Modeling Organizational Performance Indicators

Viara Popova and Alexei Sharpanskykh

***Abstract*—Performance measurement and analysis is crucial for steering the organization to realizing its strategic and operational goals. Relevant performance indicators and their relationships to goals and activities need to be determined and analyzed. Current organization modeling approaches do not reflect this in an adequate way. This paper attempts to fill the gap by presenting a framework for modeling performance indicators within a general agent-based organization modeling framework.**

***Index Terms*—agents, enterprise architectures, information systems, modeling**

I. Introduction

MEASURING and analyzing current organizational performance plays an important role in turning organizational goals to reality. The performance is usually evaluated by estimating the values of qualitative and quantitative performance indicators (e.g., profit, number of clients, costs). It is essential for a company to determine the relevant indicators, how they relate to the formulated company goals and how they depend on the performed activities. In practice such analysis is usually done in an informal, ad-hoc way. This paper introduces a framework for modeling performance indicators (PIs) and the relationships between them, which constitutes a part of a general framework for agent-based organization modeling and analysis.

In the general framework, organizations are considered from different perspectives (views). *Process-oriented view* describes the flow of control as well as static structures of tasks and resources. *Performance-oriented view* is characterized by a goal structure, a PIs structure, and relations between them as well as relations between goals and tasks, PIs and processes, goals and roles or agents. *Organization-oriented view* defines the organizational roles, each associated with a set of tasks and characterized by authority and responsibility relations on tasks, resources and information. Commitment, obligation and power relations and sets of competences required for agent allocation to roles are also defined. *Agent-oriented view* identifies different types of agents with their capabilities and principles for allocating agents to roles based on the matching between agent capabilities and competences required for roles.

Manuscript received September 29, 2006.

V. Popova and A. Sharpanskykh are with Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, phone: +31-20-5987756; fax: +31-20-5987653; e-mail: {popova, sharp}@cs.vu.nl.

Concepts and relations within every view are formally defined using dedicated languages which allow formal representation of both quantitative and qualitative aspects. To provide the formal meaning for the concepts and relations, to ensure the integrity of organization models and to enable different formal types of analysis, the axiomatic basis for these languages is given. The possible analysis includes: analysis by formal verification and validation techniques for different aspects of organization structure and behavior, and analysis by simulation of different scenarios of organizational behavior. These techniques can be applied for: analyzing an organization model abstracted from agents for the purpose of identifying inconsistencies, redundancies and errors; analyzing a simulated organization model with agent allocation in a certain scenario; analyzing a model based on empirical data generated by execution of processes in real organizations. The formal representation also enables semantic integration of ontologies for enterprise modeling, implemented in information systems of organizations aiming at cooperation or integration.

The formal language and axiomatic basis for *modeling PIs* within the *performance-oriented view* are described in this paper as well as the performance evaluation process and methodological issues of creating and revising performance-oriented models. Some verification techniques specific for performance-oriented organization models are briefly discussed. The presentation is organized as follows. Section II discusses related work on modeling organizations and PIs. In Section III the main concepts are defined. The relationships between them are defined in Section IV as well as the semantic aspects of the introduced language with its axiomatic basis. Section V discusses the evaluation of organizational performance. Methodological guidelines are given in Section VI. Section VII concludes the paper with a summary.

II. Related Work On Organization Modeling

A number of approaches have been proposed for creating a model for an organization in areas such as organization theory, computational enterprise modeling and artificial intelligence. Organization models that are normally used in *organization theory* are represented by informal or semi-formal graphical representations of specific organizational aspects [8] (e.g., decision making, power relations). The drawbacks of such models are lack of generality and relations between different specific models and graphical format that cannot be effectively processed, combined and analyzed. A class of models based on *system dynamics* is an exception devoid of these limitations [3]. These models are based on numerical variables and equations describing how they change over time. Such models are computationally effective but lack ontological expressivity and the possibility for higher abstract representation needed to conceptualize organizational relations and phenomena.

A solution to this problem has been proposed in the area of *computational enterprise modeling* [1] where a number of frameworks for enterprise engineering are introduced containing enterprise reference architectures (sets of modeling concepts), modeling languages and methodologies for creating a detailed (semi-)formal representation of organizational structure, behavior and environment. In many of these frameworks different views on organizations are distinguished, for which specialized models are created. Such declarative enterprise models may be represented in information systems that automate different organization processes and allow interoperability between parts of one enterprise and between enterprises. Although some of these models are built based on formal models of the concepts (formal ontologies), they allow only limited possibilities for computational analysis.

In order to enable more sophisticated types of analysis, techniques from mathematics and computer science may be used. In particular, *operations research* proposes mathematical methods for identifying best possible solutions to problems related to coordination and execution of the operations within an organization that optimize the overall organizational performance. At the same time other formal techniques exist, which are dedicated for analyzing particular aspects of an organization considered from a certain viewpoint (e.g., Petri-nets techniques used for modeling and analyzing business processes). Such approaches can be useful and efficient but the scope of their application is limited to a particular view, based on a limited number of concepts. To perform a profound evaluation of organizational performance and to enable analysis and prediction of organizational behavior under different influences, more sophisticated verification techniques should be used on concepts and relations across several views.

Furthermore, techniques from the area of *artificial intelligence* have been applied for modeling and analyzing multi-agent organizations [6]. In such models, software, hardware or human agents are allocated to roles that stand in relations to each other and are described by sets of functionalities performed by an organization. Although agent-based organizational models enable some formal analysis, most of these models are not able to capture the richness of the conceptual basis, relations and processes in organizations.

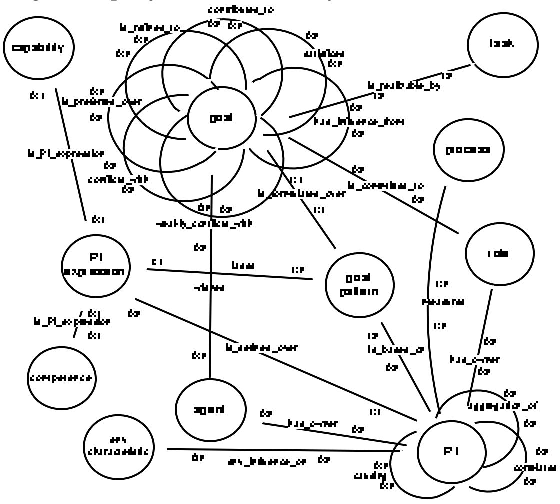
The goal of this work was to develop a framework for organizational modeling and analysis that possesses positive features of the modeling approaches described above, and provides means for elaborated manifold computational (agent-based) analysis. In particular, the modeling languages defined in this framework are expressive enough to convey structures and processes of organizations of most types within different views on organizations. Furthermore, the framework enables computational analysis techniques for different aspects of organization structure and behavior that employ concepts and relations across different views on organizations.

The focus of this paper is on modeling PIs as part of the performance-oriented view of the general framework. The area of *performance measurement* is an active field of research in management science attracting interest from both academic and practitioner circles. It is therefore surprising that the notion of a PI is nearly invisible in the current methodologies for enterprise modeling. We are only aware of one methodology, GRAI [2], which explicitly models PIs however only in the context of decision making and without the relationships among PIs and between PIs and other notions, e.g. goals. [6] on the other hand define quality variables which can be related to the PIs in this approach in order to model partial degree of satisfaction of a goal. Based on them objective functions are defined which are used in the formulation of goals. A major difference w.r.t. our approach is that in [6] probabilistic reasoning is used to determine the satisfaction of goals which shapes the definitions of objective functions and goals. It should be noted that sometimes measures as customer satisfaction, profit, etc. (typical PIs) are visible in other models as well, e.g., in the formulation of goals but they remain implicit and the relations between them are usually not discussed. [8] proposes a formal specification language for PIs used as inspiration for our approach. We extend the definition of a PI with a list of characteristics which are used in the analysis. The definition of causality relation is extended to express different degrees of influence. Inference rules are defined for adding relations to the PI structure. More importantly this paper goes further than [8] by connecting the concept of PI with other central organizational concepts, goal, task, role and agent, to enables more sophisticated analysis.

III. Performance Oriented Concepts

Every organization exists for the achievement of one or more goals. This varies depending on the type of organization, e.g. the main goal of a manufacturing company can be the realization of maximal profit, the goal of a non-profit organization can be to effectively protect wild animals. Being aware of these goals is a prerequisite to taking measures for their satisfaction. To ensure continued success, the organization should monitor its performance w.r.t. its goals. The notions of a goal and a PI are therefore essential. They are the main building blocks of the performance-oriented view of our approach. Fig.1 gives a graphical representation of the concepts and relationships of the *performance-oriented view*. This paper describes a modeling approach and a formal language for PIs as part of the performance-oriented view. Other concepts and views will be considered elsewhere [7].

The proposed approach is applied for modeling and analyzing a company from the security domain in the frames of



PI expression – a PI or a mathematical statement over a PI containing >, ≥, =, < or ≤. A PI expression can be evaluated to a numerical, qualitative or Boolean value for a time point, for the organization, unit or agent. For example using the defined PIs we can formulate PI expressions: P1 ≤ 48h; P2 = high.

Within the performance-oriented view also goal patterns and goals are defined which are addressed in details in [7]. Here a brief overview is given. PI expressions are used to define goal patterns which are properties that can be checked to be true or false for the organization, unit or individual at a certain time point or period. For example a goal pattern based on P1 can be GP1:'achieved that P1≤48h'. Goals are objectives that describe a desired state or development and are defined by adding to goal patterns information such as desirability and priority. A goal based on GP1 can be G1:'It is required to achieve that P1≤48h'. Goals can be hard (satisfaction can be established) or soft (satisfaction cannot be clearly established). For soft goals degrees of *satisficing* are defined. Goals can be organizational (i.e., belong to an organization/unit/role) or individual (i.e., belong to an agent). Individual goals may comply with, be disjoint or conflict with organizational goals. This can be determined by analyzing the relations between the PIs, on which the goals are based (see Section VI). Goal can be refined into subgoals forming a goals hierarchy. Information about the satisfaction of lower level goals can be propagated to determine satisfaction of high level goals.

The performance-oriented concepts are related to *other views* in the following way. Goals are realized by performing organizational functions described by *tasks*. *Processes* are specific instances of tasks temporally ordered in a workflow and performed by roles. A *role* represents a predefined set of functionalities performed within the organization which can be allocated to *agents*. Roles and agents can be committed to organizational or individual goals respectively. Roles are characterized by sets of *competences*, required to perform a certain task. Competences can be credentials (i.e., material or digital objects certifying accomplishments; e.g., diplomas, certificates), and skills (i.e., abilities that can be demonstrated, e.g., typing speed, flexibility). Skills are formulated as PI expressions over individual PIs. Agents are autonomous entities, characterized by their individual goals and capabilities. Individual goals of agents are based on individual PIs. *Capabilities* can be credentials or skills that are possessed by agents. Skills are formulated as PI expressions over individual PIs. An agent can only play a role if it has the capabilities to match the competences required for the role.

IV. A Formal Meta-Model of the Performance-Oriented View

The formal language used for specifying the meta-model for the performance-oriented view is a variant of the first order sorted predicate language. In this language, for each concept a special sort is introduced, containing all the names of concept instances (e.g., sort PI contains all the names of PIs). The characteristics (attributes) of the concepts are represented by relations (predicates) with arguments: a concept name, an attribute name and a value the attribute (e.g., ***has_attribute_value***: PI × ATTRIBUTE × VALUE). In the following for readability such predicates are used in the more compact form: concept.attribute=value. Furthermore, a number of other relations on concepts are defined in the meta-model (Fig.1). In order to provide formal meaning for the meta-model and to enable formal verification (e.g., consistency or integrity checking), the axiomatic basis is defined with the definitions of relations. Here the relations on PIs are considered.

causing: PI × PI × {very_pos, pos, neg, very_neg}: The first PI causes change in the same direction (positive) or opposite direction (negative) to the second PI. Very_positive describes the situation when small change in one PI causes big change in the other. Similarly for very_negative. The distinction between small and big change can be subjective and therefore should be defined carefully by the designer using input from domain experts. It is specific for each PI and is specified in the model by the threshold values assigned to PIs. When the value of a PI increases or decreases, positive or negative difference can be calculated and compared to the threshold value to determine whether it is considered a small or big change. This infomal explanation of the causality relation can be formalized as follows using the Temporal Trace Language [10] (p1 and p2 are variables over sort PI):

causing(p1, p2, pos) iff:
∀γ ∀t ∀a,b:PI_VALUE state(γ,t)|= [p1.value=a ∧ p2.value=b] ⇒
∀t1>t [∀c:PI_VALUE state(γ,t1)|= [p1.value=c ∧ c>a] ⇒
∃t2≥t1 ∃d:PI_VALUE state(γ,t2)|= [p2.value= d ∧ d>b]] & [∀e:PI_VALUE
state(γ,t1)|= [p1.value=e ∧ e<a] ⇒
∃ t2≥t1 ∃f:PI_VALUE state(γ,t2)|= [p2.value= f ∧ f<b]]

causing(p1, p2, very_positive) iff:
∀γ ∀t ∀a,b:PI_VALUE state(γ,t)|= [p1.value=a ∧ p2.value=b] ⇒
∀t1>t [∀c:PI_VALUE state(γ,t1)|= [p1.value=c ∧ c>a ∧ c-a<p1.threshold] ⇒
∃t2≥t1 ∃d:PI_VALUE state(γ,t2)|= [p2.value= d ∧ d>b ∧ d-b>p2.threshold]] &
[∀e:PI_VALUE state(γ,t1)|= [p1.value=e ∧ e<a ∧ a-e<p1.threshold] ⇒
∃t2≥t1 ∃f:PI_VALUE state(γ,t2)|= [p2.value= f ∧ f<b ∧ b-f >p2.threshold]]

The causality relations for the negative and very_negative cases are defined in a similar manner.

correlated: PI × PI × {pos, neg}: The first PI is correlated positively or negatively to the second PI, i.e., changes in the first PI result in changes in the second one in the same (pos) or opposite (neg) direction and the other way round. This is defined by the following axiom:

correlated (p2, p1, pn), where pn:{pos, neg} iff:
causing(p1, p2, pn) & causing(p2, p1, pn)

aggregation_of: PI × PI: The first PI is an aggregation of the second PI. If the aggregation relation exists between PIs, then these PIs are also positively correlated with each other.

∀ p1,p2:PI: aggregation_of(p1, p2) ⇒ correlated (p1, p2, pos)

Both PIs in the aggregation relation have the same type and unit. This is expressed by the following axiom:

∀ p1,p2:PI: aggregation_of(p1, p2) ⇒ p1.type=p2.type & p1.unit=p2.unit

The aggregation relation exists for example between PIs of the same type with time frame attributes related by the aggregation relation, e.g., PI ‘revenue for a year’ is an aggregation for PI ‘revenue for a month’. Aggregation relation between PIs can be defined based on the relations of PIs to processes and organizational roles. More specifically, the PIs of the same type are related by aggregation, when their owners (roles, agents) are related by the structural aggregation relation is_part_of, e.g., is_part_of(group1,deptA). For example PI ‘number of planners in deptA’ is an aggregation of PI ‘number of planners in group1’. Similarly if PIs of the same type measure the same aspect of execution of process instances of tasks related by is_subtask_of relation, e.g., is_subtask_of(collect_data, create_plan) then aggregation relation exists between these PIs.

Using the standard procedure from the sorted first-order predicate logic, terms and formulae over sort PI can be built, expressing different types of mathematical relations between PIs. For example, organizational_profit=organizational_revenue-organizational_costs; (PI1>3 & PI2=4.5) $\Rightarrow$ P3 > 5.2.

<u>*Examples:*</u>

PI name: **P1** (as defined earlier)

PI name: **P4**; *Def*: time to examine short-term plan proposal for correctness
Type: continuous; *Min value*: 0; *Max value*: ∞; *Unit*: hour; *Threshold*: 2 min; *Hardness*: hard; *Source*: job descriptions

PI name: **P5**; *Def*: number of produced short-term plans per planner per day
Time frame: day; *Type*: discrete; *Min value*: 0; *Max value*: ∞; *Unit*: plan; *Threshold*: 1 plan; *Hardness*: hard; *Source*: job descriptions

PI name: **P6**; *Def*: number of short-term plans produced by planning dept per week; *Time frame*: week; *Type*: discrete; *Min value*: 0; *Max value*: ∞; *Unit*: plan; *Threshold*: 5 plans; *Hardness*: hard; *Source*: domain knowledge

causing(P1, P4, pos)   aggregation_of(P6, P5)

PIs relate to tasks, processes, roles, agents by the relations:

has_owner: PI × {ROLE, AGENT}: A PI measures/describes the performance of a role or agent. Roles can be atomic or composite at any level including the level of the organization.

measures: PI × PROCESS: A PI expresses an aspect of the performance of the process execution, e.g. ‘time to produce a daily plan’ measures the time performance of the execution of the process ‘produce a daily plan’, ‘production costs’ measures the cost performance of the general process ‘production’.

Environmental conditions influence the execution of processes of an organization, thereby, also influence values of PIs related to these processes. This influence can be positive or negative and is specified by the following relation:

env_influence_on: ENV_CHARACTERISTIC × PI × {pos, neg}: An environmental characteristic of the sort ENV_CHARACTERISTIC influences a PI in a positive or negative way (i.e., contributes to the increase/decrease of a PI). For example, a large amount of rain contributes negatively to the amount and quality of harvest.

is_defined_over: PI_EXPRESSION × PI: A PI expression is defined over a PI

Other types of relations between PI, processes and roles, related to power, supervision, authorization, etc. will be discussed in the organization-oriented view.

V. PERFORMANCE EVALUATION

Every task in an organization contributes to the satisfaction of one or more organizational goals through performing its process instances. Each goal is formed based on a certain PI(s) which can be measured (directly or indirectly) during or after the process execution depending on the goal evaluation type – in the end or during a certain period of time (evaluation period defined as goal horizon). The satisfaction (degree of satisficing) of the goal(s) is determined by comparing the measured value(s) with the corresponding goal expression(s). Further, the obtained goal satisfaction (satisficing) measure is propagated by applying the rules defined in [7], upwards in the goal hierarchy for determining the satisfaction (degree of satisficing) of higher level goals. Thus, the organizational performance is evaluated by determining the satisfaction (degree of satisficing) of key organizational goals. The same principles can be applied for evaluation of agent performance.

As illustration of the proposed performance evaluation procedure consider the following example. For estimating the performance of the organization from the case study, the satisfaction of key goal G3.1 ‘it is required to achieve that within 48h from receiving relevant operational data, an up-to-date short-term plan exists’ has to be determined. This goal is refined into two goals: G3.1.1 ‘it is required to achieve that within 48h from receiving a new contract a new short-term plan is produced’ and G3.1.2 ‘it is required to achieve that within 48h from receiving data about necessary changes an updated short-term plan is produced’. These goals are related to tasks: G3.1.1 is realized by the task ‘generate new short-term plan’ and G3.1.2 is realized by the task ‘update short term plan’. By measuring the actual execution of the process instances of these tasks, it is determined that the values of both PIs of goals G3.1.1 and G3.1.2 (‘time to create short term plan’ and ‘time to update short term plan’) do not exceed 48 hours. Thus, goals G3.1.1 and G3.1.2 are satisfied. Due to the refinement relation G3.1 is also satisfied and contributes positively to the overall performance evaluation.

VI. METHODOLOGICAL ISSUES

Methodological issues discussed in this Section concern the construction and the revision of the PIs structures. As it was discussed in Section V, organization’s PIs can be extracted from different sources. To build a structure of PIs, relations between them should be identified, for which: (1) original documents can be analyzed for finding explicit references to such relations; (2) knowledge of domain experts and existing libraries of relations between PIs may be used; (3) PIs attributes and relations between these attributes (e.g., relations between time-related attributes and attributes that relate PIs to the organization and task structures) can be exploited (see Section IV); (4) from the existing relations in the PIs structure new relations may be inferred; (5) data mining techniques may be applied to the data collected during the organization operation; (6) intuitions of the modeler may be used after testing by domain experts or simulations; (7) relations to the task structure and the goal structure may be exploited.

As it follows fom the definitions in Section IV all the considered types of relations between PIs can be reduced to causality relations. Technique (4) allows inference of some missing causality relations from the existing PIs structure. In general the inference rules are specified in the form causing(p1, p2, s1) & causing(p2, p3, s2) $\Rightarrow$ causing(p1, p3, s3), where p1, p2 belong to the sort PI and s1, s2, s3 are of sort SIGN={very_neg, neg, pos, very_pos}. More specific (instantiated) inference rules are generated based on Table I, in which s3 values are given in the cells on the intersection of columns containing s1 values with rows containing s2 values. These inference rules can also be used for the verification of integrity of the PIs structure.

TABLE I.
INFERENCE RULES FOR CAUSAL RELATIONSHIPS

s2 \ s1	Very neg	Neg	Pos	Very pos
Very neg	Very pos	Very pos	Very neg	Very neg
Neg	Very pos	Pos	Neg	Very neg
Pos	Very neg	Neg	Pos	Very pos
Very pos	Very neg	Very neg	Very pos	Very pos

Further let us consider technique (7). The task structure of an organization may provide insight to discover relations between PIs. Often refinement relations specified in the task structure correspond to causality relations in the PIs structure. For example, based on the refinement relation between the task 'create a correct plan' (related by its process instance to PI 'time for creating a correct plan') and its subtask 'check a plan' (related to PI 'time to check a plan'), the PIs 'time to check a plan' and 'time for creating a correct plan' are related by positive causing relation. Refinement relations might also be reflected by other types of relations in the PIs structure.

Further, as it follows from the goal definition given in [7], goals and PIs form two highly interrelated structures – changes in one structure almost always imply changes in the other structure. Thus, the PIs structure and the goal structure may be created simultaneously. Usually, high level goals of a company are of a strategic (long-term) type. Such goals are often made operational by refining them into lower level tactical (short-term) goals. The identified in such a way refinement relation, by analogy with the task refinement, can be reflected in the PI structure by the corresponding relation between PIs, on which the considered goals are based. In general, the refinement and aggregation processes can be performed based on information about relations in an organization structure, task structure, temporal dependencies and relations between PIs.

In order to guarantee consistency and correspondence of the two structures to each other a number of consistency checks can be performed. If goals are related by refinement relation, then the corresponding PIs are related by a causality relation. Furthermore, if the PI expressions for goals related by refinement, contain comparison functions (e.g. >,<) or measures of degrees (such as 'high', 'low'), or goal patterns are specified by functions such as 'increased'/'decreased', then the specific type of causality may be determined (at least if it is positive or negative). For example, the goal 'It is required to limit the duration of the reviewing process to 1 month' (PI expression: 'duration of the reviewing process < 1 month') has one of the subgoals specified as 'It is desired to increase the number of reviewers'. Then the PIs 'number of reviewers' and 'duration of the reviewing process' should be related by negative causality relation in the PIs structure.

The identification of conflict relations between goals is of particular importance for the design and evaluation of organizations. To identify such conflicts, the goal patterns and the PIs structure can be used. More specifically, by knowing the type of the causality relation between PIs and the types of the goal patterns, the presence of a conflict between goals can be determined. For example, the goal 'It is required to maximize the time for checking the proposed plan for accuracy' and the goal 'It is required to minimize the time for producing a correct plan' are in conflict, since the PIs 'time for examining the plan for accuracy' and 'time for producing an accurate plan' are related by positive causality relation and the corresponding goal patterns are based on opposite types: maximize and minimize. If a conflict between high level goals is found, then via the refinement the cause of the conflict can be found at the lowest level of the goal structure. For this the relations between PIs and the domain knowledge are exploited.

VII. CONCLUSION

This paper presents a framework for modeling PIs and relationships between them which constitutes a part of a general organizational meta-model. The proposed framework provides formal tools for analyzing organizational and individual performance and relating current performance to the organizational goals and activities that can be used in enterprise information systems. This allows building structures that can be used for complex analysis within the performance-oriented view and between views. Some possibilities for analysis are mentioned here but will be elaborated and applied on larger case studies elsewhere. Other views and how they are related to each other will also be presented separately.

REFERENCES

[1] P. Bernus, L. Nemes, and G. Schmidt, *Handbook on Enterprise Architecture*, Springer-Verlag, 2003.

[2] G. Doumeingts, B. Vallespir, and D. Chen, "Decisional Modelling using the GRAI Grid", in: *Handbook on Architectures of Information Systems*, Springer-Verlag, 1998, pp. 313-338.

[3] J.W. Forrester, *Industrial dynamics*, Waltham, MA: Pegasus Communications, 1961.

[4] B. Horling and V. Lesser, "A Survey of Multi-Agent Organizational Paradigms". *The Knowledge Engineering Review*, Vol 19(4), Cambridge University Press, 2005, pp. 281-316.

[5] R.S. Kaplan, and D.P. Norton, "The balanced scorecard – measures that drive performance", *Harvard Business Review*, January-February 1992, pp. 71-79.

[6] E. Letier, and A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering", *Proc 12th ACM SIGSOFT Int symp on Foundations of software engineering*, 2004, pp. 53 – 62.

[7] V. Popova, and A. Sharpanskykh, "Performance-oriented organisation modelling", Technical Report 062109AI, Vrije Universiteit Amsterdam, http://hdl.handle.net/1871/10213

[8] V. Popova and J. Treur "A specification language for organisational performance indicators", in: *Proc 18th Int Conf IEA/AIE 2005*, *LNAI* vol. 3533, Springer, 2005, pp. 667-677.

[9] W.R. Scott, *Institutions and organizations*. 2nd edn, SAGE Publications, Thousand Oaks London New Delhi, 2001.

[10] A. Sharpanskykh, and J.Treur, "Verifying interlevel relations within multi-agent systems", in *Proc. of the 17th European Conference on AI, ECAI'06*, IOS Press, 2006, pp. 290-294.