SIMULATION

http://sim.sagepub.com

Simulation and Analysis of a Shared Extended Mind

Tibor Bosse, Catholijn M. Jonker, Martijn C. Schut and Jan Treur

SIMULATION 2005; 81; 719
DOI: 10.1177/0037549705060260

Simulation and Analysis of a Shared Extended Mind

Tibor Bosse
Catholijn M. Jonker
Martijn C. Schut
Vrije Universiteit Amsterdam
Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
tbosse@cs.vu.nl
C.Jonker@nici.ru.nl
schut@cs.vu.nl

Jan Treur
Vrije Universiteit Amsterdam, Department of Artificial Intelligence and
Utrecht University
Department of Philosophy
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands
treur@cs.vu.nl

Some types of animals exploit patterns created in the environment as external mental states, thus obtaining an extension of their mind. In the case of social animals, the creation and exploitation of such patterns can be shared, which supports a form of shared extended mind or collective intelligence. This article explores this shared extended mind principle for social animals in more detail. The focus is on formal analysis and formalization of the dynamic properties of the processes involved, both at the local level (the basic mechanisms) and the global level (the emerging properties of the whole), and their relationships. A case study in social ant behavior in which the shared extended mind plays an important role is used as an illustration. For this case, simulations are described based on specifications of local properties, and global properties are specified and verified.

Keywords: Extended mind, simulation, collective behavior, ant colonies

1. Introduction

Various studies [1-6] have described how behavior is often supported not only by an internal mind in the sense of internal mental structures and cognitive processes but also by processes based on patterns created in the external environment that serve as external mental structures. Examples of this pattern of behavior are the use of “to do lists” and “lists of desiderata.” Having written these down externally (e.g., on paper, in your diary, in your organizer or computer) makes it unnecessary to have an internal memory about all the items. Thus, internal mental processing can be kept less complex. The only thing to remember is where these lists are available. Other examples of the use of the extended mind are doing mathematics or arithmetic, where external (symbolic, graphical, material) representations are used (e.g., [7]).

Clark and Chalmers [3] point at the similarity between cognitive processes in the head and some processes involving the external world. This similarity can be used as an indication that these processes can be considered extended cognitive processes or the extended mind:

> If, as we confront some task, a part of the world functions as a process which, *were it done in the head*, we would have no hesitation in recognizing as part of the cognitive process, then that part of the world *is* (so we claim) part of the cognitive process. Cognitive processes ain’t (all) in the head! . . . Of course, one could always try to explain my action in terms of internal processes and a long series of “inputs” and “actions,” but this explanation would be needlessly complex. If an isomorphic process were going on in the head, we would feel no urge to characterize it in this cumbersome way. [3, pp. 8, 10]

We will call this criterion the “isomorphism” criterion. As the patterns in the external world have to be created and sensed, interaction with the external world will be more intensive, compared to the case where internal mental states are created and exploited.

Especially in the case of social animals, external mental states created by one individual can be exploited by another

SIMULATION, Vol. 81, Issue 10, October 2005 719-732

DOI: 10.1177/0037549705060260

individual, or, more generally, the creation, maintenance, and exploitation of external mental states are activities in which a number of individuals can participate (e.g., presenting slides on a paper with multiple authors to an audience). Further examples can be found everywhere, varying from roads and traffic signs to books or other media, as well as to many other kinds of cultural achievements. In this multiagent case, the extended mind principle serves as a way to build a form of social or collective intelligence that goes beyond (and may even not require) social intelligence based on direct one-to-one communication. In such cases, the external mental states cross and, in a sense, break up the borders between (the minds of) the individuals and become *shared extended mental states*.

An interesting and currently often-studied example of collective intelligence is the intelligence shown by ant colonies [8-10]. Indeed, in this case, the external world is exploited as an extended mind by using pheromones. While they walk, ants drop pheromones on the ground. The same or other ants sense these pheromones and follow the route in the direction of the strongest concentration. Because pheromones evaporate, such routes may vary over time. This context is chosen in this article to illustrate the shared extended mind principle.

The main contribution of this article is a detailed analysis of this shared extended mind principle and a formalization of its dynamics. The principle is illustrated by a case study of social behavior based on a shared extended mind (a simple ant colony). The analysis of this case study comprises multiagent simulation based on identified local dynamic properties, identification of dynamic properties for the overall process, and verification of these dynamic properties. The shared extended mind principle and its formalization as introduced in this article allow one to perform simulation and explanation of behavior on a more abstract level—in terms of mental states instead of the physical materialization. This provides a simpler, more abstract, and perhaps more understandable and elegant interpretation of the simulation models than based on the physical counterparts. This is made possible by interpreting the external world states involved according to a new ontology. Considering part of the external world as an extended mind allows one to give another interpretation to external physical processes and states. Physical state properties such as "pheromone is present at d" can be interpreted (and even renamed) as, for example, "it is believed that d is the direction home." In fact, this double interpretation still gives two possibilities: for empirical data, the physical interpretation can be chosen, whereas for modeling the other, mental interpretation can be kept in mind.

More specifically, in this article, section 2 is a brief introduction of the basic concepts used in the modeling approach and formalization. It introduces two modeling languages: one (the *leads to* language) used for simulation and one (the *temporal trace language* [TTL]) for more complex properties that can be used in analysis. For the former language, a software environment for simulation has been developed; for the latter language, a software environment has been developed that enables automatic checking of specified properties against given traces.

In section 3, the extended mind principle is formalized by an isomorphic mapping between a cognitive process using external mental states and a similar process based on internal mental states. This mapping formalizes how the processes of the agent in interaction with the world indeed can be interpreted as (comparable to) an agent with internal mental processes, thus formalizing the "were it done in the head" criterion quoted earlier in the citation from Clark and Chalmers [3]. In section 4, a simulation model is presented for the ant case study. This simulation model is specified using local properties: temporal rules that express in a local manner the basic mechanisms of the case. These rules are specified and formalized in the *leads to* language introduced in section 2 and are therefore directly executable in the software environment that has been developed. Some of the simulation outcomes are included in section 4. Whereas section 4 has a local perspective on the basic mechanisms, section 5 takes the global perspective of emergent properties of the multiagent process as a whole. A number of relevant global dynamic properties are identified and formalized in the language TTL. It is discussed how these global dynamic properties have been checked against simulation traces. Moreover, some of the logical relationships between them are discussed. Section 6 is a discussion of the results.

2. State Properties and Dynamic Properties

Dynamics will be described in the next section as evolution of states over time. The notion of state as used here is characterized on the basis of an ontology defining a set of physical and/or mental (state) properties that do or do not hold at a certain point in time. States can be taken as global states, but also more local perspectives, based on a subset of the overall ontology, can be expressed in a state (e.g., an internal agent state). As an example, the internal state property "the agent A has pain," or the external world state properties "it is raining" and "the environmental temperature is 7°C," may be expressed in terms of different ontologies. To formalize state property descriptions, an ontology is specified as a finite set of sorts, constants within these sorts, and relations and functions over these sorts. The example properties mentioned above then can be defined by nullary predicates (or proposition symbols), such as itsraining, or by using n-ary predicates (with $n \geq 1$), such as has_pain(A) and has_temperature(environment, 7). For a given ontology Ont, the propositional language signature consisting of all *state ground atoms* (or *atomic state properties*) based on Ont is denoted by APROP(Ont). The *state properties* based on a certain ontology Ont are formalized by the propositions that can be made (using conjunction, negation, disjunction, implication) from the ground atoms. A *state* S is an indication of which atomic state properties are true and which are false; that is, a mapping S: APROP(Ont) → {true, false}.

To describe the internal and external dynamics of the agent, explicit reference is made to time. Dynamic properties can be formulated that relate a state at one point in time to a state at another point in time. A simple example is the following informally stated dynamic property for belief creation based on observation:

> If the agent observes at t1 that it is raining, then the agent will believe that it is raining.

To express such dynamic properties and other, more sophisticated ones, TTL is used (cf. [11]). In this language, explicit references can be made to time points and traces. Here a *trace* or *trajectory* over an ontology Ont is a time-indexed sequence of states over Ont. The sorted predicate logic temporal trace language TTL is built on atoms referring to, for example, traces, time, and state properties. For example, "in trace γ at time t property p holds" is formalized by state(γ, t) |= p. Likewise, "in trace γ at time t property p does not hold" is formalized by state(γ, t) $|\neq$ p. Here, |= is a predicate symbol in the language, usually used in infix notation, which is comparable to the Holds-predicate in situation calculus. Dynamic properties are expressed by temporal statements built using the usual logical connectives and quantification (e.g., over traces, time, and state properties). For example, consider the following dynamic property:

> In any trace γ, if at any point in time t1 the agent A observes that it is raining, then there exists a time point t2 after t1 such that at t2 in the trace, the agent A believes that it is raining.

In formalized TTL form, it looks as follows:

$$\forall t1\ [\ \text{state}(\gamma, t1) \models \text{observes}(A, \text{itsraining}) \Rightarrow \\ \exists t2 \geq t1\ \text{state}(\gamma, t2) \models \text{belief}(A, \text{itsraining})\ ]$$

Language abstractions by introducing new (definable) predicates for complex expressions are possible and supported.

To specify simulation models, a simpler temporal language has been developed, based on TTL. This language (the *leads to* language) enables one to model direct temporal dependencies between two state properties in successive states. This executable format is defined as follows. Let α and β be state properties of the form "conjunction of atoms or negations of atoms," and let e, f, g, and h be nonnegative real numbers. In the *leads to* language, $\alpha \twoheadrightarrow_{e,\, f,\, g,\, h} \beta$ means the following:

> If state property α holds for a certain time interval with duration g, then after some delay (between e and f), state property β will hold for a certain time interval of length h.

A specification of dynamic properties in the *leads to* format has an advantage because it is executable and can often easily be depicted graphically. Moreover, the language offers the possibility to express both qualitative and quantitative aspects of a process to be simulated. Therefore, it combines the advantages of logic-oriented approaches, such as those in Barringer et al. [12] and Forbus [13], with those of mathematical approaches, such as those in Port and van Gelder [14], in the context of simulation modeling and analysis [15]. For a more precise definition of the *leads to* format, see Bosse et al. [16].

3. Explanation and the Isomorphism Principle

In section 1, the isomorphism principle was introduced, based on the apparent similarity between cognitive processes in the head and some processes involving the external world. For an illustration of this principle, see Figures 1 and 2. In these figures, the circles denote state properties, the arrows denote dynamic properties, and the dotted box indicates the borders of the agent.

Figure 1 depicts a simple case of an agent with behavior based on an internal mental state property m1, whereas Figure 2 depicts another agent with the same behavior based on an external mental state property m2. In both cases, the internal (m1) or external (m2) state property acts as a mediator in the trajectory between input (c1) and output (e1). Thus, in a way, both m1 and m2 can be considered an agent's belief. Note that the internal processing of the agent in Figure 2 is chosen as simple as possible: stimulus response. Hence, this agent is assumed not to have any internal states. This is in line with the ideas of Clark and

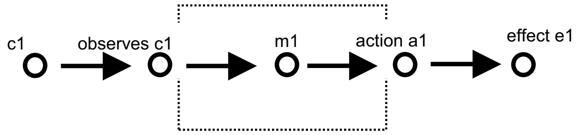

Figure 1. Behavior based on an internal mental state

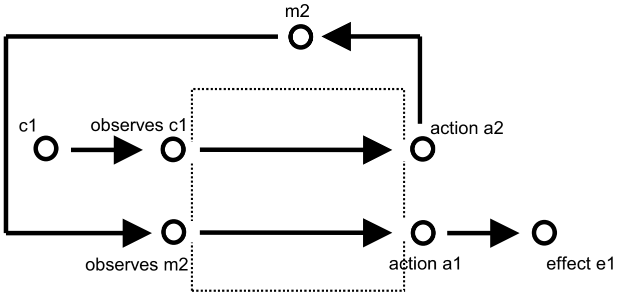

Figure 2. Behavior based on an external mental state

Chalmers [3], who claim that the explanation of cognitive processes should be as simple as possible. However, the interaction between this agent and the external world is a bit more complex than in Figure 1: one extra action is needed to create the external mental state m2, and one additional observation is needed to observe it.

To make the similarity between the two different cognitive processes more precise, the following mapping from the upper graph into the lower graph can be made (see Fig. 3).

c1 → c1
observes c1 → observes c1
m1 → m2
action a1 → action a1
effect e1 → effect e1

This mapping, indicated by the vertical dotted arrows in Figure 3, preserves the temporal (*leads to*) relationships (the solid arrows) and provides an (isomorphic in the mathematical sense) embedding of a cognitive process based on the internal mind into a cognitive process based on the extended mind. Remember the quotes from Clark and Chalmers [3], cited in section 1. Clark and Chalmers use the isomorphism to a process "in the head" as one of the criteria to consider external and interaction processes as cognitive or mind processes. This "isomorphism" criterion is formalized in Figure 3 for a simple example of such an isomorphism. Note that the process from m1 to action a1, modeled as one step in the internal case, is mapped onto a process from m2 via observes m2 to action a1, which is modeled as a two-step process in the external case. So the isomorphism is embedding in one direction and is not a bidirectional isomorphism, simply because the observation state for m2 (and the same for the action a2) has no counterpart in the internal case. For a more detailed treatment of the isomorphism and an extension of the mapping to formally defined dynamic properties, see Bosse and Treur [17].

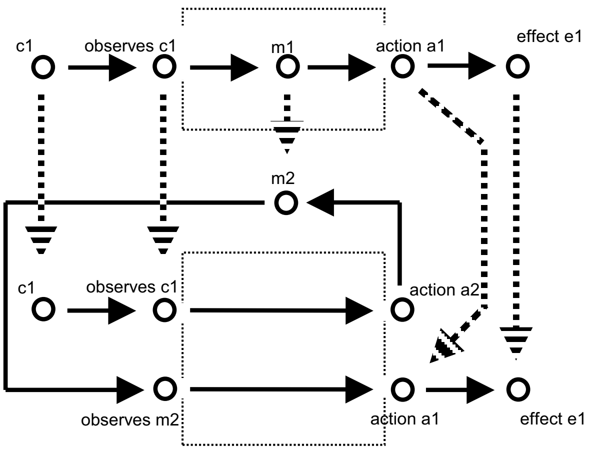

Figure 3. Internal and external mental states and their isomorphism relationship

Behavior often is explained by considering the basic underlying causal relations or mechanisms. Such basic mechanisms can be formally modeled by *leads to* relations. The isomorphism principle and its formalization, as depicted in Figure 3, allows one to replace an explanation of behavior in terms of basic mechanisms involving frequent interactions (observations and actions) with the external world with an explanation that leaves out these interactions and bases itself directly on the mental states. This explanation is simpler, more abstract, and perhaps more elegant than the more complicated "cumbersome" explanation based on the interactions. This is made possible by introducing a new ontology for the external world states involved. Considering part of the external world as an extended mind allows one to give another interpretation to external physical processes and states. Physical state properties such as "pheromone is present at d" are renamed as, for example, "it is believed that d is the direction home." Why would one introduce extra language to refer to the same fact in the world? Given the literature on reduction, where often it is claimed that mental state properties can be and actually should be replaced by their physical realizers, at first sight, such an opposite move may seem a bit surprising. For example, Kim [18, pp. 214-6) claims that ontological simplification is one of the reasons to reduce mental state properties to physical state properties. In the extended mind case at hand, the converse takes place; the question is, what is the advantage of this "ontological complication"? A number of arguments in support of this can be given. Clark and Chalmers [3] claim that this allows application of other types of explanation and other methods of scientific investigation:

> We allow a more natural explanation of all sorts of actions. . . . In seeing cognition as extended one is not merely making a terminological decision; it makes a significant difference to the methodology of scientific investigation. In effect, explanatory methods that might once have been thought appropriate only for the analysis of "inner" processes are now being adapted for the study of the outer, and there is promise that our understanding of cognition will become richer for it. [3, p. 10]

In Jonker, Treur, and Wijngaards [19], it is explained in some detail and illustrated by examples why, in various cases in other areas (such as computer science), such an antireductionist strategy often pays off. Advantages in terms of insight, transparency, and generality include the following: additional higher level ontologies can improve understanding as they may allow simplification of the picture by abstracting from lower level details, more insight is gained from a conceptually higher level perspective, analysis of more complex processes is possible, and the same concepts have a wider scope of application, thus obtaining unification. For more details and support for this antireductionist argument, see Jonker, Treur, and Wijngaards [19].

4. A Simulation Model of a Shared Extended Mind

Dynamic properties can be specified at different aggregation levels, varying from (local) dynamic properties for the basic mechanisms and (global) properties of a process as a whole. This section introduces the local dynamic properties for the basic mechanisms; they are used to specify a simulation model. The world in which the ants live is described by a labeled graph, as depicted in Figure 4.

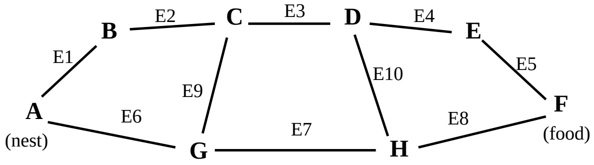

Figure 4. An ant's world

Locations are indicated by A, B, . . . , and edges by E1, E2, ... The ants move from location to location via edges; while passing an edge, pheromones are dropped. The objective of the ants is to find food and bring this back to their nest. In this example, there is only one nest (at location A) and one food source (at location F).

The example concerns multiple agents (the ants), each of which has input (to observe) and output (for moving and dropping pheromones) states and a physical body that is at certain positions over time, but no internal mental state properties (they are assumed to act purely by stimulus-response behavior). An overview of the formalization of the state properties of this single-agent conceptualization is shown in Table 1. In these local properties, a is a variable that stands for ant, e for edge, i for pheromone level, l for location, and n for number of neighbor locations. Note that in some of the state properties, the direction of an ant is incorporated (e.g., ant a is at location l coming from e, ant a is at edge e to l2 coming from location l1). This direction is meant to relate to the orientation of the ant's body in space, which is a genuine state property, but for convenience, this is expressed by referring to the past or future states involved.

In Table 2, the local dynamic properties are shown that were used to model the example. On the left, the dynamic properties are given in formal (*leads to*) format. In each dynamic property, the values 0, 0, 1, 1 were chosen for the timing parameters e, f, g, h (see section 2). For simplicity, these parameters were left out of the table. On the right, for each dynamic property, an informal description is provided.

A special software environment has been created to enable the simulation of executable models. Based on an input consisting of dynamic properties in the *leads to* format, the software environment generates simulation traces. Examples of such traces can be seen in Figures 5 through 8. Time is on the horizontal axis, and the state properties are on the vertical axis. A dark box on top of the line indicates that the property is true during that time period, and a lighter box below the line indicates that the property is false. These traces are based on all local properties identified. To limit complexity, in the example depicted in Figures 5, 6, and 7, only 3 ants are involved. The trace in Figure 8 shows an example with 2 ants. However, similar experiments have been performed with populations of 50 and 100 ants. Since the abstract way of modeling used for the simulation is not computationally expensive, these simulations can be performed relatively quickly. To be precise, they took 35 seconds (for 50 ants and 80 time steps), 70 seconds (100 ants, 80 time steps), 100 seconds (50 ants, 200 time steps), and 200 seconds (100 ants, 200 time steps).

Figures 5, 6, and 7 are all parts of the same trace. Figure 5 shows the observations and locations of the ants; Figure 6 shows the performed actions of ant1 in more detail; Figure 7 shows how the pheromone levels at edges E1 and E6 are changing. As can be seen in Figure 5 and 6, there are two ants (ant1 and ant2) that start their search for food immediately (at time point 0), whereas ant3 comes into play a bit later, at time point 3. These time points were specified manually (see property LP2). When ant1 and ant2 start their search, none of the locations contain any pheromones yet, so basically they have a random choice where to go. In the current example, ant1 selects a rather long route to the food source (via locations A-B-C-D-E-F), while ant2 chooses a shorter route (A-G-H-F). Note that, in the current model, a fixed route preference (via the attractiveness predicate) has been assigned to each ant for the case when there are no pheromones yet. After that, at time point 3, ant3 starts its search for food. At that moment, there are trails of pheromones via both edges E1 and E6, but these trails contain exactly the same number of pheromones (see Fig. 7). Thus, ant3 also has a choice among edges E1 and E6 and chooses in this case to go to E1. Meanwhile, at time point 18, ant2 has arrived at the food source (location F). Since it is the first to discover this location, the only present trail leading back to the nest is its own trail. Thus, ant2 will return home via its own trail. Next, when ant1 discovers the food source (at time point 31), it will notice that there is a trail leading back that is stronger than its own trail (since ant2 has already walked there twice: back and forth, not too long ago). As a result, it will follow this trail and will keep following ant2 forever. Something similar holds for ant3. The first time that it reaches the food source, ant3 will still follow its own trail, but some time later (from time point 63), it will also follow the other two ants. To conclude, eventually the shortest of both routes is shown to remain, while the other route evaporates. Other simulations, particularly for small ant populations, show that it is important that the decay parameter of the pheromones is not too high. Otherwise, the trail leading to the nest has evaporated before the first ant has returned, and all ants get lost.

Table 1. Formalisation of state properties

	body positions in world:
pheromone level at edge e is i	pheromones_at(e, i)
ant a is at location l coming from e	is_at_location_from(a, l, e)
ant a is at edge e to l2 coming from location l1	is_at_edge_from_to(a, e, l1, l2)
ant a is carrying food	is_carrying_food(a)
	world state properties:
edge e connects location l1 and l2	connected_to_via(l1, l2, e)
location 1 is the nest location	nest_location(l)
location 1 is the food location	food_location(l)
location l has n neighbors	neighbors(l, n)
edge e is most attractive for ant a coming from location l	attractive_direction_at(a, l, e)
	input state properties:
ant a observes that it is at location l coming from edge e	observes(a, is_at_location_from(l, e))
ant a observes that it is at edge e to l2 coming from location l1	observes(a, is_at_edge_from_to(e, l1, l2))
ant a observes that edge e has pheromone level i	observes(a, pheromones_at(e, i))
	output state properties:
ant a initiates action to go to edge e to l2 coming from location l1	to_be_performed(a, go_to_edge_from_to(e, l1, l2))
ant a initiates action to go to location l coming from edge e	to_be_performed(a, go_to_location_from(l, e))
ant a initiates action to drop pheromones at edge e coming from location l	to_be_performed(a, drop_pheromones_at_edge_from(e, l))
ant a initiates action to pick up food	to_be_performed(a, pick_up_food)
ant a initiates action to drop food	to_be_performed(a, drop_food)

Table 2. Simulation model

LP1 (Initialization of Pheromones) start → pheromones_at(E1, 0.0) and pheromones_at(E2, 0.0) and pheromones_at(E3, 0.0) and pheromones_at(E4, 0.0) and pheromones_at(E5, 0.0) and pheromones_at(E6, 0.0) and pheromones_at(E7, 0.0) and pheromones_at(E8, 0.0) and pheromones_at(E9, 0.0) and pheromones_at(E10, 0.0)	At the start of the simulation, at all locations, there are 0 pheromones.
LP2 (Initialization of Ants) start → is_at_location_from(ant1, A, init) and is_at_location_from(ant2, A, init) and is_at_location_from(ant3, A, init)	At the start of the simulation, all ants (in this case, ants 1, 2, and 3) are at location A. The exact time point an ant is added to the simulation can be specified manually.
LP3a (Initialization of World) start → connected_to_via(A, B, l1) and . . . and connected_to_via(D, H, l10)	This property expresses which locations are connected to each other and via which edges they are connected.
LP3b (Initialization of World) start → neighbors(A, 2) and . . . and neighbors(H, 3)	This property expresses for each location how many neighbors it has.

Continued on next page

Table 2. ***Continued from previous page***

LP4 (Initialization of Attractive Directions) start → attractive_direction_at(ant1, A, E1) and . . . and attractive_direction_at(ant3, E, E5)	This property expresses, for each ant and location, which edge is most attractive for the ant if it arrives at that location. This criterion is used only when there are edges with equal pheromone levels.[1]
LP5a (Selection of Edge) observes(a, is_at_location_from(A, e0)) and attractive_direction_at(a, A, e1) and connected_to_via(A, l1, e1) and observes(a, pheromones_at(e1, i1)) and connected_to_via(A, l2, e2) and observes(a, pheromones_at(e2, i2)) and e1 ≠ e2 and i1 = i2 → to_be_performed(a, go_to_edge_from_to(e1, A, l1))	If an ant observes that it is at location A, and both edges connected to location A have the same number of pheromones, then the ant goes to its attractive direction.
LP5b (Selection of Edge) observes(a, is_at_location_from(A, e0)) and connected_to_via(A, l1, e1) and observes(a, pheromones_at(e1, i1)) and connected_to_via(A, l2, e2) and observes(a, pheromones_at(e2, i2)) and i1 > i2 → to_be_performed(a, go_to_edge_from_to(e1, A, l1))	If an ant observes that it is at location A, and one edge connected to location A has the highest number of pheromones, then the ant goes to that edge.
LP5c (Selection of Edge) observes(a, is_at_location_from(F, e0)) and connected_to_via(F, l1, e1) and observes(a, pheromones_at(e1, i1)) and connected_to_via(F, l2, e2) and observes(a, pheromones_at(e2, i2)) and i1 > i2 → to_be_performed(a, go_to_edge_from_to(e1, F, l1))	If an ant observes that it is at location F, and one edge connected to location F has the highest number of pheromones, then the ant goes to that edge.
LP5d (Selection of Edge) observes(a, is_at_location_from(l, e0)) and neighbors(l, 2) and connected_to_via(l, l1, e1) and e0 ≠ e1 and l ≠ A and l ≠ F → to_be_performed(a, go_to_edge_from_to(e1, l, l1))	If an ant observes that it is at a location (which is not A or F) with 2 neighbors, then it continues in the direction it was traveling to.
LP5e (Selection of Edge) observes(a, is_at_location_from(l, e0)) and attractive_direction_at(a, l, e1) and neighbors(l, 3) and connected_to_via(l, l1, e1) and observes(a, pheromones_at(e1, 0.0)) and connected_to_via(l, l2, e2) and observes(a, pheromones_at(e2, 0.0)) and e0 ≠ e1 and e0 ≠ e2 and e1 ≠ e2 → to_be_performed(a, go_to_edge_from_to(e1, l, l1))	If an ant observes that it is at a location with 3 neighbors, and all edges connected to that location have the same number of pheromones, then the ant goes to its attractive direction.
LP5f (Selection of Edge) observes(a, is_at_location_from(l, e0)) and neighbors(l, 3) and connected_to_via(l, l1, e1) and observes(a, pheromones_at(e1, i1)) and connected_to_via(l, l2, e2) and observes(a, pheromones_at(e2, i2)) and e0 ≠ e1 and e0 ≠ e2 and e1 ≠ e2 and i1 > i2 → to_be_performed(a, go_to_edge_from_to(e1, l1))	If an ant observes that it is at a location with 3 neighbors, and one edge connected to that location has the highest number of pheromones, then the ant goes to that edge.
LP6 (Arrival at Edge) to_be_performed(a, go_to_edge_from_to(e, l, l1)) → is_at_edge_from_to(a, e, l, l1)	If an ant goes to an edge e from a location l to a location l1, then later the ant will be at this edge e.
LP7 (Observation of Edge) is_at_edge_from_to(a, e, l, l1) → observes(a, is_at_edge_from_to(e, l, l1))	If an ant is at a certain edge e, going from a location l to a location l1, then it will observe this.
LP8 (Movement to Location) observes(a, is_at_edge_from_to(e, l, l1)) → to_be_performed(a, go_to_location_from(l1, e))	If an ant observes that it is at an edge e from a location l to a location l1, then it will go to location l1.

Continued on next page

---

[1] To obtain interesting simulation traces, different attractive directions were assigned to different ants. However, another possibility (that is supported by the software) is to assign attractive directions at random.

Table 2. ***Continued from previous page***

LP9 (Dropping of Pheromones) observes(a, is_at_edge_from_to(e, l, l1)) → to_be_performed(a, drop_pheromones_at_edge_from(e, l))	If an ant observes that it is at an edge e from a location l to a location l1, then it will drop pheromones at this edge e.
LP10 (Arrival at Location) to_be_performed(a, go_to_location_from(l, e)) → is_at_location_from(a, l, e)	If an ant goes to a location l from an edge e, then later it will be at this location l.
LP11 (Observation of Location) is_at_location_from(a, l, e) → observes(a, is_at_location_from(l, e))	If an ant is at a certain location l, then it will observe this.
LP12 (Observation of Pheromones) is_at_location_from(a, l, e0) and connected_to_via(l, l1, e1) and pheromones_at(e1, i) → observes(a, pheromones_at(e1, i))	If an ant is at a certain location l, then it will observe the number of pheromones present at all edges that are connected to location l.
LP13 (Increment of Pheromones) to_be_performed(a1, drop_pheromones_at_edge_from(e, l1)) and ∀l2 not to_be_performed(a2, drop_pheromones_at_edge_from(e, l2)) and ∀l3 not to_be_performed(a3, drop_pheromones_at_edge_from(e, l3)) and a1 ≠ a2 and a1 ≠ a3 and a2 ≠ a3 and pheromones_at(e, i) → pheromones_at(e, i*decay+incr)	If an ant drops pheromones at edge e, and no other ants drop pheromones at this edge, then the new number of pheromones at e becomes i*decay+incr. Here, i is the old number of pheromones, decay is the decay factor, and incr is the amount of pheromones dropped.
LP14 (Collecting of Food) observes(a, is_at_location_from(l, e)) and food_location(l) → to_be_performed(a, pick_up_food)	If an ant observes that it is at location F (the food source), then it will pick up some food.
LP15 (Carrying of Food) to_be_performed(a, pick_up_food) → is_carrying_food(a)	If an ant picks up food, then as a result, it will be carrying food.
LP16 (Dropping of Food) observes(a, is_at_location_from(l, e)) and nest_location(l) and is_carrying_food(a) → to_be_performed(a, drop_food)	If an ant is carrying food and observes that it is at location A (the nest), then the ant will drop the food.
LP17 (Persistence of Food) is_carrying_food(a) and not to_be_performed(a, drop_food) → is_carrying_food(a)	As long as an ant that is carrying food does not drop the food, it will keep on carrying it.
LP18 (Decay of Pheromones) pheromones_at(e, i) and ∀a,l not to_be_performed(a, drop_pheromones_at_edge_from(e, l)) → pheromones_at(e, i*decay)	If the old amount of pheromones at an edge is i, and there is no ant dropping any pheromones at this edge, then the new amount of pheromones at e will be i*decay.

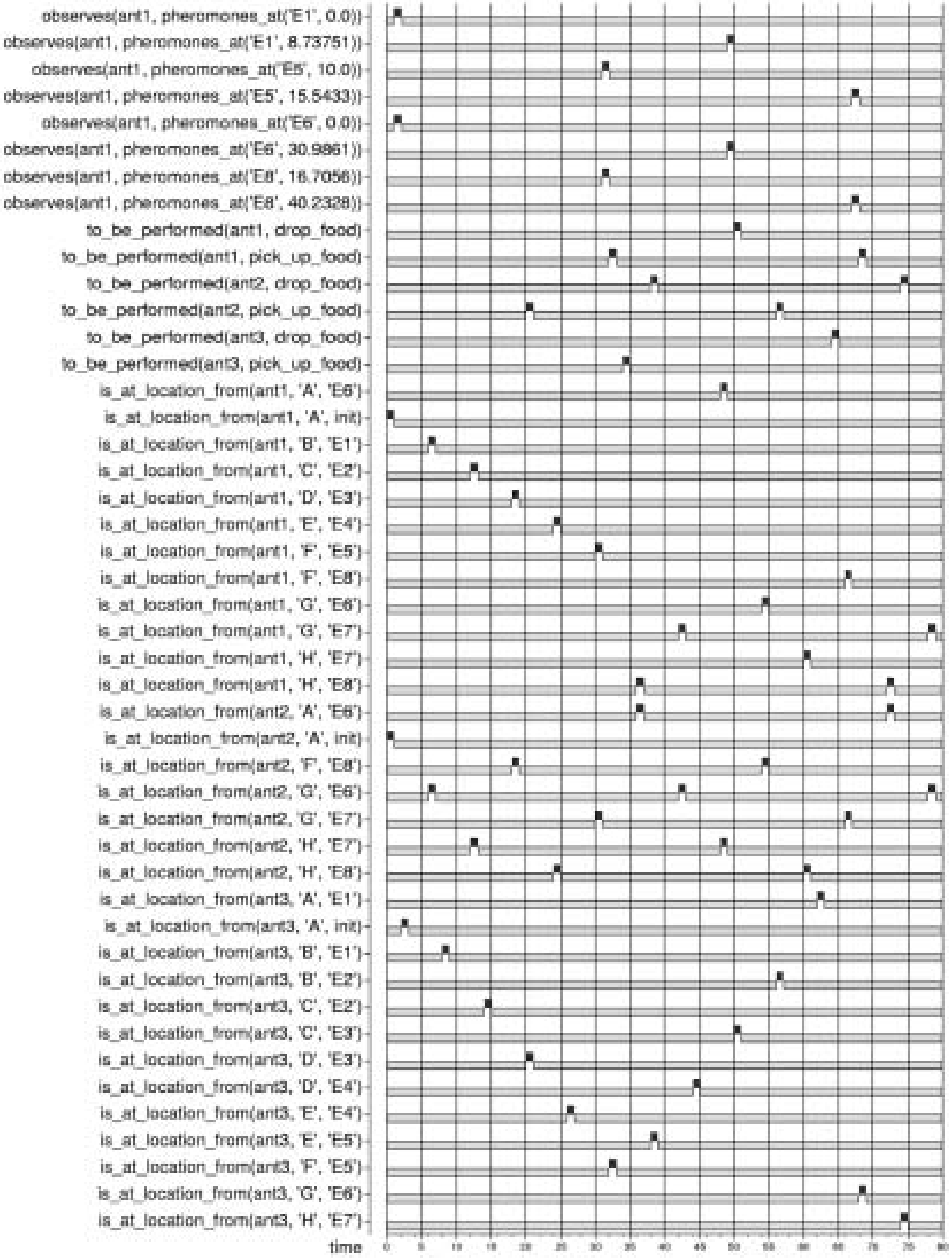

Figure 5. Simulation trace of the dynamics of the ants' behavior

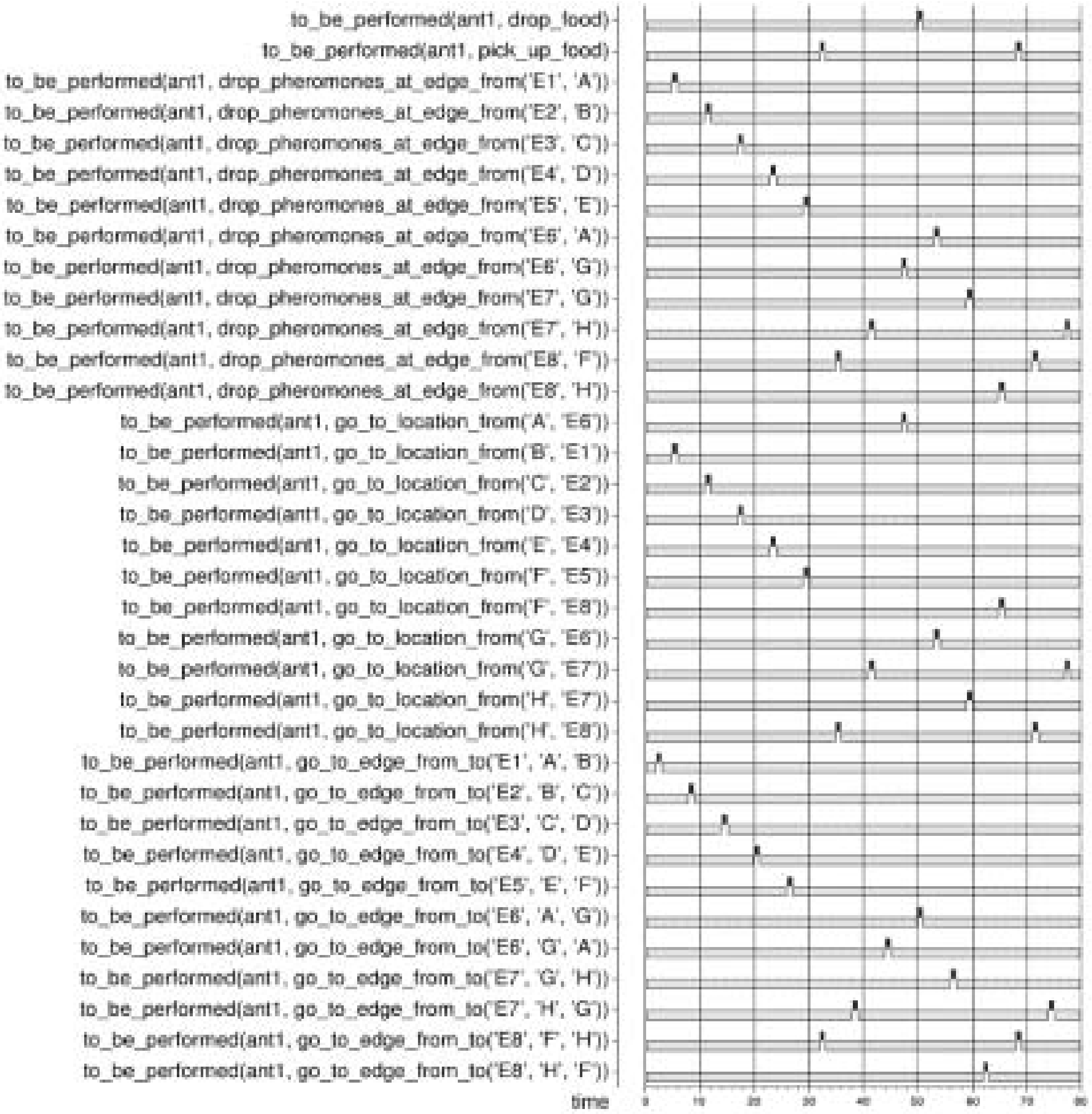

Figure 6. Simulation trace of the performed actions of the ants

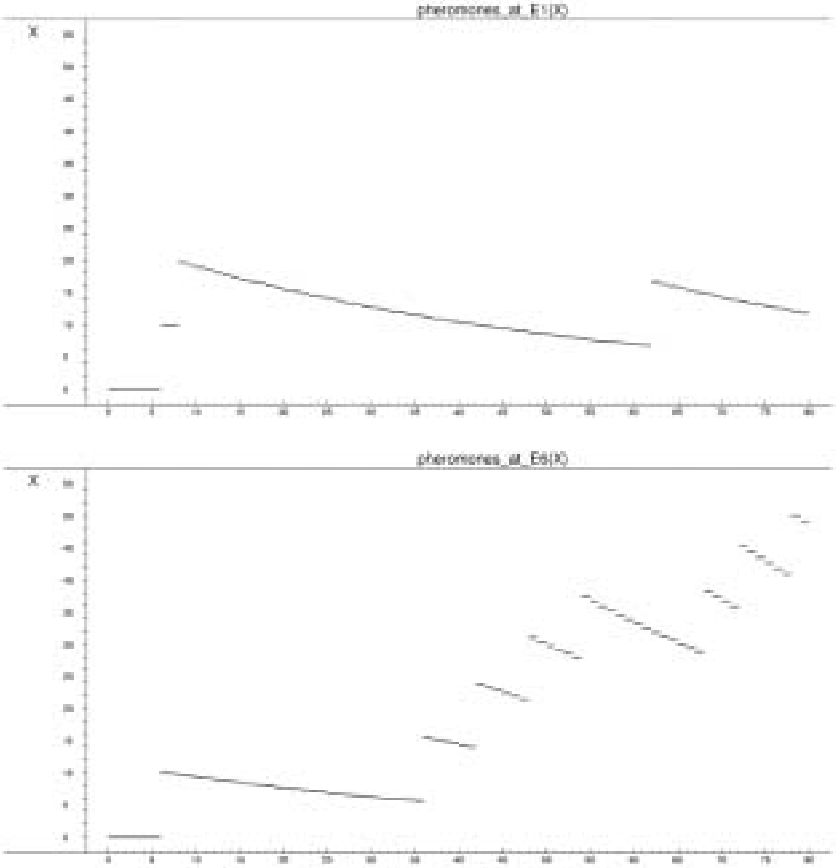

Figure 7. Simulation trace of the dynamics of the pheromone levels at edges E1 and E6

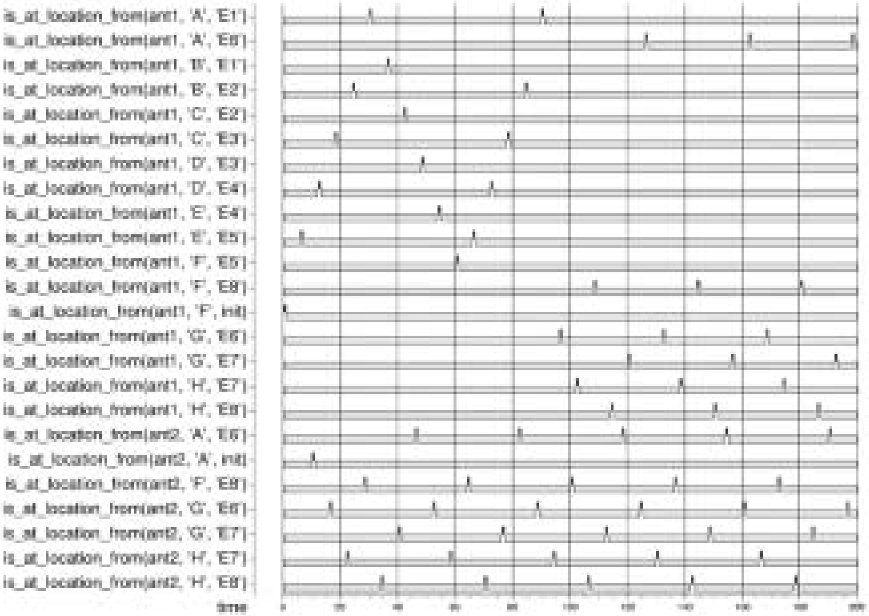

Figure 8. Simulation trace of the dynamics of the ants' behavior, where ant1 departs from the food location, while ant2 depart slightly later from the nest location

Figure 8 describes a different situation. In that figure, there is one ant (ant1) that starts its search, departing from the food location, and one ant (ant2) that starts slightly later (at time point 10), departing from the nest location. The first ant (ant1) takes the long way home (via locations F-E-D-C-B-A), while the second ant (ant2) immediately takes the short route (via locations A-G-H-F) to the food. Figure 8 shows that after some time, both ants follow the short route. Thus, also for this example, we may conclude that eventually the shortest of both routes is shown to remain, while the other route evaporates.

5. Global Properties and Verification

In the previous section, dynamic properties at the lowest aggregation level (the local dynamic properties) were addressed, and simulation based on these properties was discussed. The current section addresses dynamic properties of a global nature and their verification. Within these properties, γ is a variable that stands for an arbitrary trace. First, a language abstraction is given:

$$\begin{gathered}\text{food\_delivered\_by}(\gamma, t, a) \equiv \exists l, e\ [\text{state}(\gamma,t) \models \\ \text{is\_at\_location\_from}(a, l, e))\ \& \\ \text{state}(\gamma,t) \models \text{nest\_location}(l)\ \&\ \text{state}(\gamma,t) \models \\ \text{to\_be\_performed}(a, \text{drop\_food})\ ]\end{gathered}$$

GP1 Food Delivery Successfulness
There is at least one ant that brings food back to the nest.
$\exists t \exists a$: food_delivered_by(γ, t, a).

GP2 Multiple Delivery
Food is delivered by more than one ant.
$\exists t1, t2\ \exists a1, a2\ [\ a1 \neq a2\ \&\ \text{food\_delivered\_by}(\gamma, t1, a1)\ \&\ \text{food\_delivered\_by}(\gamma, t2, a2)\ ]$

Other language abstractions are as follows:

$$\begin{gathered}\text{attractive\_route\_to}(\gamma, a, x) \equiv \\ \exists l\ \exists e\ \forall t\ [\ \text{state}(\gamma, t) \models \text{attractive\_direction\_at}(a, l, e)\ \& \\ \text{state}(\gamma, t) \models \text{connected\_to\_via}(l, x, e)\ ]\end{gathered}$$

That is, the attractive route of the ant (in the case of equal pheromone levels) passes through location x.

$$\begin{gathered}\text{reaches\_end\_attractive\_route}(\gamma, t, a) \equiv \\ \exists l, e\ [\ \text{state}(\gamma, t) \models \text{is\_at\_location\_from}(a, l, e)\ \& \\ \text{attractive\_route\_to}(\gamma, a, l)\ \&\ \forall e'\ \text{state}(\gamma, t) \not\models \\ \text{attractive\_direction\_at}(a, l, e')\ ]\end{gathered}$$

GP3 Reaching End of Attractive Route
Ants reach the end of their attractive route.
$\forall a\ \exists t\ \text{reaches\_end\_attractive\_route}(\gamma, t, a)$

GP4 Returning to Nest
Ants get back to the nest from the end of their attractive routes.
$\forall a\ \forall t1\ \exists e, t2 > t1\ \exists l\ [\ \text{reaches\_end\_attractive\_route}(\gamma, t1, a) \Rightarrow \text{state}(\gamma, t2) \models \text{is\_at\_location\_from}(a, l, e)\ \&\ \text{state}(\gamma, t2) \models \text{nest\_location}(l)\ ]$

GP5 From Food to Nest
Ants get back to the nest from locations of food.
$\forall a, e\ \forall t1\ \exists t2 > t1\ \exists l, l', e'$
$[\ \text{state}(\gamma, t1) \models \text{is\_at\_location\_from}(a, l, e)\ \&\ \text{state}(\gamma, t1) \models \text{food\_location}(l)\ ] \Rightarrow$
$\text{state}(\gamma, t2) \models \text{is\_at\_location\_from}(a, l', e')\ \&\ \text{state}(\gamma, t2) \models \text{nest\_location}(l')$

These and a number of other properties have been formalized, and using a checking software environment have been (automatically) *verified in simulation traces*. This is the first manner for verification. A second way of verification is to establish *logical relationships* between properties (by mathematical proof). This also has been performed in a number of cases. For example, under a number of assumptions, the following relationships hold:

$$\begin{gathered}\text{GP4} \Rightarrow \text{GP5} \\ \text{GP3}\ \&\ \text{GP4} \Rightarrow \text{GP2}\end{gathered}$$

The assumptions include the following:

- Attractive routes are not branching and are not crossing each other or themselves.
- At least two ants exist for which the attractive routes end at a food location and are short enough compared to the evaporation rate of pheromones to return.
- GP5 is only valid in the infinite future since food sources are not depleted. In practice, the simulations stop, invalidating GP5 for the ants that are still on their way to the nest.

Furthermore, an additional premise of temporal completion (see [20]) is needed. For example, any of the following trivial (nonintended) world situations would disturb the ants: an ant comes to a location that contains a pheromone that is there without any reason (no ant dropped it), or on its way back, an ant comes to a location without a pheromone (the pheromone immediately disappeared). It is clear that the above properties can only be proven under the assumption that nothing unexpected will happen. To put it differently, proofs can be given under the assumption that the set of local properties determines the whole range of events. This assumption has been added as a premise to establish the logical relationships between the properties.

6. Discussion

Clark and Chalmers [3, section 5] provide four criteria for an extended mind: (1) the external information is a constant in the agent's life—when the information is relevant, he or she will rarely take action without consulting it; (2) the external information is directly available; (3) the agent endorses retrieved external information; and (4) the external information has been endorsed at some point in the past and is there as a consequence of this endorsement. How do these criteria apply to the ants' case? First, indeed, an ant always senses the pheromone before choosing a direction. Second, at each location, the pheromone is immediately accessible for sensing. Third, the decision for the direction is indeed always based on the pheromone. Finally, the external information is endorsed in the past: the pheromone was dropped at the direction from whence one or more ants traveled.

The extended mind perspective introduces an additional, cognitive ontology to describe properties of the physical world, which essentially is an antireductionist step, providing a more abstract and better manageable, higher level conceptualization. For example, considering part of the external world as an extended mind allows one to give another interpretation to external physical processes and states. Physical state properties such as "pheromone is present at d" can be reconceptualized as, for example, "the group as a whole believes that d is a relevant path." In Jonker, Treur, and Wijngaards [19], a number of arguments can be found for why such antireductionist steps can be useful in explanation and theory development; also see section 3 above.

In this article, following the extended mind perspective, the first steps have been made toward a high-level conceptualization of physical processes. The main contribution of this article is the formalization and logical analysis of this high-level conceptualization. The formalization enables simulation and automated checking of dynamic properties of traces or sets of traces and allows one to logically relate dynamic properties of different aggregation levels to each other. All this would have been more difficult in the case of an algorithmic or physically oriented modeling perspective, involving, for example, differential equations and gradients of concentrations. As a next step, the authors are currently investigating to what extent collective processes such as ant behavior can be interpreted and formalized as single-agent processes. The first results of this research, including a formal mapping between a single-agent and a multiagent conceptualization, are described in Bosse and Treur [17]. Moreover, work is currently in progress to model other examples of the shared extended mind (outside the domain of ants). In this research, the focus is on organisms with more complex cognitive capacities (humans in particular). Meanwhile, work is in progress to elaborate the isomorphism principle mentioned in section 3 in more detail.

Regarding details of the simulation, the authors are currently exploring whether the behavior prescribed by the attractiveness of a route can be replaced by random route selection. In addition, experiments with food sources at different distances from the nest will be undertaken to determine the relation between evaporation rate and ants finding their way home. Therefore, these food sources will be made depletive. Also, the effect of using different types of pheromones will be studied. Finally, an advanced visualization environment is currently being developed to make the simulation traces more readable.

7. Acknowledgments

The authors are grateful to Lourens van der Meij for his contribution to the development of the software environment and to two anonymous referees for their comments to an earlier version of this article.

8. References

[1] Clark, A. 1997. *Being there: Putting brain, body and world together again*. Cambridge, MA: MIT Press.

[2] Clark, A. 2001. Reasons, robots and the extended mind. *Mind & Language* 16:121-45.

[3] Clark, A., and D. Chalmers. 1998. The extended mind. *Analysis* 58:7-19.

[4] Dennett, D. C. 1996. *Kinds of mind: Towards an understanding of consciousness*. New York: Basic Books.

[5] Kirsh, D., and P. Maglio. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science* 18:513-49.

[6] Menary, R., ed. Forthcoming. *The extended mind*. Amsterdam: John Benjamins.

[7] Bosse, T., C. M. Jonker, and J. Treur. 2002. Simulation and analysis of controlled multi-representational reasoning processes. In *Proceedings of the Fifth International Conference on Cognitive Modelling, ICCM'03*, Universitäts-Verlag Bamberg, pp. 27-32.

[8] Bonabeau, J., M. Dorigo, and G. Theraulaz. 1999. *Swarm intelligence: From natural to artificial systems*. New York: Oxford University Press.

[9] Deneubourg, J. L., S. Aron, S. Goss, J. M. Pasteels, and G. Duerinck. 1986. Random behavior, amplification processes and number of participants: How they contribute to the foraging properties of ants. In *Evolution, games and learning: Models for adaptation in machines and nature*, edited by D. Farmer, A. Lapedes, N. Packard, and B. Wendroff, 176-86. Amsterdam: North Holland.

[10] Dorigo, M., and T. Stützle. 2004. *Ant colony optimization*. Cambridge, MA: MIT Press.

[11] Jonker, C. M., and J. Treur. 2002. Compositional verification of multi-agent systems: A formal analysis of pro-activeness and reactiveness. *International Journal of Cooperative Information Systems* 11:51-92.

[12] Barringer, H., M. Fisher, D. Gabbay, R. Owens, and M. Reynolds. 1996. *The imperative future: Principles of executable temporal logic*. New York: Research Studies Press Ltd. and John Wiley.

[13] Forbus, K. D. 1984. Qualitative process theory. *Artificial Intelligence* 24 (1-3): 85-168.

[14] Port, R. F., and T. van Gelder, eds. 1995. *Mind as motion: Explorations in the dynamics of cognition*. Cambridge, MA: MIT Press.

[15] Law, A. D., and W. D. Kelton. 2000. *Simulation modeling and analysis*. New York: McGraw-Hill.

[16] Bosse, T., C. M. Jonker, L. van der Meij, and J. Treur. 2005. LEADSTO: A language and environment for analysis of dynamics by simulation. In *Proceedings of the Third German Conference on Multi-Agent System Technologies, MATES'05*, edited by T. Eymann, F. Kluegl, W. Lamersdorf, M. Klusch, and M. N. Huhns, 165-78. Lecture Notes in AI, vol. 3550. New York: Springer- Verlag.

[17] Bosse, T., and J. Treur. 2005. Formal interpretation and analysis of collective intelligence as individual intelligence. In *Proceedings of the Sixth International Workshop on Multi-Agent-Based Simulation, MABS'05*, pp. 51-65.

[18] Kim, J. 1996. *Philosophy of mind*. Boulder, CO: Westview.
[19] Jonker, C. M., J. Treur, and W. C. A. Wijngaards. 2002. Reductionist and antireductionist perspectives on dynamics. *Philosophical Psychology Journal* 15:381-409.
[20] Engelfriet, J., C. M. Jonker, and J. Treur. 2002. Compositional verification of multi-agent systems in temporal multi-epistemic logic. *Journal of Logic, Language and Information* 11:195-225.

Tibor Bosse *is a PhD student at the Vrije Universiteit Amsterdam, Department of Artificial Intelligence, Amsterdam, The Netherlands.*

Catholijn M. Jonker *was an associate professor at the Vrije Universiteit Amsterdam, Department of Artificial Intelligence, Amsterdam, the Netherlands. She is currently a full professor at the Radboud Universiteit Nijmegen, Nijmegen Institute for Cognition and Information, Nijmegen, The Netherlands.*

Martijn C. Schut *is an assistant professor at the Vrije Universiteit Amsterdam, Department of Artificial Intelligence, Amsterdam, The Netherlands.*

Jan Treur *is a full professor at the Vrije Universiteit Amsterdam, Department of Artificial Intelligence, Amsterdam, the Netherlands, and was also a guest professor at the Utrecht University, Department of Philosophy, Utrecht, The Netherlands.*