P1: MRM/SJS P2: MRM/SJS QC: MRM/SJS T1: MRM
KI076-12 May 25, 2004 10:47 Char Count= 0

PART II. A DEVELOPMENTAL PERSPECTIVE ON LANGUAGE AND DISCOURSE

12. TEACHING AND ARTIFICIAL LIFE

TZUR SAYAG AND SIDNEY STRAUSS
Tel Aviv University

This chapter is being written with Ruth Berman in mind. It deals with aspects of cognition and its development that have captured her heart and mind over the course of her academic career in Israel. Her pioneering work in the structure of the Hebrew language has inspired two generations of students and colleagues, both in Israel and abroad who are continuing to develop theory and research in that area. The purview of her work also includes language acquisition, which was an important departure from her prior research area, not only for her, but also for the area of the acquisition of Hebrew as a first language. And then she made yet another change, this time to children's understanding and production of texts. Ruth was virtually alone in Israel, a pioneer, in all these endeavors, something that motivated many to follow the path she set. Along with her groundbreaking work on the Hebrew language, she forged research communities both in Israel and abroad. And she did so with an easy smile on her lips and friendship in her heart.

Our work looks at theory concerning a natural cognition – not language, but teaching. We are attempting to establish the nature of the cognitive prerequisites for teaching, and much of what Ruth has done serves as a guide for us, as it has for so many.

Teaching is a domain that can be investigated under the examining eyes of cognitive scientists. Strauss (in press; Strauss, & Ziv, 2001; Strauss, Ziv, & Stein, 2002) argued that because teaching is remarkably complex, universal, evolutionarily beneficial, typically found in human beings, mostly invisible, not taught, and easily learned by young children, it can be thought of as a natural cognition. We believe that these claims along with the everlasting interest in human learning, some of which is, at least at first glance, a mirror image of teaching, make a strong case for formal research focused on this remarkable social phenomenon – teaching.

Strauss (in press) proposed a wide-ranging research agenda whose purpose was to pinpoint the cognitive prerequisites that enable teaching. A part of the research agenda includes the evolutionary perspective, where teaching could be speculated to be an evolutionary solution for survival. A second part of this agenda involves the use of computational models that might bring about rigorous definitions and descriptions of teaching that are currently lacking. Our work in this chapter brings these two strands together in the case of teaching, and we do so through the Artificial Life (ALife) research paradigm.

ALife is the study of all phenomena of the living world through their reproduction in artificial systems (Langton, 1995). It is a relatively new research paradigm that is based on computer-based experiments in which virtual populations of living creatures, that are capable of perceptual input, can interact with and evolve in virtual environments. The special properties of this paradigm allow the scientist massive flexibility rarely found in other non-analytical research methods. ALife experiments combine both a computational model approach and a strong relation with models of evolution. The core engine upon which an ALife individual changes is learning and an ALife population changes is natural selection.

The main point in this short chapter is that ALife could be used to investigate some issues in teaching theory and research. Furthermore, we suggest that using ALife could potentially reveal hidden properties of teaching, properties that are difficult to observe in our natural environment. The idea to use ALife models of learning and evolution as a set of tools for the investigation of cognitive developmental issues, such as aspects of Piaget's theory, was recently suggested by Parisi and Schlesinger (2002). Our arguments closely follow some of their ideas.

This chapter has four sections. Section 1 gives a brief description of teaching research and formal methods. In section 2, we describe neural nets, artificial life, and evolutionary algorithms. Section 3 presents both the benefits and pitfalls of the formal approach to human behavior research. In section 4 we show how ALife can be used for teaching research. And we summarize the chapter in section 5.

TEACHING RESEARCH AND FORMAL METHODS

Teaching is a complex and elusive concept. One indication of its ambiguity is that, from the time of the ancient Greeks to this day, there is no single, widely accepted definition of teaching. Along with the advantages of ambiguity, this state of affairs presents a drawback for research, e.g., the lack of a rigid, accepted working definition makes it difficult to compare results from experiments because they are sometimes based on different definitions. One example that comes to mind is the on-going debate about whether teaching exists among non-human primates. Surely, with an agreed-upon definition of teaching, this debate would be reduced to a trivial problem. Of course, the definition one chooses is based on one's theoretical framework, which complicates matters.

As opposed to the natural language, intuitive and flexible way of defining concepts, formal methods postulate a different approach. In a formal framework, concepts and terms are carefully stated in a symbolic and non-ambiguous manner. While a formal approach avoids some of the problems just mentioned, it is far from being ideal. Nevertheless, despite its attendant problems, as we note in Section 3, the formal method is unquestionably an extremely powerful research approach.

One interesting area, where using formal methods made major breakthroughs, is research on language and language acquisition. By using *formal language theory* as a mathematical description of language and grammar, and integrating results from this theory and linguistics research, Chomsky was able to develop his Universal Grammar theory, which is one of the seminal achievements in the cognitive sciences over the past 50 years. These achievements were also based on the availability of detailed descriptions of natural languages that scholars from linguistics and developmental psychology, such as those by Berman (1978), made available.

When comparing the task of initiating a formal computational research program on teaching to building a computational model for, say, language, we face a decidedly challenging task. In language research, at least some of the basic building blocks are well defined, which lend themselves to formalization. Teaching theory and research, on the other hand, lag far behind. Currently, we have no clear idea about the nature of teaching, what it is comprised of, and what the basic components are of any teaching event or behavior, not to say how teaching is learned.

EVOLUTION, NEURAL NETS LEARNING, ARTIFICIAL LIFE AND GENETIC ALGORITHMS

ALife is the study of all phenomena of the living world through their reproduction in artificial systems (Langton, 1995). This means that one can simulate living phenomena in a computer, although sometimes physical artifacts (robots) are constructed that exhibit some of the behaviors of real organisms. Simulations are a new way of expressing scientific theories and hypothesis about the causes, mechanisms, and processes that underlie observed phenomena and, as such, they allow us to explain those phenomena.

The typical ALife experiment is based on two distinct processes, individual learning and evolution. Within the ALife framework, learning is the process in which tan agent's neural network adjusts its internal synaptic weights according to a combination of the current network input, output and a learning rule, which may vary. The second process is the evolutionary one. In a nutshell, we allow this process to select each generation's best fit individuals and create the next generation from them. The new generation is created by applying two sub-processes. The first is the combination of the synaptic weights of the parents' network into a new set of weights, and the second is the usage of mutation, a small random change that also changes the network's weights. These two processes interact. For a detailed discussion of the interaction between these two processes see David, Ackley, & Littman (1991) and Nolfi, Elman, & Parisi (1990).

ALife simulations address all sorts of phenomena of the living world, including the behavior, cognitive abilities and mental life of organisms. We cannot provide a detailed descriptions of all aspect artificial life, so instead, we focus on three main components of the framework. A more detailed description is found in Ruppin, (2002) and Mitchell & Forrest (1995).

The typical ALife framework is composed of three main objects: a population of agents, an environment (sometimes called an arena), and a fitness function.

Agents are neural networks (NNs). In a nutshell, a NN is a naïve implementation of a nervous system. A NN has at least 3 layers of basic units: (1) An input layer (sometimes connected to actual sensors) that receives the perceptual input, (2) one or more hidden layers that can be seen as a processing layer, and (3) an output layer which is usually connected to external controls, such as motors.

The internal units of the NN are connected to each other with virtual wires. Each wire has what is called a weight (some real number). Input is received via the input units and is transmitted via the internal units to the output layer that, in turn, causes some action to be performed. NNs learn to compute functions. For every input pattern accepted by the input neurons, the hidden layer performs some rather simple combination of its connection weights and produces an output to be carried to the output units. Sometimes, the network is trained by allowing it to compare its response with the "correct" response to the specific input. In this case, the connection weights are updated via a learning rule, with future similar input resulting in "correct" output.

The second component in the ALife framework is the environment. This is usually some sort of bounded arena where agents live. Its exact description must be given. As noted above, this is a computer program, and it cannot cope with partially defined terms.

The third component of an ALife experiment is called the fitness function. Understanding what the fitness function is requires explaining how the typical ALife simulation works. Initially, a population of agents with randomly initialized connection weights is placed in the environment. They are called generation0. At every time step, each agent performs one action (move forward for instance). After a predefined number of steps, agents are evaluated and receive a fitness score according to the fitness function. The top two agents are selected and allowed to reproduce, thus creating a new agent. And much like in real life, the newly created agent's NN is a combination of its parent's networks. Before reproduction is completed, the agent is duplicated and a small part of his NN is randomly mutated. The newly created agents are similar to their ancestors because they inherited most of their networks; however, they are different since a combination and mutation was performed. The new population, called generation1, is placed in the environment and the process continues. The simulation stops when the fitness function no longer shows signs of improving. The fitness function is some computable function, such as the number of times the agent has slammed itself into a wall when the task is to avoid walls and obstacles.

The ALife framework's simplicity along with the comparable simplicity of the genetic algorithm may lead to a somewhat misleading conclusion about its ability to solve complex tasks. Below is a short list of three remarkably sophisticated behaviors that agents perform resulting from these frameworks.

P1: MRM/SJS P2: MRM/SJS QC: MRM/SJS T1: MRM
KI076-12 May 25, 2004 10:47 Char Count= 0

Avoiding obstacles

Consider the task of moving around some arbitrary flat space at high speed trying to avoid external walls and other obstacles. Initially, you have no idea what your orientation is, where in that space you are or the location of the obstacles. Agents have been shown to perform extremely well in this task after proper training. These agents seem to first adjust their orientation to a point where the wall is to their right and then proceed at full speed keeping the external wall at a constant distance. For a detailed description of a similar experiment, see Kodjabachian & Meyer (1998).

Flocking behavior in birds

Reynolds (1987) investigated how flocks of birds fly, without central direction; i.e., a leader. He created a virtual bird with basic flight capabilities, called a boid. The computerized world was populated with a collection of boids, flying in accordance with the following three rules:

- Collision Avoidance: avoid collision with nearby flock-mates.
- Velocity Matching: Attempt to match velocity with nearby flock-mates.
- Flock Centering: Attempt to stay close to nearby flock-mates.

Each boid comprises a basic unit that sees only its nearby flock-mates and "flies" according to these rules.

These three rules were sufficient for the emergence of flocking behavior. The boids flew as a cohesive group, and when obstacles appeared in their way they spontaneously split into two groups, without any central guidance, rejoining again after clearing the obstruction.

Reynolds' (1987) model demonstrates the basic architecture of ALife systems, i.e., a large number of elemental units that are relatively simple can interact with a small number of nearby neighbors with no central controller. High-level, emergent phenomena resulting from these low-level interactions are observed.

Predator-prey

Floreano, Nolfi, & Mondada (1998) report an extremely interesting co-evolution based EAA experiment where two different populations (different basic skills) evolve and compete (see Floreano et al., 1998 and Nolfi et al., 1993 for detailed technical descriptions of this experiment). The complexity of predator and prey strategies observed in the resulting population is remarkable. Predators seem to wait for prey agents to come closer before they attempt to attack; prey and predator seem to change their strategies in accordance to their opponent's strategy.

ALIFE BENEFITS AND PITFALLS

A major advantage of the ALife framework over traditional research tools is that if one expresses one's theory or hypotheses in the form of a computer program, one is

P1: MRM/SJS P2: MRM/SJS QC: MRM/SJS T1: MRM
KI076-12 May 25, 2004 10:47 Char Count= 0

forced to be explicit, complete, and detailed. Otherwise the program won't run or the simulation will not produce the expected results. And because teaching is such an ambiguous concept, the use of the ALife framework forces us to avoid this ambiguity.

One pitfall of the Alife approach is that there are many simplifications involved in these models. While enabling one to address systems that would be otherwise too complex to investigate formally, it may well be that interesting and even vital components of a system are missed in that simplification.

ALife is a sophisticated search procedure within the task solution space. But as powerful as it is, it can never break the optimal performance boundary of the controlling network. It can be shown that a neural network simply cannot learn some tasks.

It should also be noted that evolutionary search for optimal solution within the ALife framework does not perform magic. An agent which is not equipped with wings that allow flying will not spontaneously start gliding in the air no matter how patient we are for this to occur. It will also not build wings by itself or develop an arbitrary perceptual ability. These are far beyond the capabilities of these rather simple controlling networks that are commonly used in typical ALife experiments. As computational power grows, we can expect to see increasingly more powerful networks allowing more and more complex behaviors to emerge; however, many tasks will always be beyond the reach of even the most sophisticated networks available, given the computational model currently used.

A COMPUTATIONAL APPROACH TO DESCRIBE THE NATURE OF TEACHING AND HOW TEACHING IS LEARNED

We now attempt to explore an ALife approach to teaching, in which we propose to explore idealized teaching situations and evaluate the effectiveness of various teaching methods. That is, we adopt the perspective of an artificial intelligence researcher or engineer in such a way that we can both explore designs for machines that are effective in solving teaching problems of scientific interest and evaluate these designs through mathematical analysis or computational experiments.

In the following section we propose to investigate the nature of teaching within the ALife framework. This topic encapsulates many smaller questions, some of which we believe can be addressed within that framework.

The nature of teaching

The questions we propose here deal with teaching both as a concept and a behavioral phenomenon. We suggest that, given a definition of teaching, the following questions, and others, may be addressed concerning the *concept* of teaching: What does it take for an agent to identify teaching behaviors in others? Is the task of identifying teaching behaviors a learnable task? What are the prerequisites for the identification of teaching potential situations?

Concerning the nature of teaching as a *behavioral phenomenon*, here are several questions that could be tested in an Alife context: What are the prerequisites (environmental- and agent-related) for teaching to take place? Does teaching behavior emerge spontaneously, given what its cognitive prerequisites are? What is the relationship between teaching and learning, e.g., is teaching an accelerator for learning?

Mapping teaching onto artificial life

In this section we discuss and develop an example to show that applying the methodology advocated by ALife research can be used to address some of the questions about teaching we just raised. In particular, we deal with the issue of mapping.

This section is organized as follows: First, we define mapping. Then we describe a common real life teaching scene – teaching to ride a bike. That is followed by presenting a partial mapping operator of some of the objects and actions that play a role bike-riding onto ALife A-objects and A-actions. And, finally, we complete the mapping to include some complex notions that were omitted before.

Mapping

Intuitively, a mapping transforms notions, objects and actions such as people and "learning" from real life into ALife notions, A-objects and A-actions that are precisely defined. Mapping is a projection of concepts from real life terms onto ALife terms.

A mapping M is called an improper or invalid mapping if one can point at least one relevant relationship between objects in real life that are not preserved in ALife, e.g., mapping two different people in real life onto one single undistinguished agent in ALife would be improper. While we hope that the direction from real life to ALife can be accounted for (it is under our control, after all), the opposite direction, i.e., accounting for relationships between objects in ALife to those in real life, is not trivial. Let's see how mapping can work in an example of teaching.

Teaching someone to ride a bicycle

One of the many tasks parents undertake is teaching their children to ride a bicycle. Let us observe some of the events occurring while Ron is teaching his six-year-old child, Danny, to ride a bike. Below is a typical incomplete description of the entire task:

1. Danny has seen his best friend riding his new bike.
2. He approaches his father, Ron, asking him to teach him to ride a bike.
3. Ron considers and replies that the weekend would be a good time to start.
4. Over the weekend, they go to the bike shop, buy a new bike and head to a nearby park for the teaching session.
5. Ron is not a professional bike trainer but he does have his own bike and can ride it. He spends some time planning how to teach Danny to ride, and he considers different aspects of riding such as balancing, safety and control that he believes are necessary for Danny to learn in order to ride his new bike.
6. Ron must also integrate his knowledge about Danny's physical, cognitive and learning abilities into his teaching plan.

P1: MRM/SJS P2: MRM/SJS QC: MRM/SJS T1: MRM
KI076-12 May 25, 2004 10:47 Char Count= 0

7. At some point, the teaching session starts, and Ron starts executing his teaching plan. While doing that, he has to make sure that Danny's attention is not drawn away, so he picks a quiet place in the park.
8. In the instructional part, Ron demonstrates some techniques and accompanies his demonstrations with verbal explanations. When he spots misunderstanding or confusion he repeats his demonstration, sometimes demonstrating a different technique more adequately. He is also attentive to Danny's frustration or fatigue and tries to deal with it. He foresees the obstacles and relaxes the balancing task by holding the bike while Danny tries to balance himself.
9. Danny, on the other hand, is watching, asking questions, listening to his father and tries to extract as much as possible. He wants to be taught. He has confidence that his father can teach him, and he attempts to follow his father's instructions.
10. Finally after a number of hours, Danny is skilled enough to ride his bike on his own, and both Ron and Danny are satisfied with the results.

Given the above ride-a-bike teaching example, consider the following description of a human teaching event. We argue that there are at least 5 different distinct sub-tasks performed by the teacher along the timeline:

1. *Identification* of a teaching potential situation. (start condition)
2. *Creating* the required environment for the knowledge transfer to be feasible (bike, park, Danny and Ron) (prerequisites)
3. *Planning* a teaching strategy.
4. *Executing* a teaching strategy while dynamically evaluating and adjusting it according to expected and unexpected events. (teaching actions)
5. *Evaluating* the learner's final knowledge to measure the success of the knowledge transfer event, and to decide that no more teaching is necessary. (stop condition).

Note that some of these sub-tasks may be trivial and take practically no time for certain tasks, e.g., the environment is setup properly.

We suggest that although teaching is an extremely complex task, at least some of the different sub-tasks can be naturally mapped onto an ALife experiment in a way that it will preserve some of our intuitive notions of teaching. If we are able to show a proper mapping of these 5 sub-tasks onto ALife, we will then have the benefits of using ALife as a research tool for careful, rigid analysis of teaching.

For simplicity sake, we now concentrate on finding a valid mapping only for executing a teaching strategy (task 4 above). The mapping of the other four is omitted. To be more explicit, we present a valid mapping between the objects and actions that play a role within the task of executing a teaching plan in real life onto ALife. More specifically, we have to come up with a mapping that precisely states for each object or action involved in Ron's execution of his teaching strategy, an A-object or an A-action in the ALife environment.

At first glance, the task of mapping seems extremely difficult. Ron's execution of the teaching strategy is remarkably complex. It is dynamic in that it changes according to Danny's performance, and involves numerous different actions and sophisticated cognitive processing in Ron's mind. It would be very helpful if we could, say, create a ride-a-bike ALife experiment and use it as basis for the mapping. However, its extreme complexity makes a full description impossible here, given space constraints. Instead, we make a slight detour from the ride-a-bike example and briefly describe a typical, relatively simple and well-studied ALife experiment that will serve as a ride-a-bike analogy to base our mapping upon.

Ruppin (2002) describes a prototypical ALife experiment in which simple agents live in a 30 × 30 cell arena. "Poison" is randomly scattered all over the arena. Consuming "Poison" results in a negative reward. "Food", the consumption of which results in a positive reward, is randomly scattered in a restricted "food zone" in the southwest corner of the arena.

The agents' behavioral task is to "eat" as much of the food as they can while avoiding the poison. The agents are equipped with a set of sensors, motors, and a fully recurrent artificial neural network controller. The agents have 5 sensor systems, 4 of which sense the grid cell the agent is located on and the three cells immediately ahead of it. Each of these probes can sense the difference between an empty cell and a cell containing a resource (either food or poison, with no distinction between them). The fifth probe can be thought of as a smell probe, which can discriminate between food and poison if either is present in the cell occupied by the agent.

The motor system allows the agents to go forward, turn 90 degrees in each direction, and attempt to "eat". Eating is a costly action, as it requires a time-step in a lifetime of limited time-steps. It has been shown that from the evolutionary algorithm agents emerge that learn to find food and avoid poison in this environment.

A partial mapping

Table 1 explicitly defines a partial mapping between the ride-a-bike example in real life and the food-poison experiment in ALife. We now spend some time explaining the details and rationale of this partial mapping, leaving the more complicated task of completing the mapping for later.

Table 1. Mapping from real life to Alife

Role	Real life	ALife
Task	Ride-A-Bike	Find "food", avoid "poison".
Participants	Ron (teacher)	A-ron – an agent from generation 10000.
	Danny (Learner)	A-danny – an agent from generation 0.
Environment and utility objects	Park, bicycles.	Arena, food and poison.
Learning	Sophisticated human learning	Artificial Neural Network learning algorithm.
Teaching strategy	Explaining Demonstrating, other teaching activities	To be shown later.

The mapping described in Table 1 preserves three important relationships between the participants and the task, namely:

1. Ron can initially ride-a-bike. A-ron can initially find food and avoid poison (A-ron is a member of an already trained population).
2. Danny initially does not know how to ride-a-bike. A-danny is initially a poor finder of food (since he is of generation0).
3. Danny is capable of learning to ride-a-bike. A-danny has been shown in food-poison ALife experiments to be skillful enough to eventually learn to find food.

This mapping preserves the relationship between the participants, Ron and Danny, who are two distinct figures in real life, A-ron and A-danny are two distinct agents in ALife. The park and the bicycle are mapped to the ALife arena as are the food-poison items. Finally, the learning process and activities Danny is undergoing in real life is mapped to the A-learning process that A-danny is undergoing in the ALife experiment. Fortunately, learning in ALife is rigidly defined as the change of connection weights in A-danny's neural network, so the mapping is, therefore, still valid.

Completing the mapping

We believe that the partial mapping described is proper; however, we still haven't accomplished what we are really after, i.e., creating a mapping that will enable us to observe the execution of teaching in ALife. We now face the complicated task of mapping the set of actions and available perceptual input that the teacher, Ron, has to his representative in ALife, A-ron.

While we cannot present a complete solution, we suggest one approach that may yield a feasible, proper mapping for this part. The idea is this: we set some constraints on Ron's teaching strategy, simplifying it as much as possible. To be more explicit, we try to simplify Ron's actions and perceptual input in real life as much as possible, while keeping his now-simplified behaviors within the confined domain of what we intuitively think of as teaching.

The first constraint we suggest is that Ron executes a very basic teaching strategy, namely, a sequence of demonstrations. Let's analyze what a demonstration is in real life. A demonstration is an explicit act that is similar to, yet different from, a non-demonstration action in the following ways:

1. The demonstration's goal is different from the non-demonstration's action's goal. Let's take opening a jar as an example. The goal of a demonstration is to show another how to perform an action, such as how to open the jar, whereas a non-demonstration action's goal is to achieve an end (opening the jar) that the action is performed to reach. The actions may be similar and in both cases the jar is opened, but the goal of the demonstration was to show how to do it to someone else, whereas the goal of the non-demonstration actions was to do it.
2. The demonstration actions may, in all likelihood, be slightly different than the non-demonstration action, e.g., it might be slower, emphasizing important aspects of how to perform the task at hand.

3. The demonstration actions sometimes require special technical alteration of equipment in order to deliver the important aspects of what is being demonstrated.

The above obviously too complex to preserve all these properties of differences, so we decided to build a mapping that will preserve the following differences:

1. A demonstration will give no reward to the performer.
2. A demonstration will retain its distinction from the non-demonstration action it is aimed to demonstrate.

For simplicity, let's further constrain Ron's teaching strategy to be composed of only a finite set of demonstration steps. For example, assume that holding the handlebar, riding in a straight line and turning right or left safely are the four available demonstrations Ron can perform during his execution of teaching.

We are now ready to complete our mapping. Specifically, we show a mapping between demonstrating in real life and A-demonstrating in ALife. Note that as opposed to the partial, rather trivial mapping in Table 1, which did not require any changes in the ALife experiment, this part does require some technical changes and rewiring of the ALife agents.

In real life, Ron has 4 distinct demonstration actions, and these are clearly distinguished from the non-demonstration actions. Following this property, we extend A-ron, such that for each action it can perform, it can also perform a demonstration action. For example, if A-ron can perform a Go-Left, we can tweak its neural network to allow it to also perform another distinct action namely A-Demonstrate-go-left. All in all, the altered, extended version of A-ron has exactly twice as many distinct actions compared to the original A-ron. Table 2 partially illustrates these categories.

What remains for us to do is to alter the ALife experiment in such way that: (1) if A-ron takes one of the demonstration actions, it will not be explicitly rewarded and (2) if A-ron takes a demonstration action, it can be identified and distinguished from the non-demonstration action.

Dealing with the reward issue is relatively simple to accomplish. The ALife environment allows us to simply ignore food items eaten in an A-demonstration action, and the same applies to his demonstrating a movement action. We can simply make

Table 2. Original and altered A-ron for A-demonstrations

Original A-ron behavioral repertoire	Altered A-ron behavioral repertoire
Go-Right	Go-Right A-Demonstrate-Go-Right
Go-Left	Go-Left A-Demonstrate-Go-Left
Go-Forward	Go-Forward A-Demonstrate-Go-Forward
Eat-Resource	Eat-Resource A-Demonstrate-Eat-Resource

sure the environment and his location are rolled back as if the action never occurred. Distinguishing the A-demonstration actions and the A-non-demonstration actions in a-Ron is a by-product of the non-trivial technical alterations in the neural network. It is worth noting that the technical alterations we omit here are non-trivial and may turn to be rather complex; however, a review of different neural networks used in ALife experiments lead us to believe that these alterations are feasible.

Let's sketch a review of what we have done here. We have two extended agents: A-ron and A-danny. One is skilled in finding food, and the other doesn't know how but can learn to find food in the arena. A-ron can also A-demonstrate every action in his behavioral repertoire.

In sum, we mapped the notion of demonstrating onto ALife, and argued that since demonstrating is a very basic teaching strategy, the notions of teaching are preserved. Furthermore, the relationship between Ron and demonstrating is kept across the mapping (since A-ron is able to perform an A-demonstrate action).

Although we are almost done, some things are still not quite right. For example, how can we make sure that A-danny is even aware of these demonstration steps and uses them as input for his learning process? Fortunately, ALife is not real life. Recall that in ALife we have full control over the perceptual input of every agent, so we can, for instance, implement some wiring such that whenever A-ron decides to take an A-demonstration action, A-danny's input and output units receive the exact input and output a-Ron was subject to. This will allow A-danny the ability to benefit from A-ron's demonstration. We emphasize, again, that while the technicalities of doing these alterations are non-trivial, these problems are solvable within the ALife framework.

Let's assume these technical problems are solvable. We now have an ALife experiment with notions of a teacher, a learner, a simple teaching strategy (demonstration) and a learnable task. We have created a well-defined computational model that preserves, at least partially, the notions of teaching. We can now explore some of the question raised earlier regarding teaching. For instance, we can explore whether evolution favors teaching over finding food with different fitness functions. We can evaluate the evolutionary benefits of teaching for the learner, e.g., is teaching helpful for the teacher? Is teaching an accelerator for learning? Will our teacher agents converge to interesting demonstration sequences? This is just a small sample of questions that are well defined within ALife. It is not idle speculation as to whether or not we can get answers for these questions. These questions should be fully answered by running the ALife experiment. The questionable part of our analysis is the validity of the reverse mapping of these results. We argue that while the mapping we described is simplistic, at least for the moment, it is rigid and feasible. Furthermore, we believe it to be is relatively easy to expand this mapping or similar mappings to a more complex and interesting one.

SUMMARY

We suggested that by using the research tools provided by the ALife framework, a better understanding of teaching could be obtained. Mapping teaching research topics onto the ALife framework is a non-trivial and challenging task that encourages formalization and concrete, rigid definitions. Furthermore, the process itself is an incredible opportunity to rethink and better state the actual research questions in terms that may be used for other research approaches. We also believe that if our mapping is sufficiently sophisticated and carefully crafted, some results obtained in the virtual ALife framework may possibly hold in the real world. That could give us an understanding of a most remarkable social phenomenon – teaching. And we might, perhaps, gain an understanding of whether or not teaching is a natural cognition that is learned by individuals and if it evolves in populations across many generations.

REFERENCES

Berman, R. A. (1978). *Modern Hebrew structure*. Tel Aviv: Universities Publishing.

David, H., Ackley, D. H., & Littman, M. L. (1991). Interactions between learning and evolution. In C. Langton, C. Taylor, J. D. Farmer, and S. Ramussen (Eds.), *Artificial Life II: Santa Fe Institute Studies in the Sciences of Complexity, 10,* 487–509. Redwood City, CA: Addison-Wesley.

Floreano, D., Nolfi, S., & Mondada, F. (1998). Competitive co-evolutionary robotics: From theory to practice. In R. Pfeifer, B. Blumberg, J-A. Meyer, and S. W. Wilson (Eds.), *From Animals to Animats V* (pp. 512–524). Cambridge, MA: MIT Press.

Kodjabachian, J., & Meyer, J. A. (1998). Evolution and development of neural controllers for loco-motion, gradient-following and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks, 9(5)*, 796–812.

Langton, C. G. (1995). *Artificial life: An overview*. Cambridge, MA: MIT Press.

Mitchell, M., & Forrest, S. (1995). Genetic algorithms and artificial life. In Langton, C. G. (Ed.), *Artificial life: An overview* (pp. 267–289). Cambridge, MA: MIT Press.

Nolfi, S., Elman, J. L., & Parisi, D. (1990). Learning and evolution in neural networks. *Technical Report CRT-9019*. Center for Research in Language. University of California, San Diego.

Nolfi, S., & Parisi, D. (1993). Auto-teaching: Networks that develop their own teaching input. In J. L. Deneubourg, H. Bersini, S. Goss, G. Nicolis, R. Dagonnier (Eds.), *Proceedings of the Second European Conference on Artificial Life* (pp. 845–862). Brussels: Free University of Brussels.

Parisi, D., & Schlesinger, M. (2002). Artificial life and Piaget. *Cognitive Development 17,* 1301–1321.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. Proceedings of SIGGRAPH 87, *Computer Graphics 21*, 25–34.

Ruppin, E. (2002). Evolutionary Autonomous Agents: A neuroscience perspective. *Nature Reviews: Neuroscience, 3*, 132–141.

Strauss, S., & Ziv, M. (2001). Requests for words are a request for teaching. *Behavioral and Brain Sciences, 24*, 1118–1119.

Strauss, S., Ziv, M., & Stein, A. (2002). Teaching as a natural cognition and its relations to preschoolers' developing theory of mind. *Cognitive Development, 17*, 1473–1487.

Strauss, S. (in press). Teaching as a natural cognition and its implications for teacher education. In D. Pillemer and S. White (Eds.), *Developmental psychology and the social changes of our time*. New York: Cambridge University Press.