Simulating Organizational Change Triggered by a Changing Environment

Mark Hoogendoorn[1], Catholijn M. Jonker[2], and Jan Treur[1]

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{mhoogen, treur}@cs.vu.nl
[2]Radboud Universteit Nijmegen, NICI,
Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
C.Jonker@nici.ru.nl

KEYWORDS

Simulation of organizational change, multi-agent based simulation

ABSTRACT

Organizational modeling gains importance in the development of multi-agent systems, both for modeling and simulation of phenomena in social science and economics, as for developing complex systems to support, e.g., virtual enterprises. This paper contributes to these efforts an analysis and example formalizations and simulation of cause and effect of organizational change. Causes of change are categorized and related to internals within organizations via trees covering different aggregation levels. Furthermore, properties which should be satisfied by organizational change are identified and specified.

INTRODUCTION

Organization is a systematic arrangement or approach, but also an organized body of people with a particular purpose, e.g. a business. Furthermore, organizational form refers to the structure chosen in which the whole consists of mutually influential parts. The word organization has meaning of old in sciences such as social science, economics, and psychology, but recently, see e.g., (Ciancarini and Wooldridge, 2001; Hannoun *et al.*, 1998, Hubner *et al.*, 2002), also in computer science and artificial intelligence.

Each organization has its own dynamics, whether the organization is real or virtual. The time scale of change differs with the domain and the dynamics of the environment of the organization. The time scale refers both to the durations of the periods between change, and speed of the process of change. Where it might be ages before the form of government of a nation changes, the change itself might be brought about within a day. The formation of an alliance like the European Union is a lengthy process that after some decades is still changing both in laws and regulations, as well as in members. Other organizations are more temporary in nature, e.g., the cooperation by city council, police, fire department, and ambulances to manage an incident is structural in the sense that all involved parties know how to organize themselves in case of incident, but the organization lasts only as long as the incident. Similarly, alliances between enterprises or countries can be temporary or actually involve a complete reorganization of all parties involved.

This paper introduces an approach for simulation and analysis of adaptive (multi-)agent systems and underlying mechanisms that is integrative in two ways:

(1) It combines in one modeling framework both *qualitative, logical* and *quantitative, numerical* aspects
(2) It allows to model dynamics at *different aggregation levels*, from a more local level (e.g., behaviors of roles within the organization) to a global level (behavior of the multi-agent organization as a whole); moreover, *interlevel relations* can be specified that express relationships between dynamic properties at different levels

Modeling dynamics at a local level often concerns expressing temporal relationships between pairs of successive states, such as described, for example, by basic steps within an adaptation mechanism. Local level specifications are the basis for the computation steps for a simulation model. From the more global perspective, more complex relationships over time can be used to model dynamics for adaptive multi-agent organizations. For example, how during a history of events to which it adapts, the system's behavior is changing.

Interlevel relations often take the hierarchical form of an AND-tree (or a number of them), with the most global property at the top (root) and the most local at the leaves. Such a hierarchical structure can be useful in the analysis of, in case, why a global property fails on a certain simulation trace. By going down in the tree and at each level checking the properties under the failing node, finally the leaf or leaves that fail(s) can be found, thus pinpointing the (local) cause of the failure. This can be useful in debugging a model, but also in the analysis

of the circumstances under which a model will function well and under which not, and the reasons why.

This paper more specifically studies the triggers of conscious organizational change. In the next section the organizational modeling approach used throughout this paper is addressed. In the section after that a distinction is made between internal and external triggers for organizational change, of which examples are given in the form of dynamic properties. Thereafter a section is devoted to showing how high-level dynamic properties (such as those specified in the section regarding internal and external triggers) can be related to lower-level properties within the organization. Formalizations for describing organizational change are presented in the following section. Furthermore, the section thereafter shows how the properties identified in the previously mentioned sections can be used for simulation. Verification of the simulation results is performed in second last section, and finally the last section is a discussion.

ORGANIZATION MODELING APPROACH

An organizational structure defines different elements in an organization and relations between them. The dynamics of these different elements can be characterized by sets of dynamic properties. An organizational structure has the aim to keep the overall dynamics of the organization manageable; therefore the structural relations between the different elements within the organizational structure have to impose somehow relationships or dependencies between their dynamics; cf. (Jonker and Treur, 2003). In the introduction to their book Lomi and Larsen (2001) emphasize the importance of such relationships:

- 'given a set of assumptions about (different forms of) individual behavior, how can the aggregate properties of a system be determined (or predicted) that are generated by the repeated interaction among those individual units?'
- 'given observable regularities in the behavior of a composite system, which rules and procedures - if adopted by the individual units- induce and sustain these regularities?'

Both views and problems require means to express relationships between dynamics of different elements and different levels of aggregation within an organization. In Lomi and Larsen (2001) two levels are mentioned: the level of the organization as a whole versus the level of the units. Also in the development of MOISE (Hannoun et al., 1998; Hannoun et al., 200; Hubner et al., 2002) an emphasis is put on relating dynamics to structure. Within MOISE dynamics is described at the level of units by the goals, actions, plans and resources allocated to roles to obtain the organization's task as a whole. Specification of the task as a whole may involve achieving a final (goal) state, or

an ongoing process (maintenance goals) and an associated plan specification.

The approach in this paper will be illustrated for the AGR (Ferber and Gutknecht, 1998) organization modeling approach. Figure 1 shows an example organization modeled using AGR. Within AGR organization models three aggregation levels are distinguished: (1) the organization as a whole; the highest aggregation level, denoted by the big oval in Figure 1, (2) the level of a group denoted by the middle size in the Figure, and (3) the level of a role within a group denoted by the smallest ovals in Figure 1. Solid arrows denote transfer between roles within a group; dashed lines denote inter-group interactions. This format will be adopted to formalize organizations. In addition, behavioral properties of elements of an organization are part of an organizational model. The format to express these will be shown in the next section, which addresses requirements that can be formulated for organizations.
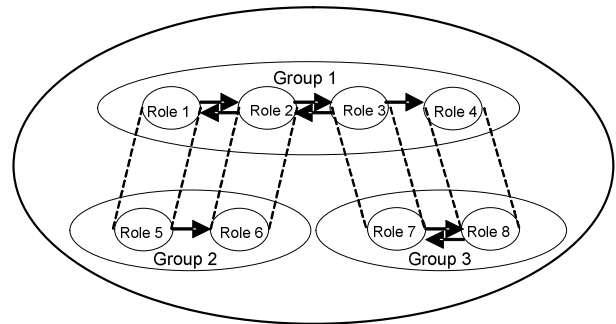

Figure 1. An AGR Organization Structure

REQUIREMENTS CHANGES AND ORGANIZATIONAL CHANGE

In real life as well as in virtual reality, the requirements an organization needs to satisfy change due to changing environmental circumstances. Such changes are caused by triggers external from the organization. The general pattern for such triggers is follows. A certain organizational goal G (e.g. sufficient demand) is no longer reached, due to an environmental change, say from E1 to E2. In the old situation requirement R1 was sufficient to guarantee G under environmental condition E1:

$$E1 \ \& \ R1 \ \Rightarrow \ G$$

Here R1 is a requirement expressing a logical/temporal relation which states that under the condition E1 the organization is able to achieve G. The change from E1 to E2 makes that requirement R1, which is still fulfilled but has become insufficient, is replaced by a new, stronger requirement R2 which expresses that under environment E2 goal G can be achieved by organization O such that
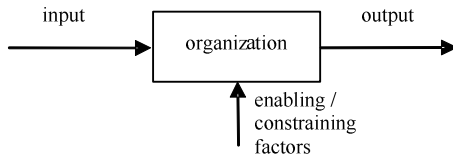
$$E2 \ \& \ R2 \ \Rightarrow \ G$$

Figure 2. Flow of information in an organization

Therefore, an organization is triggered to change to fulfill R2 and thus fulfill goal G again.

Jaffee (2001) distinguishes several of these external triggers for organizational change. This paper presents a classification (see Figure 2) of those triggers based on the flow of information for an organization. The input type of external trigger includes the triggers the organization notices on its input, for example changes in the resources or suppliers. *Enabling / constraining factors* are external triggers such as government rules and technology that concern processes within the organization. Finally, *output* can influence the input of an organization and can therefore affect the triggers received by an organization. Output information itself is however not considered a trigger for organizational change.

Input Changes

The input of an organization can originate from a variety of different sources. Each of these sources can cause a change of requirements, and possibly trigger an organization to change.

A first source is formed by the *suppliers* who can increase their price of a product P, which is used by the organization for the production, at time t from M_1 to M_2. A formal form of this trigger is specified in E1 using the Temporal Trace Language (TTL) (Jonker and Treur 2002). In the definition of the property state(γ, t) |= observation_result(price(P, R), pos) denotes that within the state state(γ, t) at time point t in trace γ the state property observation_result(price(P, R), pos) holds, denoted by the satisfaction relation |= Furthermore, the relation observation_result is used for an element that is external for the organization, and belief is used if the element is within the internal knowledge of the organization.

E1(P, M, t): Supplier Price
$\exists$R:REAL   state(γ, t) |= observation_result(price(P, R), pos) & R ≤ M

Before the environmental change, E1(P1, M_1, t) specifies the relevant property of the environment. After the change of supplier price however, this property no longer holds whereas E1(P1, M_2, t) does hold. The overall goal to be maintained within the organization is to keep demand of product P above a threshold D. A formal specification of the goal is presented in OP1.

OP1(P, D, t): Sufficient demand
$\exists$I:INTEGER state(γ, t) |= observation_result(customer_demand(P, I), pos) & I ≥ D

The requirement imposed for the organization is to maintain the goal of keeping demand for product P1

above D, given the environmental condition of the price M for product P2 which is needed for the production of P1. This requirement is specified below in property R1.

R1(P1, P2, M, D): Maintain demand
$\forall$t :TIME
[state(γ, t) |= belief(needed_for_production_of(P1, P2), pos) & E1(P1, M, t)] $\Rightarrow$ OP1(P2, D, t)

Before the change in the environment, requirement R1(P1, P2, M_1, D) was sufficient to ensure the goal being reached. After the change however, the requirement is still satisfied but might be insufficient to ensure the goal. This is due to the fact that the environmental condition in the antecedent does not hold, and hence, cannot be used to entail G (although the requirement is fulfilled all the time). The requirement is therefore withdrawn and replaced by the requirement R1(P1, P2, M_2, D) which is not necessarily satisfied and might require an organizational change to enable such a fulfillment.

Secondly, an input trigger can be formed by *resources* that run out, becoming a lot more expensive. Therefore, the requirement for an organization triggered in such a way is to reduce the usage of the particular resource. This can for example be accomplished by focusing on a completely different, more viable product, or producing the same goods using different resources.

Another source is formed by the *customers* whose demands decreases for the good being produced. The organization can change direction (and thus change the organization) or keep producing the same good but decrease the output (and therefore also change the organization).

Finally, *competitors* might change their production methods causing a more efficient production process for products within the same product group as P, lowering their price from C_1 to C_2. This can be specified using property E2 introduced below.

E2(P, C, t): Competition Price
$\exists$R:REAL, G:PRODUCT_GROUP
state(γ, t) |= belief(belongs_to_product_group(P, G), pos) & state(γ, t) |= observation_result(competitor_price(G, R) & R ≥ C

The goal is again to maintain customer demand above the level D, for which OP1 can be reused. The requirement as presented below therefore changes from R2(P, C_1, D) to R2(P, C_2, D).

R2(P, C, D):Maintain Demand
$\forall$t :TIME   E2(P, C, t) $\Rightarrow$ OP1(P, D, t)

Changes in Enabling / Constraining Factors

Besides triggers on the input of an organization, another type of trigger exists: the enabling and constraining factors. First of all, the enabling factors within the organization include *technology*. In case the technology available to produce a product P changes from T1 to T2, the profit margin should remain at least at the same level D for a company.

OP2(P, D, t): Sufficient Profit Margin
    $\exists R$:REAL state(γ, t) |= belief(profit_margin(P, R), pos) & R ≥ D

E3(P, T, t): New Technology
    $\exists R$:REAL state(γ, t) |=
observation_result(technology_available_for(T, P), pos)

R3(P, T, D): Maintain Profit
    $\forall t$:TIME E3(P, T, t) $\Rightarrow$ OP1(P, D, t)

All properties have been specified similar to those presented in the previous subsection. Before the environmental change of available technology E3(P, T1, t) was the case whereas E3(P, T2, t) is the new environment. Secondly, constraining forces include *government regulations and labor aspects*. Government regulations for workers might affect human resource practices and composition of the workforce. Concerning labor aspects, the union might demand a reduction from 40 to 36 hours a week, which naturally causes organizational change. All these aspects should however not decrease overall profitability of the organization.

REQUIREMENTS REFINEMENT BASED ON INTERLEVEL RELATIONS

To fulfill requirements at the level of the organization as a whole as discussed in the previous section, parts of the organization need to behave adequately (see also the central challenges put forward by Lomi and Larsen (2001) as discussed before). Based on this idea, in this paper dynamics of an organization are characterized by sets of dynamic properties for the respective elements and aggregation levels of the organization. An important issue is how organizational structure relates to (mathematically defined) relationships between these sets of dynamic properties for the different elements and aggregation levels within an organization (cf. Jonker and Teur, 2003). Preferably such relations between sets of dynamic properties would be of a logical nature; this would allow the use of logical methods to analyze, verify and validate organization behavior in relation to organization structure. Indeed, following (Jonker and Treur, 2003), in the approach presented below, logical relationships between sets of dynamic properties of elements in an organization turn out an adequate manner to (mathematically) express such dynamic cross-element or cross-level relationships.

A general pattern for the dynamics in the organization as a whole in relation to the dynamics in groups is as follows:
        dynamic properties for the groups &
        dynamic properties for inter-group interaction
        $\Rightarrow$ dynamic properties for the organization
Moreover, dynamic properties of groups can be related to dynamic properties of roles as follows:
        dynamic properties for roles &
        dynamic properties for transfer between roles
        $\Rightarrow$ dynamic properties for a group
The idea is that these are properties dynamically relating a number of roles within one group.

A generic overview of the logical relationships between dynamic properties at different aggregation levels is
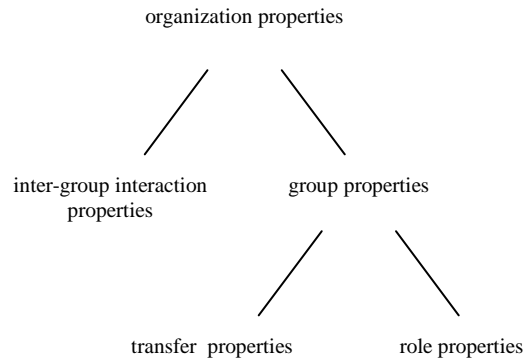

Figure 3. Overview of inter-level relations between dynamic properties

depicted as an AND-tree in Figure 3. It is possible that each level shown in the tree (for example organization properties) again consists of multiple levels. The logical relationships put forward above can be formalized further as shown in (Jonker and Treur 2003).

If the properties for roles, transfers and inter-group interactions are in executable format, and used for simulation (e.g., based on the paradigm of Executable Temporal Logic), then a generated trace will satisfy these properties, and, hence satisfy all group and organization properties as well. Among others, this gives means to validate an organization model. For simulation and validation TTL (for Temporal Trace Language, cf. Jonker and Treur 2002) is used throughout this paper.
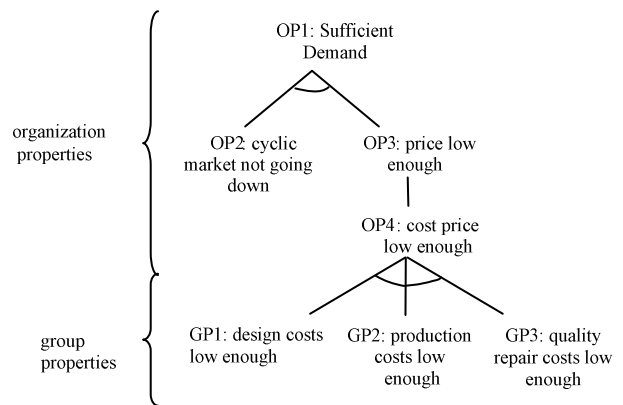

Figure 4. Hierarchy of organizational and group properties

Figure 4 shows an example of a hierarchy of dynamic properties for an organization producing certain products, the properties follow field observations at the Ford Motor Company in 1980 described in (Womack *et al.*, 1991). The overall organizational goal is to maintain sufficient demand for the goods being produced, as was also the case in OP1 in the section called requirements changes and organizational change. The organization has separate departments for design, production and quality control, which are modeled as groups in the organization. The highest levels represent organizational properties or goals whereas the lowest level shown here

represents group properties. Note that group properties could in turn be related to lower level (role) properties. A definition for each of the properties in Figure 4 is presented below. Notice that this hierarchy could easily be extended by other aspects (e.g., of quality of the products as a reason for the demand decreasing or not).

First, for OP1 see the section called requirements changes and organizational change. Furthermore, the cyclic market is not going down for a product P at time t in case the demand for the product group as a whole (i.e. all goods produced by different companies in this particular category) is not going down.

OP3(P, t): Cyclic market not going down
∀G:PRODUCT_GROUP, l1,l2:INTEGER
[state(γ, t) |= belief(belongs_to_product_group(P, G), pos) &
state(γ, (t-1)) |= observation_result(customer_demand(G, l1), pos) &
state(γ, t) |= observation_result(customer_demand(G, l2), pos)] ⇒ l2 ≥ l1

Here belief stands for internal knowledge within the organization. Prices are considered low enough for a product P at time t in case the price for the product is equal or below the average price level within the product group.

OP4(P, t): Price low enough
∀G:PRODUCT_GROUP, R1,R2:REAL
 [state(γ, t) |= belief(belongs_to_product_group(P, G), pos) &
 state(γ, t) |= observation_result(average_price(G, R1), pos) &
 state(γ, t) |= belief(price(P, R2), pos)] ⇒ R2 ≤ R1

Whether the price is low enough depends on the cost price for the particular product P at time t.

OP5(P, t): Cost price low enough
∀G:PRODUCT_GROUP, R1,R2:REAL
 [state(γ, t) |= belief(belongs_to_product_group(P, G), pos) &
 state(γ, t) |= observation_result(average_cost_price(G, R1), pos) &
 state(γ, t) |= belief(cost_price(P, R2), pos) ⇒ R2 ≤ R1

Finally, the group properties can be specified according to the departments within the organization that make up the cost price for the production of product P at time t. First, property GP1 specifies that the design cost for product P should be low enough:

GP1(P, t): Design costs low enough
∀G:PRODUCT_GROUP, R1,R2:REAL
 [state(γ, t) |= belief(belongs_to_product_group(P, G), pos) &
 state(γ, t) |= observation_result(average_design_cost (G, R1), pos) &
 state(γ, t) |= belief(design_cost(P, R2), pos)] ⇒ R2 ≤ R1

Furthermore, the production costs for product P should be low enough as well:

GP2(P, t): Production costs low enough
∀G:PRODUCT_GROUP, R1,R2:REAL
 [state(γ, t) |= belief(belongs_to_product_group(P, G), pos) &
 state(γ, t) |= observation_result(average_production_cost (G, R1), pos) &
 state(γ, t) |= belief(production_cost(P, R2), pos)] ⇒ R2 ≤ R1

Finally, quality repair costs should be low enough for product P:

GP3(P, t): Quality repair costs low enough
∀G:PRODUCT_GROUP, R1,R2:REAL
[state(γ, t) |= belief(belongs_to_product_group(P, G), pos) &
state(γ, t) |= observation_result(average_quality_repair_cost (G, R1), pos)
& state(γ, t) |= belief(quality_repair_cost(P, R2), pos)] ⇒ R2 ≤ R1

A MODEL TO OBTAIN ORGANIZATIONAL CHANGE

In the previous sections organizational models and requirements specifications have been addressed. This section addresses the process underlying organizational change. Such a process comes into play in case certain organizational goals G (e.g., sufficient demand) are not reached any longer, due to an environmental change, say from E1 to E2. To localize why the new environmental property does not imply the organizational goal G to be met (i.e., why the new requirement R2 is not fulfilled) the property hierarchy as presented in the previous section is used (e.g., the property design costs low enough not being satisfied). After the cause has been determined, a process can be initiated to change the organization in such a way that this property is satisfied. Example rules are shown below, specifying the introduction of lean production. The rules have been specified using the *leadsto* format which is of the form α →$_{e,f,g,h}$β that states that if α holds for duration g then β will holds for duration h with a delay between e and f. Not that the state(γ, t) is omitted in *leadsto* properties. Agents that were allocated to the roles in the production process that just were deleted are allocated to the newly formed roles. Agents formerly allocated to a role in quality repair are fired. Once the system is organized in this fashion, quality repair in a separate department becomes obsolete, and quality repair cost is down to 0 as the production workers are now performing the task. In fact, production is even higher (Womack *et al.*, 1991).

CP1(P, D, t): Lean Production Structure
∀I,I2:INTEGER, A:AGENT, R:ROLE
[observation_result(customer_demand(P, I), pos) & I ≥ D &
 observation_result(customer_demand(P, I2), pos) & I2 < D &
 belief(allocated_to(A, R, quality_repair), pos)

→$_{0,0,1,1}$
[belief(exists_role(spec_production_worker), neg) &
 belief(exists_group(quality_repair_group), neg) &
 belief(exists_role(multi_task_production_worker), pos) &
 belief(allocated_to(A, R, quality_repair), neg) &
 belief(allocated_to(A, multi_task_production_worker, production_group),
  pos)]

Besides changes in the structure of the organization, changes also occur in the behavior. The change rule presented below specifies the behavior of the roles that have been added to the organization. Such behavior is expressed as an executable dynamic property in the same *leadsto* format as specified before, hence *leadsto* expressions can contain *leadsto* expressions. The behavior involves the immediate reporting of errors as well as the correction of errors.

CP2(P, D, t): Lean Production Behavior
∀I,I2:INTEGER
[observation_result(customer_demand(P, I), pos) & I ≥ D &
 observation_result(customer_demand(P, I2), pos) & I2 < D

→$_{0,0,1,1}$
 belief(role_property(d1, multi_task_production_worker,
  production_group), pos) &
 belief(has_expr(d1, leadsto(err, report_error, efgh(0,0,1,1))), pos) &
 belief(role_property(d2, multi_task_production_worker,
  production_group), pos) &
 belief(has_expr(d2, leadsto(and(report_error, responsibble_for_err),
  correct_err, efgh(0,0,1,1))), pos)

The following property states that in case the belief concerning the new behavior exists, it will show in the organization as well:

CP3: Show Behavior
∀D:IDENTIFIER, R:ROLE, G:GROUP, A:ANTECEDENT,
  C:CONSEQUENT, E,F,G,H: REAL
[belief(role_property(D, R,G), pos) &
 belief(has_expr(D, leadsto(A, C, efgh(E,F,G,H))), pos) &A]

$\twoheadrightarrow_{E,F,G,H}$ C

SIMULATION OF AN EXAMPLE ORGANIZATIONAL CHANGE PROCESS

This section presents a simulation which is performed by means of a case study from (Womack *et al.* 1991). First, the case study itself is introduced, after which the simulation results are presented.

The Ford Case

The case study involves the Ford Motor Company who has been one of the leading car manufacturers since the introduction of mass production in 1913. Many other car manufacturing companies adopted the ideas of mass production, particularly in the United States and Europe. Mass production has several characteristics. First of all, production workers have specialized jobs, such as attaching one small item to the car. Secondly, product quality is relatively bad; in case a misfit of a part is observed on the conveyor belt, the error is not corrected immediately, but this is done in a quality repair location after production has finished. The size of the quality repair department depends on the frequency of errors. Furthermore to guarantee production continuation, stocks of parts need to be kept in special warehouses. Finally, a relatively large work force is present to deal with work not directly related to production such as cleaning.

In 1980 the Ford Motor Company suffered a major crisis. The company began to loose vast amounts of money and large chunks of market share. A *monitor* within the Ford organization, having the property hierarchy of section called requirements refinement based on interlevel relations in mind, was triggered by the following values. Over the last three months demand for Ford cars was on average 500,000 cars, whereas in the current month the amount decreased to 432,000, therefore, the property OP1:*sufficient demand* was not satisfied. Simulation results of the resulting processes within the Ford organization are presented in the next section.

Simulation results

Figure 5 shows a graphical representation of the most interesting part of the trigger and change process. The left side shows the relevant atoms, the right part represents a time-line indicating when an atom is true (dark box) or false (lighter box). The different phases within the change process are addressed below.

Figure 5 shows that during the first time-periods, demand for Ford cars and other manufacturers is stable: each car manufacturer sells 500,000 cars:

observation_result(customer_demand(ford, 500000), pos)

Hence, the goal OP1 for Ford to maintain sufficient demand is satisfied by the current organization because it satisfies requirement R1 which implies goal OP1 from the current environment E1. At time point 4 however, demand for Ford cars drops:

observation_result(customer_demand(ford, 432692), pos)

The organization is now triggered and derives that goal OP1 to maintain sufficient customer demand is not satisfied:

belief(OP1_sufficient_car_demand, neg)

Apparently, the current organization is no longer sufficient to guarantee satisfaction. This is due to the fact that the environment E1 is no longer the case, and hence, R1 no longer guarantees satisfaction of the goal OP1.An analysis is performed to pinpoint the exact reasons of this failure. The tree such as presented in the section called requirements refinement based on interlevel relations is used for this purpose. Based upon the demand for other car brands, the organization derives that the market is not going down:

belief(OP3_cyclic_market_not_going_down, pos)

The organization can therefore observe (and also conclude) that the price is not low enough, which is ultimately observed to be caused by quality repair cost not being low enough:

belief(GP3_quality_repair_cost_low _enough, neg)

Now that the exact reason for goal OP1 not being satisfied is determined, a new requirement R2 is set which implies the goal OP1 from the new environment E2. To enable satisfaction of this property, the organization needs to be changed. Using the properties specified in the section called a model to obtain organizational change, introduce lean production within the organization is chosen to enable fulfillment of requirement R2:

belief(lean_production_method(ford), pos)

As a result, the organization is changed according to the rules for lean production, this shows in the trace by deletion of the quality repair group:

belief(exists_group(quality_repair_ group), neg)

It is also shown by the introduction of new roles called the multi-task team production worker:

belief(role_belongs_to_group(multi_task_
                    prod_worker, production_group), pos)

These roles have specific dynamic properties concerning the reporting and correcting of errors in the production process:

belief(role_property(d1, multi_task_prod_worker,
                    production_group), pos)
belief(role_property(d2, multi_task_prod_worker,
                    production_group), pos)
belief(has_expr(d1, leadsto(err, report_error, efgh(0,0,1,1)), pos)
belief(has_expr(d2, leadsto(and(report_error,
        responsibble_for_err), correct_err, efgh(0,0,1,1))), pos)

The first property states that each multi-task production worker should, when observing an error, report this error at once. The second property specifies that if an error is reported, then the roles responsible for that error should correct this error at once. After all of these changes have been implemented, quality repair cost of Ford are down to 0 and demand for Ford cars is back to the old value again. Hence, the new organization
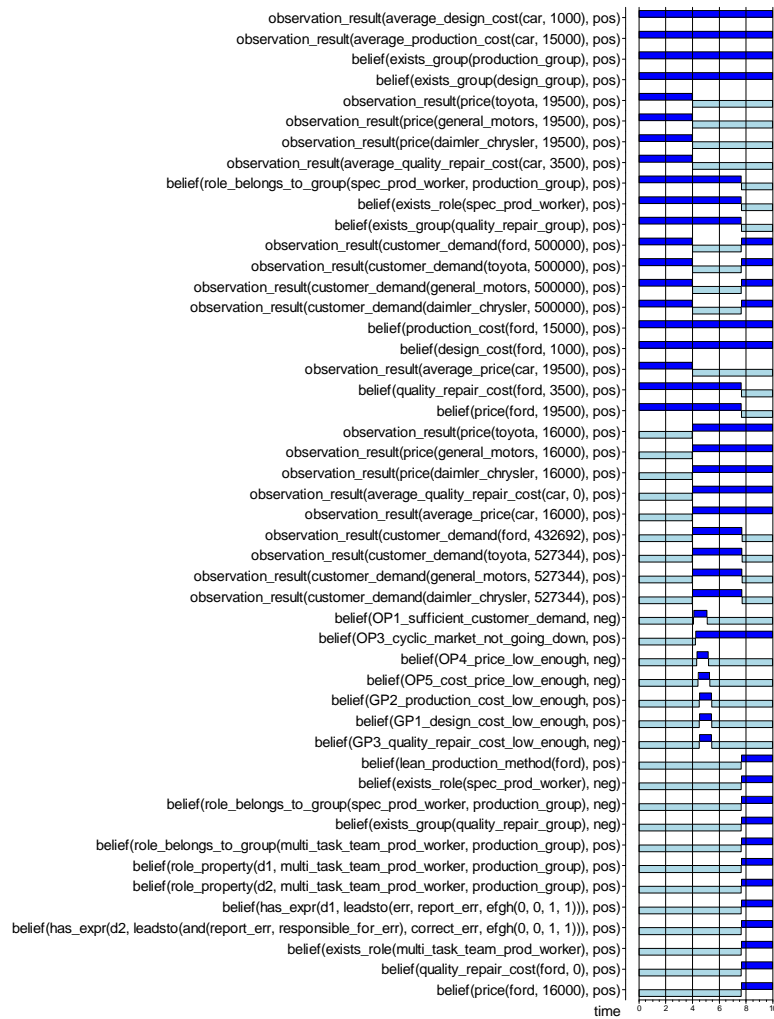
Figure 5. Simulation results for the case study

VERIFICATION

To check whether the change as performed in the Ford case study from the previous section indeed complies to the properties that have been specified in the sections before the case study, this section performs a verification of these properties against the trace presented in the previous section. The verification has been performed using a software tool called *TTL Checker*.

The results of the checks of the first set of properties, those from Section 3 and 4, are already visible in the trace itself since the Monitor actually checks the satisfaction of these properties in the simulation. Some brief results of the checks of these properties against the trace (contrary to the monitoring being performed within the simulation) are described here. First, property E2(ford, 19500, t) which specifies the environmental level of the price for the product group cars, is checked against the trace. This environmental property holds during the interval [0,4]. The goal property OP1(ford,

500000, t) that defines the goal of sufficient demand for Ford holds during the interval [0,4] as well. After time point 4, the environment changes, and OP1(ford, 500000, t) does not hold any longer. E2(ford, 16000, t) is now the new environment in which the organization is participating. The property is satisfied during the interval [4,10]. After time point 7.7, the new requirement forced by the environment to regain the goal OP1(ford, 500000, t) is satisfied: R2(ford, 16000, 500000) resulting in satisfaction of OP1 again from this time point on.

DISCUSSION

Organizational change mechanisms can involve not only quantitative numerical aspects but also qualitative, logical aspects (for example, a role switch between agents within an organization). If formalization is used for organizational change, this is often based on mathematical models using differential equations. In contrast, agent-based simulation models traditionally make use of qualitative, logical languages. Most of these languages are appropriate for expressing qualitative relations, but less suitable to work with more complex numerical structures as, for example, in differential

equations. Therefore, integrating such mathematical models within the design of (multi-)agent-based simulation models is difficult. To achieve this integration, this paper has presented an approach which bridges the gap between quantitative approaches and the type of languages typically used in agent-based simulation.

This paper focuses on external triggers for organizational change. Triggers are related to specific goals that play the role of design requirements which the organizational change should comply to. These requirements tend to be high-level goals and therefore lack the detail needed for specifying how an organization should change. Therefore, refinements are introduced in the form of hierarchies of requirements. Such hierarchies relate objectives of the organization (e.g., high demand for cars) to organizational change properties at different organizational levels (e.g., change in some departments). Thus, they relate triggers at the level of the organization to properties at the level of parts (groups) within the organization. For example, the cause of why a certain type of car is not selling according to the goals that have been set is related to the costs of quality repair. Requirements hierarchies help to localize where to change the organization. Furthermore, change properties have been introduced originating from Organization Theory (Womack *et al.* 1991), which can be used for specific cases in which an organization is not reaching it goals.

Using the lowest level, executable properties, simulations are performed and by means of formal verification approaches the simulation is shown to have satisfied all desired properties.

REFERENCES

Bosse, T., Jonker, C.M., and Treur, J., Analysis of Design Process Dynamics. In: R. Lopez de Mantaras, L. Saitta (eds.), Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'04 , 2004, pp. 293-297.

Ciancarini, P., Wooldridge, M. (eds.), Agent-Oriented Software Engineering, Lecture Notes in Computer Science, vol. 1957, Springer-Verlag, Berlin, 2001.

Ferber, J. and Gutknecht, O., A meta-model for the analysis and design of organisations in multi-agent systems. In: Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98), IEEE Computer Society Press, pp. 128-135.

Hannoun, M., Sichman, J.S., Boissier, O., and Sayettat, C., Dependence Relations between Roles in a Multi-Agent System: Towards the Detection of Inconsistencies in Organization. In: J.S. Sichman, R. Conte, and N. Gilbert (eds.), Multi-Agent Systems and Agent-Based Simulation, Lecture Notes in Artificial Intelligence, vol. 1534, Springer-Verlag, 1998, pp. 169-182.

Hannoun, M., Boissier, O., Sichman, J.S., and Sayettat, C., MOISE: An organizational model for multi-agent systems. In: M. C. Monard and J. S. Sichman (eds.), Advances in Artificial Intelligence, LNAI, vol. 1952, Springer-Verlag, Berlin, 2000, pp. 152-161.

Hubner, J.F., Sichman, J.S., and Boissier, O., A Model for the Structural, Functional and Deontic Specification of Organizations in Multiagent Systems. In: Proc. 16th Brazilian Symposium on Artificial Intelligence (SBIA'02), Porto de Galinhas, Brasil, 2002.

Jaffee, D., Organization Theory: Tension and Change, McGraw-Hill Publishers, New York, 2001.

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

Jonker, C.M., and Treur, J., Relating Structure and Dynamics in an Organisation Model. In: J.S. Sichman, F. Bousquet, and P. Davidson (eds.), Multi-Agent-Based Simulation II, Proc. of the Third Int. Workshop on Multi-Agent Based Simulation, MABS'02. Lecture Notes in AI, vol. 2581, Springer Verlag, 2003, pp. 50-69.

Lomi, A., and Larsen, E.R.. Dynamics of Organizations: Computational Modeling and Organization Theories, AAAI Press, Menlo Park, 2001.

Womack, J.P., Jones, D.T., and Roos, D., The Machine That Changed The World: The Story of Lean Production, HarperCollins Publishers, New York, 1991.

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, such as 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 modelling multi-agent systems and organisations and her work on the analysis and modelling of the dynamics of behaviour of complex systems. These research topics now contribute to the research programme of the Cognitive Artificial Intelligence division of the Nijmegen Institute for Cognition and Information.

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 programme on component-based design of knowledge-based and agent systems. In the last five years the research programme focussed on modelling dynamics of agent systems in practical application areas, and related to other disciplines such as Biology, Cognitive Science, Organisation Theory, and Philosophy of Mind.