Comput Math Organiz Theor (2007) 13:147–184
DOI 10.1007/s10588-006-9004-5

Modeling centralized organization of organizational change

Mark Hoogendoorn · Catholijn M. Jonker · Martijn C. Schut · Jan Treur

Published online: 1 September 2006

Abstract Organizations change with the dynamics of the world. To enable organizations to change, certain structures and capabilities are needed. As all processes, a change process has an organization of its own. In this paper it is shown how within a formal organization modeling approach also organizational change processes can be modeled. A generic organization model (covering both organization structure and behavior) for organizational change is presented and formally evaluated for a case study. This model takes into account different phases in a change process considered in Organization Theory literature, such as unfreezing, movement and refreezing. Moreover, at the level of individuals, the internal beliefs and their changes are incorporated in the model. In addition, an internal mental model for (reflective) reasoning about expected role behavior is included in the organization model.

Keywords Organizational change · Formal organizational modeling · Organizational simulation · Multi-agent organizations · Organization verification

M. Hoogendoorn (✉) · M. C. Schut · J. Treur
Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
e-mail: mhoogen@cs.vu.nl

M. C. Schut
e-mail: schut@cs.vu.nl

J. Treur
e-mail: treur@cs.vu.nl

C. M. Jonker
Radboud University Nijmegen, Nijmegen Institute for Cognition and Information, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
e-mail: C.Jonker@nici.ru.nl

1 Introduction

Within the literature on Organization Theory changing organizations play a dominant role (Robbins, 1998; Hyczynski and Buchanan, 2001; Jaffee, 2001). As change processes involve many factors ranging from making the employees aware of changes to come and taking away resistance to change to the design of efficient organizational structures. Changes can concern rather simple processes of slight changes in one or more role descriptions. They may affect only a part of the organization or practically the whole organization. Roles or big parts of the organization may be deleted, new ones created. The realization of the organization probably changes, e.g., agents fulfilling other roles than before, agents leaving the organization, agents joining the organization (Glaser and Morignot, 1997). A change may be initiated by the environment or by the organization itself. The organization of a change process may involve agents from outside the organization (e.g., consultation) or from inside. In this paper, the process of (business) organizational change is analyzed in more detail. Methods used in this analysis are those of formalization, simulation and verification. To organize change processes, a generic organization model for organizational change is introduced and formalized. This organization model incorporates both multi-agent co-operation aspects and individual cognitive aspects in the form of the internal mental states (e.g., beliefs) of those involved in the change.

A specific area in which organizational change is inherent, is in the organization that is needed to cope with a big upcoming event. Such an event can be a planned event in the area of sports or concerts, for example, but also an incident that can grow out to a disaster. The latter area is the focus of the project CIM (for Cybernetic Incident Management); cf. (Abbink et al., 2004; Hoogendoorn et al., 2004, 2005). A common characteristic for incidents and big planned events is that the organizational structures start almost at zero, i.e., no activity, and hence no organization, but (have to) grow out to a scale and form of organization that is able to address large and complex processes by multiple parties and multiple agents. To test ideas on organizational change modeling and to get more insight in cases with these characteristics, the organization of a big sports event has been chosen: the famous Dutch 11 cities ice skating tour (10.000s of people all performing 200 km of ice skating on one day, going from city to city). In this case study the usefulness of the developed organization model for organizational change is evaluated.

To model the organizational change process, the theory presented in Lewin (1951) and Robbins (1998) has been used as inspiration, and has been evaluated on its usefulness in an operational (modeling) sense. The three phases unfreezing, moving, refreezing distinguished have been incorporated in the generic model for organizational change developed. The case study shows that this theory indeed can be integrated in an organizational change modeling approach in a useful manner.

This paper is organized as follows: Section 2 gives an overview of organizational change literature, and introduces the stages that can be identified in an organizational change process. Section 3 introduces the approach which has been used to model the stages in organizational change in a formal way. The model itself is specified in Section 4 both from a structural as well as from a behavioral perspective. Section 5 presents a language used to specify an organizational change and Section 6 presents results of a case study which has been performed to show how the approach can be applied. In Section 7 formal verification is performed upon the simulation results to show that the simulated organization indeed satisfies the desired properties. Finally, Section 8 draws conclusions based on the results presented in this paper.

2 Organizational change literature

Organizational change is a well studied topic in recent literature on sociology, psychology, and economics. Change within organizations has become part of everyday life, some organizations are even continuously undergoing change. Changing an organization is not a simple process, often difficulties are encountered within such a change process. Research has shown that over 70 percent of the change programs in organizations do not achieve the intended goal (Hall, Rosenthal, and Wade, 1993; Bashein, Marcus, and Riley, 1994). Boonstra (2004) criticizes typical explanations given for these failures in that they pay insufficient attention to the complexity of the change process itself. Three types of organizational change are distinguished within the introduction of his book: First, *planned organizational change*, which addresses questions with respect to problems that require change in technical and instrumental aspects in which the problems and solutions are known. Secondly, *organizational development* which is said to be suitable when "the changes to be made are far-reaching, the problems not entirely unambiguous but still recognizable, and there is some idea as to the direction in which the solutions must be sought". Cummings and Worley (2001) define organizational development as "a system-wide process of applying behavioral science knowledge to the planned change and development of strategies, design components, and processes that enable organizations to be effective". The final type of organizational change distinguished is *transformational change*, in which the change processes include "renewal processes involving actors from various organizations". In Ackerman (1986) transformational change is said to be the emergence of a totally new state of being out of the remains of the old state.

Both in planned change and organizational development an approach is taken in which a move is performed from one stable state to another. The change processes involve the phases in which an organization is unfrozen, changed, and refrozen. These phases within the organizational change process originate from the ideas of Kurt Lewin (1951). He states that there are two opposing forces at work when changing an organization: forces that resist the change, and forces that drive towards the newly desired organization. Figure 1 presents the phases and forces within organizational change in a graphical manner (from Robbins, 1998). The unfreezing phase begins at the moment that change becomes necessary and consists of the process of changing the resisting and driving forces in such a way that change becomes possible (i.e., the driving forces outweigh the resisting forces). Both Schein (1993) and Hosking (1999) stress the importance of communication within this unfreezing phase to enable a successful change. According to Cummings (2004) organizational development has discovered a long list of causes for resistance to change, such as structural inertia, work habits, fear of the unknown, powerful interests, and members' security needs. Forces that drive an organization to change can be found in Jaffee (2001) and for example include change on the supply side, customer behavior, available technology (see e.g.

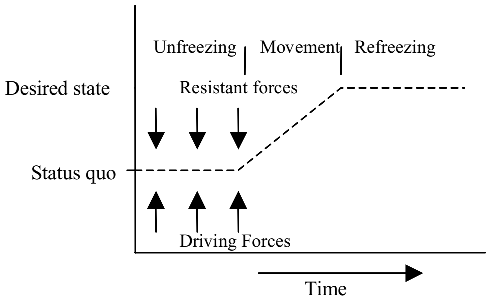

Fig. 1 Movement of an organization from a status quo to a desired state (Robbins, 1998)

Orlikowski and Hofman, 1997), etc. The actual change of the organization is contained in the movement phase in which the organization is moved from the current state to the desired stated. The refreezing phase involves freezing the newly formed organization so that there is no possibility to return to the former status quo or to continue changing in another unwanted direction. The whole re-organization process is completed when all phases have been completed. The unfreezing can be performed by increasing the driving forces and/or by decreasing the resisting forces. In their book, Cummings and Worley (2001) state that Lewin's model remains closely identified with the fields of planned change and organizational development.

Since the model of Lewin is a highly generic model, effort in organizational development research has gone into making it more concrete. Lippitt et al. for example arrange Lewin's model in seven steps: within the unfreezing phase they identify scouting, entry and diagnosis. The movement phase is split up into planning and action, and finally, stabilization and evaluation, and termination are placed within the refreezing phase.

Particularly of interest for this paper are further refinements regarding the actors within organizational change. Kotter (1998) has defined characteristics for change managers to prevent organizations from falling into pitfalls due to bad change management. These include having industrial and organizational knowledge, relations in the firm and industry, and reputation and track record. Power is an important aspect related to actors in organizational change processes as well, since the resisting and driving forces of the actors need to be changed to enable an organizational change. This particular research branch is called power dynamics. Research started in 1946 when Kurt Lewin introduced T-groups in a laboratory training setting and was mainly based on group-based approaches where people learn about group dynamics, leadership and interpersonal relationships. Bradshaw and Boonstra (2004) identify several different notions of power. Firstly, *manifest-personal power* which takes the viewpoint that a person can have power over other people and can make them do something they would not do otherwise. Research concerning this form of power research is said to have started with the work of Dahl (1975), Emerson (1962) and Wrong (1968). In *manifest-structural power*, power is no longer viewed from the personal perspective, but from a group perspective. Bacharach and Lawler (1980) is named as a reference for this notion of power. Negotiations are said to be an important part of the models regarding manifest structural power. *Latent-Cultural Power* sees organizing as "a process of the creation and reproduction of shared meanings that are largely latent or unconscious", they also refer to Alvesson (1993) for more details about the notion of latent-cultural power. Finally, *latent-personal power* which is said to be relatively new in organization theory. This type of power is said to differ from latent-cultural power is several different ways. First of all, power is said to be scattered throughout the organization, even individuals at the bottom of the organization can deploy their power. Secondly, power relations are assumed to become part of the psyche of the individual.

As the theory of Lewin is still considered being the underlying theory for organizational change research and considered valid, this paper tries to model the theory in a generic sense as a first step towards modeling and understanding complex organizational change processes. Further extensions might focus on the idea sketched above such as on more complex power relationships, the role of different characteristics for change managers, and the different ways to enable unfreezing an organization.

3 Modeling approach for organizations

Before being able to model the organizational change processes identified by Lewin, a methodology is required which enables modeling organizations in general. This Section presents such a methodology which allows modeling of organizations from two perspectives. First, the structural perspective, merely specifying the structural blueprint of an organization, and secondly, the behavioral perspective which specifies the behavior of an organization and the actors within such an organization.

3.1 The structural description of an organization

For the structural description of actual multi-agent organizations, the AGR (for agent/group/role) model approach has been adopted (Ferber and Gutknecht, 1998). In that approach, an organization is viewed as a framework for activity and interaction through the definition of groups, roles and their relationships. But, by avoiding an agent-oriented viewpoint, an organization is regarded as a structural relationship between a collection of agents. Thus, an organization can be described solely on the basis of its structure, i.e. by the way groups and roles are arranged to form a whole, without being concerned with the way agents actually behave, and multi-agent systems will be analyzed from the outside, as a set of interaction modes. The specific architecture of agents is purposely not addressed in the organizational model. The three primitive definitions are:

- The *agents*. The model places no constraints on the internal architecture of agents. An agent is only specified as an active communicating entity which plays roles within groups. This agent definition is intentionally general to allow agent designers to adopt the most accurate definition of agent-hood relative to their application.
- *Groups* are defined as atomic sets of agent aggregation. Each agent is part of one or more groups. In its most basic form, the group is only a way to tag a set of agents. An agent can be a member of *n* groups at the same time. A major point of these groups is that they can freely overlap.

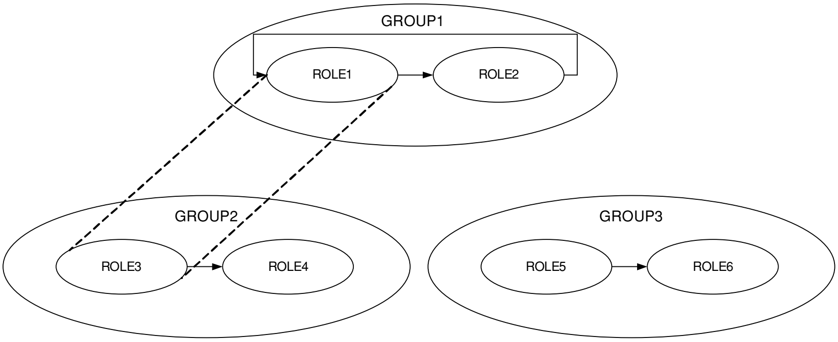

Fig. 2 Example organization modeled within AGR

- A *role* is an abstract representation of an agent function, service or identification within a group. Each agent can handle multiple roles, and each role handled by an agent is local to a group.

Figure 2 presents an example of an organization modeled in AGR. The large ovals denote groups whereas the smaller ovals denote the roles within the organizations. Furthermore, the solid arrows denote interactions between roles, and the dashed lines represent inter-group interactions. Agents realizing the roles are not depicted.

To enable simulation and reasoning about such an organizational model, the Structural Language SL is used, based on the set of *sorts* (a class or type of objects) that is shown in Table 1. These sorts enable talking about structural elements in the organization model. Additionally, Table 2 shows a set of *predicates* within SL that define relations between the introduced sorts.

3.2 The behavioral description of an organization

In this section a method to express dynamics within an organizational model is addressed. To formally specify dynamic properties at the different aggregation levels that are essential in an organization, an expressive language is needed. To this end the Temporal Trace Language is used as a tool; cf. (Jonker and Treur, 2002). For the properties occurring in the paper informal, semi-formal or formal representations are given. The formal representations are based on the Temporal Trace Language (TTL), which is briefly described as follows; for more formal details, see Appendix A.

Table 1 Sorts in *SL*

Sort	Description
ROLE	Sort for a role within an organization.
AGENT	Sort for an agent that can be allocated to a certain role.
GROUP	Sort for a group within an organization.
TRANSFER	Sort for a connection between two roles within one group.
GROUP_INTERACTION	Sort for a connection between two roles in a different group.

Table 2 Predicates defined in *SL* to describe the structure of an organization

Predicate	Description
exists_role: ROLE	A role exists within an organization.
allocated_to: AGENT × ROLE × GROUP	An agent is allocated to a role within a group.
exists_group: GROUP	A group exists within the organization.
role_belongs_to_group: ROLE × GROUP	A role belongs to a group.
intra_group_connection: ROLE × ROLE × GROUP × TRANSFER	A role is connected to another role (directed) within a certain group by means of a transfer connection. The source and destination roles are allowed to be equivalent.
inter_group_connection: ROLE × GROUP × ROLE × GROUP × GROUP_INTERACTION	A role within a group is connected to a role within another group by means of a group interaction connection.

A state ontology Ont is a specification (in order-sorted logic) of a vocabulary. A state for ontology Ont is defined as an indication of which state properties expressed in ontology Ont hold in the state and which do not hold. The set of all states is modeled by the sort STATE. A fixed *time frame* T is assumed which is linearly ordered. A *trace* or *trajectory* γ over a state ontology Ont and time frame T is an indication of which state occurs at which time point, for example if a discrete time frame based on natural numbers is taken, a trace is a sequence of states γ_t (t $\in$ T). The set of all traces over state ontology Ont is modeled by the sort TRACE. Depending on the application, the time frame T may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers), or any other form, as long as it has a linear ordering. A *dynamic property* over state ontology Ont is a temporal statement that can be formulated with respect to traces based on the state ontology. Such temporal statements can express, for example, a temporal relationship between the fact that in a given trace a certain state property holds at a certain time point and another state property holds at some other time point. For more formal details, see Appendix A.

The Temporal Trace language can be used to specify behavioral properties at different aggregation levels, according to the organizational structure. Within the AGR approach the aggregation levels are the level of the roles, the level of the groups and the level of the organization as a whole (see Fig. 3). The lower level properties can often be modeled in simpler formats than the higher level properties. In particular, it is often possible to model the properties at the leaves of the tree in the form of directly executable properties, i.e., by direct temporal dependencies between state properties in two successive states. To model direct temporal dependencies between two state properties, not the expressive language TTL, but the simpler *leads to* format is used. This is an executable format that can be used to obtain a specification of a simulation model in terms of local dynamic properties (the leaves of the tree in Fig. 3). The format is defined as follows. Let α and β be conjunctions of elementary state properties, and e, f, g, h non-negative real numbers. In the *leads to* language $\alpha \twoheadrightarrow_{e,f,g,h} \beta$, 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.

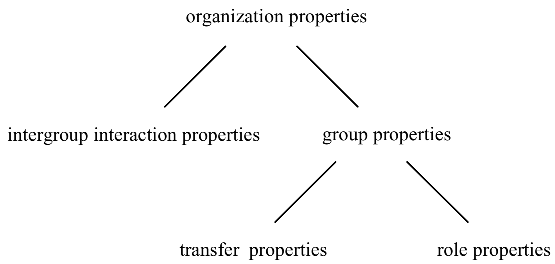

Fig. 3 Overview of interlevel relations between dynamic properties

For a precise definition of the *leads to* format in terms of the language TTL, see (Jonker and Treur, 2002). A specification of dynamic properties in *leads to* format has as advantages that it is executable and that simulation results can be depicted graphically.

Table 3 shows the predicates within the Behavioral Language BL which allows the specification of the behavioral part of the organization at different aggregation levels, using the TTL language as described above. The sort DYNPROP expresses an identifier of a dynamic property whereas DYNPROPEXP expresses the dynamic property itself in terms of TTL.

Based on the sort DYNPROPEXP it is possible to put more constraints on particular types of properties. The constraints for the different properties are defined below. The formal representations of these properties can be found in Appendix B.

Role dynamic properties. Role properties involve only one role, namely the role for which the property holds. Therefore, a role property should only contain elements that are part of the ontology of that role. The group is also part of the definition of the ontology since roles in different groups can have the same name and might have a different ontology. Role properties can be divided into different types which in turn can be defined more restricted than the general definition. An example of such a refinement is an executable role dynamic property.

Transfer dynamic properties. Transfer properties relate the output of a role to the input of a destination role, therefore the restriction on this dynamic property is that it should be expressed in terms of the output ontology of the source role combined with the input ontology of the destination role.

Table 3 Predicates defined in BL to define the dynamics within an organization

Predicate	Description
role_property: DYNPROP × ROLE × GROUP	A role within a group has a role property
transfer_property: DYNPROP × ROLE × ROLE × GROUP	Within a group, a transfer property with an identifier holds between two roles
group_property: DYNPROP × GROUP	A group has a certain group property
group_interaction_property: DYNPROP × ROLE × GROUP × ROLE × GROUP	An interaction property with an identifier holds between two roles in different groups
organization_property: DYNPROP	A certain or property holds for the organization
has_expression: DYNPROP × DYNPROPEXP	A specific dynamic property has an expression

Group dynamic properties. Group dynamic properties are dynamic properties expressed in terms of the state ontologies of (some of) the roles within the group. The most common type of group property relates an output state of a role within the group to an input state of another role within that group.

Intergroup interaction dynamic properties. Group interaction properties involve the input of a role within one group which is related to the output of a role within another group.

Organization dynamic properties. For the organization dynamic properties the same holds as for group properties: states of multiple roles (this time in different groups) can be involved; there is no further specific definition for this type of property.

4 Organizing organizational change

The term organizing organizational change makes it explicit that organizational change is a behavior process of that organization. Therefore, when formalizing organization dynamics, also the process of change must be formally specified as one of the possible ways of behavior of the organization. As all organizational behavior is described in terms of the behavior properties of the roles in that organization, also the whole process of organizational change is attributed to a set of roles in that organization. This section presents an organization model of organizational change that is based on the three stages of change introduced by Lewin.

4.1 Structure and informal behavior of the change organization

Modeling the forces indicated in Lewin's model entails attributing these forces to roles. Given an existing organization model that does not model organizational change, there are two basic choices that can be made: assigning these forces to roles already in the model, or extending the model with additional organizational elements. The first can be a part of the second approach by first extending the existing model with additional organizational elements, and then applying the first approach. Although the first approach can be a part of the second, when modeling an organization in which the realizing agents cannot reason about the change or even about the role that they are playing (e.g., when modeling an ant hill), only the first approach can be followed and the roles must be modeled as adaptive roles to ensure the possibility of change. In this article, the realizing agents can reason about roles and organizations. The second approach is chosen to most explicitly show the organizational change process. In both cases the behavioral specification of the organization elements needs extension, resulting in an organization model that incorporates organizing organizational change.

Consider, as an example, the organization as presented in Section 3.1, Fig. 2 which is also shown at the bottom part of Fig. 4(a). An organizational change might for example concern the removal of Group 3, which in turn could imply that one of the agents realizing the organization will be fired. It might further entail a re-allocation of agents over roles in groups. The organization in its state before change resists change (resisting forces outweigh the driving forces). To formally model this phenomenon,

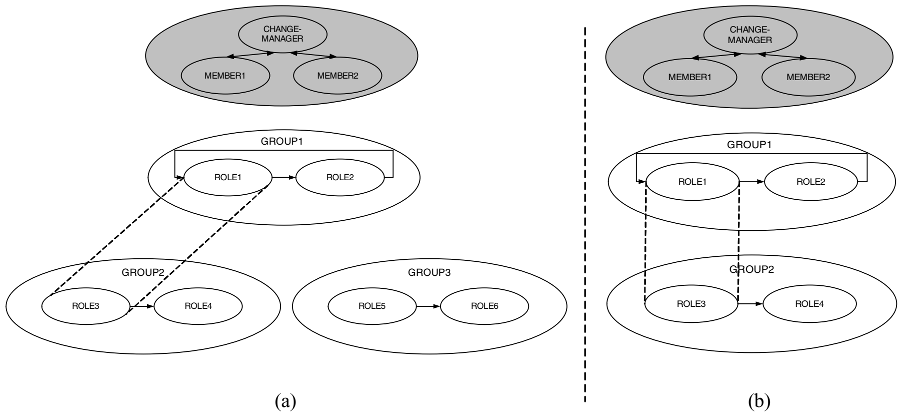

Fig. 4 (a) Organization before the change (b) Organization after the organizational change

the resisting and driving forces must be attributed to roles. Attributing them to the existing roles is counterintuitive, because different roles have been identified to specify different behaviors. The resisting and driving behaviors are of a different category. The way chosen in this article, is to recognize that all agents part of the realization of the organization have one thing in common: they are all members of the organization. Some members of the organization might be in favor of change, some against, and this might change over time. This is modeled by adding the role Member to the organization model, and attributing driving and resisting forces to that role. Given that the organization changes from one stable situation to a new stable situation, there is a need to model the focus existing in the organizational change. For this reason the role of Change Manager is added to the organizational model. The Change Manager is attributed with driving forces. This role can be realized by an agent from an external company, i.e. a consultant type of role, or by an agent from within the organization. In Fig. 4(a), the new roles are grouped together in an organizational element called the Change Group, the members are represented by Member One, Member Two, etc.

The Change Group is depicted in grey in Fig. 4(a) to indicate that in stable situations this group is inactive. The Change Manager can be of several different types, for example there can be a global Change Manager, that is allowed to change the entire organization. It is however also possible to have a local Change Manager that is only allowed to change a certain part within an organization and therefore can only communicate with a sub-group of the members within the Change Group. Because the Change Manager can be a representative of the company itself or of an external company there is no predefined shared allocation between this role and another. Every realizing agent of the organization is (next to the role it was already allocated to) also allocated to one instance of the Member role of the Change Group. The Change Group has a meta-view on the organization, and can, therefore, be seen as a meta-group. The start of an unfreezing phase (meaning a change is due) is characterized by a sudden activity of the Change Manager within the Change Group. The Change Manager might, for example, inform (all or some of) the instances of the Member role of the impending organizational change and the reasons for this change. Aside from the resulting reduction of resisting forces that this information might bring about, this interaction can also be used to model the preparation for the movement phase.

At the end of a well-performed unfreezing stage, maybe all Member role instances, but at least every Member role instance whose realizing agent is somehow involved in the change, now has beliefs about which role its realizing agent may have to play in the new organization. These beliefs include the expected role behavior. The end of the unfreezing phase may be characterized by the presence of these beliefs in the respective member role instances or communication of this presence to the Change Manager. Note that this does not say anything about all activities required to accomplish these shared beliefs.

The start of the movement phase, after a well-performed unfreezing phase, is characterized by the Change Manager informing all Members of when the actual change in organization is to take place. At the indicated moment, all Member roles are to consider in their beliefs the new organization form to be the current organization form. The movement phase is used to achieve (for example, by being informed) that all involved will get the appropriate beliefs on the new structure and their roles in this structure. As a result, the affected parts of the organization will start behaving according to the behavior specification of the new organization form. This process is modeled by means of the shared allocation of agents. Behavior that has become obsolete within the organization will disappear over time.

The start of the refreezing phase is characterized by regular functioning of the new organization form and a de-activation of the Change Group, see Fig. 4(b). The refreezing phase is complete when the behavior of the organization shows the routines that correspond to the expected behavior of the new, now current, organization.

Next to the structural properties of the organization model of organizational change, also the behavioral properties of the roles involved should be described to get a complete model. The next sections describe the behavioral properties of the main roles; the Change Manager and the Member.

4.2 Dynamic properties for the behavior of the change organization

The Change Manager is active in all stages of the organizational change. The properties in this section are described in a domain independent manner, more describing the global behavior than the actual behavior. Examples of more specific properties can be found in Section 6. First, properties regarding the unfreezing phase are presented, after which the behavior during the movement phase is described. Finally, the behavior during the refreezing phase is described.

4.2.1 Dynamic properties for the unfreezing phase

First of all, the following property states the global behavior during the unfreezing phase, namely that once there is an upcoming change, eventually enough key Members (fraction e) within the Change Group will be unfrozen which takes the form of a communication of acceptance of the new organization model.

GP1(ChangeGroup): Unfreezing organization

if at time t the Change Manager within the Change Group has access to a plan which specifies a condition C for a decision to reorganize based on a new organizational model OM
and condition C is met at time t,
and the Change Manager uses fraction e
then at a later point in time t2, at least fraction e of the key Members within the Change Group will have informed the Change Manager upon their acceptance of the new organization model OM.

This property can be fulfilled by means of several lower level properties. First of all, the Change Manager informs all Member within the Change Group that are involved in the change based on the new organizational model.

RP1(ChangeManager): Communicate change

if at time t the Change Manager within the Change Group has access to a plan which specifies a condition C for a decision to reorganize based on a new organizational model OM
and condition C is met at time t
and Member M1 is involved in the change to organizational model OM at time t
then at a later point in time t2 the Change Manager will inform Member M1 about the upcoming change to organizational model OM.

Furthermore, ideally once a Member is informed about such an upcoming change, the member will eventually communicate the acceptance of the new organizational model OM and thus will show to be unfrozen. Fraction e is given as a parameter for the property. Note that these properties describe a successful unfreezing phase where at least fraction e of the Members accepts the change.

GP2(ChangeGroup, e): Confirm change acceptance

for at least a fraction e of the key Members in the Change Group
if at time t a Member M1 is informed about the new organizational model OM
then at a later point in time t2 Member M1 will inform the Change Manager of its acceptance of the change to the new organizational model OM.

The property above is again specified in a general sense, as there might be a whole process involved in convincing the Member of the improvements that come with the new organizational model OM. Hence, there are two ways in which property GP2 can be fulfilled by the Members. First, the Member can immediately agree with the organizational model, and as a result be unfrozen at once.

RP2(Member): No resistance to change

if a Member M1 is informed about a new organizational model M at time t,
then at a later point in time t2 Member M1 will inform the ChangeManager upon its acceptance of the new organizational model OM
and there does not exist a time t' between t and t2 at which Member M1 has expressed resistance to the change to the organizational model OM.

Another option is that a Member expresses temporary resistance to the change.

GP3(ChangeGroup): Belief Change after Resistance

for all Members M1 in the Change Group
if Member M1 is informed about a new organizational model OM at time t,
then at a later point in time t2 Member M1 will inform the Change Manager of its resistance to the change to the new organizational model OM,
and at a time t3 later than t2 Member M1 will inform the Change Manager of its acceptance of the new organizational model OM.

Opposition to change can be split up into several lower level properties. First, the Member opposes the change.

RP3(Member): Oppose to change

if a Member is informed about a change to a new organizational model OM at time t,
then at a later point in time t2 Member M1 will inform the ChangeManager of its resistance to the change to the new organizational model OM.

In response the Change Manager puts forward a communication that hopefully will convince the Member that organizational model OM is an appropriate option for him. Note that these terms are kept abstract on purpose as there are many ways to convince such Members in organizational change literature, and depending on the particular case a choice can be made (see also Section 6).

RP4(ChangeManager): Convince member

if Member M1 informs the ChangeManager of its resistance to the change to the new organizational model OM at time t,
then at a later point in time t2 the ChangeManager will put forward additional arguments to Member M1 for the change to the organizational model OM.

Once this information is received by the Member it is assumed that he will be unfrozen.

RP5(Member): Member Convinced

if Member M1 receives additional arguments for organizational model OM at time t,
and Member M1 is convinced by the additional arguments for organizational model OM at time t,
then at a later point in time t2 Member M1 will inform the ChangeManager upon its acceptance of the organizational model OM

There is also the possibility that a Member does not get convinced, which means that the Member again communicates resistance. The Change Manager can put forward more arguments in response (or use another method from organizational change theory). The possibility exists that not enough key Members of the organization communicate the acceptance of the organizational model OM, resulting in an organization which is not unfrozen. To show the relation between the different properties for a *successful* unfreezing phase, Fig. 5 shows a property tree.

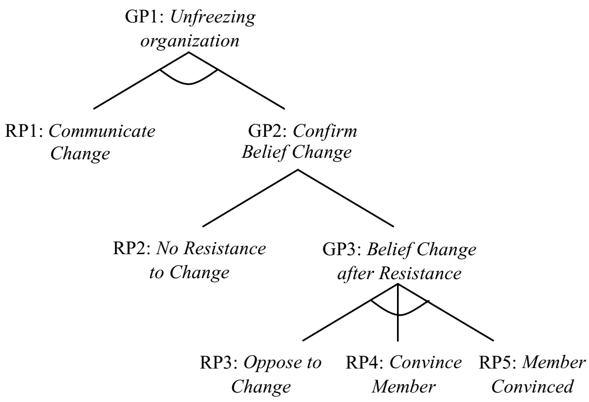

Fig. 5 Unfreezing property hierarchy specified by means of an AND/OR tree

The tree depends upon the number of Members involved in the change of the organization, this tree covers only one Member.

4.2.2 Dynamic properties for the movement phase

The movement phase is rather straightforward after the unfreezing phase, in case a fraction e of the key Members have communicated their acceptance of the organizational change towards the organizational model OM, and the condition for the change to occur holds, the roles within the groups of the organization will show the behavior as specified in the organizational model OM. Property OP1 specifies this movement and is referred to as an organizational property as it also includes roles outside of the Change Group.

OP1: Successful move

if at time t the Change Manager within the Change Group has access to a plan which specifies a condition C for a decision to reorganize based on a new organizational model OM
and condition C is met at time t,
and the Change Manager uses fraction e
and at least fraction e of the key Members involved in the change have informed the Change Manager of their acceptance of the change to organizational model OM at time t,
and organizational model OM specifies behavior B for a role R within group G at time t,
then at a later point in time t2 role R within group G behaves according to the behavior specification B.

Satisfaction of this high level property can be accomplished by means of a group property for the Change Group and group interaction properties between the Change Group and the other groups within the organization. First, the group property states that all Members involved in the change will receive an announcement of the organizational model being activated, as expressed in GP4 below.

GP4(ChangeGroup, e): Change activation

if at time t the Change Manager within the Change Group has access to a plan which specifies a condition C for a decision to reorganize based on a new organizational model OM
and condition C is met at time t,
and at least fraction e of the key Members involved in the change have informed the Change Manager of their acceptance of the organizational model OM at time t,
then at a later point in time t2 all Members involved in the change have received the announcement of organizational model OM being active.

This property is entailed by two lower level properties. First, the Change Manager announces the activation of the of the organizational model OM based on the conditions specified.

RP6(ChangeManager): Announce change

if at time t the Change Manager within the Change Group has access to a plan which specifies a condition C for a decision to reorganize based on a new organizational model OM
and condition C is met at time t,
and at least fraction e of the key Members involved in the change have informed the Change Manager of their acceptance of the change to organizational model OM at time t,
then at a later point in time t2 the Change Manager announces the new organizational model OM being active.

And furthermore, this information is received by the Members via transfer property TP1.

TP1(Change Manager, Member): Transfer announcement

if at time t the Change Manager announces organizational model OM being active
and at time t Member M1 is involved in the change to organizational model OM
then at a later point in time t2 Member M1 will receive this announcement on his input.

Finally, the group interaction properties state that after the announcement has been received by a Member role, the roles with which the Member that receives the announcement of activation will show the behavior as specified in the organizational model OM, expressed in GIP1.

GIP1(Member, ChangeGroup, R, G): New organization active

if at time t Member M1 is informed about a new organizational model OM being active,
and Member M1 has a shared allocation with a role R within group G at time t,
and role R has a behavior description B in organizational model OM at time t,
then at a later point in time t2 role R within group G behaves according to behavior B.

Figure 6 shows the property hierarchy for the movement phase.

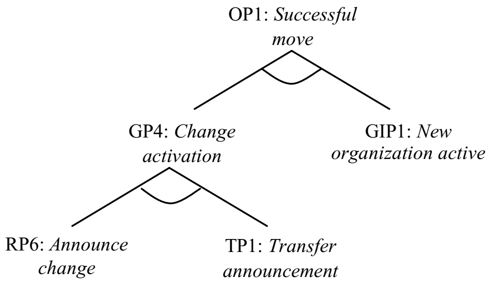

Fig. 6 Movement property hierarchy specified by means of an AND tree

4.2.3 Dynamic properties for the Refreezing Phase

The final step in the model of Lewin entails refreezing the organization. Within the model presented in this paper, this is expressed in the following way. There are two conditions to start the refreezing phase. First, the organizational model OM has been activated. Second, all roles are actually behaving according to the behavior specification. During the refreezing phase, key Members inform the Change Manager about what roles are showing the correct behavior. In case enough of these key Members (i.e. a fraction e1) communicate that a critical mass of roles (i.e. fraction e2) indeed show the correct behavior for a sufficient period of time p2, after a conditioning phase of length p1, the refreezing phase is said to be ended successfully. This property is expressed as OP2.

OP2: Successful refreezing

if before time t the ChangeManager has informed the Members that a new organizational model OM is active,
and at time t all the roles within organization are just behaving according to behavior specification B within the organizational model OM,
and the Change Manager uses a conditioning period p1, a critical period of length p2, and fractions e1 and e2,
then there exists a time point t2 (t2 > t + p2) such that at t2 the Change Manager is informed by at least fraction e1 key elements that behavior B is efficiently performed by at least fraction e2 of the roles within the organization over the last period p2

The property can be accomplished by means of a group interaction property and a group property. First, the group property states that from the time point the behavior is first shown by a role R within group G, there exists a time point at which the role R has shown the correct behavior for the minimum duration p, set by the Change Manager. The fraction e2 and periods p1 and p2 are specified as parameters.

GP5(G, e2, p1, p2): Show proper behavior

for at least a fraction e2 of the roles within group G,
if at time t a role R within group G just shows behavior B
then there exists a time point t2 (t2 > t + p1 + p2) such that at all time points between t2 and t2−p2 role R within group G shows behavior B

Some roles will immediately satisfy this property within the group, as specified by property RP7. This means that the behavior shown is always according to the specified behavior.

RP7(R, p1, p2): Immediately show behavior

if at time t a role R within group G just shows behavior B
then for all time points t2 such that $t < t2 \leq t + p1 + p2$ role R within group G shows behavior B

Of course it is also possible that the role R within group G falls back into its old habits, not complying to the behavior specification within the new organizational model. After correction however, the role shows the correct behavior again in case of successful refreezing. Such temporarily falling back into old habits is specified in property GP6.

GP6(G, p1, p2): Show behavior after correction

for at least a fraction e2 of the roles within group G
if at time t a role R within group G just shows behavior B
and there exists a time point $t1 > t$ and $t1 < t + p1$ at which role R within group G does not show behavior B
then there exists a time point t2 ($t2 \leq t1 + p1 + p2$) such that at all time points between t2 and $t2-p2$ role R within group G shows behavior B

Property GP6 is entailed by three lower level properties. First, RP8 expresses the improper behavior of the role R:

RP8(R, p1, p2): Improper behavior

if at time t a role R within group G just shows behavior B
then there exists a time point $t2 < t + p1$ at which role R within group G does not show behavior B

To correct this improper behavior, another role within the same group can correct role R by reminding the role of the proper behavior B:

RP9(R): Correct improper behavior

if at time t a role R within group G does not show the required behavior B
then at a later point in time another role R2 within group G will remind role R within group G of the proper behavior.

In a successful refreezing phase the correction indeed works, and role R returns to the correct behavior again (RP10). In case the role R is not properly refrozen such a correction might not work and therefore role R will continue to show the unwanted behavior.

RP10(R): Behave correct again

if at a time point t role R within group G is reminded by role R2 within group G of the proper behavior he should show
then for all later points in time $t2 > t$ role R within group G shows behavior B as long as no new reorganization has been announced

Finally, GIP2 specifies that after having shown the correct behavior for a period longer than length p, the Member within the Change Group communicates this to the Change Manager.

GIP2(R, G, Member, ChangeGroup, p1, p2): Communicate correct behavior

if between time point t and t2 (where t2 > t + p2) role R within group G shows the behavior according to B
and role R within group G has a shared allocation with Member M1
then at time t2 + 1 Member M1 informs the Change Manager within the Change Group that behavior B is efficiently performed by role R within group G over the last period p2.

The property hierarchy for the refreezing phase is shown in Fig. 7.

5 Change language

Since communication between the Change Manager and the Members within the Change Group also concerns changes to the current organization (i.e., a new organizational model), this section describes functions for describing such changes to be made. The sorts that have been used for this language are shown in Table 4, and are basically the sorts that make it possible to use the structural and behavioral languages SL and BL introduced in Section 3. Moreover, the sort ACTION models actions that can be performed. If a conjunction of elements of ORG_ELEMENT is deleted or added, then all conjuncts are removed from or added to the model.

The functions and predicates that can be used to describe organizational change are shown in Table 5. The modify function is basically a combination of the delete and add function, but because it is most likely that change includes modification of certain elements it is more intuitive to include it as a function. The add function possibly takes a conjunction of ORG_ELEMENT as an input (denoted as ORG_PART), this however is impossible for the delete because this would not result in a unique

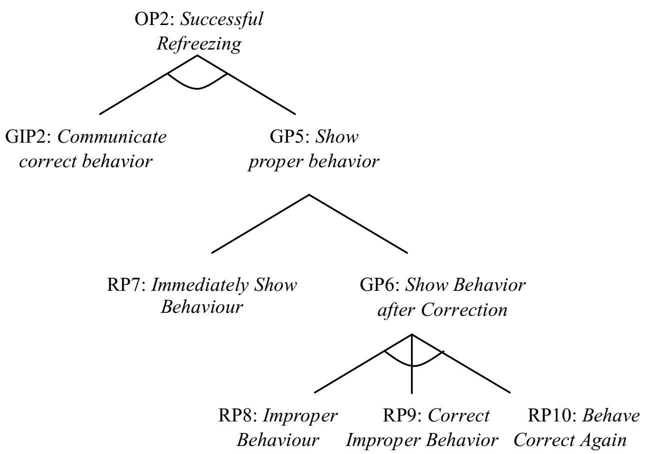

Fig. 7 Refreezing property hierarchy specified by means of an AND/OR tree

Table 4 Sorts used for the functions to describe organizational change

Sort	Description
ORG_BEHAVIOR_ELEMENT	Defined by the behavioral language *BL*
ORG_STRUCTURE_ELEMENT	Defined by the structural language *SL*
ORG_ELEMENT	Union of the sorts ORG_BEHAVIOR_ELEMENT or and ORG_STRUCTURE _ELEMENT
ORG_PART	Conjunctions of elements from ORG_ELEMENT
ACTION	Sort for actions

Table 5 Functions and predicates used to describe organizational change.

Function or Predicate	Description
add: ORG_PART → ACTION	Add takes an ORG_PART and creates the action to add that part
delete: ORG_ELEMENT → ACTION	Delete takes an ORG_ELEMENT and creates the action to delete it
modify: ORG_PART × ORG_PART → ACTION	The first element models the current organization, the second specifies the modifications that need to be done. An action is constructed by means of this
to_be_performed: ACTION	Predicate that a certain action is to be performed. This can be add, delete or modify

system configuration. The performance of the actions is done internally within the role, resulting in a communication that the structure is in place.

An organization model for organizational change as described informally in Sections 2 and 3, involves a number of issues:

- changing internal (belief) states of all those involved in the changing organization
- changing organization structure
- taking up new roles by agents
- internal state properties of the agents involved incorporate beliefs on organization structure as well as beliefs on dynamic properties characterizing role behavior
- internal state properties (beliefs) play a role as part of the dynamic properties characterizing role behavior

A language to express dynamic properties of a changing organization has to be a rich language able to express all these aspects in combination. Such a language is defined in Appendix C as an extension of TTL (Jonker and Treur, 2002) called meta-TTL. Note that in this language not only dynamic properties are defined on top of state properties, but also state properties (in particular beliefs) are defined on top of dynamic properties. This makes it possible to express a dynamic property built using a belief state property which itself refers to a dynamic property, and so on. So on the top level this is a dynamic property built on state properties (the beliefs), which themselves refer to state properties concerning the organization structure and to a dynamic (leads to) property again. An example property is the following, describing that a role performs the behavior it believes that is expected from the role:

If at time t
  a role believes that
    this role has as part of its behavior description that
        upon input v the output action w is done,
 and
  v occurs as input,
then
  at a next point in time this role will provide output w.

More formal details can be found in Appendix C.

6 Simulation of the case study: the eleven cities tour example

This Section presents a case study to illustrate the usage of the organizational change model as presented in the previous Sections. First, the organization under investigation is explained and thereafter simulation results are presented as well as domain specific properties that have been used to enable a simulation.

6.1 Case study description

The organization model of organizational change has been applied to the organization that is responsible for the famous Frisian skating tour called the Eleven Cities Tour. The association is called “De Friesche Elf Steden” in Dutch.

Although the association has fixed parts in the organization, it also has an annual dynamics in its structure. The association has a board consisting of 3 members namely the Chairperson, the Treasurer, and the Secretary. The Board has two responsibilities: running the association smoothly at all times and organizing the tour. Most of the year only the board is active, but there is also a permanent group which contains all members of the eleven cities tour society, which includes the people within the Board as well. This off-season organization is shown in Fig. 8. Once a year, at the beginning of winter, the organization changes its structure by formation of Region groups and the election of Region Heads for the coming winter season to enable monitoring of the ice conditions. This change process takes place within the Eleven Cities Tour Society group where the Member with a shared allocation to the Chairperson in the board is in charge of the change process. In the real organization, 21 Region groups are formed, for the case study however only the groups for the cities of Woudsend and Sneek are assumed to be created. The Region groups consist of more roles than the Region Head role (Monitor roles), however these roles have been left out of the case study for the sake of clarity. The election of the Region Heads is always a difficult part of the organizational change, as many people resist to the election of certain people because they think these people are not suitable for the job, or because they prefer another candidate, but in this case study we only consider suitability. Once the Regions have been formed and the Region Heads have been appointed, they start their work of monitoring the ice condition along the route. After certain conditions are met, such as a certain period of frost, another change occurs within the organization: A group called Region Representatives is formed which consists of representatives of all Region groups and representatives of the Board. This group discusses the conditions

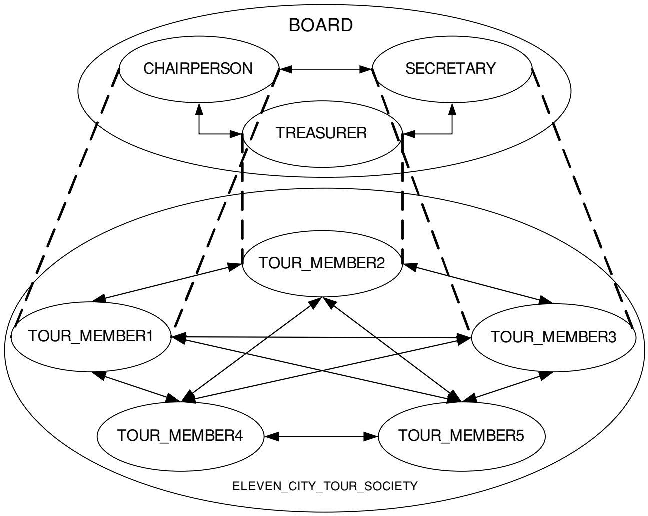

Fig. 8 Off-season eleven cities tour organization

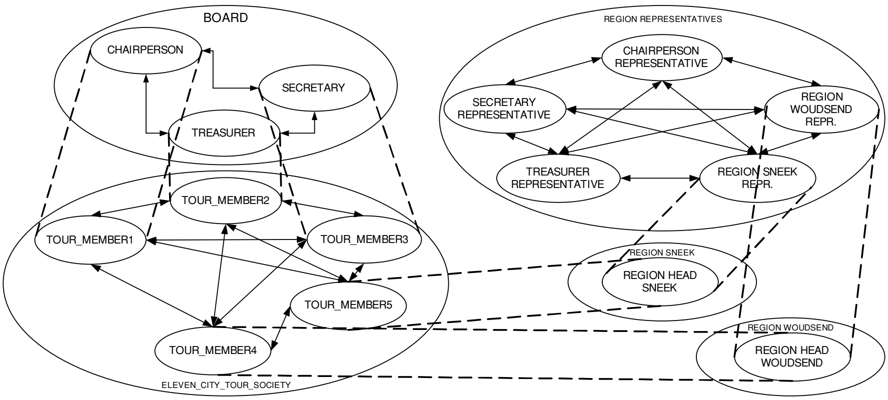

Fig. 9 Eleven cities tour organization after formation of the Regions and Region Representatives groups

along the entire trajectory of the tour. If the conditions are good, this group organizes the Tour. The organization after formation of the Regions and Region Representatives group is shown in Fig. 9. Note that the shared allocations between the members of the Board and the representatives of these in the Region Representatives group have been omitted to keep the Figure clear. To ensure that indeed all roles within the Region Representatives group show the desired behavior, the Chairperson Representative monitors whether the representatives of each of the Regions are indeed behaving according to the specification and do not fall back into their prior behavior.

Finally, at the end of the winter, the Chair of the Meeting of Region Heads thanks all participants and deactivates all roles in that group as well as all Region Head role instances. At this point in time the agents are de-allocated from their roles, and the roles immediately cease to exist. The involved agents only remain allocated to the continuous roles/role instances in the Board and Eleven City Tour Society group.

6.2 Simulation results

Based on the generic properties as specified in Section 4, a domain specific simulation model for the eleven cities tour has been created. All of the properties that underline the basis of this model have been specified in the *leadsto* format as introduced in Section 3. Since this format is executable, simulations can be performed using the *leadsto* software tool (Bosse et al., 2005). This Section presents a selection of the simulation results, and gives example of the domain specific properties that have been used for the simulation. Furthermore, several events are put into the model to see how well the organization changes in case this is required. The results have been ordered based on the different phases in organizational change distinguished by Lewin.

6.2.1 Initial organization

The initial organizational setup for the simulation is shown in Fig. 10. On the left hand side of the Figure statements are shown about the organization whereas the right hand side presents a timeline where a black box indicates that the statement is true at that particular time point. The Eleven Cities Tour Society group is called Change Group within the initial organization since this group's only function is organizational change, hence it is considered a Change Group. Note that the Figure only presents part of the initial organization: only a selection of the intra and inter group interactions, and only the beliefs of the Change Manager are shown. The Figure for example shows the presence of the role Chairperson:

internal(GlobalChangeManager|ChangeGroup)|belief(exists_role(Chairperson))

Fig. 10 Initial setup of the organization for the simulation

Furthermore, the existence of the group Board is shown:

internal(GlobalChangeManager|ChangeGroup)|belief(exists_group(Board))

The role Chairperson is specified to be part of the Board group:

internal(GlobalChangeManager|ChangeGroup)|belief(role_belongs_to_group(Chairperson, Board))

Intra group interaction is part of the Board group as well, the Secretary can for example communicate with the Chairperson:

internal(GlobalChangeManager|ChangeGroup)|belief(intra_group_connection(Secretary, Chairperson, Board, t1))

And finally, inter group connections are part of the beliefs of the Change Manager. The inter group connection shown is the one between the Chairperson in the Board and the Change Manager within the Change Group:

internal(GlobalChangeManager|ChangeGroup)|belief(inter_group_connection(Chairperson, Board, GlobalChangeManager, ChangeGroup, gi1))

This inter group connection is based on shared allocation, which means that the agent playing the role of Chairperson within the Board also plays the role of Change Manager within the Change Group. Within the Figure, the Change Group consists of five Member roles and one Change Manager. The additional Members in the Change Group are played by agents that are not yet part of the organization, but can be used by the Change Manager for the fulfillment of new roles to be played.

6.2.2 Unfreezing phase for region formation

After the initial setup of the organization, an event is put into the simulation which requires an organizational change, namely the onset of winter meaning that it is time to form the regions within the organization. The first phase within this change process is unfreezing. The occurrences during this phase are shown in Fig. 11. The event requiring change is the Chairperson within the board observing that it is time to form the regions:

input(Chairperson|Board)|time_to_form_regions

An inter-group interaction property in the form of a leadsto rule now fires which specifies that if the Chairperson within the Board observes it is time to form the regions, the Change Manager activates the Change Group and announces the organizational model for the region structure:

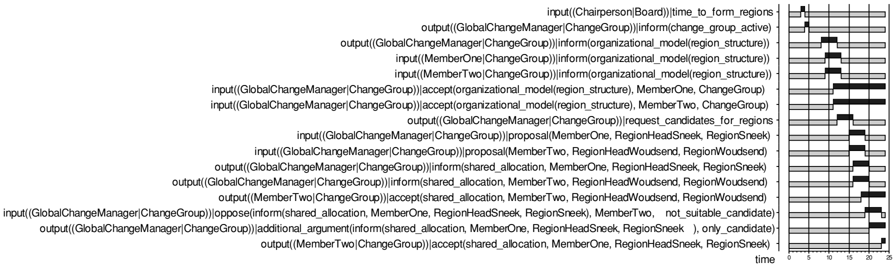

Fig. 11 First unfreezing phase during simulation

GIP_specific(Chairperson, Board, ChangeManager, ChangeGroup): Form Regions when winter

if at time t the Chairperson within the Board observes that it is time to form the Region groups
then at time t + 1 the Global Change Manager within the Change Group informs the Members within the Change Group that the group is now active
and at time t + 1 the Global Change Manager within the Change Group announces the new organizational model regarding the Regions.

The results of this rule show in the trace by the following elements:

output(GlobalChangeManager|ChangeGroup)|inform(change_group_active)
output(GlobalChangeManager|ChangeGroup)|inform(organizational_model(region_structure))

Only a reference i.e. the statement region_structure, to the whole specification of this organizational structure is presented in the Figure for the sake of clarity. None of the Members oppose the change as all are skating fanatics that long for a tour and all are convinced that winter has started. For them the onset of winter naturally means the formation of regions, so all communicate the acceptance of the organizational model, for example Member One :

input(GlobalChangeManager|ChangeGroup)|accept(organizational_model(region_structure), MemberOne, ChangeGroup)

The unfreezing for this particular organizational structure is therefore accomplished, following RP2 as described in Section 4. Another element of the change is to allocate the appropriate agents to the specific roles within the new organizational model: appoint the Region Heads. For this a more complicated unfreezing phase is performed. First, the Change Manager within the Change Group requests candidates for the newly formed roles:

output(GlobalChangeManager|ChangeGroup)|request_candidates_for_regions

The Members within the ChangeGroup receive the request and propose candidates for the positions, based upon their availability during the winter:

input(GlobalChangeManager|ChangeGroup)|proposal (MemberOne, RegionHeadSneek, RegionSneek) input(GlobalChangeManager|ChangeGroup)|proposal(MemberTwo, RegionHeadWoudsend, RegionWoudsend)

After receiving the proposals, the Change Manager decides upon an optimal allocation. Since there are two roles that need to be fulfilled and there is one proposal per role, these allocations are chosen and communicated:

output(GlobalChangeManager|ChangeGroup)|inform(shared_allocation, MemberOne, Region HeadSneek, RegionSneek)
output(GlobalChangeManager|ChangeGroup)|inform(shared_allocation, MemberTwo, Region HeadWoudsend, RegionWoudsend)

Member Two is however not convinced about the suitability of Member One for the role of Region Head Sneek and opposes to the organizational model following a domain specific instantiation of RP3 in Section 4:

RP3_specific(MemberTwo): Oppose to Change

if	at time t Member Two is informed about a change to an organizational model M in which a Member M1 has a shared allocation to a Role R1
and	Member Two observes M1 is unsuitable for the Role R1 at time t
then	at time t + 1 Member Two opposes to the change to the organizational model stating that M1 is not suitable for R1

The result of this rule is shown in the trace by the following statement:

input(GlobalChangeManager|ChangeGroup)|oppose(inform(shared_allocation, MemberOne, RegionHeadSneek, RegionSneek), MemberTwo, not_suitable_candidate)

As a result a domain specific instantiation of RP4 fires which is specified below.

RP4_specific(ChangeManager): Convince member

if	at time t a Member M1 communicates opposing to the change to organizational model M to the Change Manager because candidate M2 is considered not suitable for the allocation to role R1
and	the Change Manager observed M2 is the only candidate for the role R1 at time t
then	at time t + 1 the ChangeManager communicates that M2 is the only candidate for role R1.

In the trace, the communication can be seen in the following format:

output(GlobalChangeManager|ChangeGroup)|additional_argument(inform(shared_allocation, MemberOne, RegionHeadSneek, RegionSneek), MemberTwo, only_candidate)

Finally, a rule RP5 is specified for this domain as well, as shown below. Since the successful organization of an eleven cities tour is most important for the Members and all roles being allocated is essential for such a successful organization, they seize to oppose to an allocation in case they are informed about the existence of only one candidate.

RP5_specific(Member Two): Member convinced

if at time t Member Two opposes to the change to the organizational model M regarding the allocation of Member M1 to Role R1
and at time t2 later than t Member Two receives the argument that Member M1 is the only candidate for role R1
then at time t2 + 1 Member Two will inform the Change Manager upon its acceptance of the change to the organizational model M

In the trace the Member indeed outputs the belief upon the shared allocation:

output(MemberTwo|ChangeGroup)|accept(shared_allocation, MemberOne, RegionHead-Sneek, RegionSneek)

Since all Members have now communicated their acceptance of the new organizational model, the unfreezing phase is performed successfully.

6.2.3 *Movement and refreezing of the region formation*

The movement and refreezing phase for the case study are much shorter than the unfreezing phase, as the new organizational model is already accepted by all Members of the organization. The two phases are shown in Fig. 12. Trigger for the ChangeManager to start the movement phase is when an acceptance on all parts of the organizational model M has been communicated to the Change Manager, as specified before in RP6. The movement phase starts with the communication of the region structure being active:

output(GlobalChangeManager|ChangeGroup)|inform(active(organizational_model(region_structure)))

The phase ends after all participants of the change have confirmed that the organizational model will be active, which they instantly do as they are already unfrozen:

input(GlobalChangeManager|ChangeGroup)|accept(active(organizational_model(region_structure)), MemberOne, ChangeGroup)
input(GlobalChangeManager|ChangeGroup)|accept(active(organizational_model(region_structure)), MemberTwo, ChangeGroup)

Finally, the refreezing phase ends after the duration set by the Change Manager. In this particular refreezing phase, all roles immediately behave correctly after the change (according to RP7 in Section 4.2.3) which is not shown in the trace for the sake of brevity. Eventually, the Change Group is deactivated:

output(GlobalChangeManager|ChangeGroup)|inform(change_group_inactive)

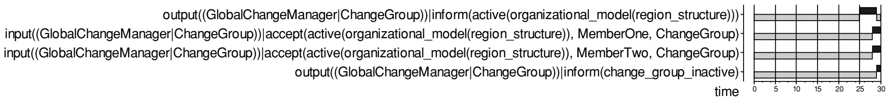

Fig. 12 First movement and Refreezing

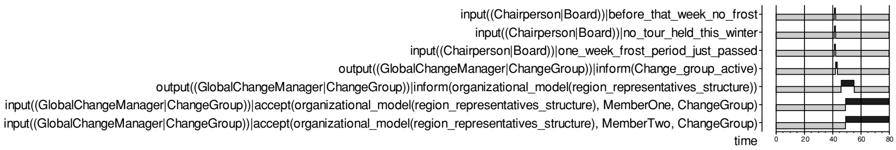

Fig. 13 Second unfreezing phase

6.2.4 Unfreezing phase for regions representatives group

The second unfreezing phase which is required to form the Region Representatives group is shown very briefly in Fig. 13. As a start of the unfreezing phase the following events are put into the simulation:

input(Chairperson|Board)|one_week_frost_period_just_passed
input(Chairperson|Board)|before_that_week_no_frost
input(Chairperson|Board)|no_tour_held_this_winter

As a result, the following inter group interaction property fires:

GIP_specific(Chairperson, Board, ChangeManager, ChangeGroup): Form Region Representatives group after frost period

if	at time t the Chairperson within the Board observes a period of one week of frost
and	at time t−(1 week) the Chairperson within the observed that there was no frost
and	at no time point this year the Chairperson within the Board observed that a tour has been held
then	at time t + (1 day) the Change Manager within the Change Group informs the Members within the Change Group that the group is now active
and	at time t + (1 day) the Change Manager within the Change Group announces the new organizational model regarding the Regions Representatives group.

The resulting communication of the Change Manager is shown in the trace: A communication of the Change Group being active again, and communication of the new organizational model:

output(GlobalChangeManager|ChangeGroup)|inform(change_group_active)
output(GlobalChangeManager|ChangeGroup)|inform(organizational_model(region_coordination_structure))

All Members accept the new structure, as they are very eager just thinking about a possible eleven cities tour, the event of the year, and are therefore immediately unfrozen, communicating their acceptance to the Change Manager. Therefore, the unfreezing is performed using RP2. Resistance can however easily be incorporated using properties such as presented in Section 6.2.2. The unfreezing process has now ended successfully.

Fig. 14 Second Movement and refreezing

6.2.5 Movement and refreezing of the region representatives group

After unfreezing the organization, the Change Manager communicates that the new organization with the new Region Representatives structure is now active, which is shown in the partial trace in Fig. 14:

output(GlobalChangeManager|ChangeGroup)|inform(active(
organizational_model(region_coordination_structure))

As a result, the organizational model becomes active in the actual organization, not only in the internal beliefs of the Members of the Change Group. Within the simulation there is a mapping between the name of general organizational structures (e.g. region coordination structure) and the actual changes on a lower level. For the Region Head Woudsend for example, the internal belief that a new role Region Representative Woudsend exists is added:

internal((RegionHeadWoudsend|RegionWoudsend))|belief(exists_role(RegionRepresentative Woudsend))

Furthermore, the group Region Representatives is added to the internal beliefs:

internal((RegionHeadWoudsend|RegionWoudsend))|belief(exists_group(RegionRepresentatives))

The role RegionRepresentativeWoudsend belongs to the group RegionRepresentatives:

internal((RegionHeadWoudsend|RegionWoudsend))|belief(role_belongs_to_group(Region RepresentativeWoudsend, RegionRepresentatives))

A belief on an inter-group connection is added between the Region Head Woudsend within the Region Woudsend and the Region Representative Woudsend within the Region Representatives:

internal((RegionHeadWoudsend|RegionWoudsend))|belief(inter_group_connection(RegionHead Woudsend, RegionWoudsend, RegionRepresentativeWoudsend, RegionRepresentatives, gi24))

Besides the structure itself, the new roles also require new behavior. When first starting to perform a new role, the new behavior associated with the role is far from automated, and requires internal beliefs on the desired behavior. Such elements are shown in the trace of Fig. 9 as well. First of all, there is an internal belief about the existence of a group interaction property gip1:

internal((RegionHeadWoudsend|RegionWoudsend))|belief(group_interaction_property(gip1, RegionHeadWoudsend, RegionWoudsend, RegionRepresentativeWoudsend, RegionRepresentatives))

The specification of the behavior required by such a property is done using a TTL expression, more particular in leadsto format:

internal((RegionHeadWoudsend|RegionWoudsend))|belief(has_expression(gip1, leads_to((input((RegionHeadWoudsend|RegionWoudsend))|report(RegionWoudsend, good)), output((RegionRepresentativeWoudsend|RegionRepresentatives))|report(Region Woudsend, good), efgh(0,0,1,1))))

This specifies that if the Region Head Woudsend receives a report that the ice is good, then this will be communicated by the Region Representative Woudsend in the Region Representatives group as well, with an efgh value of (0,0,1,1). Around time point 65 the antecedent of this rule becomes true, however, the consequent is not true after 1 time point within the Region Representatives group. As a result, the Chairperson Representative within the Region Representatives group reminds the role of the desired behavior gip1 (according to RP9 in Section 4.2.3):

output((CharipersonRepresentative|RegionRepresentatives))|remind(gip1)

After having received this reminder, the Region Representative Woudsend does behave according to gip1 and outputs the consequent:

output((RegionRepresentativeWoudsend|RegionRepresentatives))|report(RegionWoudsend, good)

This refreezing therefore takes the form of GP6 (Section 4.2.3) and the properties below it in the property tree. In exceptional years, all Region Representatives report that the ice is good, and the Chairperson within the board announces the date the tour will take place:

output((Chairperson|Board))|let_the_tour_be_held_on_date

7 Verification of the case study simulation

As verification of the organization process of the Eleven Cities Tour is concerned, a distinction is made between two types of verification. Firstly, guarantees are given that concern the tour itself (so-called content properties). For example, it the circumstances permit so (if the ice is thick enough over the whole trajectory) then a tour should be organized as soon as possible. Secondly, guarantees on the organization of organizational change for setting up the tour are verified (so-called organizational change properties). This Section presents both verification types.

Logical relationships between properties, as depicted in the tree of Section 4, can be very useful in analyzing the dynamic properties of an organization. For example, if for a given trace of the system some global property OP is not satisfied, then by a refutation process it can be concluded that either one of the group properties, or one of the group interaction properties in the tree does not hold. If, after checking these properties, it turns out that a group property does not hold, then either one of the role properties or the intra group interaction properties is not satisfied. By this refutation analysis it follows that if OP does not hold for a given trace, then, via the intermediate properties, the cause of this malfunctioning can be found in the set of leaves of the tree of Section 4.

In order to determine which one of the properties encountered in this refutation process actually is refuted, some mechanism is needed to check if a certain property holds for a given trace. To this end, the simulation software described in Section 6 automatically produces log files containing the traces. In addition, software has been developed that is able to read in these log files together with a set of dynamic properties (in TTL format), and to perform the checking process. Traces are thus analyzed with an automated logic-based checker. This checker takes as input a property of interest about the trace and logically validates whether the property holds in the given trace. If the property holds in the trace, the checker outputs success otherwise it outputs failure. But the software determines not only whether the properties hold for the trace or not, but in case of failure, it also pinpoints which parts of the trace violate the properties. The results of different checks that have been performed are described below.

7.1 Content properties

The overall goal of the Eleven Cities Tour organization is to arrange for a tour to be organized when possible, i.e., when the ice along the tour is thick enough to ensure a safe passage. This following property expresses this goal: the tour has to be organized whenever possible, ensuring a safe passage over the ice for all skaters.

OP3: Organize tour in case of good conditions

if the ice conditions in all regions are good
then it is announced that the tour will be held

This property has been checked against the simulation trace that was presented in the previous Section and is indeed satisfied within that trace. Other content properties to consider in this context are, for example, the organization daily decides on the possibility and date (if appropriate) of a tour: ‘it giet oan’ (in Frisian language a go decision) decisions, and in wintertime, the organization daily monitors the weather. However, only OP3 is addressed in this paper.

Table 6 Checked Properties (Yes = satisfied, No = not satisfied)

Stage	Property	Change 1	Change 2
Unfreezing	GP1	Yes	Yes
	GP3	Yes	No
Moving	OP1	Yes	Yes
Refreezing	OP2	Yes	Yes
	GP6	No	Yes

7.2 Organizational change properties

The properties as presented in the previous Section depend on some organizational structure to ensure the fulfillment of each property and all of them combined. For this purpose, the aim of this paper is exactly this: a way to specify and model such an organization itself has been presented, as well as the actual process of setting up the organization. As such, this organization can support the organizational properties as presented above.

For the purpose of verifying the organizational change in the Eleven Cities Tour simulation, automatic checking of the high-level properties presented in Section 4.2 has been performed on the generated trace. The results are shown in Table 6. In the simulation trace, there are 2 change moments: the formation of the regions structure and the formation of the region representatives group. For the last changes, there is no resistance to the organization change, while there is for the first one. The automated checker has verified that all properties specifying a successful phase are indeed satisfied, hence, both changes have passed a successful unfreezing, movement, and refreezing phase. There is however a difference in how this success was accomplished. In the first change, property GP3 was satisfied, specifying that there was resistance to the change which was taken away. In the second change however, the change went without resistance; property GP3 was not satisfied in that change. In the refreezing phase of the first change, property GP6 was not satisfied as no improper behavior was encountered. In the second change however, improper behavior did show, after which the behavior was corrected, satisfying property GP6. The following setting were used for checking. For the unfreezing phase e was set to 1.0. Regarding the refreezing phase both e1 and e2 have been set to 1.0, for p1 a value of 10 was used, and finally, p2 was set to 20.

8 Conclusions

Organizations often have to survive in a dynamic world. To enable organizations in practice to adapt to the dynamics of the world, certain facilities, structures and capabilities are needed that support organizational change. This paper shows how the organization of organizational change processes can be modeled within a formal organization modeling approach. A generic organization model for organizational change was presented and formally verified for a case study concerning the organization of a major event in the Netherlands: the eleven cities tour. The formal verification sets it apart from existing work on organization modeling, e.g., (Fox and Gruninger, 1998; Steen Lankhorst and Wetering, 2002). Previous work of the authors on organizational change (Jonker Schut and Treur, 2003) considered change as an instantaneous event instead of a process of change as is done in this paper. Additionally, previous work did not include the distinction between formal languages for expressing the change process. The change model in this paper takes into account different phases in a change process (unfreezing, movement and refreezing) considered in Lewin (1951), which is still considered valid in current organizational change literature, see e.g. (Robbins 1998; Orlikowski and Hofman, 1997). In Orlikowski and Hofman (1997) a distinction is made between anticipated change (for which the model of Lewin is said to be suitable), emergent change and opportunity-based change. In this paper only anticipated change is being modeled and therefore the other two types of change are not addressed. In change processes the internal (mental) states of those involved in the organization are important. Therefore, also internal states of individuals have to be part of a model for organizational change. In particular, beliefs and their changes have been incorporated in the model. In addition, an internal model for (reflective) reasoning about expected role behavior was included. Hence, a model was created that combines organization aspects and cognitive aspects.

Appendix A: The temporal trace language TTL: more formal details

A state ontology is a specification (in order-sorted logic) of a vocabulary. A state for ontology Ont is an assignment of truth-values {true, false} to the set At(Ont) of ground atoms expressed in terms of Ont. The *set of all possible states* for state ontology Ont is denoted by STATES(Ont). The set of *state properties* STATPROP(Ont) for state ontology Ont is the set of all propositions over ground atoms from At(Ont). A fixed *time frame* T is assumed which is linearly ordered. A *trace* or *trajectory* γ over a state ontology Ont and time frame T is a mapping γ : T → STATES(Ont), i.e., a sequence of states γ_t (t ∈ T) in STATES(Ont). The set of all traces over state ontology Ont is denoted by TRACES(Ont). Depending on the application, the time frame T may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers), or any other form, as long as it has a linear ordering. 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, the input state of some role r within a group g at time point t is denoted by

state(γ , t, input(r | g))

analogously

state(γ, t, output(r | g))
state(γ, t, internal(r | g))

denote the output state and internal state.

These states can be related to state properties via the formally defined satisfaction relation |=, comparable to the Holds-predicate in the Situation Calculus: state(γ, t, output(r | g)) |= p denotes that state property p holds in trace γ at time t in the output state of role r within group g. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic with sorts

TIME or T for time points, Traces for traces and F for state formulae, using quantifiers over time and the usual first-order logical connectives such as ¬, ∧, ∨, ⇒, ∀, ∃. In trace descriptions, notations such as state(γ, t, output(r | g))| = p are shortened to output(r | g)|p.

Appendix B: Constraints on the language elements

Role dynamic properties

If has_expression(p:DYNPROP, d:DYNPROPEXP)
and role_property(p, r:ROLE, g:GROUP)
then element_of(d, DYNPROPEXP(r|g, ONT(r|g)))

The group is also part of the definition of the ontology since roles in different groups can have the same name and might have a different ontology.

Role properties can be divided into different types which in turn can be defined more restricted than the general definition. An example of such a refinement is an executable role dynamic property. This special type is defined as follows:

if has_expression(p:DYNPROP, d:DYNPROPEXP)
and role_property(p, r:ROLE, g:GROUP)
then element_of(d, DYNPROPEXP((r|g), role_input_ontologies(r|g)
∪ role_ouput_ontologies(r|g)))

Transfer dynamic properties

If has_expression(p:DYNPROP, d:DYNPROPEXP)
and transfer_property(p, r1:ROLE, r2:ROLE, g:GROUP)
then element_of(d, DYNPROPEXP({r1|g, r2|g}, role_output_ontologies(r1|g)
∪role_input_ontologies(r2|g)))

Group dynamic properties

If has_expression(p:DYNPROP, d:DYNPROPEXP)
and group_property(p, g:GROUP)
then element_of(d,DYNPROPEXP(g, ONT(g)))

Intergroup interaction dynamic properties

If has_expression(p:DYNPROP, d:DYNPROPEXP)
and group_interaction_property(p, r1:ROLE, g1:GROUP,r2:ROLE, g2:GROUP)
then element_of(d, DYNPROPEXP({r1|g1, r2|g2}, role_input_ontologies(r1|g1)
∪ role_output_ontologies(r2|g2)))

Appendix C: Changing organizations formalized in meta TTL

This is the formal part from Section 5.

C.1 Sorts and Subsorts in meta-TTL

The sorts that are included in meta-TTL are shown in Table C.1. The subsort relation

Table C.1 Sorts in meta TTL

Sort	Description
TRACE	for traces
STATE	for states within a trace.
T	time frame.
STATOMS	expressions for state atoms.
CONSTATOMS	expressions for conjunctions of state atoms.
STATPROPEXP	expressions for state properties.

STATOMS $\subseteq$ CONSTATOMS holds.

The function

and: CONSTATOMS $\times$ CONSTATOMS $\rightarrow$ CONSTATOMS

is used to build conjunctions of state atoms; it is also written as $\wedge$ in infix notation Furthermore, the relation <: T $\times$ T for time ordering is used , and the function

state: TRACE $\times$ T $\times$ PART $\rightarrow$ STATE

that indicates the state of part of the considered system within a trace at some point in time.

For the changing organization it is needed to use names and expressions for dynamic properties within other formulae. Therefore two sorts

DYNPROP names for dynamic properties

DYNPROPEXP expressions for dynamic properties

have been introduced in the Appendix A.

Moreover,

holds: STATE $\times$ STATPROPEXP $\rightarrow$ DYNPROPEXP

indicates the dynamic property that a state property expression is true in a state; this predicate holds is often written as |= in infix notation.

C.2 Example formalization in change language

By means of an example the use of the functions combined with the language is shown below.

to_be_performed(delete(exists_role(RoleTwo))) $\wedge$
to_be_performed(delete(role_belongs_to_goup(RoleTwo, Group1))) $\wedge$
to_be_performed(delete(intra_group_connection(RoleOne, RoleTwo, Group1, t1))) $\wedge$
to_be_performed(delete(intra_group_connection(RoleTwo, RoleOne, Group1, t1))) $\wedge$ } SL

to_be_performed(delete(transfer_property(tp1, RoleOne, RoleTwo, Group1))) $\wedge$
to_be_performed(delete(has_expression(tp1, {expression1}))) $\wedge$
to_be_performed(delete(transfer_property(tp2, RoleTwo, RoleOne, Group1))) $\wedge$
to_be_performed(delete(has_expression(tp2, {expression2}))) } BL

The example models the deletion of Role One from Group1. Both specification languages have been used to model this change as is shown by the braces at the side.

C.3 Building properties for the changing organization

In a change process it is needed that the roles have beliefs about the organization structure. Therefore all organization structure representations described in Section 4 are included ; some examples are shown in Table C.2,

Table C.2 Examples of included organization structure representations

exists_role : ROLE → STATPROPEXP
role_belongs_to_group: ROLE × GROUP → STATPROPEXP
role_property: DYNPROP × ROLE × GROUP → STATPROPEXP
has_expression: DYNPROP × DYNPROPEXP → STATPROPEXP
allocated_to: AGENT × ROLE × GROUP → STATPROPEXP

Moreover, to express beliefs, the following language construct is used:

belief: STATPROPEXP → STATPROPEXP

An example of its use is: belief(exists_role(s) ∧ role_belongs_to_group(s, g))
Furthermore it is needed that the roles have beliefs about the behavioral properties that are expected from a certain role. Therefore first a representation

leads_to: CONSTATOMS × CONSTATOMS → DYNPROPEXP

is introduced for a simple type of such properties. A more general type of dynamic property is built using:

& : DYNPROPEXP × DYNPROPEXP → DYNPROPEXP

and similarly for other logical connectives such as not, ⇒, ∀, ∃.
Thus within the sort DYNPROPEXP two types of expressions are built:

- temporal statements based on atoms of the form state(γ, t, P) |= p for state properties p
- leads to statements of the form leads_to(V, W) with V and W conjunctions of atoms

Although the latter type of expressions can be mapped to (are definable in terms of) the former type of expressions, for simplicity they are kept separate.

An example of an expression that can be built using the constructs above is the following
∃t state(γ, t, internal(r)) |= belief(exists_role(s) ∧ role_belongs_to_group(s, g)) ∧
belief(role_property(d1, s, g)) ∧
belief(has_expression(d1, leads_to(a∧b, c)))

This expression states that
there will be a time that
within role r there is the belief that
the organization structure includes role s in group g, and
this role has dynamic property d1 which
is expressed by leads_to(a∧b, c).

Another example property is the following, describing that a role performs the behavior it believes that is expected from the role:

If at time t
a role believes that
this role has as part of its behavior description that
upon input v the output action w is done,
and
v occurs as input,
then
at a next point in time this role will provide output w.

Here the nesting is visible in the informal structured text representation using tabs. The formalization of this property also shows a nesting as indicated.

[state(γ, t, internal(RegHead)) | = belief(role_property(d, RegHead, RegGroup1)) $\wedge$
[belief(has_expression(d, leads_to(v, w)))
[& state(γ, t, input(RegHead)) | = v]
[$\Rightarrow$ $\exists$t' $\geq$ t state(γ, t', output(RegHead)) | = w

Acknowledgments This research has been performed as part of two projects: CIM, for Cybernetic Incident Management, and DEAL, for Distributed Engine for Advanced Logistics. Both projects are funded by the Dutch Ministry of Economic Affairs. Furthermore, the authors would like to thank the anonymous reviewers for their useful comments that helped to improve the paper.

References

Abbink H, Dijk R van, Dobos T, Hoogendoorn M, Jonker CM, Konur S, Maanen PP van, Popova V, Sharpanskykh A, Tooren P van, Treur J, Valk J, Xu L, Yolum P (2004) Automated support for adaptive incident management. In: Walle B van de, Carle B (eds) Proc. of the first international workshop on information systems for crisis response and management, ISCRAM'04, pp 69–74

Ackerman LS (1986) Development, transition, or transformation: the question of change in organization. OD Practitioner 18(4):1–9

Alvesson M, (1993) Cultural perspectives on organizations. Cambridge University Press, New York

Bacharach SB, Lawler EJ (1980) Power and politics in organizations. Jossey-Bass, San Francisco

Bashein ML, Marcus ML, Riley P (1994) Business process reengineering: preconditions for success and failure. Inform Syst Manag 9:24–31

Boonstra JJ (ed) (2004) Dynamics of organizational change and learning. Wiley

Bosse T, Jonker CM, Meg L van der, Treur J, (2005) LEADSTO: a language and environment for analysis of dynamics by simulaTiOn. In: Eymann T et al (eds) Proc. of the third german conference on multi-agent system technologies, MATES'05. Lecture notes in artificial intelligence. Springer Verlag, vol. 3550. pp. 165–178

Bradshaw P, Boonstra JJ (2004) Power dynamics in organizational change. In: Boonstra JJ (ed) Dynamics of organizational change and learning. Wiley, pp 279–299

Cummings TG, Worley CG (2001) Organization development and change. South Western College Publishing

Cummings TG (2004) Organization development and change. In: Boonstra JJ (ed) Dynamics of organizational change and learning. Wiley, pp 25–42

Dahl RA (1975) The concept of power. Behav Sci 2:201–215

Emerson RM (1962) Power dependence relations. Amer Socio Rev 27:31–41

Ferber J, Gutknecht O (1998) A meta-model for the analysis and design of organizations in multi-agent systems. In: Proceedings of the third international conference on multi-agent systems (ICMAS'98). IEEE Computer Society Press, pp 128–135

Ferber J, Gutknecht O, Jonker CM, Müller JP, Treur J (2001) Organization models and behavioural requirements specification for multi-agent systems. In: Demazeau Y, Garijo F (eds) Multi-agent system organizations. Proceedings of MAAMAW'01

Fox MS, Gruninger M (1998) Enterprise modelling. AI Magazine 19(3);AAAI Press, pp. 109–121

Glaser N, Morignot P (1997) The reorganization of societies of autonomous agents. In: Boman M, Velde W van de (eds) Multi-agent rationality, 8th european workshop on modelling autonomous agents in a multi-agent world, Lecture notes in computer science, Springer, vol 1237 pp 98–111

Hall G, Rosenthal T, Wade J (1993) How to make reengineering really work. Harv Busin Rev 71(6):119–131

Hoogendoorn M, Jonker CM, Konur S, Maanen PP van, Popova V, Sharpanskykh A, Treur J, Xu L, Yolum P (2004) Formal analysis of empirical traces in incident management. In: Macintosh A, Ellis R, Allen T (eds) Applications and innovations in intelligent systems xii, proceedings of AI-2004, the 24th SGAI international conference on innovative techniques and applications of artificial intelligence. Springer Verlag, pp 237–250

Hoogendoorn M, Jonker CM, Popova V, Sharpaskykh A, Xu L (2005) Formal modelling and comparing of disaster plans. In: Carle B, Walle B van de (eds) Proceedings of the second international conference on information systems for crisis response and management ISCRAM '05, pp 97–107

Hosking DM (1999) Social construction as process: some new possibilities for research and development. Con Trans 4(2):117–132

Huczynski A, Buchanan D (2001) Organizational behaviour. Prentice Hall

Jaffee D (2001) Organization theory—tension and change. McGraw-Hill companies

Jonker CM, Treur J (2002) Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness. Intern J Cooper Inform Syst 11:51–92

Jonker CM, Treur J (2003) Relating structure and dynamics in an organization model. In: Sichman, JS Bousquet F, Davidson P (eds) Multi-agent-based simulation II. Proceedings of the third international workshop on multi-agent based simulation, MABS'02. Lecture notes in AI, Springer Verlag, vol 2581 pp 50–69

Jonker CM, Schut MC, Treur J (2003) Modelling the dynamics of organizational change. In: Klusch M, Omicini A, Ossowski S, Laamanen H (eds) Cooperative information agents VII. Proceedings of the seventh international workshop on cooperative information agents, CIA 2003. Lecture notes in AI, Springer Verlag, vol. 2782. pp. 336–344

Kotter JP (1999) Leading change. Harvard Business School Press, Boston

Lewin K (1951) Field theory in social science. Harper & Row, New York

Lippit R, Watson J, Westley B (1958) The dynamics of planned change. Harcourt, New York

Orlikowski W, Hofman D (1997) An improvisational model of change management: the case of groupware technologies. Sloan Manag Rev 38(2):11–22

Robbins SP (1998) Organizational behaviour. Prentice Hall, New Jersey

Schein EH (1993) On dialogue, culture, and organizational learning. Organiz dynam 22(2):40–51

Steen MWA, Lankhorst MM, Wetering RG van de (2002) Modelling networked enterprises. In: proceedings of the 6th international enterprise distributed object computing conference (EDOC). IEEE Computer Society, pp. 109–119

Wrong DH (1968) Some problems in defining social power. Amer J Soc 73:673–681

Mark Hoogendoorn is a PhD student at the Vrije Universiteit Amsterdam, Department of Artificial Intelligence. He obtained his Masters degree in Computer Science in 2003 at the same university, graduating on a project related to multi-agent negotiation as part of the MAGNET research group at the University of Minnesota. In his PhD research he focuses on organizational change within multi-agent systems, applying his research in various domains, including incident management, logistics, and the naval domain.

Catholijn M. Jonker is a full professor in Artificial Intelligence and Cognitive Science at the Nijmegen Institute for Cognition and Information of the Radboud Universiteit Nijmegen in the Netherlands. She studied computer science at Utrecht University. She completed her PhD on the topic of Negations and Constraints in Logic Programming at the same university. After completing a post-doc position on the same topic at the Universität Bern, she became an assistant professor at the Vrije Universiteit Amsterdam and switched her research topic to agent technology. During the time at the Vrije Universiteit her interest in cognitive

science increased, which she combined with her work on modeling multi-agent systems and organizations and her work on the analysis and modeling of the dynamics of behavior of complex systems. These research topics now contribute to the research program of the Cognitive Artificial Intelligence division of the Nijmegen Institute for Cognition and Information.

Martijn C. Schut is Assistant Professor at the Department of Artificial Intelligence, Vrije Universiteit, Amsterdam, The Netherlands. He received a MSc from the Vrije Universiteit (NL) and a PhD from the University of Liverpool (UK). His research interests concern the emergence of organizational dynamics in distributed multi-agent systems.

Jan Treur received his Ph.D. in Mathematics and Logic in 1976 from Utrecht University. Since 1986 he works in Artificial Intelligence, from 1990 as a full professor and head of the Department of Artificial Intelligence at the Vrije Universiteit Amsterdam. In the 1990s he headed a research program on component-based design of knowledge-based and agent systems. In the last five years the research program focused on modeling dynamics of agent systems in practical application areas, and related to other disciplines such as Biology, Cognitive Science, Organization Theory, and Philosophy of Mind.