

An Adaptive Multi-Agent Organization Model Based on Dynamic Role Allocation*

Mark Hoogendoorn and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{mhoogen, treur}@cs.vu.nl <http://www.cs.vu.nl/~{mhoogen, treur}>

Abstract

Organizations involving multiple agents require adaptation mechanisms to guarantee robustness, especially in critical domains. This paper presents an organizational template to aid analysis and design of organizations with adaptation mechanisms based on dynamic role reallocation. The adaptive organization model can be used both for qualitative and quantitative domains, as is shown in two application cases made to evaluate the applicability of the model.

1. Introduction

Robustness of a multi-agent organization functioning in critical domains is essential. Unpredictability can both be in the internal functioning of the system itself (e.g., an incorrect functioning agent), or external to the system (e.g., a sudden increase in environmental pressure). To enable a multi-agent organization to be robust, capabilities are required that allow the organization to adapt in order to continue functioning adequately.

An approach could be to model a multi-agent system in which each of the agents have those specific capabilities, and show the effectiveness of the system as a whole in this domain. However, it is hard to generalize results obtained beyond the specific domain and the specific agents occurring in this domain. Recently, an abstraction level higher than the concept agent has become in use: the organizational level (see e.g. [3], [12]). On this level, templates can be specified to aid analysts in modeling appropriate multi-agent organization models. These templates, for example, include specification of roles, possibly in the form of required behavior. In a given application,

* A preliminary, shorter version of this work was presented at the International Conference on Intelligent Agent Technology, IAT'06.

agents can be allocated to such roles. The templates can be reused each time a new domain is analyzed for which the characteristics comply to the ones specified for the template. Once the correctness of the template is proven (given certain domain assumptions) for a desired property, each model which complies to the specified template will satisfy that property as well, making the approach reusable. Of course, for each new case in which the template is used, an instantiation with domain-specific knowledge is still required.

This paper presents such an organizational model or template for the analysis of multi-agent organizations with the ability to adapt to unpredictable circumstances, maintaining the robustness of the system. The essential part of the organizational model is the specification of roles, since those can be seen as the engines of the organization. The approach taken distinguishes a number of aggregation levels, starting with the highest level dynamic property desired (i.e., robustness) and refining this property in a number of steps until the level of role behavior has been reached. Interlevel relations between dynamic properties at the different aggregation levels have been specified and verified using the model checker SMV [21]. The applicability of the model has been evaluated by using it to analyze two application case studies in different domains, one qualitative, and one quantitative.

The remainder of this paper is organized as follows. Section 2 introduces the modeling approach used to specify the organization model, which includes both a structural and a behavioral specification of the organization. The structural model within the template is specified in Section 3, whereas Section 4 presents the behavioral model, without taking into account adaptation. The adaptation model is presented in Section 5. Section 6 presents a qualitative application of the template in the domain of incident management and presents simulation results. In Section 7 a quantitative specialization of the model is specified which is applied in the domain of social insects in Section 8. Section 9 is a discussion.

2. Modeling Approach

This Section presents the modeling approach used to specify the adaptive multi-agent organization model. First, the framework used to model organization structure will be explained, and thereafter the method for describing the behavior of such an organization. For describing the structural part of the model for adaptive multi-agent organizations, the AGR (Agent-Group-Role) modeling approach introduced in [7] is used. Here three basic concepts are used to model a multi-agent organization: agent, group, and role. An agent is an active communicating entity which plays roles within groups. Groups are sets of roles; a role is an abstract representation of an agent function or service.

The approach presented in [8] which is partly based on AGR is used to specify behavioral or dynamic properties on multiple aggregation levels: following AGR, the functioning of the particular roles, the functioning of groups, and of the organization as a whole. In general, behavioral properties are expressed in terms of temporal relations over input states and output states of roles over time.

Properties at the different levels can be structured by means of interlevel relations in a hierarchy. At the lowest level role properties describe the behavior of an individual role whereas transfer properties describe the dynamics of (intragroup) transfer between roles. For the roles within a given group, such role properties, together with the transfer properties, entail the group properties that characterize the behavior of the group as a whole. The group properties for the different groups, together with the inter-group relationship properties (for transfer between groups), entail the overall organization properties. Properties about the environment are treated the same way as groups and roles.

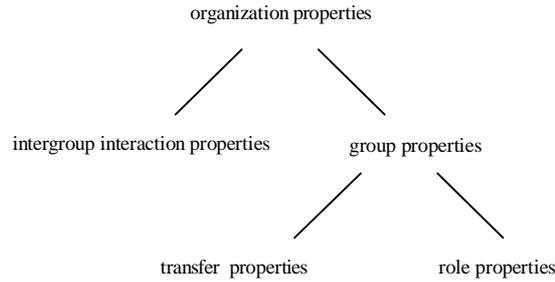


Figure 1. Overview of interlevel relations between dynamic properties

Specification of dynamic properties is done in the *Temporal Trace Language* (TTL) [16]. This language can be classified as a reified predicate-logic-based temporal logic (see, e.g., [10], [11]), in contrast to, for example, modal-logic-based temporal logics as the ones discussed in, e.g., [9]. The language is briefly introduced here. For more details, see [4].

States and Traces

In TTL, ontologies for world states are formalised as sets of symbols in sorted predicate logic. For any STATE ontology Ont , the ground atoms form the set of *basic state properties* $BSTATPROP(Ont)$. Basic state properties can be defined by nullary predicates (or proposition symbols) such as n -ary predicates (with $n > 0$) like $has_temperature(environment, \tau)$. The *state properties* based on a certain ontology Ont are formalised by the propositions (using conjunction, negation, disjunction, implication) made from the basic state properties; they constitute the set $STATPROP(Ont)$.

In order to express dynamics in TTL, in addition to state properties, important concepts are *states*, *time points*, and *traces*. A *state* S is an indication of which basic state properties are true and which are false, i.e., a mapping $S: BSTATPROP(Ont) \rightarrow \{true, false\}$. The set of all possible states for ontology Ont is denoted by $STATES(Ont)$. Moreover, a fixed *time frame* τ is assumed which is linearly ordered. Then, a *trace* γ over a state ontology Ont and time frame τ is a mapping $\gamma: \tau \rightarrow STATES(Ont)$, i.e., a sequence of states $\gamma_t (t \in \tau)$ in $STATES(Ont)$. The set of all traces over ontology Ont is denoted by $TRACES(Ont)$, i.e., $TRACES(Ont) = STATES(Ont)^\tau$.

Dynamic Properties

Patterns in world dynamics are described by dynamic properties. The set of *dynamic properties* $DYNPROP(Ont)$ is the set of temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont , a certain state of the world at time point t is denoted by $state(\gamma, t)$. These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate \models , comparable to the `Holds`-predicate in event calculus [17] or situation calculus [23]. Thus, $state(\gamma, t) \models p$ denotes that state property p holds in trace γ at time t . Here state properties are considered objects and denoted by term expressions in the TTL language. Likewise, $state(\gamma, t) \not\models p$ denotes that state property p does not hold in trace γ at time t . Based on these statements, dynamic properties can be formulated in a formal manner in a sorted predicate logic, using the usual logical connectives such as \neg , $\&$, \vee , \Rightarrow , and the quantifiers \forall , \exists (e.g., over traces, time and state properties). For example, consider the following dynamic property for a pattern concerning belief creation based on observation:

if at any point in time t_1 the agent A observes that it is wet outside,
then there exists a time point t_2 after t_1 such that at t_2 in the trace the agent A believes that it is wet outside

This property can be expressed as a dynamic property in TTL form (with free variable for trace γ) as follows:

$$\forall t:T [state(\gamma, t) \models observes(itswet) \Rightarrow \exists t' \geq t \ state(\gamma, t') \models belief(itswet)]$$

To model direct temporal dependencies between two state properties, the simpler *leads to* format is used. This is an executable format defined as follows. Let α and β be state properties of the form ‘conjunction of literals’ (where a literal is an atom or the negation of an atom), and e, f, g, h non-negative real numbers. In the *leads to* language $\alpha \xrightarrow{e, f, g, h} \beta$, means:

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 .

3. Adaptive Organization Model: Organizational Structure

This Section presents the structural model within the adaptive organization model; see Figure 2. Here, small ovals denote roles, bigger ovals denote groups, solid arrows denote transfers between roles, and dashed lines denote inter-group interaction. The model can be composed from two parts, of which the structure of one (the lower layer in Figure 2) is dependent upon the specific domain of application, and the other structure (top layer, depicted in gray), consisting of the *Change Group*, by which adaptation of the organization takes place, is generic for any type of application. All the agents participating in the organization are assumed to have an *Adaptor* role in this group. This role has the ability to monitor the existing agent-role allocations, and role, group and organization properties (hence, the group has a meta-view on the organization). In case it is observed that a property is not satisfied, an Adaptor role makes decisions about change, thereby possibly requesting other Adaptor roles to perform a role for which errors have been observed. The exact specification of the Adaptor role is addressed in Section 5.

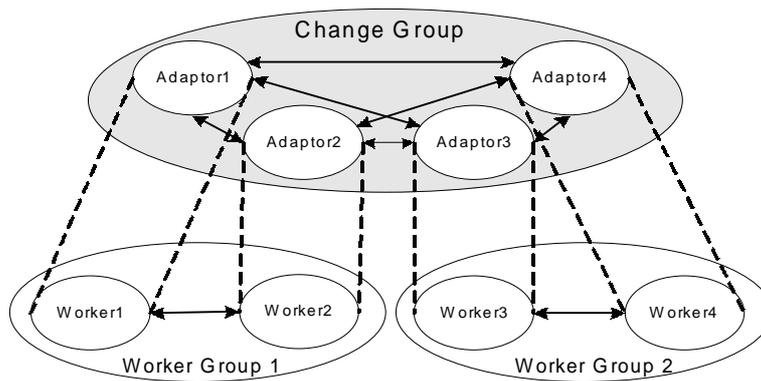


Figure 2. Adaptive Organizational Model

In the lower part of Figure 2, *Worker Groups* are shown of which each one addresses a particular part of the tasks needed to be performed within an organization. In incident management there would, for example, be a fire fighting group, medical group, and a police group. Notice that names in small ovals such as Worker1, ..., Worker4, used here denote the roles in these groups, not the agents allocated to these roles. The adaptativity in the model is in the flexibility of allocations of agents to these roles, not in the change of the roles themselves. Such an allocation change may involve, for example, that for an agent A that was allocated to role Worker1 within Worker Group 1, its allocation is changed to an allocation to role Worker4 in Working Group 2.

4. Adaptive Organization Model: Organizational Behavior

The behavioral model within the adaptive organization model takes the form of a hierarchy of dynamic properties at the different aggregation levels of the organization (see Figure 1 and 3). The relationships between the different levels within the hierarchy have been verified using the SMV model checker; cf. [21]. The highest organizational properties express what one wants a particular organization (as a whole) to establish, e.g., based on *performance indicators* of an organization. This section shows how such organizational properties are refined into more specific properties for particular aspects or parts of an organization (Section 4.1). These are further refined to the aggregation level of the particular groups within the organization (Section 4.2).

4.1. Organization-Level Properties

The highest level organization properties, expressing the well-being or satisfactory functioning of the organization, can take various forms; e.g., see the specific applications presented in Sections 6 and 8. In the adaptive organization model presented here the assumption is that robustness of the organization is a main organization property to be achieved. An organization is said to be robust in case all relevant aspects of the organization are well maintained, despite environmental or internal fluctuations. Therefore, to achieve the goals of the organization a number of aspects X_1, \dots, X_n can be distinguished that have to be maintained; e.g., [1] p. 58, 83. Examples of such aspects in the context of incident management are fire fighting, health care, and traffic care. Thus, a main property for the whole organization is that the organization functions well for the combination of these aspects X_1, \dots, X_n . The organization property OP expresses that at all points in time proper maintenance of the combination of aspects is satisfied:

$$OP = \forall t:TIME \text{ state}(\gamma, t, O) \models \text{satisfied}(\text{combination}(X_1, \dots, X_n)).$$

Here

$$\text{state}(\gamma, t, O) \models \text{satisfied}(\text{combination}(X_1, \dots, X_n))$$

denotes that within the state $\text{state}(\gamma, t, O)$ at time point t of the organization O in trace γ the state property $\text{satisfied}(\text{combination}(X_1, \dots, X_n))$ holds, with the infix predicate \models denoting the satisfaction relation between a state and a state property. Notice that such a state property can have different truth values at different points in time.

Other relevant organization properties (e.g., survival) are assumed to be entailed by this primary organization property OP. The organization property OP is refined using properties for different aspects of the organization: For any of the aspects X , the property $OAP(X)$, expresses that at all points in time aspect X is maintained in a satisfactory manner:

$$\text{OAP}(X) = \forall t:\text{TIME} \text{ state}(\gamma, t, O) \models \text{satisfied}(X).$$

These properties are assumed to relate to the overall organization property by

$$\forall X \text{ OAP}(X) \Leftrightarrow \text{OPI}$$

In other words, as long as all aspect properties are satisfied, the organization as a whole functions in a satisfactory manner. To this end it is assumed that:

$$\text{satisfied}(\text{combination}(X_1, \dots, X_n)) \Leftrightarrow \forall X \text{ satisfied}(X).$$

Then the bi-implication above can be rephrased as

$$\forall X \text{ state}(\gamma, t, O) \models \text{satisfied}(X) \Leftrightarrow \text{state}(\gamma, t, O) \models \forall X \text{ satisfied}(X)$$

which is one of the axioms for the predicate \models within the logical language TTL.

4.2. Group-Level Properties

In order to achieve the robustness of the organization, depending on circumstances, the organization needs to spend a certain effort on each of the distinguished aspects. As circumstances may change, it is here that adaptive control is possible and needed; such control can be insufficient, leading to one or more not well-maintained aspects, or sufficient, leading to all aspects well-maintained. In the organizational structure within the model, for each of the aspects a *Worker Group* is included to provide sufficient effort at each point in time as required to maintain this aspect, given the circumstances at that point in time.

GPI(X, G) : group provides required effort

For all time points t the effort provided by group G for aspect X is sufficient for the aspect.

$\forall t:\text{TIME}, E:\text{EFFORT}$

$$\begin{aligned} & [\text{state}(\gamma, t, G) \models \text{group_relates_to}(G, X) \wedge \\ & \text{provides_group_effort_for}(G, E, X) \\ & \Rightarrow \text{state}(\gamma, t, O) \models \text{satisfies_required_effort_for}(E, X)] \end{aligned}$$

Here the antecedent denotes that within the state $\text{state}(\gamma, t, G)$ at time point t of group G in trace γ the state property

$$\text{group_relates_to}(G, X) \wedge \text{provides_group_effort_for}(G, E, X)$$

holds, expressing that group G relates to aspect X and provides effort E . Moreover,

$$\text{state}(\gamma, t, O) \models \text{satisfies_required_effort_for}(E, X)$$

expresses that at time t in trace γ the effort E is the effort required to satisfy aspect X . It is assumed as part of the organization model that when the effort provided by group G relating to X satisfies the required effort for aspect X , then X is considered satisfied:

$$\begin{aligned} & \text{group_relates_to}(G, X) \wedge \\ & \text{provides_group_effort_for}(G, E, X) \wedge \\ & \text{satisfies_required_effort_for}(E, X) \rightarrow \text{satisfied}(X) \end{aligned}$$

Therefore, group effort property $GP1(G, X)$ relates to the corresponding aspect property $OAP(X)$ as follows:

$$GP1(X, G) \Rightarrow OAP(X).$$

The current roles within the group G are the ones that actually provide the effort for X . Each role has a particular effort it can provide, based on the role specification. In order to provide the required effort, sufficient effort of specific roles within a group is needed that together deliver enough combined effort; this is expressed in $GP2$. Here the sorts $ROLECOMBINATION$ and $EFFORTCOMBINATION$ denote sorts for combinations of (a finite number of) roles and of efforts, respectively. The latter sort is a subsort of $EFFORT$.

GP2(X, G) : roles provide required effort

For all time points t the total effort E_1, \dots, E_n provided by the roles R_1, \dots, R_n within group G addressing aspect X provides a combined effort satisfying the effort required for X .

$\forall t: \text{TIME}, RC: \text{ROLECOMBINATION}, EC: \text{EFFORTCOMBINATION}$:

$$\begin{aligned} & [\text{state}(\gamma, t, O) \models \text{group_relates_to}(G, X) \wedge \\ & \quad \text{group_has_roles}(G, RC) \wedge \\ & \quad \text{provides_effort_combination}(RC, EC) \\ & \Rightarrow \text{state}(\gamma, t, G) \models \text{provides_group_effort_for}(G, EC, X) \ \& \\ & \quad \text{state}(\gamma, t, O) \models \text{satisfies_required_effort_for}(EC, X)] \end{aligned}$$

This property relates to the previous one as follows:

$$GP2(X, G) \Rightarrow GP1(X, G)$$

4.3. Role-Level Properties

One role property is present on the lowest level not devoted to adaptation: Each of the active worker roles performs a certain amount of work. This is expressed as follows:

RP1(R) Worker Contribution

For all t the Worker role R delivers an effort E .

$\forall t: \text{TIME} \exists E: \text{EFFORT}$

$$\text{state}(\gamma, t, R) \models \text{provides_role_effort}(R, E)$$

5. Adaptive Organization Model: Organizational Adaptation

This Section presents the adaptation properties for the organization model. First, the adaptation properties expressing how the organization can achieve or maintain its goals under changing circumstances are introduced. Thereafter, the Adaptor role properties are presented which form the engines of the adaptation process.

5.1. Adaptation Properties

Within the organization the aspects distinguished are monitored all the time, in the sense that it is verified whether the provided effort is expected to stay sufficient for the required effort. To this end a signaling property is specified, based on desired effort. The property indicates those cases and time points that the effort observed for a certain aspect is close to becoming insufficient to satisfy the effort required for that aspect. The margin between the time point of signaling not satisfying the desired effort and the time point that the required effort is at risk of not being satisfied, is assumed large enough to have time to adapt. The adaptation mechanism within the organization has to guarantee that the effort will satisfy the desired effort again within a certain duration, without dissatisfying the required effort in the meantime; this to prevent property GP1 not being satisfied. This adaptation is expressed by the group adaptation property AP1.

AP1(X, G, d): Group adaptation for desired effort

For all time points t , in case the current effort E provided by group G for aspect X is not satisfying the desired effort, then at a later point in time t_2 (where $t_2 > t$ and $t_2 < t+d$) the organization has changed such that the effort provided satisfies the desired effort and in between will still satisfy the required effort.

$\forall t: \text{TIME}, G: \text{GROUP}, E1, E2: \text{EFFORT}$:

$$\begin{aligned} & [[\text{state}(\gamma, t, O) \models \text{group_relates_to}(G, X) \ \& \\ & \quad \text{state}(\gamma, t, G) \models \text{provides_group_effort_for}(G, E1, X) \ \& \\ & \quad \text{state}(\gamma, t, O) \models \text{satisfies_required_effort_for}(E1, X) \ \& \\ & \quad \text{state}(\gamma, t, O) \models \text{not_satisfies_desired_effort_for}(E1, X)] \\ & \Rightarrow \exists E2: \text{EFFORT}, t_2 > t \quad [t_2 < t+d \ \& \\ & \quad \text{state}(\gamma, t_2, G) \models \text{provides_group_effort_for}(G, E2, X) \ \& \\ & \quad \text{state}(\gamma, t_2, O) \models \text{satisfies_desired_effort_for}(E2, X) \ \& \\ & \quad \forall t_1 [t \leq t_1 \leq t_2 \Rightarrow \text{state}(\gamma, t_1, O) \models \text{satisfies_required_effort_for}(E2, X)]] \end{aligned}$$

This property relates to the previous properties as follows:

$$\text{AP1}(X, G, d) \Rightarrow \text{GP1}(X, G)$$

The group property for adaptation can be related to adaptation properties of individual roles taken as follows.

AP2(X, G, d) : Role adaptation for desired effort

For all time points t , in case the current effort combined from role efforts E_1, \dots, E_n provided by the roles R_1, \dots, R_n in G is not satisfying the desired effort, then at a later point in time t_2 (where $t_2 > t$ and $t_2 < t+d$) the organization has changed such that the effort combined from efforts provided by the roles within G satisfies the desired effort and in between will still satisfy the required effort.

$\forall t:\text{TIME}, \text{RC1}:\text{ROLECOMBINATION}, \text{EC1}:\text{EFFORTCOMBINATION}$
[[state(γ, t, G) |= group_relates_to(G, X) \wedge
 group_has_roles($G, \text{RC1}$) \wedge
 provides_effort_combination($\text{RC1}, \text{EC1}$) &
state(γ, t, O) |= satisfies_required_effort_for($\text{EC1}, X$) &
state(γ, t, O) |= not satisfies_desired_effort_for($\text{EC1}, X$)]
 \Rightarrow
 $\exists t_2 > t, \text{RC2}:\text{ROLECOMBINATION}, \text{EC2}:\text{EFFORTCOMBINATION}$
[$t_2 < t+d$ & state(γ, t_2, O) |= group_relates_to(G, X) \wedge
 group_has_roles($G, \text{RC2}$) \wedge
 provides_effort_combination($\text{RC2}, \text{EC2}$) &
state(γ, t_2, O) |= satisfies_desired_effort_for($\text{EC2}, X$) &
[$\forall t' \leq t_2$ [$t' > t \Rightarrow$
 $\exists \text{RC3}:\text{ROLECOMBINATION}, \text{EC3}:\text{EFFORTCOMBINATION}$
state(γ, t', O) |= group_relates_to(G, X) \wedge
 group_has_roles($G, \text{RC3}$) \wedge
 provides_effort_combination($\text{RC3}, \text{EC3}$) &
state(γ, t', O) |= satisfies_required_effort_for($\text{EC3}, X$)]]]

This property relates to the others as follows:

$$\text{AP2}(X, G, d) \Rightarrow \text{AP1}(X, G, d)$$

$$\text{AP2}(X, G, d) \Rightarrow \text{GP2}(X, G)$$

The next Section presents role properties which enable the adaptation.

5.2. Adaptor Role Properties

By an Adaptor role M , decisions about taking up or switching between Worker roles are made. As input information is used about the effort E currently being delivered by the different Worker groups G for a certain aspect X as expressed in

$$\text{provides_group_effort_for}(G, E, X).$$

In the model the decision mechanism is indicated by a relation expressing that an aspect has urgency:

$$\text{has_urgency}(X_1, E_1, \dots, X_n, E_n, X)$$

indicating that aspect X needs to be addressed in the context of efforts E_i , for aspects X_i . This relation can be specified as only deriving one aspect to be addressed (i.e., the most important aspect)

or multiple aspects (e.g., all aspects currently not being addressed properly). The relation takes into account which effort E suffices for the required effort to be delivered for aspect X:

satisfies_required_effort_for(E, X)

and which effort E suffices for the desired effort for aspect X:

satisfies_desired_effort(E, X).

A simple form of an urgency relation that is taken by default is:

has_urgency(X₁, E₁, ... X_n, E_n, X_i) \leftrightarrow not satisfies_desired_effort(E_i, X_i).

This expresses that all aspects for which the desired effort is not satisfied are urgent. Based on the input on urgency, the Adaptor role M generates in an intermediate state an indication of the aspect that needs to be addressed in the current situation.

RP1(M) Aspect Urgency

At any t, if at t Adaptor role M observes the group efforts for each of the aspects, and has a urgency relation that indicates X an urgent aspect at that time, then at some t' \geq t it will generate that X needs to be addressed.

$\forall t, X_1, \dots, X_n, E_1, \dots, E_n, X, M$

state(γ, t, G_1) |= provides_group_effort_for(G_1, E_1, X_1) & ... &

state(γ, t, G_n) |= provides_group_effort_for(G_n, E_n, X_n) &

state(γ, t, M) |= has_urgency($X_1, E_1, \dots, X_n, E_n, X$)

$\Rightarrow \exists t' \geq t$ state(γ, t', M) |= to_be_addressed(X)

Based on this, the appropriate role(s) R within the Worker Group(s) WG for the aspect(s) is/are determined, and that a candidate is to be found for the role:

RP2(M) Role Change Determination

At any t, if at t Adaptor role M generated that X is an urgent aspect, and role R in WG is responsible for this aspect, then at some t' \geq t it will generate that a candidate for role R in WG has to be found.

$\forall t, X, R, WG, M$ [state(γ, t, M) |= to_be_addressed(X) &

state(γ, t, M) |= role_responsible_for(R, WG, X)

$\Rightarrow \exists t' \geq t$ state(γ, t', M) |= to_be_found_candidate(M, ChangeGroup, R, WG)]

Finding the right Adaptor to be allocated to the role is the next step in the process. Assumed is that there exists shared knowledge of the capabilities of the Adaptors of the organization. An Adaptor may only have a partial view on this, and simply choose a local optimum. The decision mechanism states that the Adaptor will perform the role itself in case it has the capabilities or otherwise appoints another Adaptor which does have the capabilities and is preferred.

RP3(M) Candidate Selection: Own Selection

$\forall t:\text{TIME}, M, R:\text{ROLE}, WG:\text{GROUP}, C1, C2:\text{CAPABILITIES}$
 $[\text{state}(\gamma, t, M) \models \text{to_be_found_candidate}(M, \text{ChangeGroup}, R, WG) \ \&$
 $\text{state}(\gamma, t, M) \models \text{required_capabilities}(R, WG, C1) \ \&$
 $\text{state}(\gamma, t, M) \models \text{has_capabilities}(M, \text{ChangeGroup}, C2) \ \&$
 $\text{state}(\gamma, t, M) \models \text{capabilities_match}(C1, C2)$
 $\Rightarrow \exists t2 > t \ [\text{state}(\gamma, t2, M) \models \text{shared_allocation}(M, \text{ChangeGroup}, R, WG)]]$

RP4(M) Candidate Selection: Request

$\forall t:\text{TIME}, M1, M2, R:\text{ROLE}, WG:\text{GROUP}, C1, C2, C3:\text{CAPABILITIES}$
 $[\ \text{state}(\gamma, t, M1) \models \text{to_be_found_candidate}(M1, \text{ChangeGroup}, R, WG) \ \&$
 $\text{state}(\gamma, t, M1) \models \text{required_capabilities}(R, WG, C1) \ \&$
 $\text{state}(\gamma, t, M1) \models \text{has_capabilities}(M1, \text{ChangeGroup}, C2) \ \&$
 $\text{state}(\gamma, t, M1) \models \text{not capabilities_match}(C1, C2) \ \&$
 $\text{state}(\gamma, t, M1) \models \text{has_capabilities}(M2, \text{ChangeGroup}, C3) \ \&$
 $\text{state}(\gamma, t, M1) \models \text{preferred_candidate}(M2, \text{ChangeGroup}) \ \&$
 $\text{state}(\gamma, t, M1) \models \text{capabilities_match}(C1, C3) \]$
 $\Rightarrow \exists t2 > t \ [\text{state}(\gamma, t2, M1) \models \text{request_shared_allocation}(M2, \text{ChangeGroup}, R, WG)]]$

Furthermore, once a shared allocation is requested, the shared allocation will be in place:

RP5(M) Candidate Selection: Response

$\forall t:\text{TIME}, M, R:\text{ROLE}, WG:\text{GROUP}$
 $[\text{state}(\gamma, t, M1) \models \text{request_shared_allocation}(M1, \text{ChangeGroup}, R, WG)]$
 $\Rightarrow \exists t2 > t, M2 : \text{ADAPTOR} \ [\text{state}(\gamma, t2, M2) \models \text{shared_allocation}(M2, \text{ChangeGroup}, R, WG) \]]$

Finally, the following relationship is assumed to hold, given that roles $R1..Rn$ are devoted to Group G addressing aspect X :

$$RP1(M) \ \& \ RP2(M) \ \& \ RP3(M) \ \& \ RP4(M) \ \& \ RP5(M) \ \& \ RP1(R1) \ \& \ .. \ \& \ RP1(Rn) \ \Rightarrow \ AP2(X, G, d)$$

This logical relationship is an assumption imposed on the domain of application. It is assumed that by adding more roles to the group involved, the effort for an aspect X can be strengthened so that the required effort is kept satisfied, and the desired effort will become satisfied again within duration d . In many qualitative and quantitative domains this assumption is fulfilled, for example, in the two domains addressed in Sections 6 and 8 in this paper. In quantitative cases it gets the form of the assumption that by adding role efforts for X , the total sum of efforts can be increased until a certain value is reached, which has some relationship to the Archimedean principle in the real numbers:

$$\forall a, b > 0 \exists n \in \mathbb{N} \quad n * a > b$$

In qualitative cases the assumption can be related to an assumption on the availability of the right capabilities within the organization, as is shown in Section 6.

6 The Interlevel Relations and Their Verification

In this section an overview of the interlevel relations between the dynamic properties at different levels of aggregation is given, as occurred at various places in Sections 4 and 5. They can be summarized and classified as shown in Table 1.

Table 1 Overview of the logical interlevel relations

From aspect properties to organisation properties	$\forall X \text{ OAP}(X) \Rightarrow \text{OPI}$
From global group properties to aspect properties	$\text{GP1}(X, G) \Rightarrow \text{OAP}(X)$
From aggregated group properties to global group properties	$\text{GP2}(X, G) \Rightarrow \text{GP1}(X, G)$
From group adaptivity properties to global group properties	$\text{AP1}(X, G, d) \Rightarrow \text{GP1}(X, G)$
From role adaptivity properties to group adaptivity properties	$\text{AP2}(X, G, d) \Rightarrow \text{AP1}(X, G, d)$
From role adaptivity properties to aggregated group properties	$\text{AP2}(X, G, d) \Rightarrow \text{GP2}(X, G)$
From role properties to role adaptivity properties	$\text{RP1}(M) \& \dots \& \text{RP5}(M) \& \text{RP1}(R1) \& \dots \& \text{RP1}(Rn) \Rightarrow \text{AP2}(X, G, d)$

A graphical representation of the property hierarchy is shown in the AND/OR tree in Figure 3.

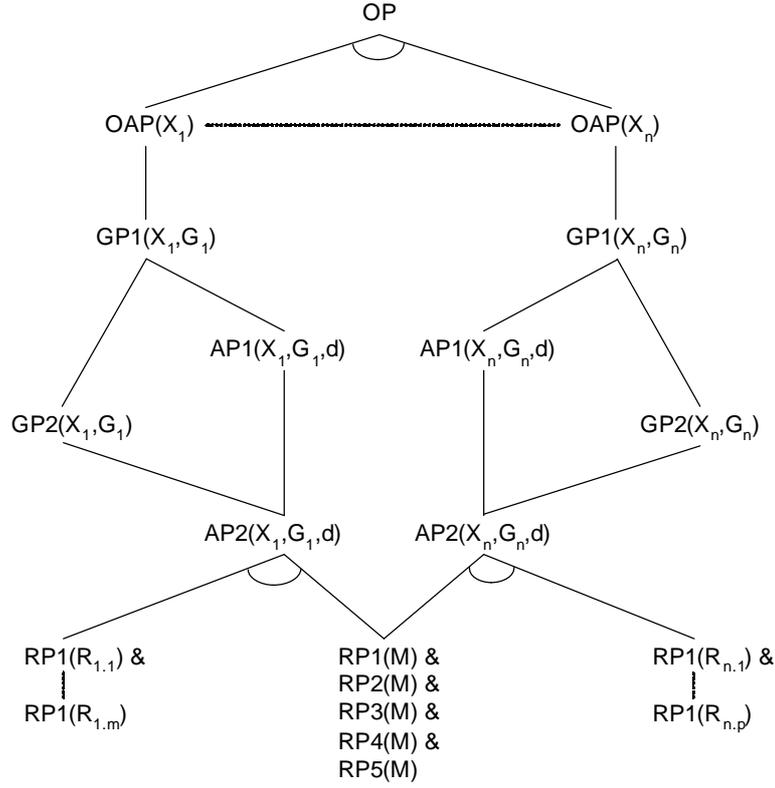


Figure 3. Property hierarchy presented in graphical form as an AND/OR tree

All relationships for the generic model as expressed within the tree have been verified using the SMV model checker under the assumptions as stated before. This has been done for every implication $A \Rightarrow B$ by rewriting property A in the transition system format that can be used by SMV as a description of a system. Moreover, B has been rewritten in CTL format, as required for the properties to be checked in SMV. As an example, this is shown for the relation between the role properties at the lowest level and the adaptivity property AP2. Here the antecedent A is a conjunction of a number of role properties for the adaptor role and for the worker roles. In SMV RP1 for the Member role has been specified as follows:

```

next(urgency) := case
    aspect = desired: 0;
    1: 1;
    esac;

```

This specifies the default urgency function, namely that in case the desired effort is not provided the aspect is considered to be urgent. This is expressed in the SMV transition system as follows: The next state for urgency is false (i.e., 0) in case the effort provided for the aspect is as desired, whereas

urgency for the aspect is true in any other case. Furthermore, RP2 concerns the search of a candidate, which is true in the next state in case the aspect is considered urgent, and otherwise false.

```
next(search_candidate) := case
    urgency : 1;
    1:0;
    esac;
```

The search of such a candidate is expressed in RP3 – RP5 and has been simplified for the sake of clarity. It is simply specified that a search for a candidate results in a candidate which has been found (which is also the case for the role properties specified).

```
next(found_candidate) := case
    search_candidate : 1;
    1:0;
    esac;
```

Furthermore, the role property for the role which now has an agent allocated it is specified as follows:

```
next(effort_added) := case
    found_candidate : 1;
    1:0;
    esac;
```

This specifies the effort added being by the newly found candidate. The added effort is set to true in case a candidate has been found, whereas it is false in case no candidate has been found. Final element of the transition system is to specify the effect of the added effort. Note that the addition of effort is not quantified, effort is simply added. This choice has been made for the sake of simplicity but has no consequences for the proof of the model. Adding effort has the following effect on the maintenance of the aspects within the organization:

```
next(aspect) := case
    aspect = failure & effort_added: required;
    aspect = required & effort_added: desired;
    aspect = desired & effort_added: desired;
    1 : aspect;
    esac;
```

This specifies that the next value for an aspect in case of effort being added is required in case the effort was previously failure; desired in case the previous aspect value was required, and desired remains desired upon additional effort. The consequent B is here property AP2. This is expressed in SMV's format based on CTL as follows.

```
AG (!(( aspect = desired) & (aspect = required))
    -> AF (aspect = desired) &
    A [(aspect = required|aspect=desired) U aspect=desired])
```

This expresses that when the considered aspect is required and not desired, then eventually the aspect will be desired again while in the meantime the value for the aspect will be either required or desired. This CTL property indeed holds for the specified model.

6. A Qualitative Application of the Organizational Model

This Section presents one of the two case studies undertaken to evaluate the applicability of the adaptive organization model presented in Sections 3, 4 and 5. This case study provides an analysis of the functioning of incident management organizations, in which adaptation of the organization by dynamic role reallocation is often observed. The analysis is based on a qualitative model for this domain made on the basis of extensive documentation of one of the disasters taking place in the Netherlands in the recent past [22]. First, domain specific variants of properties are introduced, after which simulation results are presented.

6.1. Domain Specific Properties

On the highest level of this qualitative incident management model, the property OP is defined as each aspect of the organization being satisfied. In the case of incident management, the aspects considered are fire fighting, health care, and traffic care:

OP(disaster)

For all time points t each aspect for incident management in the organization is satisfied.

$$\forall t:\text{TIME} \quad [\text{state}(\gamma, t, O) \models \text{satisfied}(\text{fire_fighting}) \wedge \text{satisfied}(\text{health_care}) \wedge \text{satisfied}(\text{traffic_care})]$$

On a lower level each individual aspect X is satisfied within the organization, for example, for the traffic care aspect of the organization:

OAP(traffic_care)

For all time points t aspect traffic care is satisfied in the organization.

$$\forall t:\text{TIME} \quad [\text{state}(\gamma, t, O) \models \text{satisfied}(\text{traffic_care})]$$

One group is responsible for the aspect traffic care: the police department. An instance of property GP1 requires a definition of satisfaction of the required effort. The effort of a group is defined as an abstract name; the required effort is always satisfied in case a route plan for ambulances is created which passes all wounded people within duration d from the start of the incident:

satisfies_required_effort_for(police_effort, traffic_care) \leftrightarrow

$$\forall t, t_0:\text{TIME} \quad [\text{present_time}(t) \wedge \text{memory}(t_0, \text{incident_started}) \wedge t_0 + d < t] \rightarrow$$

$$\exists t_2 \exists R:\text{ROUTE_PLAN} \quad [t_2 < t_0 + d \wedge$$

$$\text{memory}(t_2, \text{proposed_route_plan}(R)) \wedge$$

$$\forall W : \text{WOUNDED} \quad \text{memory}(t_2, \text{passes_wounded}(R, W))]$$

Here it is assumed that there are memory states. The desired effort is defined by:

$$\begin{aligned} &\text{satisfies_desired_effort_for}(\text{police_effort}, \text{traffic_care}) \leftrightarrow \\ &\forall t, t_0: \text{TIME} [\text{present_time}(t) \wedge \text{memory}(t_0, \text{incident_started}) \rightarrow t_0 + rd > t] \vee \\ &\exists t_2 \exists R: \text{ROUTE_PLAN} [\text{memory}(t_2, \text{proposed_route_plan}(R)) \wedge \\ &\quad \forall W : \text{WOUNDED} \text{memory}(t_2, \text{passes_wounded}(R, W))] \end{aligned}$$

Here $0 < r < 1$. In other words, the desired effort states that the correct route plan should be present before the required deadline already. The desired effort is always satisfied in the time interval from the start of the incident until rd after this start. It is not satisfied in the time interval starting rd after the start of the incident where no route plan was proposed yet. In view of property AP1 this means that after a correct route plan has not been generated by the police department within rd from the start of the incident, adaptation will be initiated at this time point, in order that the required effort will still be guaranteed before d after the start of the incident. As soon as indeed a route plan is proposed, the required effort remains satisfied and the desired effort becomes satisfied again.

Failure of the satisfaction of desired effort means that there is no role within the police department which has generated the correct route plan. By property AP2, this ultimately results in an adapted police department with roles which do perform the desired effort. To enable this change, the Adaptor within the Change Group uses the standard default definition of the urgency relation, in this case specifically for the police:

$$\begin{aligned} &\text{has_urgency}(\text{fire_fighting}, \text{fire_brigade_effort}, \text{health_care}, \text{health_effort}, \text{traffic_care}, \text{police_effort}, \text{traffic_care}) \leftrightarrow \\ &\text{not satisfies_desired_effort_for}(\text{police_effort}, \text{traffic_care}) \end{aligned}$$

expressing that the traffic care aspect has urgency in case no route plan is generated within the desired duration.

6.2. Simulation Results

In order to show how a multi-agent organization functions using the organizational model as presented above, simulation runs have been performed based on observations at the Volendam bar fire as described in [22]. In order to be able to simulate these adaptation processes, the lowest level properties (i.e. role properties) as presented in the Sections above have been translated into the executable subset of TTL called *leadsto* [5] which is used as an input for a simulation tool as described in [5]. The full *leadsto* specification is presented in Appendix B. Figure 4 shows the result of the simulation using this tool. In the Figure the left side shows the atoms that occur during the simulation run whereas the right side shows a timeline where a dark gray box indicates an atom being true whereas a light gray box indicates false.



Figure 4. Simulation results using the adaptive organization model

As can be seen in the trace, at time point 0 the bar fire starts: `incident_started(bar_fire_volendam)`. Three wounded people are present at the scene, at different locations, namely “zuideinde”, “pellersplein”, and “zeestraat”:

```
wounded_location(wounded_1, zuideinde)
wounded_location(wounded_2, pellersplein)
wounded_location(wounded_3, zeestraat)
```

Note that in reality much more wounded are present. Based on these circumstances an Adaptor role observing the current state of affairs at the scene derives that both the desired and required effort concerning traffic care are being delivered by the police, since they have until time point 4 to come up with a correct plan:

```
internal(adaptor_role_1|ChangeGroup)|satisfies_desired_effort(police_effort, traffic_care)
internal(adaptor_role_1|ChangeGroup)|satisfies_required_effort(police_effort, traffic_care)
```

Underlying this derivation is the following *leadsto* property:

```

 $\forall I1, I2: \text{INTEGER}, RO: \text{ROLE}, IN: \text{INCIDENT}$ 
[ internal(RO|ChangeGroup)|memory(time(I1), incident_started(IN))  $\wedge$ 
  present_time(I2)  $\wedge$  (I2  $\leq$  I+R*D) ]
 $\xrightarrow{0,0,0.1,0.1}$ 
internal(RO|ChangeGroup)|satisfies_desired_effort(police_effort, traffic_care)
```

The constants for R and D have been set to 0.5 and 6 respectively. Besides this property for desired effort a similar property exists for required effort. The only difference between these two properties

is the multiplication with the factor R in the condition, which is not performed in the required effort property. In order for this rule to fire, the Adaptor roles have a memory state. For example, at time point 1 the Adaptor role stores the previously observed start of the incident:

```
internal(adaptor_role_1|ChangeGroup)|memory(time(0), incident_started(bar_fire_volendam))
```

At time point 2 the route planner within the police group proposes a route plan which consists of merely one drive up route which is the location ‘zuideinde’:

```
output(route_planner|police)|proposed_route_plan(zuideinde)
```

This plan however only passes the wounded person at the location ‘zuideinde’ and not the other wounded:

```
passes_wounded(zuideinde, wounded_1)
```

Since the requirement is that the route plan should pass all wounded, the current proposed plan does not satisfy the requirements. However, due to the fact that the police has 4 time points before the desired effort needs to be provided, it takes until time point 4 before this failure is addressed (they could also have thought out a new, correct, route plan before the fourth time point). At that time point, an Adaptor role is unable to derive that the desired effort for traffic care is satisfied. Such a desired effort can be derived by means of two rules, the first rule being the *leadsto* property presented above (i.e., there is still remaining time to come up with a good solution) whereas the second rule specifies that a correct traffic plan has been generated which passes all wounded:

```

∀I1,I2:INTEGER, RO:ROLE, IN:INCIDENT, RP:ROUTE_PLAN
[[ internal(RO|ChangeGroup)|memory(time(I), incident_started(IN)) ∧
  internal(RO|ChangeGroup)|memory(time(I2), proposed_route_plan(RP)) ∧
  ∀W:WOUNDED [passes_wounded(RP, W)]]
→0,0.0.1,0.1
[ internal(RO|ChangeGroup)|satisfies_desired_effort(police_effort, traffic_care) ∧
  internal(RO|ChangeGroup)|satisfies_required_effort(police_effort, traffic_care) ] ]

```

Since not all wounded are being passed, this rule cannot fire either. As a result, the Adaptor role derives that the police effort does not satisfy the desired effort regarding traffic care by means of a *closed world assumption* (i.e., everything which cannot be derived is assumed false). The required effort is however still satisfied because this can still be derived by means of the first variant presented in this Section. The fact that the desired effort is not longer satisfied causes an urgency for the traffic care task:

```
internal(adaptor_role_1|ChangeGroup)|has_urgency(fire_fighting,..., traffic_care)
```

As a result, the role immediately derives that traffic care needs to be addressed:

```
internal(adaptor_role_1|ChangeGroup)|to_be_addressed(traffic_care)
```

Since the route planner is the role responsible within the police department for this task, a candidate must be found to take over the role:

```
internal(adaptor_role_1|ChangeGroup)|to_be_found_candidate(adaptor_role_1, ChangeGroup, route_planner, police)
```

The capabilities required for the role are navigation skills, a skill present at the particular Adaptor role, which therefore starts a shared allocation with the role itself (following the properties specified in Section 5.2) according the following *leadsto* property:

```
∀R1,R2:ROLE, G:GROUP, C:CAPABILITIES  
[ [ internal(R1|ChangeGroup)|to_be_found_candidate(R1, ChangeGroup, R2, G) ∧  
  internal(R1|ChangeGroup)|required_capabilities(R2, G, C) ∧  
  internal(R1|ChangeGroup)|has_capabilities(R1, ChangeGroup, C)  
  →0,0.0.1,0.1  
  internal(R1|ChangeGroup)|shared_allocation(R1, ChangeGroup, R2, G) ]
```

As a result, the shared allocation occurs in the trace:

```
shared_allocation(adaptor_role_1, ChangeGroup, route_planner, police)
```

Resulting from this new shared allocation, the role outputs a new route plan which described a route that circles the scene and therefore passes all the wounded:

```
output(route_planner|police)|proposed_route_plan(circle_scene)
```

Thereafter, the desired effort is satisfied again. Note that during the entire adaptation process the required effort was always fulfilled since the requirement stated by the guidelines says that a route plan that passes all wounded should be present within 6 time points, which is the case within the simulation. Would there however not have been any adaptation, the required effort would not have been satisfied after time point 6.

7. Quantitative Specialization of the Adaptive Organization Model

For domains that can be quantified, the adaptive organization model can be specialized. As a starting point each aspect X can be quantified by some value V (real or integer number), indicated by

$$\text{has_value}(X, V).$$

For each aspect a lower bound $V1$ and upper bound $V2$ is specified, indicated by

$$\text{lower_bound}(X, V1) \text{ and } \text{upper_bound}(X, V2).$$

The aspect is satisfied whenever its value is between these values:

$$\text{satisfied}(X) \leftrightarrow \forall V1, V2, V: \text{VALUE} \quad [\text{has_value}(X, V) \wedge \text{lower_bound}(X, V1) \wedge \text{upper_bound}(X, V2)] \rightarrow V1 \leq V \wedge V \leq V2]$$

Each of these aspects has a particular type of role attached to it, in which work is performed which contributes to that particular aspect. On the highest level, each aspect simply needs to be satisfied, expressed by the property OP in the following manner:

OPquantitative

For all time points t each aspect X has a value V which is below the upper bound $V2$ and above the lower bound $V1$.

$$\forall t: \text{TIME} \quad \text{state}(\gamma, t) \models \forall X: \text{ASPECT}, V1, V2, V: \text{VALUE} \\ [\text{has_value}(X, V) \wedge \text{upper_bound}(X, V2) \wedge \text{lower_bound}(X, V1)] \rightarrow V1 \leq V \wedge V \leq V2]$$

On the lower level of $OAP(X)$, the same is expressed per aspect X . The effort required to maintain each of the aspects throughout the organization can change over time. A value to be maintained might for example express that a certain percentage of environmental pressure needs to be dealt with, which means more effort in case of more environmental pressure. The group properties which express the effort being delivered by the groups addressing the aspects can again be reused from the model. However, the definitions for required effort and desired effort can be tailored towards the quantitative perspective. Here the assumption is made that V depends on E in a monotonic manner (when E is increasing, either V is increasing or decreasing). First of all, the required effort for each group is satisfied in case the current effort is between the minimum effort required (based either on the upper or lower bound of the aspect value) and the maximum effort (again from either the upper or lower bound of the aspect value).

satisfies_required_effort_for(E,X) ↔
 $\forall V1, V2: \text{VALUE}, E1, E2: \text{EFFORT}$
 $[[\text{upper_bound}(X, V1) \wedge \text{lower_bound}(X, V2) \wedge$
 $\text{required_effort_for_value}(E1, V1) \wedge \text{required_effort_for_value}(E2, V2) \wedge$
 $\text{is_max_of}(E_{\text{max}}, E1, E2) \wedge \text{is_min_of}(E_{\text{min}}, E1, E2)]] \rightarrow E_{\text{min}} \leq E2 \wedge E \leq E_{\text{max}}]$

Regarding the desired effort, the effort should be farther away from the bounds set. In other words, a parameter for a value ϵ with $0 < \epsilon < 0.5$ is added, as follows:

satisfies_desired_effort_for(E,X) ↔
 $\forall V1, V2: \text{VALUE}, E1, E2: \text{EFFORT}$
 $[[\text{upper_bound}(X, V1) \wedge \text{lower_bound}(X, V2) \wedge \text{required_effort_for_value}(E1, V1) \wedge$
 $\text{required_effort_for_value}(E2, V2) \wedge \text{is_max_of}(E_{\text{max}}, E1, E2) \wedge \text{is_min_of}(E_{\text{min}}, E1, E2)]]$
 $\rightarrow E_{\text{min}} + \epsilon (E_{\text{max}} - E_{\text{min}}) \leq E \wedge E \leq E_{\text{max}} - \epsilon (E_{\text{max}} - E_{\text{min}})]$

The decision properties for the Adaptor role again are reused from the generic properties as specified in Section 5, and also the default urgency relation:

has_urgency($X_1, E_1, \dots, X_n, E_n, X_i$) ↔ not satisfies_desired_effort_for(E,X)

In other words, an aspect is considered to be urgent in case the effort is outside the bounds of the desired effort.

8. A Quantitative Application of the Organizational Model

As a further evaluation of the applicability, the quantitative specialization of the organizational model has been applied in the domain of social insects and, more specifically, to analyze the functioning of honeybee colonies. In honeybee colonies several aspects need to be maintained in order for the population to be robust enough to be successful (in this case to survive), which include foraging, brood care, and undertaking. For the aspect brood care for example, the larvae need to have sufficient food, which requires more effort in case more larvae are present. If the larvae are insufficiently fed, the population size will eventually drop, endangering population survival. For each of these aspects, a specific Worker Group is present within the honeybee colony, i.e. all brood carer roles are part of the brood care group, etc. Furthermore, it is known from the biological domain that all honeybees within the organization have the capabilities to play each of the roles (see [6]).

Application of the quantitative specialization of the model is for the highest levels straightforward as these can simply be reused. On the lowest level however, the urgency of an aspect is defined by a threshold and trigger mechanism in each of the Adaptor roles, as described in biological literature; e.g., [6]. The mechanism informally works as follows. Each bee has a specific threshold for each of the aspects to be maintained in the organization. For the aspect the Adaptor is triggered most, relative to the threshold value it has, it will start a shared allocation with a role within a worker group

devoted to that aspect. Following what is known in biological literature [6], for this case such a mechanism can be specified as follows:

$$\begin{aligned} & \mathbf{has_urgency}(X_1, E_1, \dots, X_n, E_n, X_i) \leftrightarrow \\ & \forall V_1, \dots, V_i, \dots, V_n: \text{VALUE}, T_1, \dots, T_i, \dots, T_n: \text{VALUE}, \\ & \quad [\text{has_value}(X_1, V_1) \wedge \dots \wedge \text{has_value}(X_n, V_n) \wedge \\ & \quad \text{has_threshold}(X_1, T_1) \wedge \dots \wedge \text{has_threshold}(X_n, T_n) \wedge \\ & \quad \text{is_max_of}(V_i/T_i, V_1/T_1, \dots, V_n/T_n)] \end{aligned}$$

As can be seen, the effort provided is not used for the decision process within the Adaptor roles, only the triggers (the value of the aspect) and the thresholds are used to determine which aspect is most important. Note that the thresholds are defined for each individual Adaptor within the ChangeGroup. For simulations, see [14].

9. Discussion

This paper presented a organizational model for the analysis and design of multi-agent organizations that are able to adapt to unpredictable events. The organization model was specified distinguishing a number of aggregation levels. At the highest level the goal for the organization as a whole is expressed and this is refined to lower aggregation levels until role properties are reached that have to be fulfilled by agents allocated to the role. The model has been formally specified and verified using the model checker SMV. Besides a generic template, also specific variants have been presented, addressing both quantitative and qualitative models. Applicability of the model was evaluated positively, using it to analyze two cases: social insects and incident management. For both cases simulations have been performed, based on translating the lowest level properties to an executable format.

Research as described in [2], [18], [19], [20] has some similarity to the approach presented in this paper: when only looking at the agents, they adapt their behavior based on an event. The difference is however that in this paper, the adaptation of the behavior of the agents over time is described using the roles they play. As a result, it abstracts from the specifics of the agent that describe this change behavior, but simply poses a requirement upon the adaptation behavior of the agent in the form of a role.

In the domain of organizational modeling for multi-agent systems several frameworks have been extended with capabilities to model organizational change as well. [13] for example introduces an approach where a Change Manager is present, deciding what to change within the organization, and following a model from a well known social scientist. Such a model is however concerned with centrally directed organizational change whereas this paper concentrates on adaptation brought about by individuals within the organization detecting unsatisfactory occurrences in the organization. In an

extension of MOISE, namely MOISE+ [15] a central director for change is present as well; decision rules as detailed as presented in this paper are not presented.

In order to incorporate new behavior which is not pre-specified, the approach presented in this paper can be enriched with adaptation of role properties or addition of roles. Such adaptations could for example include a new specification of role behavior. This is however future work and is not addressed in this paper.

10. References

- [1] Ashby, W. R., Design for a Brain – The origin of adaptive behaviour, John Wiley and Sons, 1960.
- [2] Barber, S.K., Goel, A., and Martin, C.E., Dynamic adaptive autonomy in multi-agent systems, *Journal of Experimental and Theoretical Artificial Intelligence*, 12:129-147, 2000.
- [3] Boissier, O., Padget, J., Dignum, V., Lindemann, G., Matson, E., Ossowski, S., Sichmann, J., and Vázquez-Salceda, J. (eds.), Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, Proc. of the First International Workshop From Organizations to Organization-Oriented Programming, OOP' 05. Lecture Notes in AI, vol. 3913. Springer Verlag, 2006
- [4] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J., Specification and Verification of Dynamics in Cognitive Agent Models. In: Proceedings of the Sixth International Conference on Intelligent Agent Technology, IAT' 06. IEEE Computer Society Press, 2006, to appear.
- [5] Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., *LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn*. In: Proceedings of MATES' 05. LNAI 3550. Springer Verlag, 2005, pp. 165-178.
- [6] Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E., Self-Organization in Biological Systems, Princeton University Press, Princeton, USA, 2001.
- [7] Ferber, J. and Gutknecht, O., A meta-model for the analysis and design of organizations in multi-agent systems, *Proceedings of ICMAS'98*, IEEE Computer Society Press, pp. 128-135, 1998.
- [8] Ferber, J., Gutknecht, O., Jonker, C.M., Müller, J.P., and Treur, J., Organization Models and Behavioural Requirements Specification for Multi-Agent Systems, in Y. Demazeau, F. Garijo (Eds.), *Multi-Agent System Organizations. Proceedings of MAAMAW' 01/2001*
- [9] Fisher, M. (2005). Temporal Development Methods for Agent-Based Systems, *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 10, pp. 41-66.
- [10] Galton, A. (2003). Temporal Logic. *Stanford Encyclopedia of Philosophy*, URL: <http://plato.stanford.edu/entries/logic-temporal/#2>.
- [11] Galton, A. (2006). Operators vs Arguments: The Ins and Outs of Reification. *Synthese*, vol. 150, 2006, pp. 415-441.
- [12] Giorgini, P., Müller, J., Odell, J. (eds.), *Agent-Oriented Software Engineering IV*, LNCS, vol. 2935, Springer-Verlag, Berlin, 2004.
- [13] Hoogendoorn, M., Jonker, C.M., Schut, M.C., and Treur, J., Modeling Centralized Organisation of Organizational Change. *Journal of Computational and Mathematical Organisation Theory*. In press, 2006.

- [14] Hoogendoorn, M., Schut, M.C., and Treur, J., Modeling Decentralized Organization Change in Honeybee Colonies, Technical Report (see <http://www.cs.vu.nl/~mhooogen/report/tr-asr-06-01.pdf>), Vrije Universiteit Amsterdam, 2006.
- [15] Hübner, J.F. and Sichman, J.S., Using the MOISE+ model for a cooperative framework of MAS reorganization, Boletim Técnico BT/PCS/0314, Escola Politécnica da USP, São Paulo, 2003.
- [16] Jonker, C.M., and Treur, J., Relating Structure and Dynamics in an Organization Model, In J.S. Sichman, F. Bousquet, and P. Davidson (eds.), *Multi-Agent-Based Simulation II, Proc. of the 3rd Int. Workshop on Multi-Agent Based Simulation, MABS' 02* Lecture Notes in AI, vol. 2581, Springer Verlag, pp. 50-69, 2003.
- [17] Kowalski, R., and Sergot, M. (1986). *A Logic-Based Calculus of Events*. New Generation Computing, 4:67--95, 1986.
- [18] Krieger, M. J. B. and Billeter, J.-B. (2000) The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, vol. 30, pp. 65-84.
- [19] Kube, C.R., and Zhang, H., (1994). Stagnation recovery behaviours for collective robotics. In *Proceedings of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS' 94)*pp. 1883-1890.
- [20] Maes, P., Modeling adaptive autonomous agents, *Artificial Life*, 1:1-37, 1994.
- [21] McMillan, K., Symbolic Model Checking: An approach to the state explosion problem, Kluwer Academic Publishers, 1993.
- [22] Ministry of the Interior, Investigation Bar Fire New Years Night 2001 (in Dutch), SDU, The Hague, 2001
- [23] Reiter, R. (2001). *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

Appendix A Example SMV Specifications

From role properties to adaptivity property AP1

```
MODULE main

VAR
  aspect:{desired, required, failure};
  urgency: boolean;
  search_candidate: boolean;
  found_candidate: boolean;
  effort_added: boolean;

ASSIGN
  init(aspect) := required;
  next(urgency) := case
    aspect = desired: 0;
    1: 1;
    esac;
  next(search_candidate) := case
    urgency : 1;
    1:0;
    esac;
  next(found_candidate) := case
    search_candidate : 1;
    1:0;
    esac;
  next(effort_added) := case
    found_candidate : 1;
    1:0;
    esac;
  next(aspect) := case
    aspect = failure & effort_added: required;
    aspect = required & effort_added: desired;
    aspect = desired & effort_added: desired;
    1 : aspect;
    esac;

SPEC
  AG ((( aspect = desired) & (aspect = required))
  -> AF (aspect = desired) &
  A [(aspect = required|aspect=desired) U aspect=desired])
&
  AG (aspect=desired | aspect = required)
```

From adaptivity property AP1 to group property GP1

```
MODULE main

VAR
  aspect:{desired, required, failure};

ASSIGN
  init(aspect) := required;
  next(aspect) := case
    aspect = desired: desired;
    aspect = required: desired;
    1: aspect;
    esac;

SPEC
  AG (aspect=desired|aspect=required)
```

Appendix B LEADSTO Specification for Simulation

```

CONSTANT D=6
CONSTANT R=0.5
start_time(0)
end_time(10)
owa()

E—SORT LOCATION
├── zuideinde
├── zeesstraat
└── pellersplein

E—SORT WOUNDED
├── wounded_1
├── wounded_2
└── wounded_3

E—SORT ROUTE_PLAN
├── zuideinde
└── circle_scene

E—SORT INCIDENT
└── bar_fire_vlendam

E—SORT ROLE
├── adaptor_role_1
└── route_planner

E—SORT GROUP
└── police

E—SORT CAPABILITIES
└── navigation

E—SORT ASPECT
└── traffic_care

E—interval
├── R: range(0, 1)
└── F: incident_started(bar_fire_vlendam)

E—interval
├── R: range(2, 3)
└── F: output((route_planner police)) proposed_route_plan(zuideinde)

E—interval
├── R: range(0, 10)
└── F: role(adaptor_role_1)

E—interval
├── R: range(0, 10)
└── F: and
    ├── wounded_location(wounded_1, zuideinde)
    ├── wounded_location(wounded_2, pellersplein)
    └── wounded_location(wounded_3, zeesstraat)

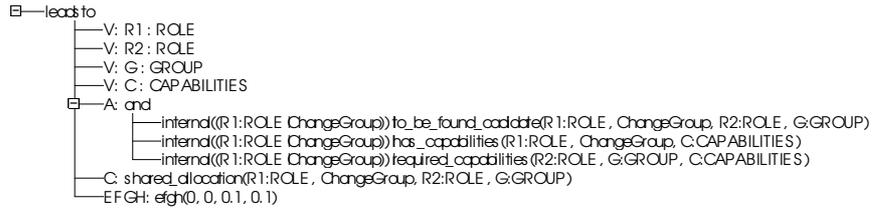
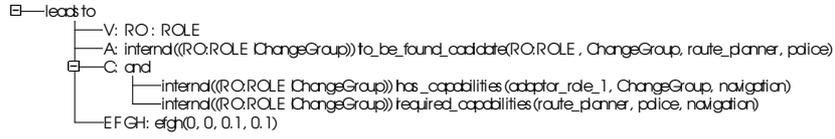
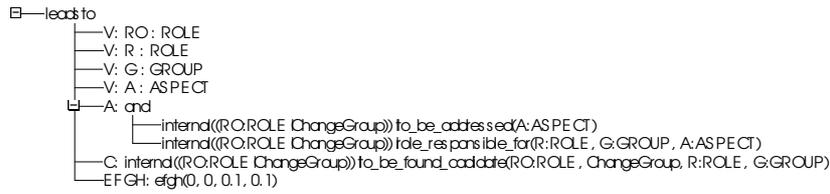
E—leads to
├── V: R2 : ROLE
├── V: RP : ROUTE_PLAN
├── V: I : INTEGER
└── A: and
    ├── output((route_planner police)) proposed_route_plan(RP:ROUTE_PLAN)
    ├── present_time(I:INTEGER)
    └── role(R2:ROLE)
    C: interval((R2:ROLE ChangeGroup) memory(time:I:INTEGER), proposed_route_plan(RP:ROUTE_PLAN))
    EFGH: efg(0, 0, 0.1, 10)

E—leads to
├── V: I : INCIDENT
├── V: T : INTEGER
├── V: R : ROLE
└── A: and
    ├── incident_started(I:INCIDENT)
    ├── role(R:ROLE)
    └── present_time(T:INTEGER)
    C: interval((R:ROLE ChangeGroup) memory(time:T:INTEGER), incident_started(I:INCIDENT))
    EFGH: efg(0, 0, 0.1, 10)

E—leads to
├── V: L : LOCATION
├── V: W : WOUNDED
└── A: and
    ├── output((route_planner police)) proposed_route_plan(L:LOCATION)
    ├── wounded_location(W:WOUNDED, L:LOCATION)
    └── C: passes_wounded(L:LOCATION, W:WOUNDED)
    EFGH: efg(0, 0, 0.1, 10)

```





OTHER : display(C, sort_atoms_time_global)

OTHER : display(C, no_show_atoms(present_time()))

OTHER : display(C, no_show_atoms(role))

OTHER : display(C, no_show_atoms(internal(_L)memory(time(5), _)))