Redesign of Organizations as a Basis for Organizational Change

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

Abstract. Within AI, in relation to application areas such as architecture, engineering and software design, (formal) design models and supporting tools have been developed. This paper explores in how far such design models can be applied to the area of organization (re)design. It therefore presents a component-based model for (re)design of organizations, as a specialization of an existing generic design model. Based on formalizations within Organization Theory that recently have been developed, it is shown how to formally describe organization models as design object descriptions, and how to specify organization goals as design requirements. Moreover, it is shown how a formal design process description can be obtained to model the redesign process for an organization that adapts to changes in the environment. The formally specified and implemented approach to organization redesign thus obtained has been tested for a well-known historical case study from the Organization Theory literature.

1 INTRODUCTION

To smoothen processes in society, often specific forms of organization are used. Organizations usually have goals to be achieved or maintained that serve as requirements for their functioning. They are created according to certain organizational structures which define elements or parts of the organization and how these are connected. The idea is that for these elements and parts certain behaviors occur that interact with each other so that the resulting overall organization behavior fulfills its goals. How to achieve a goal may depend on the environment of the organization, which often is dynamic, therefore the environmental circumstances impose requirements to the organization which change over time. To adapt to such changes in requirements, often the organization has to change itself. This re-organization problem, to find a changed organization form that fulfills the new requirements, can be considered a (re)design process.

Within the area of AI and Design, in the last decade formally specified generic models for (re)design processes have been developed; e.g., [1, 3]. To apply such a generic model for redesign to the area of organizations, specialized types of knowledge are required (in formalized form) which include: (1) knowledge concerning the goals of an organization; (2) knowledge on how to derive refined requirements from such goals given a variable environment; (3) knowledge on what the current design object description is, and (4) knowledge on what components for a design object satisfy which requirements. Output of a redesign process is a new design object description as a modification of the existing one and a specification of changed (new) design requirements.

The redesign process as formally modeled in [3] involves generation and modification steps for the specification of the requirement set and for the design object description. Within a formal model of a redesign process, formalizations are needed of design objects, design requirements, and of the dynamics of redesign processes. In this paper, for the area of organizational (re)design, formalizations of these aspects are proposed in the context of a component based model for (re)design of organizations. Formalized organization models [4,7,8,10,12] are used for design object descriptions. Furthermore, formalizations of organizational behavior are used for design requirements specifications [7,8,10,12]. Finally, for design process dynamics a formalization is used as put forward in [1]. Based on these ingredients, this paper introduces a formalized modeling approach to organization redesign. Such a formal approach contributes to the domain of organization redesign in that it facilitates formal modeling, simulation and verification of the redesign process. The approach is supported by tools to model and analyze such redesign processes.

Section 2 gives the components based model for the design and redesign process and describes the types of domain specific knowledge needed in such a process. Section 3 addresses the formalization of design object descriptions by means of an organization model format in which different components and aggregation levels can be distinguished. In Section 4 the relation between goals, a changing environment and requirements is described, including example cases described in Organization Theory. Section 5 presents the method of refinement of such requirements and shows a specific example. Thereafter, Section 6 presents examples of design object that are known to satisfy certain design requirements, and Section 7 presents generic properties which enable an evaluation of the successfulness of the whole (re)design process. Section 8 presents simulation results of the model, and finally Section 9 is a discussion.

2 A COMPONENT-BASED MODEL FOR (RE)DESIGN OF ORGANIZATIONS

This Section presents a component-based generic model for design of organizations based on requirements manipulation and design object description manipulation. The component-based model presented draws inspiration from [3] and was specified within the DESIRE [2] framework. The model for design is composed of three components:

- RQSM, which stands for Requirement Qualification Set Manipulation. Such requirements are for example acquired by elicitation in cooperation with managers within a company. Within RQSM the appropriate requirements are determined in relation to the goals set for the organization and the current environmental conditions. After having selected a set of requirements, these are refined to more specific ones.
- DODM, for Design Object Description Manipulation, creates a design object description based on the (specific) requirements received from RQSM. In order to determine such a design object description, a number of alternative solutions known to satisfy the requirements are generated and

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

[2] Radboud University Nijmegen, Nijmegen Institute for Cognition and Information, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands, email: C.Jonker@nici.ru.nl

according to certain strategic knowledge one of those is selected.

- Design Process Coordination (DPC) is the coordinating component for the design process. The component determines the global design strategy (e.g., [3]) and can evaluate whether the design process is proceeding according to plan.

Figure 1 shows the top level composition within the component-based design model. The figure shows the three main components and links between them that express information flow. Note that the small box at the left hand side refers to the input of a component whereas the one at the right hand side refers to the output.

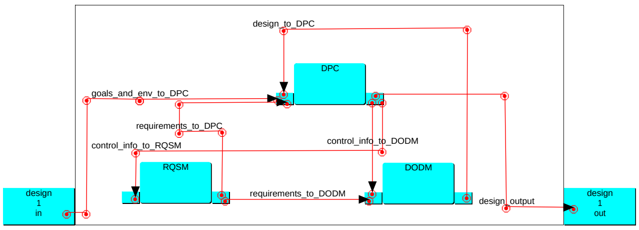

Figure 1. Top level of the design model

Next, each of the three main components is treated in more detail. As the model is a generic design model for organizational design, no application- or domain-specific knowledge is included in the model itself, but the model has slots where the different types of knowledge needed can be added. In later sections each of these types of knowledge is specified for a case study which has been performed based on literature from Organization Theory.

2.1 RQSM

The component RQSM is composed from two sub-components, namely Requirements Sets Generation and Requirements Set Selection. This is shown in Figure 2.

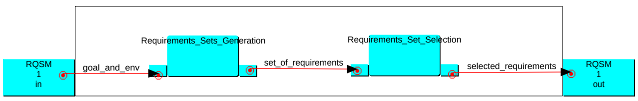

Figure 2. Components within RQSM

The component Requirements Sets Generation receives as an input the current environmental conditions. The sub-component contains knowledge on what requirements entail fulfillment of the organizational goal given a certain environmental conditions that holds. Such knowledge can be depicted in the form of an AND/OR tree as shown in Figure 3.

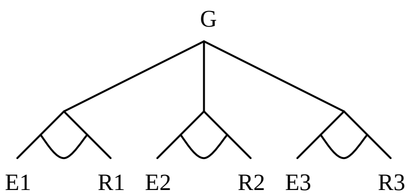

Figure 3. Example AND/OR tree relating environmental conditions and requirements to a goal

If for example E1 is observed, requirement R1 is an example of a requirement that, when it is fulfilled by the organization being designed, guarantees to satisfy goal G under environmental conditions E1. If the environment however changes to a situation described by E2, the requirement has to change as well; the example tree shows how R1 can be changed to requirement R2 that guarantees G under the new environmental conditions E2. After a requirement is determined, it can be refined in order to obtain requirements on a more specific level. Making such a requirement more specific can result in several options being generated. For example, it might be possible to establish a certain market share by having the best quality products but also by having the lowest priced products. After having refined each of the requirements, all possible sets of refined requirements are forwarded to the component Requirements Set Selection.

After the component Requirements Set Selection has received the alternative sets of requirements its task is to select one of those alternatives, and to forward it to the component DODM which will in turn find a suitable organization design for such a requirement set. Selection of one alternative can be performed by means of several methods. Sometimes explicit ranking of the different alternatives is available, but this is however not frequently the case. More in general, strategic knowledge is required which enables selection of the best alternative. Such strategic knowledge can for example be based on the source of requirements: requirements that originate from users can for example be preferred over those derived by default rules which are in turn preferred over requirements derived from previous requirements (see [9]).

2.2 DODM

DODM receives input in the form of a set of refined requirements from RQSM, which is handled by again two sub-components, Design Object Description Generation and Design Object Description Selection.

Design Object Description Generation receives input in the form of requirements and delivers as output descriptions for possible alternative design objects of the organization, such that the (specific) requirements as received from RQSM are satisfied. In order to be able to relate a requirement to an appropriate design of an organization, knowledge is needed that specifies what possible part of a design object within a design contributes to fulfillment of a certain specific requirement. If, for example, the requirement is set that products of the highest quality should be produced, then an example design which fulfills such a requirement is an organization in which there is a department dedicated to checking the quality of products and repairing of production errors. Again, there can be many possibilities available which result in satisfaction of the requirements. All these alternatives are forwarded to the component Design Object Description Selection.

In order for the component Design Object Description Selection to choose the optimal design several criteria can be used. One organization design might for example be less expensive in operation than another design, or a design might be known in general to work better than another design. In order to make such a selection, the component has (strategic) knowledge concerning these aspects, it might for example know the typical price for hiring an agent for a particular role, etc. Eventually, the component outputs a new design for the organization.

2.3 DPC

The component DPC is the component which determines the global design strategy and oversees whether the design process proceeds according to plan. Two different tasks are therefore distinguished. First of all, DPC checks whether a design object description determined by DODM satisfies the refined requirements. It might for example be the case that the combination of two suitable design object parts causes a conflict. In case the refined requirements are not satisfied control information is passed to DODM stating that an alternative should be found (e.g., taking a different branch of an OR tree). In case these refined requirements are satisfied whereas the high-level requirements are not, the requirements refining process has failed, therefore control information is given to RQSM to refine the requirements in another way (again by for example taking another OR branch).

3 ORGANIZATION MODELS AS DESIGN OBJECTS

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. [13]. In the introduction to their book Lomi and Larsen [14] 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 [14] two levels are mentioned: the level of the organization as a whole versus the level of the units. Also in the development of MOISE [7,8,10] 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 [6] 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 4, (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 4. Solid arrows denote transfer between roles within a group; dashed lines denote inter-group interactions. This format will be adopted to formalise organization models as design object descriptions. In addition, behavioral properties of elements of an organization are part of a design object description. The format to express these will be shown in Section 4, which addresses design requirements that can be formulated for organizations.

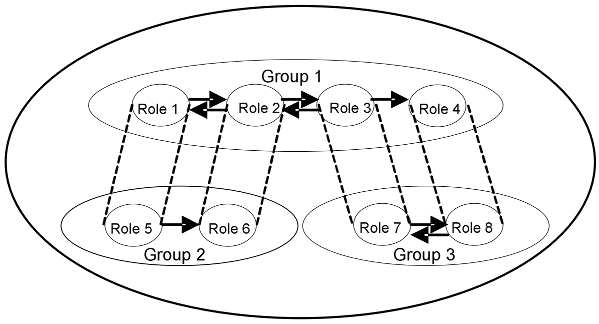

Figure 4. An AGR Organization Structure

4 RQSM: CHANING REQUIREMENTS UPON ENVIRONMENTAL CHANGE

Requirements an organization needs to satisfy change due to changing environmental circumstances: requirements changes are caused by triggers external to the organization which occurs in RQSM. The general pattern 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 ⇒ G. Here R1 is a requirement expressing a 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 to be replaced by a new, stronger requirement R2 which expresses that under environment E2 goal G can be achieved; therefore: E2 & R2 ⇒ G. Thus, an organization is triggered to change to fulfill R2 and as a consequence fulfill goal G again.

Jaffee [11] distinguishes several of these external triggers for organizational change. They can be classified in triggers in the organization's *input*, (e.g., changes in the resources or suppliers), and triggers in *enabling / constraining factors* such as government/labor rules and (new) technology. 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. Examples of input triggers are *resources* that run out, becoming a lot more expensive, c*ustomers* whose demands decrease for the good being produced, and *competitors* changing their production methods causing more efficient production for products within the same product group. Another example of an input-base external trigger is the case that at time t *suppliers* increase their price of a product P, which is used by the organization for the production, from M_1 to M_2. A formal form of this environmental condition is specified in E1 using the Temporal Trace Language (TTL) [12]. In the definition of this property state(γ, t) |= environmental_condition(price(P, R), pos) denotes that within the state state(γ, t) at time point t in trace γ the state property environmental_condition(price(P, R), pos) holds, formalized as infix predicate |= .

E1(P, M, t): Supplier Price

∃R:REAL state(γ, t) |= environmental_condition(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 the demand of product P above a threshold D. A formal specification of the goal is presented in OP1.

OP1(P, D, t): Sufficient demand

∃I:INTEGER state(γ, t) |= environmental_condition(customer_demand(P, I), pos) & I ≥ D

The requirement imposed for the organization is to maintain the goal of keeping demand for product P2 above D, in the new situation given the environmental condition of the price M for product P1 which is needed for the production of P2. This requirement is specified below in property R.

R(P1, P2, M, D): Maintain demand

∀t :TIME
[state(γ, t) |= needed_for_production_of(P1, P2) & E1(P1, M, t)] ⇒ OP1(P2, D, t)

Before the change in the environment, requirement R1 which is R(P1, P2, M_1, D) was sufficient to ensure the goal being reached. After the change however, this requirement is still satisfied but might be insufficient to ensure the goal. This is due to the fact that the environmental condition E1 in the antecedent of E1 & R1 ⇒ G does not hold, and hence, cannot be used to entail G (although the requirement R1 is fulfilled all the time). The requirement is therefore withdrawn and replaced by the requirement R2 which is R(P1, P2, M_2, D). This R2, however, is not necessarily satisfied and may require an organizational change to enable fulfillment.

5 RQSM: REFINING REQUIREMENTS BASED ON INTERLEVEL RELATIONS

To fulfill requirements at the level of the organization as a whole as discussed in Section 4, parts of the organization need to behave adequately (see also the central challenges put forward by Lomi and Larsen [14] as discussed in Section 2). 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 (the design object description determined in DODM) relates to (mathematically defined) relationships between these sets of dynamic properties for the different elements and aggregation levels within an organization (cf. [13]). 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 [13], 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.

Figure 5 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 [17]. The overall organizational goal is to maintain sufficient demand for the goods being produced, as was also the case in OP1 in Section 4. 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 at the aggregation level of the organization as a whole, whereas the lowest level shown here represents properties at the aggregation level of the groups. Note that the fact that these are group properties already restricts the design of the object in DODM, which makes the process less complex.

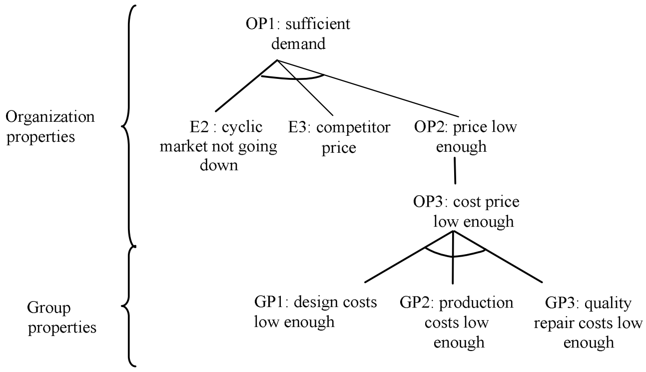

Figure 5. Hierarchy of Organizational and Group properties

A definition for each of the properties in Figure 5 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 Section 4. Furthermore, an environmental condition is that 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.

E2(P, t): Cyclic market not going down

∀G:PRODUCT_GROUP, I1,I2:INTEGER
[state(γ, t) |= belongs_to_product_group(P, G) &
state(γ, (t-1)) |= environmental_condition(customer_demand(G, I1), pos) &
state(γ, t) |= environmental_condition(customer_demand(G, I2), pos)] ⇒
I2 ≥ I1

Furthermore, an environmental condition E3 poses a requirement on the price of competitors in the form of the average price of products within the product group to which product P belongs. These prices should not be higher than V:

E3(P, V, t): Competitor Price

∀G:PRODUCT_GROUP, V1:REAL
[state(γ, t) |= belongs_to_product_group(P, G) &
state(γ, t) |= environmental_condition(average_price(G,V1), pos) &
V1 ≥ V]

In order to achieve the goal OP1 given environmental conditions E2 and E3, the price of the products being produced by the organization should be low enough, which is the requirement posed on 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 (i.e. prices are ≤ V as set above).

OP2(P, V, t): Price low enough

∀G:PRODUCT_GROUP, V1:REAL
[state(γ, t) |= price(P, V1)] ⇒ V1 ≤ V

Whether the price is low enough depends on the cost price for the particular product P at time t, which purely depends on the costs for the different groups within the organization, as expressed in the group properties (GP's)

OP3(P, V, t): Cost price low enough

∀V1,V2,V3:REAL
[state(γ, t) |= design_cost(P, V1) &
state(γ, t) |= production_cost(P, V2) &
state(γ, t) |= quality_repair_cost(P, V3)]
⇒ V1+V2+V3 ≤ V

Finally, the group properties can be individually specified such that the cost of such a group are below a certain value. Note that it is a refinement choice how to divide such cost, it could for example be the case that it is decided to allow only a small percentage of the cost for quality repair whereas production and design cost are allowed equal cost. Individually, each group should meet such requirements. First of all, design costs should be low enough:

GP1(P, V1, t): Design costs low enough
∀Q:REAL
[state(γ, t) |= design_cost(P, Q)] ⇒ Q ≤ V1

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

GP2(P, V2, t): Production costs low enough
∀Q:REAL
[state(γ, t) |= production_cost(P, Q)] ⇒ Q ≤ V2

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

GP3(P, V3, t): Quality repair costs low enough
∀Q:REAL
[state(γ, t) |= quality_repair_cost(P, Q)] ⇒ Q ≤ V3

After having generated all options in RQSM, selection knowledge is used for selecting one of the available options. In this paper, such selection knowledge is not further addressed. The output of RQSM is however of the form selected_basic_refinement_set(RS) where RS is a name for a requirements set. The elements within this set are defined as follows: in_selected_basic_refinement_set(R, RS) where R is a requirement, as the ones shown above, and RS is the selected basic refinement set.

6 DODM: CONSTRUCTING DESIGN OBJECTS

As already stated in the introduction of the model in Section 2, DODM contains a library of templates for (parts of) design objects which are known to satisfy certain specific requirements (of the form as specified in the last paragraph of the previous section). In this specific case, two of such templates are within the library of DODM. First of all, a template in which mass production is the system used to produce certain goods. Such a system is known to produce goods at a reasonable production cost but at a high quality repair cost. The template for mass production includes a production worker group, in which specific production workers are present (e.g. attaching a wheel to a car). Furthermore, a quality repair department of considerable size is present with quality repair worker roles.

Another template for a design object within the library is an organization in which lean production is the production system in use. Lean production has a quality repair cost of 0, since there is no separate quality repair department. The production costs are at the same level as the production costs for mass production organizations. In the lean production method (see e.g. [17]), multi-task production workers are present which perform several tasks, and also handle errors in case they are observed. As a result of such immediate error detection and correction, a quality repair department is not present within a lean production model.

Figure 6 shows an example AND/OR tree for DODM (focusing at lean production as a solution) in which options for changes in a design object not satisfying the requirement that design costs are low enough. The specific changes in the design object are presented below. First of all, the highest level property states that design costs will at least at the required level within a duration d:

CP1(P, D, t):Lower Quality Repair Costs
∀V1,V2:REAL
[state(γ, t) |= selected_basic_requirement_in(GP3(P, V1, t), RS) &
state(γ, t) |= DOD_includes(D, quality_repair_cost(P, V2)) & V1 < V2]
⇒ ∃t2:TIME > t, V3:REAL
[t2 < t + d & state(γ, t2) |= DOD_includes(D, quality_repair _cost(P, V3))
& V3 ≤ V1]

On a lower level, property CP2(P, D, t) specifies the introduction of lean production into an organization. This reduces the quality

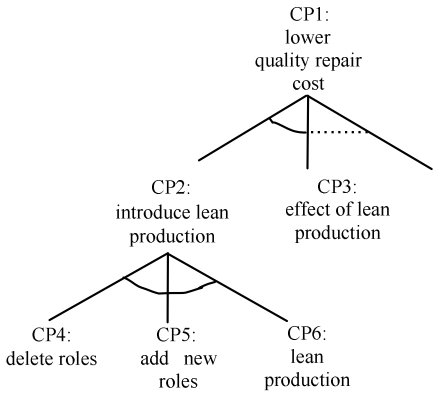

Figure 6. Redesign options specified in the form of an AND/OR tree

repair cost to 0 as shown by CP3(P, D, t). Note that more options are possible for reducing quality repair cost, shown by the dotted line in the figure, these are however not addressed in this paper.

CP2(P, D, t): Introduce Lean Production
∀V1,V2:REAL
[state(γ, t) |= selected_basic_requirement_in(GP3(P, V1, t), RS) &
state(γ, t) |= DOD_includes(D, design_cost(P, R2)) & V1 < V2]
⇒ ∃t2:TIME > t
[t2 < t + d &
state(γ, t2) |= DOD_includes(D, lean_production_method(P))]

CP3(P, D, t): Effect of Lean Production
[state(γ, t) |= DOD_includes(D, lean_production_method(P))
⇒ state(γ, t) |= DOD_includes(D, quality_repair_cost(P, 0))]

Introducing a lean production system entails that within the production process the specialized roles for mass-production and quality repair department are deleted.

CP4(P, D, t): Delete Roles
∀R1,R2:REAL
[state(γ, t) |= DOD_includes(D, lean_production_method(P))
⇒ ∃t2:TIME > t [t2 < t + d &
state(γ, t2) |= ¬ DOD_includes(D, exists_role(spec_production_worker))) &
state(γ, t2) |= ¬ DOD_includes(D, exists_group(quality_repair_group))]

Moreover, roles are created that perform multiple tasks, and teams are created such that the roles combined in the team have all the abilities to make a car.

CP5(P, D, t): Add New Roles
∀R1,R2:REAL
[state(γ, t) |= DOD_includes(D, lean_production_method(P))
⇒ ∃t2:TIME > t, ∀A:AGENT, R:ROLE [t2 < t + d &
state(γ, t2) |= DOD_includes(D, exists_role(multi_task_production_worker)) &
state(γ, t2) |= DOD_includes(D, previously_allocated_to(A, R, quality_repair)) &
state(γ, t2) |= DOD_includes(D, allocated_to(A, multi_task_production_worker,
production_group))]

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. CP6 expresses that the measures as described in CP4 and CP5 results in a lean production method for the product P:

CP6(P, D, t): Lean Production
∀A:AGENT, R:ROLE
[state(γ, t) |= ¬ DOD_includes(D, exists_role(spec_production_worker)) &
state(γ, t) |= ¬ DOD_includes(D, exists_group(quality_repair_group)) &
state(γ, t) |= DOD_includes(D, exists_role(multi_task_production_worker)) &
state(γ, t) |= DOD_includes(D, previously_allocated_to(A, R, quality_repair)) &
state(γ, t) |= DOD_includes(D, allocated_to(A,
multi_task_production_worker, production_group))]

⇒ ∃t2:TIME < t + d state(γ, t2) |= DOD_includes(D, lean_production_method(P))

After such options for (re)design of the object have been generated based on the requirements, selection knowledge is used to select one of the options that have been generated. Again, such selection knowledge is not addressed in this paper. Eventually, DODM outputs a design object description of the form selected_DOD_output(D) where D is the design object description. Furthermore to identify properties of the DOD or its parts, output of the form in_selected_DOD_output(P,D) is generated where P is a property of (a part of) the DOD and D is the selected DOD. This is based on the internal information represented in the form of DOD_includes(D, P).

7 (RE)DESIGN PROCESS EVALUATION

This Section addresses the evaluation of the whole design process. Such a design process is successful when both RQSM and DODM show the proper behavior.

RQSM shows the proper behavior in case it first of all generates requirements, and secondly these requirements indeed result in the goal set for the organization being met. Such properties are formulated in a formal form below.

RQSM_generate

If RQSM receives new environmental conditions on its input, then RQSM eventually generates a set of requirements

∀t:TIME, γ:TRACE, E:ENV_COND
state(γ, t, input(RQSM)) |= environment_property(E) & ¬∃t':TIME < t [state(γ, t', input(RQSM)) |= environment_property(E)]
⇒ ∃t2:TIME > t, G:GOAL, RS:REQUIREMENT_SET
[state(γ, t2, output(RQSM)) |= main_requirement(G) &
state(γ, t2, output(RQSM)) |= selected_basic_refinement_set(RS)]

RQSM_successful

If RQSM generates requirements, then the combination of these requirements entail the goal set for the organization.

∀t:TIME, γ:TRACE, RS :REQUIREMENT_SET, G :GOAL
[state(γ, t, output(RQSM)) |= main_requirement(G) &
state(γ, t, output(RQSM)) |= selected_basic_refinement_set(RS)]
⇒ entails_goal(RS, G)

DODM shows the proper behavior in case it first of all generates a design object description in case a new requirement set is received. Besides simply generating such a design object description, the object also needs to satisfy the requirements received on its input.

DODM_generate

If DODM receives a new requirements set on its input, then DODM eventually generates a design object description as output.

∀t:TIME, γ:TRACE, RS :REQUIREMENTS_SET
[state(γ, t, input(DODM)) |= selected_basic_refinement_set(RS) &
¬∃t':TIME < t [state(γ, t', input(DODM)) |= selected_basic_refinement_set(RS)]
⇒ ∃t2:TIME, D:DESIGN_OBJECT_DESCRIPTION
state(γ, t2, output(DODM)) |= selected_DOD_output(D)]

DODM_successful

If DODM generates a design object description as output, then the design object description satisfies the requirements set on the input of DODM.

∀t:TIME, γ:TRACE, R :REQUIREMENT_SET, D:DESIGN_OBJECT_DESCRIPTION
[state(γ, t, input(DODM)) |= selected_basic_refinement_set(R) &
state(γ, t, output(DODM)) |= selected_DOD_output(D)]
⇒ fulfills_requirements(D, R)

8 SIMULATION RESULTS

In order to show the functioning of the model presented above, simulation runs have been performed based on the properties as identified in Sections 4-6 using the component-based design presented in Section 2. As a scenario for the case study, a sudden decrease of competitor price is inserted as an event into the simulation (following [17]). Figure 7 shows a partial trace of the simulation results. In the figure, the left side shows the atoms that occur during the simulation whereas the right side shows a timeline where a dark gray box indicates an atom being true at that particular time point and a light gray box indicates the atom is false.

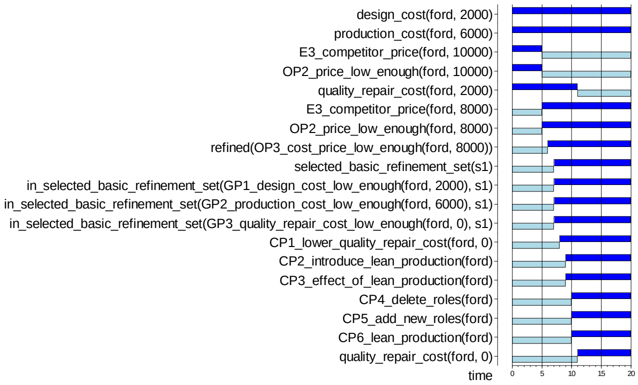

Figure 7. Case Study simulation results

The figure shows the following. Initially, the different cost factors for the ford design object are the following:

design_cost(ford, 2000)
production_cost(ford, 6000)
quality_repair_cost(ford, 2000)

This perfectly fulfills the requirement that price is considered to be low enough in case it is at most 10000 as expressed in OP2 at that time point:

OP2_price_low_enough(ford, 10000)

This requirement is sufficient to guarantee the goal OP1 (as expressed in Figure 5) due to the environmental condition E3 that competitor price for products within the same product group as ford are at that exact same level:

E3_competitor_price(ford, 10000)

And furthermore, the cyclic market should not be going down (E2) which is left constant during this simulation. Suddenly however, the environment changes, the price of competing cars drops to 8000:

E3_competitor_price(ford, 8000)

The current property OP2 is now insufficient to guarantee the overall goal OP1 being satisfied, therefore a redesign process is activated. RQSM determines a new requirement for the design object, namely that prices should be below 8000, the competitor car price:

OP2_price_low_enough(ford, 8000)

Other options might be possible as well, but are not addressed in the simulation. The requirement is refined, first of all by expressing that the cost price should be low enough:

refined(OP3_cost_price_low_enough(ford, 8000))

This results in a selected basic refinement that quality repair cost should become 0 whereas design and production cost can remain 2000 and 6000 respectively, as shown in the requirements part of the selected refinement s1:

in_selected_basic_refinement_set(GP1_design_cost_low_enough(ford, 2000), s1)
in_selected_basic_refinement_set(GP2_production_cost_low_enough(

ford, 6000), s1)
in_selected_basic_refinement_set(GP1_quality_repair_cost_low_enough(
ford, 0), s1)

Since these are basic refinements, they are passed to DODM in order to find templates appropriate for these basic requirements. DODM observes that quality repair costs for the current design object are too high, and therefore starts to use the tree as expressed in Figure 6, refining the exact changes to be performed on the design object more and more. First the introduction of the lean production system is chosen, as expressed in CP2. Thereafter, the more concrete changes are determined, namely the deletion of the specialized production worker roles, the addition of new multi-task roles, and the insertion of the new behavior of those roles:

CP4_delete_roles(ford)
CP5_add_new_roles(ford)
CP6_lean_production(ford)

Note that in the simulation the actual contents of such properties are more concrete (in the form of current DOD descriptions), however these are not presented here for the sake of brevity. Finally after the actual changes have been performed for the design object, quality repair costs drop to 0, and the goal is therefore satisfied again:

quality_repair_cost(ford, 0)

In order to see whether the properties as expressed in Section 7 hold for the simulation trace, first of all, the RQSM_generate and DODM_generate properties have been checked against the trace shown in Figure 7 using a software tool called the TTL Checker [12]. Both properties were shown to hold for the trace. In order to see whether the refinement process within RQSM is properly performed, the tree used for the simulation as presented before in Section 5 has been formally proven by means of the SMV model checker [15]. The result indeed show that the lowest level properties entail the goal given the environmental conditions. Furthermore, to prove the successfulness of DODM, the property hierarchy shown in Figure 6 has also been proven by the SMV model checker which shows that introducing lean production in a design object indeed results in a lowering quality repair cost to 0, which satisfied the property DODM_successful. As a result, the DODM_successful property is satisfied as well as the RQSM_successful property in case the components indeed generate the output based on these property hierarchies.

9 DISCUSSION

Organizations aim to behave in the best possible way to meet their organizational goals. An organization continuously monitors whether events occur that endanger fulfillment of these goals. Once such an event is observed, causing a certain organizational goal to become unreachable, this acts as a trigger for organizational change. In this paper it was shown how such a change process can be addressed as a (re)design process. Traditionally, within the area of AI and Design, most often the areas architecture, engineering and software design have been addressed as application areas. This paper explores how a component-based formal generic model for design developed within the area of AI and Design can be specialized into a model for organization (re)design.

Based on formalizations that recently have been developed within the area of Organization Theory and AI (or computational organization theory), it has been shown how to describe organization models as design object descriptions, and how to specify organization goals as design requirements. Furthermore, different types of specialized knowledge have been identified (1) about a main organization goal and how it relates for given environmental conditions to organization requirements, (2) about how the organization requirements can be refined, (3) about design object descriptions, and (4) about which components for a design object description satisfy which requirements. Moreover, it has been shown how a design process description can be obtained to model the redesign process for an organization. Together with the generic design model which includes slots for such types of knowledge, these types of knowledge constitute a specialized component-based model for (re)design of organizations. Example properties have been taken from a well known example in Organization Theory describing the introduction of lean production within an organization [17].

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, design requirement refinement is 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. Relating high-level goals for an organization as well as goals for organizational redesign to low-level executable properties. Formal verification has been performed and the results show satisfaction of the non-leaf properties in the property tree.

When comparing the approach to previous work in the redesign of organizations the main strength is the formal description of the whole redesign process. In the field of management for example, an overview of which can be found in [5], merely informal descriptions are given about redesign processes. In Systems Theory (see e.g. [16]) goal oriented behavior is addressed. The gap observed between the actual state of the system and the desired state causes redesign, which strokes with the approach taken in this paper. Formalizations by means of property hierarchies are however not present, therefore formal verification as done is this paper cannot be performed.

REFERENCES

[1] 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.

[2] Brazier, F.M.T., Jonker, C.M., and Treur, J., Principles of Component-Based Design of Intelligent Agents. Data and Knowledge Engineering, vol. 41, 2002, pp. 1-28.

[3] Brazier, F.M.T., Langen, P.H.G. van, and Treur, J., Strategic knowledge in design: a compositional approach, Knowledge-Based Systems 11:405-415, 1998.

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

[5] Douglas, C., Organization redesign: the current state and projected trends, Management Decision 37(8), 1999.

[6] 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.

[7] 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 (Proc. of the 1st. Int. Workshop on Multi-Agent Based Simulation, MABS'98), Lecture Notes in Artificial Intelligence, vol. 1534, Springer-Verlag, 1998, pp. 169-182.

[8] 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, Lecture Notes in Artificial Intelligence, vol. 1952, Springer-Verlag, Berlin, 2000, pp. 152-161.

[9] Haroud, D., Boulanger, S., Gelle, E., and Smith, I.F.C., Strategies for conflict management in preliminary engineering design, In: Proceeding of the AID 1994 Workshop Conflict Management in Design, 1994.

[10] 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. Extended abstract in: C. Castelfranchi and W.L. Johnson (eds.), Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'02. ACM Press, 2002, pp. 501-502.

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

[12] 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.

[13] 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.

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

[15] McMillan, K., Symbolic Model Checking: An approach to the state explosion problem, Kluwer Academic Publishers, 1993.

[16] Rapoport, A., General System Theory, Abacus Press, 1986.

[17] 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.