Modeling Decentralized Organizational Change in Honeybee Societies

Mark Hoogendoorn, Martijn C. Schut, and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{mhoogen, schut, treur}@cs.vu.nl

Abstract. Multi-agent organizations in dynamic environments, need to have the ability to adapt to environmental changes to ensure a continuation of proper functioning. Such adaptations can be made through a centralized decision process or come from the individuals within the organization. In the domain of social insects, such as honeybees and wasps, organizations are known to adapt in a decentralized fashion to environmental changes. An organizational model for decentralized organizational change is presented that can aid in analyzing and designing such organizations. The model is specified by dynamic properties at different aggregation levels. At the lowest level such properties characterize the behavior of individual roles, which can be related to higher level properties that express important elements such as survival of an organization. A honeybee colony is used as a case study.

1. Introduction

The concept of organization has been studied in sciences such as social science and economics, but recently also in artificial intelligence [3;4;7]. With the desire to analyze and design more complex systems consisting of larger numbers of agents (e.g., in nature, society, or software), the need arises for a concept of higher abstraction than the concept agent. To this end, organizational modeling is becoming a practiced stage in the analysis and design of multi-agent systems. Hereby, the environment in which the multi-agent organization participates has to be taken into consideration. An environment can have a high degree of variability which might require organizations that change to adapt to the environment's dynamics, to ensure a continuous proper functioning of the organization. Hence, such change processes are a crucial function of the organization and should be part of the organizational model.

An organizational model incorporating organizational change can be specified in two ways: from a centralized perspective, in which there is a central authority that determines the changes to be performed within the organization, taking into account the current goals and environment. In [5] for example, an organizational model is presented which is specified from such a perspective, whereby a single entity, called the change manager, is responsible for directing the change. A second possibility is to create a model for organizational change from a decentralized perspective, in which each agent decides for himself if and how to change its own role allocations. In the

latter approach, it is much more difficult for the organization as a whole to change in a coherent way, still satisfying the goals set for the organization, as there is no overall view of the organizational change. The approach might however be the only possibility for an organization to perform change as a central authority for performing change could be non existing or infeasible due to the nature of the organization. In the domain of social insects, such as honeybees and wasps, organizations are known to adapt in a decentralized fashion to environmental changes. This paper presents a model for decentralized organization change appropriate for such phenomena as occur in Nature. Such a model can aid developers of multi-agent systems in creating and analyzing such an organization. The description of the model abstracts from the actual tasks being performed by the organization. The scope of the model is broader than simply being able to model social insects: the mechanisms incorporated in the model facilitating decentralized organizational change may work in other types of organizations as well. In [1] for example, a comparable approach is used for finding an optimal allocation of cars to paint booths.

To evaluate the model being proposed, as a case study the honeybee (Apis Mellifera) has been investigated. For this domain the model has been instantiated. The instantiated model has been validated against properties as acquired from biological experts. A number of different roles have been identified in the literature (see e.g., [8;11]). For the sake of brevity only five will be addressed here: (1) a *brood carer* takes care of feeding the larvae within the bee hive; (2) a *patroller* guards the hive by killing enemies entering the hive; (3) a *forager* harvests food to be stored in the hive; (4) an *undertaker* cleans the hive of corpses, and (5) a *resting worker* simply does nothing.

Switching between roles is triggered by changes in the environment observed by the bees. Such observations differ per bee. Each role has a specific trigger, for which a bee has a certain threshold that determines whether this is the role it should play. The bee always plays the role for which it is most triggered. For example, bees are triggered to start playing the *brood carer* role when they observe the larvae emitting a too high level of hunger pheromones. Once they are allocated to the role, they start getting food from the combs and feed the larvae that are emitting the pheromones. A trigger for the *patroller* role is the amount of enemies observed around the hive. *Foragers* that have returned from their hunt for food, communicate the location where they found the food by means of the honeybee dance (see [2]). For other bees currently not playing the *forager* role, such a dance is a trigger to start playing the *forager* role. The more corpses there are, the more bees are being triggered to switch from their current role to being *undertaker*. Bees perform the *resting worker* role in case they are not sufficiently triggered for any other role.

Section 2 presents the methodological approach used. The model for decentralized organizational change is described in Sections 3 (properties at organization level) and 4 (role properties). Results of a simulation of the organizational model instantiated with domain-specific knowledge of the bee colony are shown in Section 5, and finally Section 6 concludes the paper.

2. Modeling Organizational Dynamics

To enable modeling an organization, an expressive language is needed that has the ability to describe the dynamics of such an organization. For this purpose TTL (Temporal Trace Language) has been adopted cf. [6]. TTL allows for the formal specification of dynamic properties on multiple levels of aggregation. The bottom level addresses role properties, describing the required behavior for each of the roles within the organization. On the top level organization properties are defined, expressing the overall goals or requirements for the organization. An advantage of using TTL is that an executable subset has been defined called *leadsto* which is of the form $\alpha \twoheadrightarrow_{e,f,g,h} \beta$ that states that if α holds for duration g then β will holds for duration h with a delay between e and f. In case role properties are expressed in this executable format, the organizational model can be simulated by putting certain (e.g., environmental) events in the model (without including agents in the model), resulting in a trace of the organizational behavior. The top level organization properties can thereafter be checked against the trace by means of an automated tool called *TTL checker* to see whether the organizational model indeed satisfies the goals or requirements set for it, given the events that have been put into the model. Using the results of these checks, statements can be made about the behavior of the organization, when the agents comply to the role properties that have been defined. More details and the semantics for TTL can be found in [9]. Examples and explanation of properties expressed in TTL are shown in Section 3 and 4.

3. Organizational Properties

The model for decentralized organizational change presented here takes the form of a hierarchy of dynamic properties at two aggregation levels: that of the organization, and that of the roles within the organization. This section describes a number of such properties as well as the relationships between them.

The highest level requirement for the organization as a whole as inspired by the biological domain experts, is survival of the population given a fluctuating environment, in other words, population size needs to stay above a certain threshold M.

OP1(M) Surviving Population
For any time t, a time point t'≥t exists such that at t' the population size is at least M.

Formal:

$\forall t \, \exists t' \geq t, v : \text{state}(\gamma, t') \models \text{total\_living\_population\_count}(v) \, \& \, v \geq M$

Here state(γ, t') |= total_living_population_count(v) denotes that within the state state(γ, t') at time point t' in trace γ the state property total_living_population_count(v) holds, denoted by the (infix) predicate |= for the satisfaction relation. Such a high-level requirement is refined by means of a property hierarchy, depicted as a tree in Figure 1. At the highest level OP1 is depicted which can be refined into a number of properties (in Figure 1 *n* properties) each expressing that for a certain aspect the society is in good condition,

characterized by a certain *value* for a variable (the *aspect variable*) that is to be maintained. The property template for an aspect X is as follows:

OP2(X, P1, P2) Organization Aspect Maintenance
For all time points t
If v is the value of aspect variable X at t, then v is between P1 and P2

$\forall t, v :$ state$(\gamma, t) \models$ has_value$(X, v) \Rightarrow P1 \le v \le P2$

Sometimes one of the two bounds is omitted, and it is only required that value v is at least P1 (resp., at most P2). For the example bee society the aspects considered are *wellfed brood*, *safety*, *food storage*, and *cleanness* (addressed, respectively, by Brood Care, Patroller, Forager, and Undertaker roles). For each of these aspects a variable was defined to indicate the state of the society for that aspect. For example, for wellfed brood, this variable concerns relative larvae hunger, indicated by the larvae pheromone rate.

In order to maintain the value of an aspect variable X, a certain effort is needed all the time. To specify this, a property that expresses the *effort* made by the organization on the aspect, is introduced. Notice that the notion of provided effort at a time point t can be taken in an *absolute* sense (for example, effort as the amount of feeding work per time unit), but it can also be useful to take it in a *relative* sense with respect to a certain overall amount, which itself can vary over time (for example, effort as the fraction of the amount of feeding work per time unit divided by the overall number of larvae). Below the latter, relative form will be taken. The general template property for aspect effort is as follows:

OP3(X, W1, W2) Sufficient Aspect Effort
For all time points t the effort for aspect X provided by the organization is at least W1 and at most W2.

$\forall t, v :$ state$(\gamma, t) \models$ provided_effort$(X, v) \Rightarrow W1 \le v \le W2$

For the bee colony, for instance, the brood care workers take care that the larvae are well-fed. The effort to maintain the hunger of larvae at a certain low level is feeding the larvae. Here provided effort for brood care is defined as the brood care work per time unit divided by the larvae population size. Brood care work is taken as the amount of the (average) brood care work for one individual brood carer times the number of brood carers.

Whether the refined properties given above will always hold, depends on the flexibility of the organization. For example, in the bee colony case, if the number of larvae or enemies increases, also the number of brood care workers, respectively patrollers should increase. If the adaptation to the new situation takes too much
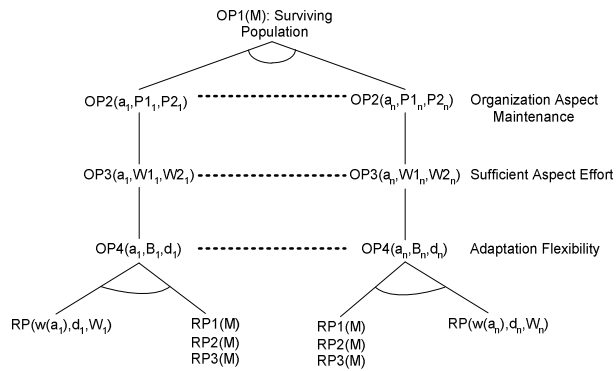

Fig. 1. Property hierarchy for decentralized organizational change

time, the property Brood Care Effort will not hold for a certain time. In principle, such circumstances will damage the success of the organization. Therefore, an adaptation mechanism is needed that is sufficiently flexible to guarantee the properties such as Brood Care Effort. For this reason, the adaptation flexibility property is introduced, which expresses that when the effort for a certain organization aspect that is to be maintained is below a certain value, then within a certain time duration d it will increase to become at least this value. The smaller this parameter d is, the more flexible is the adaptation; for example, if d is very large, the organization is practically not adapting. The generic property is expressed as follows:

OP4(X, B, d) Adaptation Flexibility
At any point in time t, if at t the effort for aspect X provided by the organization is lower than B, then within time duration d the effort will become at least B.

$\forall t, v1 \;[\; [\text{state}(\gamma, t) \;|= \text{provided\_effort}(X, v1) \;\&\; v1 < B \;] \;\Rightarrow$
$\exists t' \geq t, v2 : [\; t' \leq t+d \;\;\&\;\; \text{state}(\gamma, t') \;|= \text{provided\_effort}(X, v2) \;\&\; v2 \geq B \;]\;]$

An assumption underlying this property is that not all aspects in the initial situation are critical, otherwise the adaptation mechanism will not work. OP3 expressing that sufficient effort being provided directly depends on this adaptation mechanism as shown in Figure 1. OP4 depends on role properties at the lowest level of the hierarchy, which are addressed in the next Section.

4. Role Properties

Roles are the engines for an organization model: they are the elements in an organization model where the work that is done is specified. The properties described in Section 3 in an hierarchical manner have to be grounded in role behavior properties as the lowest level properties of the hierarchy. In other words, specifications of role properties are needed that entail the properties at the organizational level described in Section 3. In the behavioral model two types of roles are distinguished: Worker roles which provide the effort needed to maintain the different aspects throughout the organization, and Member roles which have the function to change Worker roles. Each Member role has exactly one shared allocation with a Worker role. The role behavior for the Worker roles within the organization is shown in Section 4.1, whereas Section 4.2 specifies the behavior for the Member roles.

4.1 Worker Role Behavior

Once a certain Worker role exists as an active role, it performs the corresponding work. What this work exactly is, depends on the application: it is not part of the organization model. The property directly relates to OP4 which specifies the overall effort provided, as shown in Figure 1. Note that Figure 1 only shows the generic form of the role property (depicted as $RP(w(a_i), d_i, W_i)$ where a_i is the specific aspect and $w(a_i)$ the Worker role belonging to that aspect) whereas in an instantiated model a role property is present for each instance of the Worker role providing the effort for the specific aspect. In a generic form this is specified by:

RP(R, d, W) Worker Contribution

For all t there is a t' with t ≤ t' ≤ t + d such that at t' the Worker role R delivers a work contribution of at least W.

∀t ∃t' ≥ t, v : [t' ≤ t+d & state(γ, t') |= work_contribution(R, v) & v ≥ W]]

Here work_contribution is part of the state ontology for the output of the role. For each of the specific roles it can be specified what the work contribution is in terms of the domain specific state ontology (e.g., the number of larvae to be fed for the brood carer role).

4.2 Member Role Behavior

By a Member role M decisions about taking up or switching between Worker roles are made. As input of this decision process, information is used about the well-being of the organization, in particular about the different aspects distinguished as to be maintained; these are input state properties indicating the value of an aspect variable X: has_value(X, v). Based on this input the Member role M generates an intermediate state property representing an indication of the aspect that is most urgent in the current situation. In the model the decision mechanism is indicated by a priority relation priority_relation(X_1, v_1, w_1, ..., X_n, v_n, w_n, X) indicating that aspect X has priority in the context of values v_i, respectively norms w_i for aspects X_1, .., X_n. This priority relation can be specialized to a particular form, as shown below by an example specialization in the last paragraph of this section.

RP1(M) Aspect Urgency

At any t, if at t Member role M has norms w_1 to w_n for aspects X_1 to X_n

and receives values v_1 to v_n for X_1 to X_n at its input,

and has a priority relation that indicates X as the most urgent aspect for the
    combination of these norms and values,

then at some t' ≥t it will generate that X is the most urgent aspect.

∀t, v1, .., vn, w1, .., wn, X
  state(γ, t) |= has_value(X_1, v_1) & … & has_value(X_n, v_n) &
        has_norm(X_1, w_1) & … & has_norm(X_n, w_n) &
        priority_relation(X_1, v_1, w_1, …, X_n, v_n, w_n, X)
⇒ ∃t'≥t state(γ, t') |= most_urgent_aspect(X)

Based on this, the appropriate role for the aspect indicated as most urgent is determined. If it is not the current role sharing an allocation with M, then another intermediate state property is generated expressing that the current Worker role sharing an allocation with M should be changed to the role supporting the most urgent aspect. In other words, the shared allocation of Member role M in the Change Group should change from one (the current) Worker role R1 in Worker Group WG1 to another one, Worker role R2 in Working Group WG2:

RP2(M) Role Change Determination

At any t, if at t Member role M generated that X is the most urgent aspect,

and Worker role R2 is responsible for this aspect, and

R1 is the current Worker role sharing an allocation with M, and R1 ≠ R2,

then at some t' ≥t it will generate that role R2 has to become the Worker role sharing an allocation with M, instead of R1.

∀t, X, R1, R2   state(γ, t) |= most_urgent_aspect(X) &
        role_responsible_for(R2, X) & role_reserved_for(R2, M) &
        state(γ, t) |= has_shared_allocation(M, R1) & R1≠R2

$\Rightarrow \exists t' \geq t$ state(γ, t') |= shared_allocation_change(M, R1, R2)

Based on this intermediate state property the Member role M generates output indicating which role should become a shared allocation and which not anymore:

RP3(M) Role Reallocation
At any t, if at t Member role M generated that Worker role R2 has to become sharing an allocation with M, instead of Worker role R1,

then at some t' $\geq$ t it will generate the output that role R1 will not share an allocation with M and R2 will share an allocation with M.

$\forall$t, R1, R2
 state(γ, t) |= shared_allocation_change(M, R1, R2)
$\Rightarrow \exists t' \geq t$ state(γ, t') |= not has_shared_allocation(M, R1) &
has_shared_allocation(M, R2)

All three role properties for the Member roles are depicted in Figure 1. The adaptation step property OP4 for all organizational aspects dependent upon it, so each of the OP4 branches depends upon RP1, RP2, and RP3 which have therefore been depicted two times in the Figure.

The generic description for the Member role behavior can be specialized one step further by incorporating a specific decision mechanism. This gives a specific definition of the priority relation priority_relation(X_1, v_1, w_1, ..., X_n, v_n, w_n, X) as has been done for the following decision mechanism based on norms used as thresholds (see e.g. [10]).

1. For each aspect X to be maintained a norm w(X) is present. For the Worker role R1 for X sharing an allocation with Member role M, each time unit the norm has a decay described by fraction r.
2. For each X, it is determined in how far the current value is unsatisfactory, expressed in a degree of urgency u(X) for that aspect.
3. For each aspect with urgency above the norm, i.e., with u(X) > w(X), the relative urgency is determined: u(X)/ w(X)
4. The most urgent aspect X is the one with highest relative urgency.

5. Simulation Results

This section discusses some of the results of simulations that have been performed based on the organizational model, in particular the role properties presented in Section 4 have been put in an executable format and have been instantiated with domain-specific information for bee colonies.

To validate the instantiated simulation model, the high-level dynamic properties from Section 3 were used (in accordance with biological experts). Proper functioning of such an organization in Nature is not self-evident, therefore two simulation runs are compared: one using the adaptation mechanism, and one without. The choice has been made to compare the result of using adaptation with no adaptation due to the fact that comparing with centralized change will obviously result in better performance of the centralized change model. Having a complete picture of the different aspects and their urgencies, gives a major advantage. Whether decentralized adaptation is more successful than no adaptation is however not a trivial matter. In case there is merely coordination by means of observables in the world (which is the case in honeybee colonies), adaptation might even be counterproductive. Note that the results presented

here are the results of a simulation of the instantiated organizational model, abstracting from allocated agents. Performing such high-level simulations of an executable organizational model enables the verification of properties against these simulation runs. Hence, it can be checked whether or not the model satisfies the properties or goals considered important. When such properties are indeed satisfied, by allocating agents to the roles that comply to the role properties, the multi-agent system delivers the desired results as well. In the two simulations, several parameters have been set to certain values, where the circumstances are kept identical for both simulations.

External world. Initially, 15 larvae and 10 workers are present for which the initial type of the latter is randomly assigned. The natural mortality age is set to 500 time steps, whereas a larva is grown up after 250 time steps. Every 20 time steps, a new larva is added to the population. The initial food stock is set to 40 units of food. Once every 100 time points an attack of 40 enemies occurs, who stay there until a patroller defeats them. In case over 200 enemies are present in the hive, each individual in the organization is removed with a probability of 0.05 per time step. In case more than 20 dead bodies are present in the hive, individuals are removed with the same probability. Food used by larvae is 0.5 per feed, for workers 1 unit of food per time step.

Larvae. Larvae have an initial pheromone level of 0.5, increasing 0.006 per time step. In case pheromone emissions exceed 0.95, the larva dies. After being fed, the emission level is set to 0.1.

Foragers. Foragers each collect 3 food units per time step.

Brood carers. Feed 1 larvae per 8 time steps, and only feed the larvae with a pheromone level above 0.55.

Undertaker. Carry 1 body per 12 time steps.

Patroller. Defeat 1 enemy per time step.

In the adaptation simulation, the Member thresholds are randomly generated, being somewhat above or below the average observed value of the various triggers. Figure 2 shows results on the performance of the two settings of the organizational model. Figure 2a shows the overall population size over time. The population size of the simulation with adaptation remains relatively stable, whereas without adaptation it drops to a colony of size 3, which is equal to the amount of larvae living without being fed. Figures 2b and 2c show information regarding brood care: Firstly, the average pheromone level, the trigger to activate the allocation to brood carer. Furthermore, the number of active brood carers in the colony is shown. In the case with adaptation their number increases significantly in the beginning of the simulation, as the amount of pheromones observed is relatively high. Therefore, a lot of the brood carer roles are allocated. For example, at time point 300, 15 out of a population of 28 are brood carers.

Despite the fact that the overall pheromone level is not decreasing rapidly, the amount of brood carer roles drops significantly after time point 300. This is due the fact that Member roles can only share an allocation with one Worker role at a time. When another role receives a higher urgency (e.g., there is a huge attack, demanding many patrollers) a switch of worker role takes place. Figure 2d shows the amount of

worker roles of the different types (except the resting workers) within the bee colony for the setting with adaptation. The amount of brood carers decreases after time point 300 due to an increase in the amount of shared allocations to the undertaker and forager roles. This results in an increase in pheromone level again, causing a higher delta for brood care again, resulting in more brood carers, etc. The pheromone level finally stabilizes around 0.5 in the organizational model with adaptation. For the setting without adaptation, the brood carers simply cease to exist due to the fact that none of the larvae are growing up. The pheromone level stabilizes at a higher level.

The properties from Section 3 have been checked by the automated TTL checker. With the following parameter settings, the properties were validated and confirmed for the organizational model with adaptation and falsified for the one without adaptation: OP1(20), OP2(broodcare,0,0.9), OP3(broodcare,0.15,10000), OP4(broodcare, 0.3, 200).
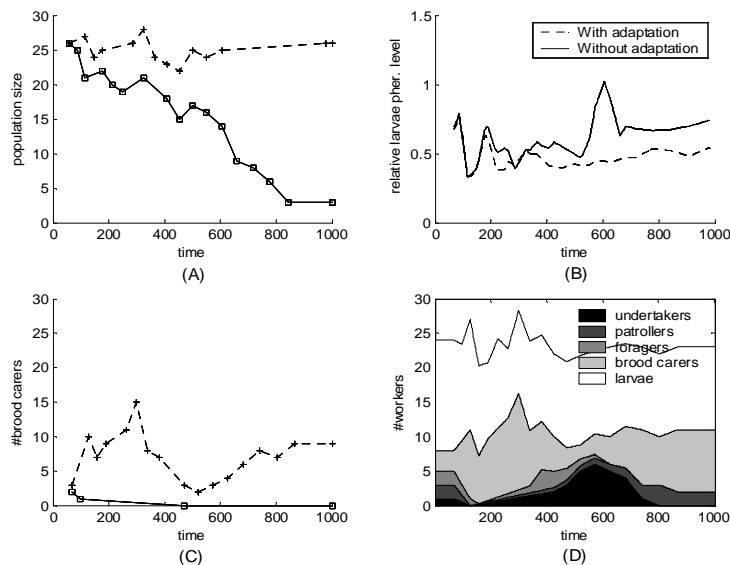

Fig. 2. Results of simulating the bee colony with and without adaptation. Note that (D) only shows the worker types for the adaptive case

6. Discussion

The organizational model for decentralized organizational change has been formally specified by means of a methodology which describes the behavior of an organization on multiple aggregation levels; cf. [6]. The model is inspired by mechanisms observed in Nature. The model was used for a honeybee colony case study. The scope of the model is not limited to being a model for social insects: in [1] the effectiveness of such approaches is shown for other domains as well. The model can therefore support organizational modelers and analysts working with multi-agent organizations in highly dynamic environments, without a central authority directing change, in

general in designing and analyzing such an organization. The formal specification of the behavior of the organization is described by dynamic properties at different aggregation levels. Once the lowest level properties within the organization are specified in an executable form, the organizational model can be used for simulation abstracting from agents (to be) allocated. Such low level properties can be indicative for the behavior of the agent allocated to that particular role. The possibility also exists to specify the role properties at the lowest aggregation level in a more abstract manner, in a non-executable format. Hierarchical relations between the properties can be identified to show that fulfillment of properties at a lower level entails the fulfillment of the higher level properties. Simulations using agents can be performed and checked for fulfillment of these properties. Properties for the behavior of roles regarding decentralized organizational change have been specified on an executable level to be able to perform simulation, and higher-level properties have been identified as well. The case study of the honeybee colony was used as an evaluation of the model. Simulation of this instantiated model showed that given the external circumstances, it was effective, given overall properties put forward by biological experts.

References

[1] Bonebeau, E. and Theraulaz, G., 2000, Swarm Smarts, Scientific American, 282 (3): 72-79.

[2] Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E., 2001, *Self-Organization in Biological Systems*, Princeton Univ. Press, Princeton, USA.

[3] Furtado, V., Melo, A., Dignum, V., Dignum, F., Sonenberg, L., 2005, Exploring congruence between organizational structure and task performance: a simulation approach. In: Boissier, O., Dignum, V., Matson, E., Sichman, J. (eds.), *Proc. of the 1st OOOP Workshop*.

[4] Giorgini, P., Müller, J., Odell, J. (eds.), 2004, *Agent-Oriented Software Engineering IV*, LNCS, vol. 2935, Springer-Verlag, Berlin.

[5] Hoogendoorn, M., Jonker, C.M., Schut, M., and Treur, J, 2007, Modeling Centralized Organization of Organizational Change, *Computational and Mathematical Organization Theory*, vol.13, 2007, pp. 147-184.

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

[7] McCallum, M., Vasconcelos, W.W., and Norman, T.J., 2005, Verification and Analysis of Organisational Change. In: Boissier, O., Dignum, V., Matson, E., Sichman, J. (eds.), *Proc. 1st OOOP Workshop*.

[8] Schultz, D.J., Barron, A.B., Robinson, G.E., 2002, A Role for Octopamine in Honey Bee Division of Labor, *Brain, Behavior and Evolution*, vol. 60, pp. 350-359.

[9] Sharpanskykh, A., Treur, J., 2005, Temporal Trace Language: Syntax and Semantics, Technical Report, Vrije Universiteit Amsterdam, Dept. of Artificial Intelligence, A'dam.

[10] Theraulaz, G., Bonabeau, E., and Deneubourg, J.L., 1998, Response thresholds reinforcement and division of labor in insect societies. Proceedings of the Royal Society of London Series B-Biological Sciences, 265: 327-332.

[11] Winston, M.L. and Punnet, E.N., 1982, Factors determining temporal division of labor in honeybees, *Canadian Journal of Zoology*, vol. 60, pp. 2947-2952.