidt for many spirited discussions, and NWO, the Netherlands Organisation for Scientific Research, for funding her doctorate, grant 021.001.089. Furthermore, Rineke Verbrugge gratefully acknowledges the NIAS, the Netherlands Institute for Advanced Sciences, and NWO, grants 051-04-120 and 400-05-710, for enabling her to work on this project during her time as a NIAS fellow of the theme group 'Games, Action and Social Software'.

7. REFERENCES

[1] D. Ackley and M. Litmann. Interactions between learning and evolution. In *Artificial Life II: Proceedings Volume X of the Sante Fe Institute Studies in the Sciences of Complexity*, 487-509, Reading, MA, Addison-Wesley, 1992.

[2] R. Andrews, J. Diederich and A.B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems* 8(6): 373-389, 1995.

[3] L.B. Booker, D.E. Goldberg and J.H. Holland. Classifier systems and genetic algorithms. *Artificial Intelligence* 40(1-3): 235-282, 1989.

[4] T. Bosse, Z. A. Memon and J. Treur. Modelling animal behaviour based on interpretation of another animal's behaviour. In *ICCM '07: Proceedings of the 8th International Conference on Cognitive Modeling,* 193-198, London, Taylor and Francis, 2007.

[5] J. Bräuer, J. Call and M. Tomasello. Chimpanzees really know what others can see in a competitive situation. *Animal Cognition* 10(4): 439-448, 2007.

[6] R. W. Byrne and A. Whiten. Cognitive evolution in primates: evidence from tactical deception. *Man* 27(3): 609-627, 1992.

[7] J. J. Grefenstette. The evolution of strategies for multiagent environments. A*daptive Behavior* 1(1): 65-90, 1992.

[8] V. Grimm and S. F. Railsback. *Individual-based Modeling and Ecology*. Princeton University Press, Princeton, NJ, 2005.

[9] B. Hare, J. Call, B. Agnetta and M. Tomasello. Chimpanzees know what conspecifics do and do not see. *Animal Behaviour* 59(4): 771-785, 2001.

[10] M. R. Karin-D'Arcy and D. J. Povinelli. Do chimpanzees know what each other see? A closer look. *International Journal of Comparative Psychology* 15(1): 21-54, 2002.

[11] H. Moll and M. Tomasello. Cooperation and human cognition: The Vygotskian intelligence hypothesis. *Philosophical Transactions of the Royal Society B* 362: 639-648, 2007.

[12] J. W. Pepper and B. B. Smuts. The evolution of cooperation in an ecological context: An agent-based model. In T. A. Kohler and G. J. Gumerman, Eds., *Dynamics in Human and Primate Societies: Agent-based Modeling of Social and Spatial Processes*. Oxford University Press, Oxford, 2000.

[13] D. Premack. Human and animal cognition: Continuity and discontinuity. *Proceedings of the National Academy of Sciences of the USA* 104: 13861-13867, 2007.