EMERGENT FRAME RECOGNITION AND ITS USE IN ARTIFICIAL CREATURES

Luc Steels
Artificial Intelligence Laboratory
Vriie Universiteit Brussel
Pleinlaan 2, B-1050 Brussels Belgium.
email: steels@arti.vub.ac.be

Abstract

Artificial creatures are autonomous mobile agents that have to react in real-time to sensors and plan and perform actions in the real world. Current effective architectures for artificial creatures are behavior-based and use variants of the subsumption architecture. The paper proposes an extension to these architectures in terms of a layer which introduces cognitive capabilities to artificial creatures. This extension is frame-based and uses emergent frame recognition to determine which frame should become active.

1. Introduction

The construction of artificial creatures has been put forward recently, particularly by Brooks, as a new goal for AI. An artificial creature is an autonomous agent that is operating in the real world. It gets information about the environment through sensors, and not through symbolic tokens fed by a human, and it directly executes its actions in the world. The construction of artificial creatures poses a set of challenging problems which are central to intelligence but could be ignored in current knowledge systems research[1]. These problems include:

- The symbol grounding problem: how can symbols used by a symbolic reasoner relate to real world objects.
- The real time problem: how can signals from sensors be interpreted in real time, how can planning be performed under extreme time constraints, and how can the resulting plans be made reactive enough to cope in real time with unexpected circumstances.
- The situatedness problem: how can behavior be defined so that it fits flexibly and dynamically with the situation being faced.
- The programming problem: how can systems be built up in a modular fashion, i.e. how can new competences be added without having to revise earlier capabilities.

The solutions so far proposed by Brooks and his associates [Brooks, 1986] constitute quite radical departures from traditional AI and are closer to earlier cybernetic models [Braitenberg, 1984]. The main assuptions behind these solutions can be summarized as follows.

- There is no attempt to build a full centralized symbolic model of the world. Instead sensors are almost directly coupled to effectors in a distributed fashion.
- Sensor interpretation, planning, and execution are not put into separate components. Instead they are organized around particular competences each of which is implemented by a particular behavior.
- The behaviors are not explicitly controlled by a scheduler that reasons about when to do what, but they are linked in a network where one behavior may suppress immediately other behaviors. This way of organization is known as the subsumption architecture.
- More complex behavior is not programmed explicitly but emerges by the dynamic interaction of the simpler behaviors with the environment and among themselves.

Some effectively working artificial creatures have been demonstrated recently based on these principles.

We have subjected these assumptions to experimental investigation in a project that is building an artificial creature for the Barcelona science museum[2]. This creature, whose current prototype is nicknamed Lola, must operate in real time in the hostile environment of a museum to welcome, entertain, guide, and instruct visitors, mostly children. The creature needs to exhibit intelligence, language communication, a particular character, and emotions. These cognitive capabilities need to be implanted on a firm basis of behaviors that include wall following, obstacle avoidance, target tracking, etc. The hardware of Lola contains about a hundred sensors, various processors, a mobile base, and speech input and output.

---

[1] This does not mean that knowledge systems arc not useful or not intelligent.

[2] This project is the collective effort of many people at the VUB AI laboratory. The core team of hardware designers and builders consists of Pranky Keppens and Piet Ruysselinck. Francis

Although the behavior-based approach gives us ways to handle the physical parts of the task, it is in itself not adequate for the other aspects. For example, there are no elegant ways to internally represent objects with their properties (behavior-based systems react directly to sensors and have no notion of an "object"), there is no way to keep memory of past events and use that in action selection, it is not possible to classify situations into categories and decide on the basis of that what action to undertake, there are no "variables" which can be bound (filled) with arbitrary objects including objects that are not in view, therefore actions need to be represented for each possible instantiation or actions can only take place over objects or situations that are directly visible to the sensors. What seems clearly needed is an additional layer on top of a behavior-based layer that takes care of the more cognitive aspects. This layer nevertheless needs to remain grounded in the physical layers below it. It also needs to satisfy a number of requirements such as real-time performance and emergence of behavior.

We are exploring a frame-based architecture [Minsky, 1987] for the cognitive layer. This means that mechanisms like frames, slots, defaults, inheritance hierarchies, procedural attachment, and links between frames are available. Frame recognition is the primary mode of operation: the frame(s) that best fit the current situation are recognized and this triggers through procedural attachment the scheduling or the inhibition of behaviors at the behavior-based layer. The choice of a frame-based architecture (as opposed to a logic-based or rule-based one) is motivated partly by the ability of frames to represent in a packaged way complete situations including the behaviors that are appropriate in the situation and the sensor/effector couplings. The availability of slots overcomes also the lack of variables in earlier behavior-based architectures.

There is however a major unsolved problem in frame theory, namely how frames are supposed to be recognized. Exhaustive search is clearly excluded in an artificial creature that works under heavy time constraints and has limited amounts of processing power and memory[3]. Another approach assumes that a lot of domain-dependent heuristics are encoded. For example, the work of Schank, et.al. [Schank and Riesbeck, 1981] on recognizing which scripts (i.e. system of frames) applies in a given story, typically centers around the introduction of domain-specific information to make the frame recognition problem tractable. This solution is not usable in the present context because (1) it would introduce extra memory usage and computation, and (2) it requires that the "right" heuristics are thought up by human designers and programmed in. There are also some other disadvantages: brittleness (when appropriate heuristics are not present the system fails to recognize the frame), non-modularity (the addition of new behaviors requires revision of earlier heuristics), and inflexibility because there needs to be a prior categorization of the set of possible situations in order to express the heuristics.

van Ackcn is one of the chief designers of the functionality. The contributions of Maja Mataric and Rodney Brooks in the startup phases of the project are gratefully acknowledged. Jorge Wasserman, director of the Barcelona science museum, has been a champion of the project from the beginning.

We have therefore developed a new approach towards frame recognition which is based on the principle of emergent functionality (Steels, 1990). No mechanisms are programmed in to help find the frame that best matches. Instead, the most appropriate frame, i.e. the one whose slots best fit with objects found in the current situation, emerges as the winner after a dynamical process in which the different frames "fight" among themselves and interact directly with the world through behaviors and their associated sensors and effectors. This process has a gestalt-like quality. Also it is completely reactive in the sense that if the situation in the world changes, frame activations change dynamically with it.

The frame recognition system that is proposed in the paper constitutes a contribution not only to research on artificial creatures but also to frame theory in general. The emergent frame recognition system could be used also in story understanding, for example, in order to find the best matching frame system and the best fillers of all slots within the system. In this paper we concentrate however on the application to artificial creatures.

The rest of the paper is in two parts. The first part develops the proposed frame recognition system. The second part discusses an example. Some conclusions end the paper.

2. Emergent frame recognition

2.1. Initial definitions

Assume a collection of frames F. Each frame f € F consists of an n-tuple f = $<s_1 \ldots s_n>$. s =<slot-name, slot-filler-descriptionx A slot-filler-description is a list of propositions[4].

Assume also a collection of objects O. Each object has a list of propositions describing the object. Some of these propositions are causally connected to external or internal sensors. A proposition has a certainty c associated with, c is a real number between 1.0 and 0.0. New objects may be created at all times. New propositions may be added to an object at all times and the certainty c of a proposition may be changed at all times, for example under influence of sensors[5]

---

[3] The frame recognition problem and its solution through exhaustive search triggered the research on massively parallel computing in AI [Fahlman, 1979]. But although there are now massively parallel computers [Hillis, 1985] that can in principle do exhaustive search to find the best fit they are technologically too heavy for artificial creatures.

[4] A future version of the theory will incorporate also descriptions with variables which would be local inside a given frame system

[5] In the implementation operational on the robot, we assume a fixed set of objects and a fixed set of propositions for efficiency reasons.

Frames have two types of relations between them: specialization relations (the isa-hierarchy) and interframe relations. Two frames f1, f2 are connected by an isa-link if they share the same slots but f2 contains additional constraints on the fillers. For example, there is a general frame for object, with specializations for living and non-living object Living objects are further subdivided into group and single individual, single individual is subdivided into adult and child, etc. Frame recognition typically proceeds in a top-down fashion. For example, it is easier to recognize that an object is present (using the infra-red sensors) than to recognize whether that object is a child (which requires a combination of pyro-sensors, pitch detection, and prior expectation). The fact that frame-recognition is typically top-down is an emergent property however, simply because frames near the top of isa-hierarchies have less conditions and are therefore easier to satisfy than frames near the bottom of the hierarchy.

Interframe relations implement inhibition and enforcement relations between frames. For example, often a frame for an emotional state will be in direct competition with frames of other emotional states. Thus the anger-frame and the happiness-frame inhibit each other, whereas the frustration-frame enforces the anger-frame. The interframe relations can also be used to implement expectations (for example a frame for an action may enforce a frame for the next action in a sequence), or weak forms of planning (a frame for an action enforces frames for actions that could make its preconditions true).

Besides a set of slots and objects filling these slots, a frame contains also a set of behaviors which take the fillers of the slots as parameters. These behaviors are identical to the behaviors found in subsumption architectures. For example, the dance frame has an associated behavior that causes the robot to dance (while *still* avoiding obstacles, etc. - the subsumption architecture remains in force at the level of behaviors).

2.2. The theory

The frame recognition system uses the metaphor of flow of energy in a network. Every frame has a particular strength, called the frame-strength. It is a real number that indicates how well the frame fits with the present situation. The frame strength can also be viewed as an indication of how much energy (strength) the frame has. The initial strength comes from a partial fit of the frame with reality (for example, some of the slots could be filled) or from possible reenforcement through interframe relations. This helps the frame to pull other objects into its slots and to compete with adversaries for the same objects. As more objects fill the slots (or better fit the slots), the frame gains progressively more strength and is therefore able to pull even more. Interframe relations further influence positively or negatively the frame strength. A frame has been recognized if the framestrength is greater than a certain treshold (close to 1.0). When this *is the* case the behaviors associated with the frame are activated.

We call the relation between an object and a slot a link. The frame strength is partly determined by the strength with which a particular object fits within a certain slot. This is called the link strength. We now first describe the influences on the link strength and then the influences on the frame strength which is based on the link strength of its constitutive slots. Initially the frame-strength of all frames is set to a very low value (e.g. 0.01).

I. Impact on link strength

The link strength is determined in a series of steps:

1. The initial link strength m_j of the link j between an object o and a slot s of a frame f *is* based on the match between o and s. This match is a weighted sum based on how many properties are present and how high the certainty is of each property. It is similar to a perceptron-like decision process. Let $c_j = m_j$

2. Objects that fit in the same slot compete (figure la.) , *cj* is therefore qualified as follows:

$$c_j = c_j + i_j \; i_f$$

where i_f is equal to a (global) parameter known as the inhibition factor which determines how big competitive influence is between objects competing for the same slot. Furthermore

$$i_j = \sum_k (c_k - c_j)$$

where k ranges over all links from objects to the same slot s.

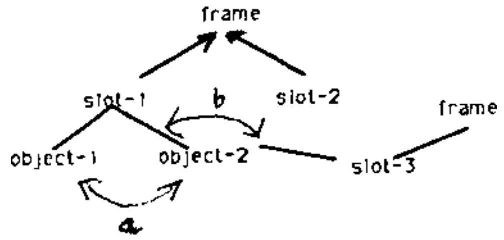

Figure 1. Types of competition between objects in a slot.

3. The link strength is influenced by the frame strength distributed proportionally over all slots in the frame leading to a qualified link strength *Sj*:

$$s_j = \frac{1}{\sum_i c_i} c_j^2 f_s$$

i ranges over all links coming out of the slot s of the same frame f. f_s, is the current frame-strength.

4. There is competition for link strength if the same object fits in more than one slot (figure lb.), leading to a qualified link strength q_j.

$$q_j = \frac{1}{\sum_i s_i} s_j^2$$

where i ranges over all links from an object o to any slot s.

5. There is a small random factor for avoiding dead-lock situations when two objects fit equally well in the same slot of the same frame and all else is equal.

$$r_j=q_j\ \rho$$

6. Frame strength is the sum of the best link strengths

$$f_s=\frac{1}{s\ f_s}\sum_m r_m$$

where m ranges over all slots in f and r_m is the best r value for m.

II. Impact on frame strength.

7. The frame strength f_s is further qualified by the excitatory or inhibitory relations between frames:

$$f_s=\sum_j \frac{f_j r_{i,j}}{n}+f_s$$

where j ranges over all frames related to f, $r_{i,j}$ is the excitatory/inhibitory relation between i and j and n is the number of frames related to f.

The computation keeps circling through the steps outlined above, starting with step 2, so that there is a continuous updating and interaction with the world.

III. Activation

There are two types of frames depending on the associated actions. The first type, called a releaser-frame, has actions that have an impact in the real world (e.g. frames activating dancing behavior, exploratory movement behavior, etc.). The frame is recognized if the framestrength is greater than a certain treshold (close to 1.0). When this is the case, the behaviors associated with the frame are activated and the frame itself gets a strongly reduced frame strength. Note that the functions that call the behaviors have variables which are determined on the basis of the contents of the slots. Thus there could be a frame for going towards a particular target. The associated behavior starts moving towards the target but the target itself is a variable based on which location fills the target slot in the frame. These frames are called releaser-frames because their operation implements the "releaser" behavior described by ethoiogists like Tinbergen. A second type of frame has actions that change the internal state of the agent, for example, frames that recognize what kind of object is there or that determine the emotional state of the agent. These frames are continuously active, in the sense that they continuously update the features they have an impact on.

2.3. Justification

The theory proposed above satisfies the general structure of systems with emergent functionality (Steels, 1990). We find the required feedback relation because link strength increases frame strength which increases link strength. How to find the best frame at a particular point in time is not programmed in but emerges by the dynamics which is at a level below the actual frames. This dynamics is influenced by the fit between a frame and the current situation and by the competition between frames. Notice that default-handling follows as a side effect of the proposed dynamics, because the link-strength between the default filler and the frame is very weak and therefore easily overridden by an object that may also fill the slot.

2.4. Implementation

Various implementations have been constructed of the system proposed here[6]. One runs on a Symbolics LISP machine and is essentially an environment for interactively exploring different frame systems. It includes a graphical interface and measuring apparatus to follow the dynamics. A frame language has been designed to make it easy to define frames.

A second implementation has been developed to run on physical mobile robots. This implementation is written in C and downloaded on on-board processors. The frames are compiled into C datastructures. The emergent frame recognition system runs in conjunction with a behavior language similar to the one described by [Brooks, 1990]. Underlying the behavior language is a scheduler which schedules the processes implied by the finite-state machines defining particular behaviors.

3. An example

In this section an example is briefly discussed. The example is necessarily simple due to space limitations. We assume two releaser frames each describing a possible action. One frame is called communicate-move-request. It has slots for communicate-with, state, and goal. Its associated action performs a communication through the speech output channel to produce a request to move out of the way. The matching state of the agent is described as talkative, in-a-hurry, etc. The goal state includes moving-towards-target. The other frame is go-around-object. It has slots for object, state and goal. The state includes features like frustration and exploring. The goal includes also the feature moving-towards-target. When this frame becomes active it causes a behavior to go around the object that is filling the object slot. There are other frame-hierarchies needed for the example, such as the hierarchy of objects displayed in figure 2.

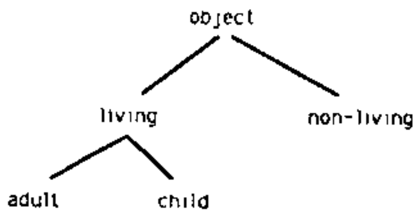

Figure 2. Frame hierarchy of objects.

---

[6] This work was done by Hin Saeris. The behavior language was implemented by Piet Ruysselinck.

Assume a situation where there are four objects: object-in-front, object-in-back, object-to-left and object-to-right. Each of these is competing for the communicate-with slot in the communicate-move-request frame and the object slot in the go-around-object frame. The state of the agent potentially fits in the state slot of the communicate-move-request and the go-around-object frame. The goal of the agent potentially fits in the goal slot of these two frames. The features associated with the objects and their associated certainties are as follows[7]:

object-in-front: object (1.0), living (0.8), child (0.4), noisy (0.5), pushy (0.2).

object-in-back: clear (1.0).

object-to-left: object (1.0), non-living (0.7), big (0.8).

object-to-right: clear (1.0).

The state of the agent includes the following features:

talkative, exploring, in-a-hurry

The goal includes

moving-towards-target

The following simulation results list the evolution of the various link strengths and frame strengths after one sweep through the algorithm:

(after step 2)
GOAL into GOAL of GO-AROUND-OBJECT is 0.7
GOAL into GOAL of COMMUNICATE-MOVE-REQUEST is 0.9
STATE into STATE of GO-AROUND-OBJECT is 0.2
STATE into STATE of COMMUNICATE-MOVE-REQUEST is 0.5
OBJECT-TO-RIGHT into OBJECT of GO-AROUND-OBJECT is 0.0
OBJECT-TO-RIGHT into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 0.030000001
OBJECT-TO-LEFT into OBJECT of GO-AROUND-OBJECT is 0.64000005
OBJECT-TO-LEFT into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 0.030000001
OBJECT-IN-BACK into OBJECT of GO-AROUND-OBJECT is 0.0
OBJECT-IN-BACK into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 0.030000001
OBJECT-IN-FRONT into OBJECT of GO-AROUND-OBJECT is 0.71
OBJECT-IN-FRONT into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 1.0

(after step 3)

GOAL of GO-AROUND-OBJECT into GOAL is 0.07
STATE of GO-AROUND-OBJECT into STATE is 0.020000001
OBJECT of GO-AROUND-OBJECT into OBJECT-IN FRONT is 0.03734O73&
OBJECT of GO-AROUND-OBJECT into OBJECT-IN-BACK is 0.0
OBJECT of GO-AROUND-OBJECT into OBJECT-TO-LEFT is 0.030340744
OBJECT of GO-AROUND-OBJECT into OBJECT-TO-RIGHT is 0.0
GOAL of COMMUNICATE-MOVE-REQUEST into GOAL is 0.089999996
STATE of COMMUNICATE-MOVE-REQUEST into STATE is 0.05
COMMUNICATE-WITH of COMMUNICATE-MOVE-REQUEST
into OBJECT-IN-FRONT is 0.09174312
COMMUNICATE-WITH of COMMUNICATE-MOVE-REQUEST
into OBJECT-IN-BACK is 8.25688 le-5
COMMUNICATE-WITH of COMMUNICATE-MOVE-REQUEST
into OBJECT-TO-LEFT is 8.25688le-5
COMMUNICATE-WITH of COMMUNICATE-MOVE-REQUEST
into OBJECT-TO-RIGHT is 8.256881e-5

(after step 4)

GOAL into GOAL of COMMUNICATE-MOVE-REQUEST is 0.050624996
GOAL into GOAL of GO-AROUND-OBJECT is 0.030625
STATE into STATE of COMMUNICATE-MOVE-REQUEST is 0.035714287
STATE into STATE of GO-AROUND-OBJECT is 0.0057142866
OBJECT-TO-RIGHT into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 8.25688le-5
OBJECT-TO-RIGHT into OBJECT of GO-AROUND-OBJECT is 0.0
OBJECT-TO-LEFT into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST *is* 2.240916c-7
OBJECT-TO-LEFT into OBJECT of GO-AROUND-OBJECT is 0.030258399
OBJECT-IN-BACK into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 8.25688le-5
OBJECT-IN-BACK into OBJECT of GO-AROUND-OBJECT is 0.0
OBJECT-IN-FRONT into COMMUNICATE-WITH
of COMMUNICATE-MOVE-REQUEST is 0.06520412
object impact from OBJECT-IN-FRONT into OBJECT
of GO-AROUND-OBJECT is 0.010801744
frame strength GO-AROUND-OBJECT is 0.22199227
frame strength COMMUNICATE-MOVE-REQUEST is 0.50514466

The global evolution of the various frames is displayed graphically in the following figures. The first figure shows how the communicate-move-request frame gradually wins over the go-around-object frame.

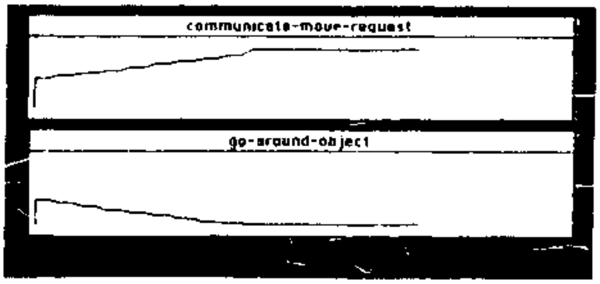

Figure 3. Evolution of frame-strengths.

---

[7] These features are themselves connected to sensors (infra-red, sound, pyro, touch, etc.) or groups of sensors which themselves have specific certainties.

The next figure shows the evolution of the strength of the links between the objects and the slots for the go-around-object frame. Notice how the strength gradually diminishes for all of them.

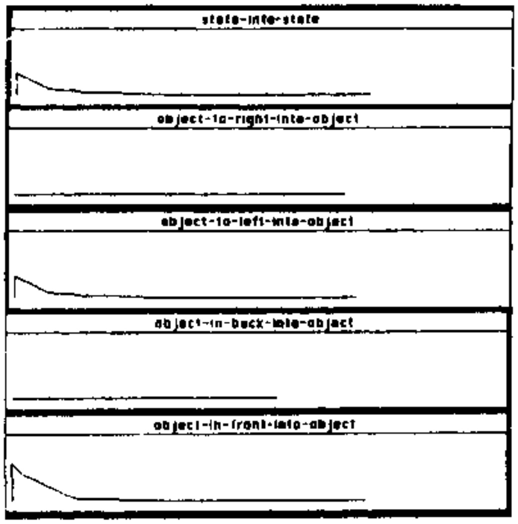

Figure 4. Evolution of link-strengths for go-around-object frame

Finally the following figures show the evolution of the link-strengths for the communicate-move-request frame. We see that the link-strength becomes progressively stronger.

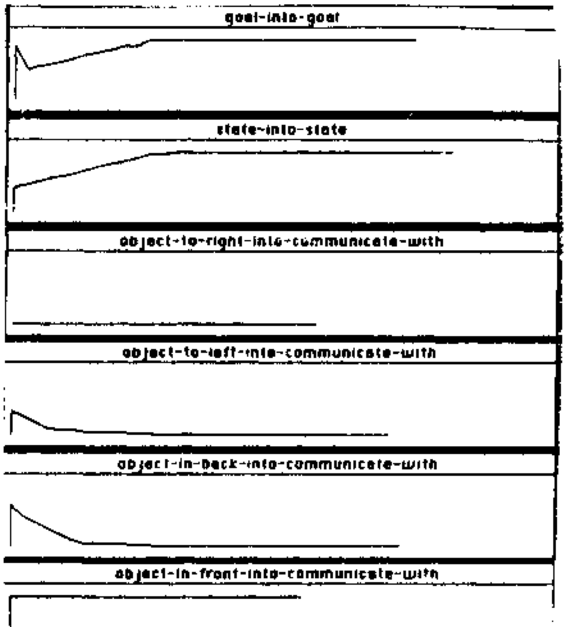

Figure 5. Evolution of link-strengths for communicate-move-request frame.

Acknowledgement

This research is partly sponsored by the Belgian government Impuls program on AI and partly by Nato Grant CRG90031. Hein Saris has made valuable contributions to the contents of this paper.

References

[Braitenberg, 1984] Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge, Massachusetts, 1984.

[Brooks, 1986] Brooks, R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation.* RA-2(l):14-23, 1986.

[Brooks, 1990] Brooks, R. (1990) The Behavior Language. MIT AI Lab. Memo.

[Fahlman, 1981] Fahlman, S. *NETL: A system for representing and using real-world knowledge.* MTT Press, Cambridge, Massachusetts, 1981.

[Hillis, 1985] Hillis, D. The Connection Machine. MIT Press, Cambridge, Massachusetts, 1985.

[Minsky, 1987] Minsky, M. *The society of mind.* Basic books. New York, 1987.

[Schank and Riesbeck, 1981] Schank, R. and C. Riesbeck (1981) Inside computer understanding: Five programs plus miniatures. Lawrence Erlbaum Ass. Hillsdale, New Jersey, 1981.

[Steels, 1990] Steels, L. (1990) Towards a theory of emergent functionality. In B. Meyer and S. Wilson (1990) *Simulation of Adaptive Behavior.* MIT Press, Cambridge, Massachusetts, 1984.