View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by VU Research Portal

VRIJE UNIVERSITEIT AMSTERDAM

VU Research Portal

Performance-oriented Organisation Modelling

Popova, V.; Sharpanskykh, A.

2006

document version
Publisher's PDF, also known as Version of record

Link to publication in VU Research Portal

citation for published version (APA)
Popova, V., & Sharpanskykh, A. (2006). *Performance-oriented Organisation Modelling*.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 22. May. 2021

Performance-oriented Organisation Modelling
Technical Report 062109AI

Viara Popova and Alexei Sharpanskykh

Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{popova, sharp}@cs.vu.nl

Abstract. Each organisation exists or is created for the achievement of one or more goals. To ensure continued success, the organisation should monitor its performance with respect to the formulated goals. In practice the performance of an organization is often evaluated by estimating its performance indicators. In most existing approaches on organization modelling the relation between performance indicators and goals remains implicit. This paper proposes a formal framework for modelling goals based on performance indicators and defines mechanisms for establishing goal satisfaction, which enable evaluation of organizational performance. Methodological and analysis issues related to goals and performance indicators are briefly discussed in the paper. The described framework is a part of a general framework for organization modelling and analysis.

1. Introduction

Organizations exist in the first place for coordinating activities among actors and for handling the complexity of interactions with the environment while pursuing certain general goals. Therefore, the viability and success of an organization depend on how effectively the organization manages its internal activities and how well its behaviour fits with the environmental conditions. The behaviour of an organization is usually guided by its strategic and tactical goals that depend on the professional orientation (i.e., domain, types of activities) and specific characteristics of the organization, interests of stakeholders and on the type of the environment in which the organization is situated. The performance of an organization is often evaluated by estimating the values of its qualitative and quantitative performance indicators [1, 8] (e.g., profits, number of clients, losses). Therefore, to ensure effectiveness of an organization, all the principal performance indicators should be reflected in its goals. While in most existing approaches on organization modelling the relation between performance indicators and goals remains implicit, this paper defines a clear and general mechanism for specifying goals, performance indicators and relations between them. Then, the performance of an organization can be evaluated by estimating the (level of) satisfaction of its goals.

In real organizations many different types of goals can be identified that are related by different types of relations. The satisfaction of some of them can be determined in a clear-cut way by evaluating conditions in goal expressions (e.g., “ensure that an order is performed in 24 hours”), whereas the satisfaction of some other goals is difficult to assess (e.g., “maximize the customer satisfaction”), since they refer to not directly measurable quantities. This paper differentiates goals in different types and proposes mechanisms for establishing the goal satisfaction.

Furthermore, the satisfaction of goals can only be determined in the framework, in which goals are related to other concepts (such as tasks or activities, roles and agents). This paper introduces such a framework for the performance-oriented modelling, which constitutes a part of a general framework for organization modelling and analysis, briefly described in this paper.

In the general framework, organizations are considered from different perspectives (or views): performance-oriented, process-oriented, organization-oriented and agent-oriented. Concepts and relations within every view are formally described using dedicated languages. To provide the formal meaning for the concepts and relations between them defined in the organizational views, to ensure the integrity of an organization model and to enable different formal types of analysis of organization models, the axiomatic basis is defined for the dedicated formal languages. Furthermore, the formal definition of organizational models and the axiomatic basis enable semantic integration of different ontologies for enterprise modelling, implemented in information systems of organizations aiming at cooperation or integration.

The formal language and the axiomatic basis specific for the performance-oriented view are described in this paper. Moreover, this paper addresses methodological issues of creating and revising performance-oriented models of organizations, and specifically describes the process of performance evaluation.

The individuals (or agents) assigned to certain positions (or roles) in an organization have personal goals based on individual performance indicators that may comply with, be disjoint or even conflict with organizational goals. The performance of individuals can be determined in the same manner as the performance of an organization. For solving conflicts between goals, conflict resolution methods are sketched in the paper.

Some of the verification techniques specific for performance-oriented organization models are briefly described in this paper and will be considered in more detail elsewhere. More sophisticated and general types of analysis of organization models (e.g., analysis of the temporal development of an organization with agents allocated to roles) employ concepts and relations from different views on organizations that are beyond the scope of this paper and will be described elsewhere.

The presentation is organised as follows. Section 2 discusses the related work on organization modelling and provides motivations for the development of a general multi-purpose framework on organization modelling and analysis. In Section 3 the proposed framework that considers different views on organizations is described. Section 4 introduces the case study that will be used to illustrate modelling and analysis techniques considered in the performance-oriented view. In the following Sections 5-8 the proposed framework for the performance-oriented modelling and analysis is described. Section 9 compares the presented approach to other existing goal-oriented methodologies. Finally, Section 10 concludes the paper with a summary and directions for future research.

2. Related Work on Organization Modelling

To a large extent the vitality and productivity of an organization situated in an environment of a certain type depend on the structure and behaviour of the organization, and how they conform to the environmental conditions. For more precise evaluation of the organizational performance, for identification of performance bottlenecks and organizational conflicts, and for estimation (prediction) of consequences of different environmental influences, organization structures and behaviours on organizational performance, detailed organizational analysis based on an organization model should be performed. Furthermore, an organization model constitutes a basis for many automated processes within enterprises (e.g., computer integrated manufacturing [3], production management [7]) and provides a foundation for the inter-enterprise cooperation. For creating a model for an organization a number of approaches have been proposed in areas such as organization theory, computational enterprise modelling, and artificial intelligence.

Organization models that are normally used in organization theory are represented by informal or semi-formal graphical descriptions that illustrate specific aspects of organizations [25, 27] (e.g., decision making, authority and power relations). The disadvantages of such models are obvious: (1) lack of generality and relations between different specific types of models, and (2) graphically depicted data can not be effectively processed, combined and analyzed. A class of models built based on the system dynamics theory is an exception in organization theory devoid of both these disadvantages [9]. Organizational models specified in system dynamics are based on numerical variables and equations that describe how these variables change over time. Although such models can be computationally effective (i.e., used for simulations and computational analysis), nevertheless they still lack the ontological expressivity and the possibility for higher abstract representation that are needed to conceptualize wide range of relations and phenomena that exist in different types of organizations. A solution to this problem has been proposed in the area of computational enterprise modelling [3, 10, 14]. In this area a number of frameworks for enterprise engineering are introduced. Usually, these frameworks include enterprise reference architectures (i.e., sets of modelling concepts), modelling languages and methodologies for creating a detailed (semi-)formal representation of organizational structure, behaviour, and the environment for different types of organizations. Within many of these frameworks different views on organizations are distinguished (e.g., information view, process view, resource view), 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 different parts of one enterprise and between enterprises. Although some of these models are developed based on formal models of the concepts (i.e., formal ontologies) [10], they allow only limited possibilities for performing computational analysis (e.g., by simulations or by specific types of verification).

In order to enable more sophisticated types of analysis, techniques from mathematics and computer science may be used. In particular, operation research proposes mathematical methods for identifying best possible solutions to problems related to coordination and execution of the operations within an organization that improve or optimize the organizational performance [24]. Usually, operations research methods propose solutions to improve performance of an entire organization, rather than concentrating only on its specific elements.

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 modelling and analyzing business processes [6]). Although such approaches can be useful and efficient, the scope of their application is limited to a particular view on an organization, based on a limited number of concepts. However, to perform a profound evaluation of the organizational performance and to enable analysis and prediction of organizational behaviour under different influences, more sophisticated verification techniques should be used that employ concepts and relations between them across different views on organizations.

Furthermore, techniques from the area of artificial intelligence have been applied for modelling and analyzing multi-agent organizations [2, 5]. In such organizational models (software, hardware or human) agents are allocated to roles that stand in certain relations to each other and are described by sets of functionalities performed by an organization. Such models can be used for example for coordinating tasks execution in a multi-agent system [15], or for enforcing certain behaviours (e.g., normative systems) upon an agent system [29]. Although agent-based organizational models enable different types of formal analysis, most of these models are not able to capture the richness of the conceptual basis, social relationships and diversity of processes that exist in organizations of different types.

The goal of this work was to develop a framework for organizational modelling and analysis that possesses positive features of the modelling approaches described above, and provides means for elaborated manifold computational (agent-based) analysis of organizational models. In particular, the modelling 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 behaviour that employ concepts and relations between these concepts across different views on organizations.

3. The Proposed Framework

The proposed framework consists of a modelling approach, which is based on a number of formal languages, analysis methods for different aspects of an organization model and a methodology, which describes the process of organization design.

In order to reduce the complexity of the modelling process and to capture essential aspects of structure and behaviour of organizations of different types, the proposed modelling approach incorporates a number of interrelated concepts that are grouped under different views, formally described by a number of dedicated logic-based languages. Each view represents a certain perspective on an organization. The following four views are distinguished in the proposed modelling approach: process-oriented, performance-oriented, organization-oriented and agent-oriented view. Note that both the components of the proposed framework and the considered modelling views can be directly related to the components of the GERAM (the Generalized Enterprise

Reference Architecture and Methodology) [14], the initiative of the IFIP-IFAC Task Force, based on which several international standards for enterprise modelling have been created. The GERAM provides a generalized template for the development of elaborated enterprise modelling frameworks, which was taken into account when the proposed framework has been developed. In the following a brief characterization of the listed organization modelling views is given.

Process-oriented view: This view describes different types of flows in an organization, i.e., the flow of tasks, the flow of resources, and the flow of information. To define these flows, appropriate static structures of tasks, resources and information are used. In particular, a task structure is a graph built based on the refinement relations between tasks. A flow of tasks may either reflect a temporal ordering of tasks, or a temporal ordering of tasks affected by control elements (i.e., conditions and decision points) that influence the path of a flow. Resources are required for the execution of tasks. A flow of resources describes a (temporal) sequence of processes such as utilization, consumption, creation, modification and replacement of resources during the execution of organizational tasks. A flow of information describes how existing information is used and new information is created and stored in an organization. Two types of information flows are distinguished: flow of control information and flow of data. Control information usually exists in the form of directives and guidelines that regulate the execution of a task. Unlike control information, data characterize objects (material and informational) of an organization.

Performance-oriented view: This view is characterized by an organizational goal structure, a performance indicators structure, and relations between them. Both these structures are built based on multiple types of relations that allow representing to a high degree the complexity and interdependency of intentions and performance measures within real organizations. In order to evaluate and to predict a level of performance of an organization, next to relations between goals and performance indicators, relations between goals and tasks, and performance indicators and tasks are explicitly defined in this view. Furthermore, organization goals are always assigned to certain organization role(s). Also agents may have individual goals that conform to or contravene goals of an organization and, thus, influence the execution of organizational tasks by agents. Both organizational and individual goals and relations between them are specified in this view.

Organization-oriented view: Within this view organizational roles are defined. Each role is associated with a subset of functionalities (a set of tasks) performed by an organization and, consequently, is committed to certain organization goal(s). Each role is characterized by authority and responsibility relations on tasks, resources and information defined in the process-oriented view. Between roles power relations with respect to certain tasks are defined. Furthermore, commitment and obligation relations originating from contracts made between roles are considered in this view. For each role a set of competences is defined that are required from agents to be allocated to the role. An organization reward system is also explicitly defined in this view.

Agent-oriented view: In this view different types of agents with their capabilities (i.e., skills and credentials) are identified. Principles for allocating agents to roles are formulated based on the matching between agent capabilities and competences defined for roles.

Note that the identified views are related to each other by means of sets of common concepts. For example, the relations between goals and roles are introduced in the performance-oriented view. Further these relations are used in the organization-oriented view to describe mechanisms of goal assignment and delegation, which also use power and authority relations from the organization-oriented view.

In all these views environmental conditions, in which the organization is functioning, are taken into account: they influence the specification of organization concepts and relations between them (e.g., the formulation of goals and the specification of tasks), thus, affecting the structure and behaviour of a particular organization model. Furthermore, the type of the environment determines a part of domain knowledge, which is represented by unconditionally valid facts and rules about the environment that directly influence all the activities within the organization. Another part of the domain knowledge is defined by intrinsic properties of the organization itself.

To formalize concepts, relations and rules specified within the described views, dedicated logic-based modelling languages are used. These languages allow formal representation of both quantitative and qualitative aspects of an organization model, described by continuous and discrete variables respectively. The semantics of the modelling languages are defined by the meta-model that describes modelling concepts, relations between them and constrains on these relations, and semantic rules over these relations.

A formal representation of the organization models enables different types of analysis: analysis by simulation of different scenarios of organizational behaviour, and analysis by formal verification and validation techniques for different aspects of an organization structure and behaviour. These techniques can be applied for:

1. analyzing an organization model abstracted from agents (i.e., without allocation of agents to roles) for the purpose of identifying inconsistencies, redundancies and errors;
2. analyzing a simulated organization model with agents allocated to roles in a certain scenario;
3. analyzing a model based on empirical data generated by execution of processes in real organizations.

Some of these analysis techniques are specific for a particular view; however, most of them make use of relationships between elements of an organization model defined across different views.

The methodological part of the framework describes a number of guidelines for designing organization models of different types (i.e., corresponding to different views). Design principles defined in these guidelines may be applied both for creating organization models from scratch, and for analyzing models of existing organizations. In order to provide freedom of choice for a designer during the organization design process, most of the design principles have the status of a recommendation, which nevertheless may be very useful for novices and inexperienced users.

This paper describes a modelling approach and a formal language for the performance-oriented view. Other views defined in the proposed framework will be considered elsewhere.

In the next Section the case study that will be used for illustration of the proposed performance-oriented framework is introduced. In Section 5 the main concepts included in the performance-oriented view are defined. The relationships between them are described and formalized using the dedicated logic-based language, thus, presenting the formal meta-model for the performance-oriented view in Section 6. Section 7 describes the semantic aspects of the introduced language in the form by introducing the axiomatic basis. Section 8 discusses how the introduced concepts are used to evaluate the performance of the organization. Methodological guidelines are given in Section 9.

4. Introduction to the Case Study

The proposed approach is applied for modeling and analyzing an organization from the security domain. The main purpose of the organization is to deliver security services (e.g., private property surveillance, safeguard) to different types of customers (individual, firms and enterprises). The organization has well-defined structure with predefined (to a varying degree) job descriptions for employees. The total number of employees in the organization is approximately 230.000 persons. The global management of the organization (e.g., making strategic decisions) is performed by the board of directors, which includes among others the directors of the different divisions (regions). Within each region a number of areas exist controlled by area managers. An area is divided into several units, controlled by unit managers. Within each unit exist a number of locations, for which the contracts with customers are signed and security officers are allocated. The allocation of employees is performed based on plans created by planning groups.

The model that corresponds to the part of the organization concerned with the planning process will be used in this paper to illustrate concepts, relations and techniques related to the performance-oriented view. Therefore, let us consider the planning process in more detail. The planning process consists of the forward (or long-term) planning and the short-term planning. The forward planning is a process of creation, analysis and optimization of forward plans that describe the allocation of security officers within the whole organization for a long term (4 weeks). Forward plans are created based on customer contracts by forward planners from the forward planning group, managed by the manager of planning. During the short-term planning, plans that describe the allocation of security officers to locations within a certain area for a short term (a week) are created and updated based on the forward plan and up-to-date information about the security employees. Furthermore, based on short term plans, daily plans are created. Within each area the short-term planning is performed by the area planning team that consists of planners and is guided by a team leader. During the planning process short-term planners interact actively with forward planners (e.g., for consultations, problem solving). Furthermore, forward planners have a number of supervision functions with respect to short-term planners.

5. 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 amount of profit, the goal of a non-profit organization can be to protect maximal number of wild animals, the goal of a sports club can be to realize maximal number of winning games, a book club might among others have the goal to maximize the fun experienced by members during the meetings, etc. Being aware of these goals is a prerequisite to taking measures for their satisfaction. To ensure continued success, the organization should monitor its performance with respect to the formulated goals. The notions of a goal and a performance indicator are therefore important and need to be taken into account in the organizational model. They are the main building blocks of the performance-oriented view of the approach presented by this paper.

Performance indicators are defined as measures, quantitative or qualitative, that can be used to give a view on the state or progress of the company, a unit within the company or an individual. The set of relevant PIs is company-specific and the used PIs are often not independent. Causal and other relationships can exist which represent the influence that change in one PI can have on another PI. Such relationships are used to build a structure of PIs.

Expressions can be formulated over performance indicators for example by defining target values. PI expressions are then 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 a time period. Goals are objectives that describe a desired state or development of the organization and are defined by adding to goal patterns information such as desirability and priority. Roles in the organization or agents can be committed to organizational or individual goals respectively. Goals are realized by performing tasks which contribute to their satisfaction. Every task in the organization should contribute to some defined goal. Goal can be refined into subgoals forming a goals hierarchy. Conflicts can be present between goals in the hierarchy and they are specified explicitly.

In the rest of this Section the concepts and their characteristics will be defined more precisely. In Section 6 the relations between the concepts will be defined and formally specified using the dedicated language. The semantic aspects and the axiomatic basis for the language will be also discussed in Section 6.

Performance indicator – quantitative or qualitative indicator that reflects the state/ progress/ performance of the company/unit/individual. Characteristics:

name

definition

time frame – (if applicable) for which time frame is the performance indicators defined – the length of the time interval for which it will be measured/evaluated, e.g. the indicator ‘yearly profit’ has time frame ‘year’, ‘number of customers per day’ has time frame ‘day’

type – continuous, discrete

scale – if relevant, the measurement scale for the PI. Different scales can be predefined and referred to here.

min value – if needed, e.g. when a predefined scale is used and only a part of this scale is relevant for the particular PI

max value – if needed

value – the current measured value of the performance indicator

unit - how is the PI measured, e.g. in hours / kg / pages / etc., if applicable

source – which was the internal or external source used to extract the PI: company policies, mission statements and job descriptions, laws, domain knowledge, etc. These sources contain (informal) statements about desired state or behaviour of the company and regulations it has to obey.

threshold – the cut-off value separating changes in the PI value considered to be small and changes considered to be big. It will be used to define the degree of influence of one PI on another (see Section 6).

hardness:

Soft – not directly measurable, e.g. qualitative. Some example: customer's satisfaction, company's reputation, safety, employees' motivation, efficiency, etc.

Hard – measurable, quantitative. Examples: number of customers, time to produce a plan, number of mistakes in the data base, etc.

The set of performance indicators that can be defined for one organization can be very large and it is often not feasible to monitor all of them. Therefore the companies select a subset of indicators that are considered enough to give a representative picture of the performance and the costs of measuring and monitoring are reasonable. Those are called *key performance indicators.* It is essential for the company to choose its key performance indicators carefully to form a balanced (w.r.t. the company activities, involved parties, etc.) and sufficiently complete set [17]. The KPIs of the organization should be reflected in its goals.

The process of extracting the PI from a source document involves asking the question: What needs to be measured or observed in order to conform to the statements from this document? Often the performance indicators are represented by nouns in the text, and modifiers such as the adjectives give information about the type and scale of measurement as well as what will be considered a desirable value of the indicator (this will be used to construct expressions over the PI, goal patters and goals). For example, one of the requirements from the job description for a planner from the case study is "to ensure the high precision of calculation when plan is created". Here "the precision of calculation during the plan creation" is a PI and "high" is its desirable value.

Often the performance indicators that can be extracted from documents such as the mission statements and the strategy definition are soft and difficult to assess. In order to evaluate such a performance indicator it is usually beneficial to find a closely related hard indicator that can be measured instead and that can give an impression on the state of the soft one. For example customer satisfaction cannot be measured directly but it is possible to design questionnaires that will be used to collect information on customer's opinion and classify it in predefined ranges such as high, medium, low, etc. The results from such a study give some impression on the actual degree of satisfaction but it is important to notice that the actual degree of satisfaction might deviate from the calculated value and cannot be measured directly. The domain knowledge that is used is that if the questionnaires are designed appropriately then there will be correlation between the customer's degree of satisfaction and the results from the study. Moreover, changes in the former will cause changes in the latter.

Example:

PI name: P1
Definition: time to produce a new short-term plan given a contract
Type: continuous
Min value: 0
Max value: ∞
Unit: hour
Threshold: 5 hours
Hardness: hard
Source: job descriptions

PI name: P2
Definition: efficiency of allocation of security officers to objects
Type: discrete
Scale: very low-low-medium-high-very high
Hardness: soft
Source: job descriptions

PI name P3
Definition: customer satisfaction
Type: discrete
Scale: very low-low-medium-high-very high
Hardness: soft
Source: mission statement

PI expression – a performance indicator or a mathematical statement over a performance indicator containing >, ≥, =, < or ≤, e.g. p1, p1>50, p1<=6. A PI expression can be evaluated to a numerical, qualitative or Boolean value at a certain time point for the organization or agent.

Examples: Using the previously defined performance indicators we can formulate some PI expressions: P1 <= 48 hours; P2 = high; P3 = high

Goal pattern – a property over one or more PI expressions that can be checked for a given state/time point or an interval for the company or an individual agent. Characteristics:

name

definition

pattern - the temporal pattern which determines the way the property should be checked:

Achieved – it should be checked whether the property is true for a specific time point

Ceased – it should be checked whether the property is false for a specific time point

Maintained – it should be checked whether the property is true for the duration of a specific time interval

Avoided – it should be checked whether the property is false for the duration of a specific time interval

Optimized – (maximized, minimized, approximated) it should be checked if the value of the PI expression has increased, decreased or approached a given target value for the duration of a given time interval

Achieved, ceased, maintained and avoided are used on PI expressions that are evaluated to a Boolean value while optimised is defined over PI expressions that are evaluated to value of any type which is ordered (for maximized and minimized) or for which a distance measure is defined (for approximated).

<u>*Examples:*</u>

Name: GP1
Definition: maintained efficiency of allocation of security officers to objects = high
Pattern: maintain

Name: GP2
Definition: achieved that time to produce a short-term plan given operational data ≤ 48
Pattern: achieve

Goal – an objective to be satisfied describing a desired state or development of the company or of an individual. Characteristics:

name,

definition,

priority – numerical estimation between 0 and 1 of the priority of the goal; alternatively {very high, high, medium, low, very low}. Priority can play a role in conflict resolution and in situations when choice can be made between tasks to be performed and/or scheduled.

evaluation type:

Achievement goal – based on achieve or cease pattern – is evaluated for a given state/time point

Development goal – based on maintain, avoid or optimize pattern – is evaluated for a given time interval

horizon - within which time interval (for development goals) or at which time point (for achievement goals) is the goal supposed to be satisfied:

Long-term goal – is supposed to be satisfied (achieved or maintained) on the long term

Mid long term goal

Short term goal

ownership:

Organizational – belongs to the organization/unit/role, follows from the highest level goals of the company. It is often assigned higher priority.

Individual – belongs to an agent; priority or such goals might depend on the company policy – some company might assign low priority to individual goals compared to organizational goals, others might decide to involve and motivate the agents by taking better into account their individual goals and thus avoiding some conflicts that might exist between individual and organizational goals. However, in many cases individual goals of agents assigned to organizational roles comply with organizational goals. Also agents may have individual goals that conform to or contravene goals of an organization and, thus, influence the execution of organizational tasks by agents.

perspective (for organizational goals) – which point of view is described by the goal: management – internal; supplier's point of view – external; customer's point of view – external; and societal point of view – external.

Even though all organizational goals are meant to belong to the organization itself, they can reflect the point of view of an external party which desires the organization to perform in a certain way. For example the society might want the organization to obey the norms and values adopted in this society. It is sometimes beneficial for the company to adopt goals desired by other parties e.g. to conform to the relevant laws.

It is also important to note that the different points of view will often be conflicting, for example while customers might want low prices, the management wants high profits, however if the prices are lowered that will decrease the profits. Such conflicts should be recognised during the design phase and made explicit in order to deal with them. For example priorities can be defined in order to specify which goal is more important to satisfy.

hardness:

Soft – satisfaction of the goal cannot be clearly established, qualitative. We use the term satisficing to indicate acceptable degree of satisfaction of a soft goal since satisfaction cannot be established in a clear-cut way. Soft goals are given the labels that correspond to degrees of satisficing or denial of them: *satisficed*, *weakly_satisficed*, *undetermined*, *weakly_denied*, *denied*. There is a natural order between these labels: satisficed > weakly_satisficed > undetermined > weakly_denied > denied. Soft goals are represented graphically as cloud shapes.

Hard – satisfaction can be established, quantitative. Hard goals can be *satisfied*, *undetermined* or *failed*. These labels are ordered as follows: and satisfied > undetermined > failed. Hard goals will be represented graphically as ovals.

negotiability:

Non-negotiable – need to be satisfied, no compromise is possible

Negotiable – negotiation is possible in order to resolve conflicts with other goals

The negotiability characteristic of a goal can be used in the process of conflict resolution at the design phase.

<u>*Example:*</u>

Goal name: G3
Informal definition: It is required to maintain high efficiency of allocation of security officers to objects.
Priority: high
Horizon: long-term
Evaluation type: development goal (maintain goal pattern)
Ownership: organizational
Perspective: management, customer
Hardness: soft
Negotiability: negotiable

Goal name: G3.1
Informal definition: It is required to achieve that within 48 hours from receiving relevant operational data, an up-to-date short-term plan exists
Priority: medium
Horizon: short-term
Evaluation type: achievement goal (achieve goal pattern)
Ownership: organizational
Perspective: management, customer
Hardness: hard
Negotiability: negotiable

Goals are realizable by tasks in an organization. Task represents a certain function performed in an organization by its role(s). A role represents a predefined set of functionalities performed within the organization. Roles and agents can be committed to goals.

These and other related concepts will only briefly be discussed here since they are more relevant to other views (process-oriented, organizational and agent-oriented). More detailed definitions of concepts and relationships belonging to these views (including power and authority relations) will be given elsewhere.

Roles are characterized by sets of competences, which are required to perform a certain task. Competences can be credentials (i.e., material or digital objects certifying accomplishments; e.g., diplomas, patents, certificates), and skills (i.e., abilities that can be demonstrated and/or tested; e.g., typing speed, flexibility, programming skills). Skills can be hard (measurable, such as typing speed) or soft (not directly measurable, such as flexibility). Skills are formulated as PI expressions over individual performance indicators.

Roles are allocated to agents, who eventually will perform tasks in an organization. Agents are autonomous entities, characterized by their individual goals and capabilities. Individual goals of agents are based on individual performance indicators. Capabilities can be credentials and/or skills that can be possessed by agents. Skills are formulated as PI expressions over individual performance indicators. An agent can only play a particular role if it has the capabilities that match the competences required in the role description. More elaborated characteristics and mechanisms (e.g. agent allocation) related to agents will be given in the description of the organization-oriented view.

6. A Formal meta-model for 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 [23]. In this language, for each type of a concept a special sort is introduced, which contains all the names of concept instances (e.g., sort PI contains all the names of performance indicators). The semantics for this language is defined in a standard way, by interpretation of sorts, constants, functions and predicates, and a variable assignment. The characteristics (or attributes) of the concepts are represented by corresponding relations (predicates) with arguments: a concept name, an attribute name and a value the attribute (e.g., ***has_attribute_value***: PI x ATTRIBUTE x VALUE). In the following for better readability such predicates will be used in the more compact form: concept.attribute=value. Furthermore, a number of other relations on concepts are defined in the meta-model (the graphical representation of the meta-model that depicts most of the identified relations is given in Figure 1). In order to provide the formal meaning for the introduced meta-model and to enable formal verification of organization models (e.g., consistency and integrity checking), the axiomatic basis is defined along the definitions or relations.

First, consider the relations on performance indicators.

causing: PI $\times$ PI $\times$ {very_positive, positive, negative, very_negative}

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 PI. 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 value assigned to each PI. 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. When small change of indicator p1 with respect to its threshold results in big change in p2 with respect to its own threshold then we can specify that causing(p1, p2, very_positive).

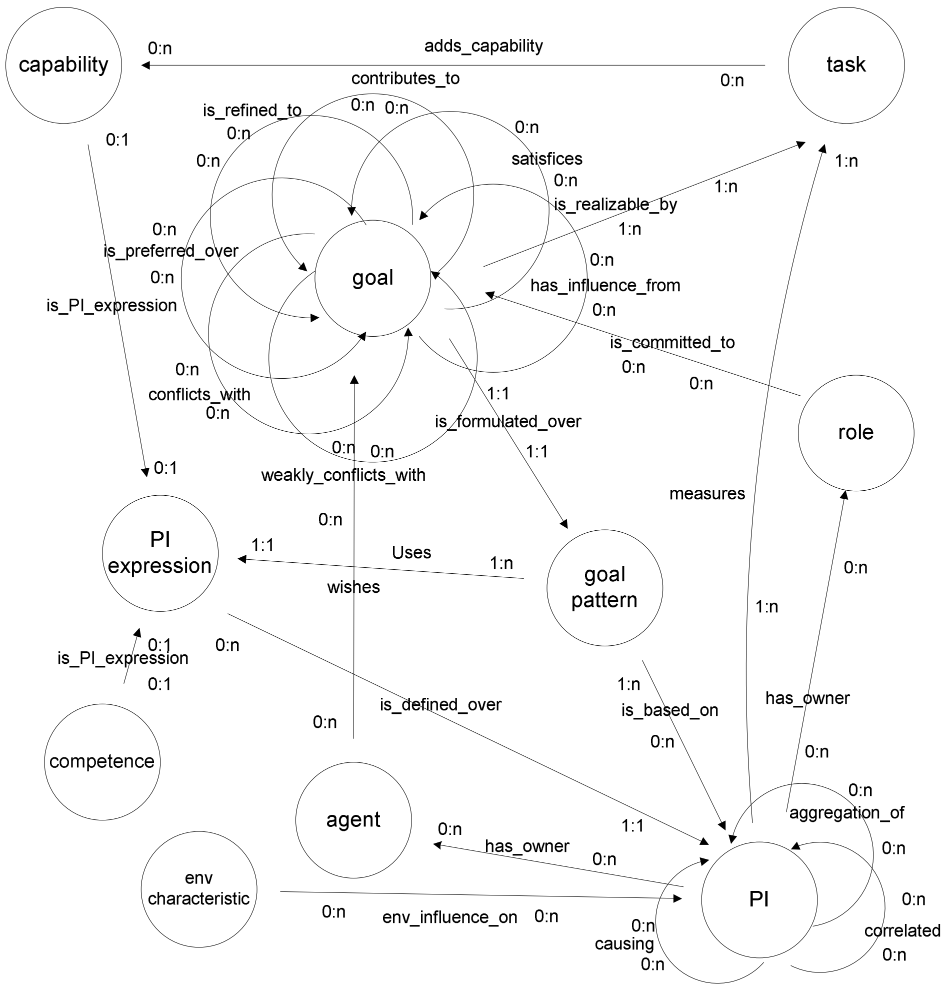

Fig. 1. A Meta-model for the performance-oriented view.

Similarly for very negative influence. This informal explanation of the causality relation can be formalized by the definitions below (p1 and p2 here and below are variables over sort PI):

causing(p1, p2, positive) iff:

$\forall\gamma\ \forall t\ \forall a,b{:}PI\_VALUE\ state(\gamma,t)|{=}\ [\ p1.value{=}a \wedge p2.value{=}b\ ] \Rightarrow \forall t1{>}t\ [\forall c{:}PI\_VALUE\ state(\gamma,t1)|{=}\ [p1.value{=}c \wedge c{>}a\ ] \Rightarrow \exists t2{\geq}t1\ \exists d{:}PI\_VALUE\ state(\gamma,t2)|{=}\ [p2.value{=}\ d \wedge d{>}b]]\ \&\ [\forall e{:}PI\_VALUE\ state(\gamma,t1)|{=}\ [p1.value{=}e \wedge e{<}a\ ] \Rightarrow \exists\ t2{\geq}t1\ \exists f{:}PI\_VALUE\ state(\gamma,t2)|{=}\ [p2.value{=}\ f \wedge f{<}b]]$

causing(p1, p2, very_positive) iff:

$\forall\gamma\ \forall t\ \forall a,b$:PI_VALUE state($\gamma$,t)|= [p1.value=a $\wedge$ p2.value=b] $\Rightarrow$ $\forall$t1>t [$\forall$c:PI_VALUE state(γ,t1)|= [p1.value=c $\wedge$ c>a $\wedge$ c-a<p1.threshold] $\Rightarrow$ $\exists$t2≥t1 $\exists$d:PI_VALUE state(γ,t2)|= [p2.value= d $\wedge$ d>b $\wedge$ d-b>p2.threshold]] & [$\forall$e:PI_VALUE state(γ,t1)|= [p1.value=e $\wedge$ e<a $\wedge$ a-e<p1.threshold] $\Rightarrow$ $\exists$t2≥t1 $\exists$f:PI_VALUE state(γ,t2)|= [p2.value= f $\wedge$ f<b $\wedge$ b-f >p2.threshold]]

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

correlated: PI × PI × {positive, negative}

The first PI is correlated positively or negatively to the second PI. The meaning of this relation is defined by the following axiom:

correlated (p2, p1, pn), where pn:{positive, negative} 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.

$\forall$ p1,p2:PI: aggregation_of(p1, p2) $\Rightarrow$ correlated (p1, p2, positive)

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

$\forall$ p1,p2:PI: aggregation_of(p1, p2) $\Rightarrow$ p1.type=p2.type & p1.unit=p2.unit

The aggregation relation exists for example between performance indicators of the same type with time frame attributes related by the aggregation relation. For example, the performance indicator “revenue for a year” is an aggregation for the performance indicator “revenue for a month”.

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 performance indicators. For example, organizational_profit=organizational_revenue-organizational_costs; (PI1>3 & PI2=4.5) $\Rightarrow$ P3 > 5.2.

Examples:

PI name: P1 (as defined earlier)
Definition: time to produce a new short-term plan given a contract

PI name: P4
Definition: time to examine the short-term plan proposal for correctness
Type: continuous
Min value: 0
Max value: ∞
Unit: hour
Hardness: hard
Source: job descriptions

PI name: P5
Definition: number of produced short-term plans per planner per day
Type: discrete
Min value: 0
Max value: ∞
Unit: plan
Hardness: hard
Source: job descriptions

PI name: P6
Definition: number of produced short-term plans for the planning department per week
Type: discrete
Min value: 0
Max value: ∞
Unit: plan
Hardness: hard
Source: domain knowledge

PI name: P7
Definition: time to estimate human capacity per location
Type: continuous
Min value: 0
Max value: ∞
Unit: hour
Hardness: hard
Source: job descriptions

PI name: P8
Definition: time to assign officers to tasks
Type: continuous
Min value: 0
Max value: ∞
Unit: hour
Hardness: hard
Source: job descriptions

PI name: P9
Definition: time to input planning data into the information system
Type: continuous
Min value: 0
Max value: ∞
Unit: hour
Hardness: hard
Source: job descriptions

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

Performance indicators are related to tasks, roles and agents by the following relations.

has_owner: PI x {ROLE, AGENT}

PI measures/describes the performance of a role or an agent. Roles can be atomic as well as composite at any aggregation level including the level of the organization.

measures: PI $\times$ TASK

PI expresses an aspect of the performance of the task execution, e.g. 'time to produce a daily plan' measures the time performance of the execution of the task 'produce a daily plan', 'production costs' measures the cost performance of the execution of the composite task 'production'.

An aggregation relation between performance indicators can be defined based on the relations of performance indicators to tasks and organizational roles. More specifically, the performance indicators of the same type are related by aggregation, when their owners (roles or agents) are related by the structural aggregation relation is_part_of (e.g., is_part_of(group1, department_A)). For example, the performance indicator "a number of engineers within the department_A" is an aggregation for the performance indicator "a number of engineers within the group1". Similarly, if the performance indicators of the same type measure the same aspect of the execution of tasks related by the is_subtask_of relation (e.g., is_subtask_of(collect_planning_data, create_short_term_plan)), then the aggregation relation exists between these performance indicators.

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

env_influence_on: ENV_CHARACTERISTIC x PI x {positive, negative}

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.

Other types of relations between PI, tasks and roles, related for example to power, supervision and authorisation (e.g., is_monitored_by: PI × ROLE, is_controlled_by: PI × ROLE) will be considered and elaborated in the organization-oriented view.

The relations for expressions over performance indicators, for goal patterns and for goals are specified below.

is_defined_over: PI_EXPRESSION × PI

PI expression is defined over the PI

is_based_on: GOAL_PATTERN × PI

The goal pattern in the first argument is defined over the PI in the second argument

uses: GOAL_PATTERN × PI_EXPRESSION

Goal pattern is defined over the PI expression

In goal patterns the symbols <, >, and = from PI expressions are interpreted as functions: PI x {NUM_VALUE, QUALIT_VALUE} → PI_EXPRESSION, where VALUE is a sort containing all numerical values, QUALIT_VALUE contains all qualitative values, and PI_EXPRESSION is a sort with all names of performance indicators expressions.

<u>Example:</u>

PI name: P2 (as defined earlier)
Definition: efficiency of allocation of security officers to objects

Goal pattern name: GP1 (as defined earlier)
Definition: maintained efficiency of allocation of security officers to objects = high

Let PI expression PE1 be P2 = high. Then: is_defined_over(PE1, P2), and uses(GP1, PE1), and is_based_on(GP1, P2).

is_formulated_over: GOAL × GOAL_PATTERN

The goal in the first argument is defined over the goal pattern in the second argument

<u>Example:</u>

Goal name: G3 (as defined earlier)
Informal definition: It is required to maintain high efficiency of allocation of security officers to objects.

is_formulated_over(G3, GP1)

Goals are related to tasks, roles and agents by the following relations:

is_realizable_by: GOAL x TASK_LIST

The goal in the first argument is realizable by the list of tasks in the second argument.

is_committed_to: ROLE × GOAL

The goal is an organizational goal and the role is committed to the satisfaction of this goal.

wishes: AGENT × GOAL

The goal is an individual goal and the agent wishes to satisfy the goal.

A goal structure can be built by refining high level goals (a top-down approach) and aggregating lower lever goals into higher level goals (a bottom-up approach). Since goals in the modelling framework can be of two types: hard and soft, different types of refinement relations should be considered.

First let us consider refinement of hard goals. Hard goals are refined into lists of hard goals (sort AND_GOAL_LIST), in which the goals are connected by AND relation. For this the following relations are introduced:

is_refined_to: GOAL × AND_GOAL_LIST

Defines a refinement of a hard goal into a list of hard goals, which contribute to its satisfaction. The refinement means that when all the goals in the list are satisfied then the goal in the first argument will be satisfied as well. If one or more goals in the list fail and no other refinement exists where all goals are satisfied, then the goal in the first argument will fail too. More formally, we introduce the predicates satisfied: GOAL and failed: GOAL to express the satisfaction state of a goal and these predicates can then be used to formulate the following axioms:

is_refined_to(g, l) & (∀gi: is_in_and_goal_list(gi,l) ⇒ satisfied(gi)) ⇒ satisfied(g)

∀ l: (is_refined_to(g, l) ⇒ ∃gi: is_in_and_goal_list(gi, l) & failed(gi)) ⇒ failed(g)

where

is_in_and_goal_list: GOAL × AND_GOAL_LIST

Expresses that the goal in the first argument is in the goal list of the second argument.

is_subgoal_of: GOAL × GOAL

The first argument is a goal which is a subgoal of the goal in the second argument meaning that it takes part in a refinement list of the second goal, e.g. is_subgoal_of(G2, G1), is_subgoal_of(G4, G1). (for hard goals only)

The relation between is_in_goal_list and is_subgoal_of is established by the following axiom expressing that if the goal G2 is refined into the list L, and G1 is one of the goals in the list L, then G1 is a subgoal of G2:

∀ G1, G2: GOAL, ∀ L: GOAL_LIST: is_in_goal_list(G1, L) & is_refined_to(G2, L)
⇒ is_subgoal_of(G1, G2)

When more than one refinements are defined, they are considered as alternatives connected by OR – they allow making a choice in what direction to go (which measures to take) in order to satisfy the desired goal. For example is_refined_to(G1, L1) and is_refined_to(G1, L2), where L1=[G2, G3] and L2=[G4, G5, G6] are lists of type AND_GOAL_LIST and all goals are hard, are alternative refinements of G1 meaning that G1 will be satisfied if G2 and G3 are both satisfied, however G1 will also be satisfied if G4, G5 and G6 are all satisfied. The figure 2 represents this example. We represent lists of type AND_GOAL_LIST graphically by drawing a double arc over the included links.

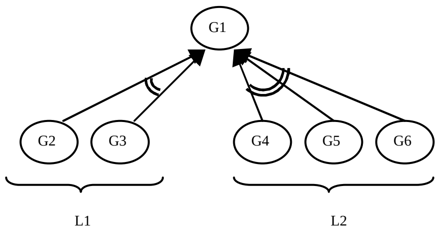

Fig. 2. Refinement of the hard goal G1 into two alternative and-lists of goals

Goal and performance indicators structures are closely related to each other. In particular, if goals are related by the refinement relation, then the performance indicators corresponding to these goals are related by a certain causality relation. This is expressed by the following axiom:

∀ G1, G2: GOAL, ∀ L: GOAL_LIST ∀ GP1, GP2: GOAL ∀ PI1, PI2: GOAL:
is_in_goal_list(G1, L) & is_refined_to(G2, L) & is_based_on(GP1, PI1) & is_formulated_over(G1, GP1) & is_based_on(GP2, PI2) & is_formulated_over(G2, GP2) ⇒ ∃pn:SIGN causing(PI1, PI2, pn),

where SIGN={very_negative, negative, positive, very_positive}.

Example:

In the planning department short-term plans are produced for the distribution of security officers to objects. A short-term plan is produced either when a new contract is signed and a plan is designed from scratch or when data for necessary changes arrives (sick leave, vacations, changes from the side of the client, etc.) and the existing plan needs to be adjusted. The earlier defined goal G3.1 expresses that all short-term plans should be available within 48 hours of receiving new relevant data. This goal can be decomposed in two sub-goals according to the situation – new plan or an update to an existing plan, meaning that if the two sub-goals are satisfied then G3.1 will also be satisfied.

Goal name: G3.1 (as defined earlier)
Informal definition: It is required to achieve that within 48 hours from receiving relevant operational data, an up-to-date short-term plan exists

Goal name: G3.1.1
Informal definition: It is required to achieve that within 48 hours from receiving a new contract, a new short-term plan is produced
Priority: medium
Horizon: short-term
Evaluation type: achievement goal (achieve goal pattern)
Ownership: organizational
Perspective: management, customer
Hardness: hard
Negotiability: negotiable

Goal name: G3.1.2
Informal definition: It is required to achieve that within 48 hours from receiving data about necessary changes in the short-term plan, an updated short-term plan is produced
Priority: medium
Horizon: short-term
Evaluation type: achievement goal (achieve goal pattern)
Ownership: organizational
Perspective: management, customer
Hardness: hard
Negotiability: negotiable

is_in_and_goal_list(G3.1.1, L)
is_in_and_goal_list(G3.1.2, L)
is_refined_to(G3.1, L)

is_subgoal_of(G3.1.1, G3.1)
is_subgoal_of(G3.1.2, G3.1)

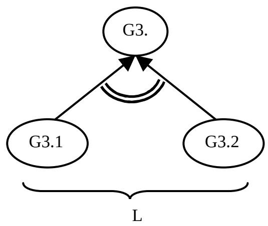

Fig. 3. Refinement of the hard goal G3 into the and-list of goals G3.1 and G3.2

Now let us consider the refinement of soft goals. Since the satisfaction of soft goals cannot be established in a clear-cut way, the process of refinement of soft goals also differs from the refinement of hard goals. The notion of refinement here carries a slightly different meaning from the one used for hard goals. It is more difficult to clearly define decomposition for soft goals. Instead we talk about positive contribution from other goals in the satisfaction of the goal to be refined. Such contribution can vary in its degree (i.e. strength) which is expressed by the following relations:

satisfices: GOAL $\times$ GOAL

The first argument points to a goal which strongly contributes in a positive way to the satisficing of the goal in the second argument. If the first goal is satisficed and any other influences are ignored then the second goal is also considered satisficed. The second goal is soft; the first can be soft or hard.

contributes_to: GOAL $\times$ GOAL

The first goal contributes positively to the satisficing of the first goal, however might not be enough to satisfice it. The second goal is soft, the first can be soft or hard.

The precise meaning of these relations is defined through the rules of propagation of the degree of satisfaction/satisficing of goals (inspired by but different from [26]), which are described in the following. The propagation rules are used for performance evaluation – to determine the degree of satisfaction/satisfycing of higher level goals based on the available information about the degree of satisfaction/satisfycing of the lower level (refinement) goals.

Table 1. The table for determining a degree of satisficing for higher-level goals based on the satisfaction label of lower level contributing goals and types of contributing links.

Type of the link Label of the contributing goal	satisfices	contributes_to
satisficed / satisfied	satisficed	weakly_satisficed
weakly_satisficed	weakly_satisficed	undetermined
undetermined	undetermined	undetermined
weakly_denied	weakly_denied	undetermined
denied / failed	denied	weakly_denied

Note that the propagated labels are not directly assigned to the higher level goal. All labels for the same goal are combined depending on the type of relations between the lower level goals. Lower level goals can be combined in lists using *and* and *balanced contribution* relations, which contribute positively to the satisficing of the higher level soft goal. Both sorts with names of "and"-lists (AND_GOAL_LIST) and "balanced contribution"-lists (BAL_GOAL_LIST) are subsorts of GOAL_LIST sort, over which the following relations are defined:

is_in_goal_list: GOAL × GOAL_LIST

The goal from the first argument is in the goal list of the second argument.

has_influence_from: GOAL × GOAL_LIST

This is the relation corresponding to is_refined_to for soft goals. The relation means that the goals in the list contribute positively to the satisficing of the soft goal in the first argument. For each goal in the list it is defined separately what the level is of its contribution (the type of the link) using the above defined relations satisfices and contributes_to. Soft goals can have contributions from either soft or hard goals and can be combined in lists of type AND_GOAL_LIST or BAL_GOAL_LIST.

Furthermore, some goal lists may be related by the *or* relation. This reflects the knowledge that these lists are in conflict or competition and if one is satisficed then the probability that the rest will also be satisficed is lower. They present design choice between alternative influences for satisficing a goal.

An and-list combines goals that need to be considered together, when their influence on the higher level goal is determined. If one goal is less than satisficed, then that decreases the overall influence of the whole list on the higher level goal. The combination of goals in an and-list implies that if all goals in the list are satisficed or satisfied then the higher level goal will also be satisfied. In order to ensure this the following constraint is enforced: at least one of the goals in an and-list is connected with a link of the type satisfices to the higher level goal. More precisely, when the links are connected by AND, the overall influence of the list is defined by the minimal label within the list using the above defined order between the labels. When the connection is OR, then the maximal label defines the overall influence.

<u>Example</u>

Consider the following hypothetical situation (see Figure 4). The soft goal G1 has two refinements – the and-list containing the soft goals G1.1, G1.2 and G1.3 and the and-list containing the soft goals G1.4 and G1.5. The two lists are considered as alternatives in an OR relation. Let all goals except G1.3 are connected to G1 through a satisfices-link while G1.3 is connected through a contributes_to-link. This can be represented as follows:

```
is_in_and_goal_list(G1.1, L1)        has_influence_from(G1, L2)
is_in_and_goal_list(G1.2, L1)        satisfices(G1.1, G1)
is_in_and_goal_list(G1.3, L1)        satisfices(G1.2, G1)
is_in_and_goal_list(G1.4, L2)        contributes_to(G1.3, G1)
is_in_and_goal_list(G1.5, L2)        satisfices(G1.4, G1)
has_influence_from(G1, L1)           satisfices(G1.5, G1)
```

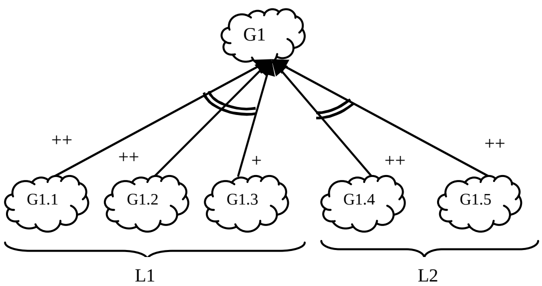

Fig. 4. Refinement of the soft goal G1 into two contributing and-lists.

Furthermore, let us have the levels of satisfycing of G1.1 to G1.5 based on measurement and observation. They are assessed to be as follows: G1.1 is satisficed, G1.2 is weakly satisficed, G1.3 is satisficed, G1.4 is also satisficed and G1.5 is weakly denied. We now can propagate this knowledge in order to find out the level of satisficing of G1. All lower level goals are assigned the appropriate labels corresponding to their level of satisficing and these labels are propagated taking into account the type of the links using the Table 1. They all result in the same label except for G1.3, which propagates the label weakly_satisficed. Therefore the labels originating from list L1 are {satisficed, weakly_satisficed, weakly_satisficed}. Taking the minimal label we conclude that L1 propagates weakly_satisficed. Analogously list L2 propagates weakly_denied. Between the two lists we take the maximal label and conclude that G1 is weakly satisficed.

Another kind of relation between goals represents balanced contribution which gives us the possibility to describe more fine-tuned ways of contributing which favour the majority influence.

For goals in a balanced contribution list the following relation is defined:

is_in_bal_goal_list: GOAL × BAL_GOAL_LIST

The goal from the first argument is in the goal list of the second argument.

The rule that is used to calculate the exact effect first quantifies the propagated labels of lower level goals (as shown in Table 1) and then takes the (weighted) average which will then be discretized again to the closest label, which will be the sought label for the higher level soft goal. The quantification scale can look as follows: *satisficed* = 2, *weakly_satisficed* = 1, *undetermined* = 0, *weakly_denied* = -1, *denied* = -2. In tie situations a common strategy can be adopted, for example assign the higher of the two closest labels (optimistic strategy). Using weights we can modify the results to emphasize some of the labels and thus increase their effect in the calculation. Weights can be assigned using different approaches depending on the available domain knowledge. One approach is to emphasise one type of influence versus the other, e.g. increase the effect of all links with strong influence (++) by giving a higher weight value while giving a lower weight value to all weaker links (+). Another approach can be applied if the designer wants to make finer distinction between links of the same type. While a strong link (++) should still have at least as high weight as a weak link (+), it is possible, for example, to give different weight values to all strong links in order to increase the strength of some and decrease the strength of others and thus to fine-tune their influence on the final result. Let the quantified labels of the goals in the balanced list be g_i and the weights defined for each goal in the list are w_i. Then the influence of the balanced list on the higher level goal is calculated using a formula of the type: $\sum_i w_i g_i \Big/ \sum_i w_i$

In order to specify the weight of a goal in a balanced list we define the following relation:

has_weight_in_list: GOAL × INTEGER × BAL_GOAL_LIST

When a goal is refined in one list only then the calculated influence (using the above defined rules) defines the satisficing label of the goal. In situations when a goal is refined in more than one list we use the following strategy: first the influences of the and- and bal-lists are calculated separately and then combined according to the rule for OR connection.

A list of type BAL_GOAL_LIST is represented by a single arc drawn over the links of the members of the list.

Relations of type satisfices are distinguished in a graphical representation by a double + on the link while contributes_to is depicted by a single + on the link.

Example

Goal name: G7
Informal definition: It is required to maintain optimal number of qualified personnel
Priority: medium
Horizon: long-term
Evaluation type: development goal (maintain goal pattern)
Ownership: organizational
Perspective: management
Hardness: soft
Negotiability: negotiable

Goal name: G7.1
Informal definition: It is required to maintain high qualification of current personnel
Priority: medium
Horizon: long-term
Evaluation type: development goal (maintain goal pattern)
Ownership: organizational
Perspective: management
Hardness: soft
Negotiability: negotiable

Goal name: G7.2
Informal definition: It is required to maintain up-to-date data of the available and needed (human) capacity and qualifications
Priority: medium
Horizon: long-term
Evaluation type: development goal (maintain goal pattern)
Ownership: organizational
Perspective: management
Hardness: soft
Negotiability: negotiable

Goal name: G7.3
Informal definition: It is required to maintain timely recruitment and dismissal of personnel according to the data on available and needed (human) capacity and qualifications
Priority: medium
Horizon: long-term
Evaluation type: development goal (maintain goal pattern)
Ownership: organizational
Perspective: management
Hardness: soft
Negotiability: negotiable

```
is_in_bal_goal_list(G7.1, L1)
is_in_bal_goal_list(G7.2, L1)
is_in_bal_goal_list(G7.3, L1)
satisfices(G7.1, G7)
satisfices(G7.2, G7)
satisfices(G7.3, G7)
has_influence_from(G7, L1)
has_weight_in_list(G7.1, 3, L1)
has_weight_in_list(G7.2, 1, L1)
has_weight_in_list(G7.3, 1, L1)
```

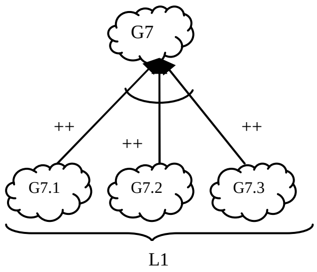

Fig. 5. Refinement of the soft goal G7 into the contributing balanced list that consists of goals G7.1, G7.2 and G7.3.

Let us assume that the degrees of satisficing of the lower-level goals G7.1, G7.2 and G7.3 are known. Let G7.1 be satisficed and G7.2 and G7.3 be weakly satisficed. The labels are quantified so that the degree of satisficing of G7.1 is considered 2 and of G7.2 and G7.3 – 1. Therefore the degree of satisficing of G7 is calculated as (3*2+1+1)/5 = 1.6 which we round up to 2 (which corresponds to satisficed in our scale). Thus the label assigned to G7 is 2.

Apart from the refinement links discussed so far, we can also define conflicts, which represent negative relations between goals or lists of goals.

conflicts_with: AND_GOAL_LIST × AND_GOAL_LIST

Represents joint negative effect between lists of goals which can be hard or soft and the meaning would be that the goals in both lists cannot be satisfied, satisficed or weakly_satisficed at the same time. The precise meaning of this relation can be defined using the following rules. If all goals in one list are satisfied or satisficed then at least one goal in the other list is failed or denied. If all goals in one list are at least weakly satisficed then at least one goal in the other list is at most weakly denied. Here we use the previously defined ordering.

Note that this relation is only defined for and-lists and that it is possible to have the situation that in each list there is one or more goals (weakly) satisfied/satisficed as long as there is in at least one of the lists at least one goal which is (weakly) failed/denied.

weakly_conflicts_with: AND_GOAL_LIST × AND_GOAL_LIST

Represents weak joint negative effect between lists of goals and the meaning would be that the goals in both lists cannot be satisfied or satisficed at the same time. More precisely we formulate the following rules. If all goals in one list are satisfied or satisficed then at least one goal in the other list is at most weakly denied. If all goals in one list are at least weakly satisficed then at least one goal in the other list is at most weakly satisficed.

Conflicts can be specified at the design phase and often represent the points of view of different stakeholders. It is important that they are acknowledged so that measures can be taken to address the problems e.g. by negotiation between the corresponding stakeholders. Sometimes conflicts cannot be addressed immediately when detected and remain in the goals structure until at later stages a solution is found. Conflicts can be defined at each two levels of the goals hierarchy, however if the hierarchy is sufficiently complete then these conflicts should be reflected on the lower levels. For example if two higher level goals are in conflict then there should also be a conflict between some of the goals in their refinements. In this way we can propagate each conflict to the lowest level of the structure. Similar reasoning can be used when only the conflicts at the lower levels of the structure are known – then they can be propagated to the higher levels as well. This rule can be used at the design phase, while building the goals structure.

Conflicts can also be used at the analysis and evaluation phases by giving insight, verification by inconsistencies analysis as well as organizational performance evaluation by propagating satisfaction labels bottom-up when only partial information is available. For example let goals g1 and g2 be in conflict at the lowest level of the goals structure and let g1 be known to be satisfied. Then if the satisfaction label of g2 is not known it can be assumed to be at most weakly denied if g2 is soft and failed if g2 is a hard goal. If however it is known that g2 is also satisfied/satisficed, then that points at an inconsistency in the model which should be corrected.

Conflicts are represented graphically using dashed lines joined by a double arc for an and-list with a single minus for weak conflicts and a double minus for strong conflicts.

Examples:

Goal name: G3.4
Informal definition: It is required to maximize the time spent on examining the plan proposal for correctness
Priority: medium
Horizon: short-term
Evaluation type: development goal (optimize goal pattern)
Ownership: organizational
Perspective: management
Hardness: hard
Negotiability: negotiable

```
is_in_and_goal_list(G3.1, L)
is_in_and_goal_list(G3.4, L1)
```

```
conflicts_with(L, L1)
```

Goal name: G8
Informal definition: It is required to minimize training for current personnel
Priority: medium
Horizon: long-term
Evaluation type: development goal (optimize goal pattern)
Ownership: organizational
Perspective: management
Hardness: soft
Negotiability: negotiable

Goal name: G9
Informal definition: It is required to minimize recruitment of new personnel
Priority: medium
Horizon: long-term
Evaluation type: development goal (optimize goal pattern)
Ownership: organizational
Perspective: management
Hardness: soft
Negotiability: negotiable

```
is_in_and_goal_list(G7, L2)
is_in_and_goal_list(G8, L3)
is_in_and_goal_list(G9, L4)
```

```
weakly_conflicts_with(L2, L3)
weakly_conflicts_with(L2, L4)
```

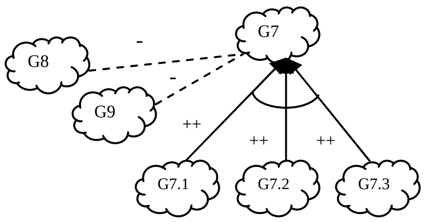

Fig. 6. Refinement of the soft goal G7, which stands in conflict relations to goals G8 and G9

It is also possible to define preferences between goals using the following relation:

is_preferred_over: GOAL × GOAL

The first goal points to a goal which has higher priority than goal indicated by the second goal. This relation builds a directed graph where the goals are the nodes and the relation defines the edges. Note that we also defined a priority value for goals. We leave the choice to the designer which of the two approaches fits better the domain knowledge available for the particular case. If more precise information about the priorities is extracted then it might be beneficial to define priority values which will provide more possibilities for analysis when a decision needs to be taken. Otherwise, when less information is available, the relation is_preferred_over will be easier to define. Here, however, additional methods need to be employed in order to prevent inconsistencies (e.g. g1 is preferred over g2, g2 is preferred over g3 but g3 is preferred over g1, which is a directed loop in the graph and can be discovered with graph-theoretical algorithms). It is even possible to use both approaches at the same time with the appropriate inconsistency prevention methods.

7. Performance evaluation

Every task performed in an organization contributes to the satisfaction of a certain organizational goal(s). Each goal is formed based on a certain performance indicator(s). This performance indicator(s) can be measured (directly or indirectly) during or after the task execution depending on the goal evaluation type, in the end or during a certain period of time (an evaluation period defined as a goal horizon). After that, by comparing the measured value(s) with the corresponding goal expression(s), the satisfaction (or a degree of satisficing) of the goal(s) is determined. Further, the obtained goal satisfaction (or satisficing) measure is propagated by applying the rules defined in Section 6, upwards in the goal hierarchy for determining the satisfaction (or a degree of satisficing) of higher level goals. Thus, the organizational performance is evaluated by determining the satisfaction of or a degree of satisficing of key organizational goals. The same principles can be applied for the evaluation of agent performance.

In situations, in which only partial information about the execution of tasks in the organization is available, by exploiting conflict relations between goals, for which satisfaction or satisficing degrees are known and goals in the undetermined state, it is possible to evaluate with a certain accuracy the degree of satisfaction of undetermined goals and make a prediction about the performance of the organization.

As illustration of the proposed performance evaluation procedure consider the following example. For estimating the performance of the organization introduced in Section 4, a degree of satisfaction of hard goal G3.1 defined in Section 5 has to be determined. This goal is refined into the and-list that consists of two hard goals: G3.1.1 and G3.1.2. These lower level roles are related to tasks specified in the task graph created for this case study. In particular, G3.1.1 is related to the task "generate a new short-term plan" and G3.1.2 is realized by the task "update the short term plan". By measuring the actual task execution during the evaluation period defined for the goal G3.1 (a month), it was determined that values for the both performance indicators corresponding to the goals G3.1.1 and G3.1.2 (time needed to create a short term plan and time needed to update a short term plan) do not exceed 48 hours. Therefore, both goals G3.1.1 and G3.1.2 are satisfied. Due to the refinement relation the goal G3.1 is also satisfied and contributes maximally to the estimation of overall performance of the organization.

8. Methodological issues

Methodological issues discussed in this Section include the construction and the revision of the performance indicators and goal structures, the identification and handling conflicts between goals. Interdependencies between different structures of an organization (e.g., goal, performance indicators and task structures) play an important role in addressing these issues.

As it was discussed in Section 5, performance indicators for organizations can be extracted from different sources (e.g., mission statements, strategy definitions, job descriptions, laws, sets of requirements for organizations being designed). Other strategies for obtaining organization performance indicators are described in [1, 17, 8].

In order to build a structure of performance indicators, 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 performance indicators may be used;
3. performance indicators attributes and relations between these attributes (e.g., relations between time-related attributes and attributes that relate performance indicators to the organization and task structures) can be exploited (briefly described in Section 5);
4. from the existing relations in the performance indicators structure some new relations may be inferred;
5. data mining techniques may be applied to the data collected during a certain period of the organization operation;
6. intuitions of the modeller may be applied that should be tested by domain experts or simulations;
7. relations to the organization task structure and to the goal structure may be exploited.

As it follows from the definitions in Section 6 all the considered types of relations between performance indicators can be reduced to the causality relations. The technique (4) allows inference of some missing causality relations from the existing performance indicators structure. In general the inference rules are specified in the form causing(p1, p2, s1) & causing(p2, p3, s2) ⇒ causing(p1, p3, s3), where p1, p2 belong to the sort PI and s1, s2, s3 are of sort SIGN={very_negative, negative, positive, very_positive}. More specific (instantiated) inference rules are generated based on the table below, in which s3 values are given in the cells on the intersection of columns containing s1 values with rows containing s2 values.

Table 2. Inference table for generating inference rules in the form causing(p1, p2, s1) & causing(p2, p3, s2) ⇒ causing(p1, p3, s3); s3 values are given in the cells on the intersection of columns containing s1 values with rows containing s2 values.

s1 / s2	Very Negative	Negative	Positive	Very Positive
Very Negative	Very Positive	Very Positive	Very Negative	Very Negative
Negative	Very Positive	Positive	Negative	Very Negative
Positive	Very Negative	Negative	Positive	Very Positive
Very Positive	Very Negative	Very Negative	Very Positive	Very Positive

The obtained inference rules can be also used for the verification of integrity of the performance indicators structure.

Further let us consider the technique (7) in more detail. The task structure of an organization may provide useful insights to determine relations between performance indicators. In many cases refinement relations specified in the task structure correspond to the causality relations in the performance indicators structure. For example, based on the refinement relation between the task “create a correct plan” (related to the performance indicator “time for creating a correct plan”) and its subtask “check a plan” (related to the performance indicator “time to check a plan”), the performance indicators “time to check a plan” and “time for creating a correct plan” are related by the positive causing relation. In some cases refinement relations can be also reflected by other types of relations in the performance indicators structure.

Further, as it follows from the goal definition given in Section 5, goals and performance indicators form two highly interrelated structures. It means that changes in one structure almost always imply changes in the other structure. Thus, the performance indicators 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 performance indicator structure by the corresponding relation between performance indicators, on which the considered goals are based. In such a way a goal-structure is created by a top-down design process. The refinement of goals may proceed until subgoals are found, which could be realized by (possibly single) lowest-level tasks from the task hierarchy. In practice, the top-down design approach is often combined with the bottom-up approach, which is performed by aggregation of goals. For example, in the goal elicitation approach described in [4] subgoals are identified by asking “how” questions about the goals already determined, and parent goals are identified by asking "why" questions. 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 performance indicators.

To fine-tune goal and task structures, and relations between them at the design phase, backwards reasoning approaches on a goal structure [19] can be used. These approaches are particularly useful for the analysis of cases of a soft goal refinement. More specifically, given that a higher level soft goal is required to be satisfied to a certain degree, and provided the type of a list into which this goal is refined (i.e., and or balanced) and types of refinement links between goals, it is possible to determine the least degree of satisfaction of the lower level goals from the refinement list. This information constitutes kind of constrains on lower-level goals that can be used for the revision or (re)formulation of goals and corresponding tasks, and relations between them.

Since goal and performance indicators structures are closely related, it is important to guarantee consistency and correspondence of these structures to each other. For this a number of consistency checks can be performed, which are based on the principles described in the following. If goals are related by the refinement relation (the subgoal relation in case of hard goals and different degrees of contribution relation in case of soft goals), then the performance indicators corresponding to these goals, are related by a certain causality relation. Furthermore, if the performance indicator expressions for goals related by refinement, contain comparison functions (such as '>','<') or measures of degrees (such as 'high', 'low'), or goal patterns are specified by change 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 one month" (the PI expression is "the duration of the reviewing process < 1 month") has one of the subgoals specified as "It is desired to increase the number of reviewers" (the goal pattern is "increase(number of reviewers)"). Since goals are related by refinement, the performance indicators "number of reviewers" and "the duration of the reviewing process" should be related by the negative causality relation in the performance indicators structure.

The identification of conflict relations between goals is of particular importance for the design and the evaluation of organizations. In order to create an effective organization, it is often advised at the early design phase to take into consideration interests and concerns (expressed as goals) of different stakeholders, who will eventually play a role within the organization and will interact with the organization. The stakeholders may have conflicting goals that should be reflected in an organization model being constructed. Furthermore, conflicts may exist in a goal set of a stakeholder. To identify conflicts between goals, the goal patterns and the performance indicators structure can be used. More specifically, by knowing the type of a causality relation between performance indicators and the types of goal patterns, the presence of a conflict between goals can be determined. For example, the goal "It is required to maximize the time spent on examining the plan proposal for correctness" and the goal "It is required to minimize the time spent on producing a correct plan" are in conflict, since the corresponding performance indicators "the time spent on examining the plan proposal for correctness" and "the time spent on producing a correct plan" are related by the positive causality relation, and the corresponding goal patterns are built based on the opposite types of functions: maximize and minimize. If during the design phase a conflict between high level goals is determined, then through the refinement a more precise cause of the conflict can be found at the lowest level of a goal structure. For this the relations between performance indicators and the domain knowledge are exploited.

For those organization models that do not allow conflicts, the consistency of a model can be archived by applying different conflict resolution techniques [20]. The common strategy for the conflict resolution is based on weakening of goal expressions (e.g., by weakening boundary conditions in the performance indicator expressions; by introducing so-called 'organizational slacks' [13]). For example, the goal "It is required to achieve that within 48 hours an up-to-date short-term plan exists" may be weakened to "It is required to achieve that within 54 hours an up-to-date short-term plan exists". When the conflicting goals are modified, the goal priority attributes should be taken into account (e.g., to determine, which goal should be modified to a greater degree or even should be deleted from a model). For example, in general organization goals have the higher priority than individual goals of agents. Therefore, in order to fit into the organization, an agent sometimes needs to adjust her/his own goals to the organizational ones. On the other hand, sometimes priorities of goals of an agent (e.g., important customer, government) can be so high that the organization decides to revise its goal structure to ensure the satisfiability of agent goals. For negotiable goals conflicts can be solved by performing negotiations among the stakeholders, to which goals are related [28].

9. Comparison to Other Goal-Oriented Approaches

Goals and performance indicators take a central position in the performance-oriented view. In this section we briefly review the most relevant body of literature to these two notions as used in the framework.

9.1 Goals modelling

In requirements engineering it has long been recognized that it is necessary to consider the organizational context of the designed information system since the system is intended to improve the existing business situation. This gave rise to a lot of research efforts in the area of enterprise modelling which aims at specifying and analysing the current organization's structure and behaviour and the desired points for change. Here the notion of a goal plays an important role [4]. Often both organizational and individual goals of the involved actors can be considered and distinction can be made between the goal originating from different stakeholders. Also in the later stages of requirements engineering the notion of a goal has been extensively used however the focus lies differently.

Some aspects of our definition of the notion of a goal are inspired and come close to existing state-of-the-art approaches in enterprise modelling and requirement engineering [4, 30]. There are however significant differences as well which will be pointed out here. Our analysis pinpointed the following approaches as most relevant which will be discussed here – CIMOSA [3], TOVE [10], i* [30, 31], Tropos [2, 12], the agent-oriented enterprise meta-model of [16], KAOS [4], the NFR framework [26] ordered roughly in an increasing degree of relevance to this discussion.

In CIMOSA the notion of objectives is used to represent business goals for a particular domain (i.e. a part of the enterprise). No relationships between the objectives are defined therefore no hierarchy of objectives is built. No distinction is made between hard and soft goals.

Also in the TOVE model no distinction is made between hard and soft goals. Goals can be decomposed in AND/OR subgoal trees and in this way a goal hierarchy is built.

The i* approach focuses on the dependencies relationships between the actors and builds the so-called Strategic Dependency Model. Separately a Strategic Rationale model is built on the level of each actor where its internal reasoning on the relationships between goals, tasks and resources can be modelled. The approach recognises both hard and soft goals and defines a (soft)goal dependency relationships between actors with respect to (soft)goals expressing the knowledge that one actor (dependum) depends on another (dependee) to make a condition in the world come true. Only the end state is specified, thus the dependee has the freedom to decide on how to achieve it. The goals are only informally specified therefore no format and unified representation is enforced. The goals hierarchy is not decoupled from the tasks hierarchy as tasks can be decomposed to goals and tasks. This intertwining of the two concepts might sometimes blur the difference between them and make the model less comprehensible. Positive and negative contribution to a different degree of tasks or goals to soft goals can be modelled using several types of contribution links.

Tropos is a methodology for agent-oriented software development based on i*. Goals are defined as to represent actors' strategic interests. Distinction is made between hard and soft goals. Goals are modelled and related from the point of view of a particular actor in a similar spirit as the Strategic Rationale model of i*. Dependencies between actors for achieving goals are modelled. Contribution, positive or negative, of goals to the fulfilment of other goals can be modelled as well as AND/OR decomposition of goals into other goals. The extension Formal Tropos [12] uses a temporal specification language inspired by KAOS (see below for a discussion on KAOS).

The agent-oriented enterprise meta-model presented in [16] defines a goal as a desired or undesired state of the environment which is described by states of objects (beliefs, authorisations, resources, etc.). Goals can be refined into alternative sets of other goals using AND/OR relationship. Distinction is made between operational and soft goals – plans can fulfil operational goals but can only contribute positively or negatively to soft goals. Goals can also be organizational or personal. A dependency relationship between organizational roles for the fulfilment of organizational goals is defined.

The KAOS methodology focuses on the process of requirements elaboration and provides support in connecting high-level goals to operations, objects and constraints to be implemented by the software. A goal is defined as a non-operational objective to be achieved by the composite system while an operational objective is called a constraint. Goals and constraints are defined formally using the patterns achieve, cease, maintain, avoid and optimize which are reused in our approach through the notion of a goal pattern. A difference here is that the goal pattern in our approach is based on an expression over a performance indicator. Soft goals are not considered in KAOS. Goals are structured and operationalized to constraints in AND/OR graphs. Temporal logic is used to define the goals and their relationships. This is possible because the goals considered in KAOS are very operational and can be defined as logical statements. Organizational goals as defined in our approach are not always operational and sometimes soft. This reflects the way organizations define their goals in practice.

The NFR framework focuses on the representation of non-functional requirements on the designed software system through interrelated goals. Three types of goals are defined: NFR goals, satisfycing goals and argumentation goals. The last two model design decisions and arguments respectively and are therefore not relevant for this discussion. The NFR goals are soft goals which can be refined using different types of relationships describing how the satisfycing or denial of the offspring relates to the satisfycing of the parent goal. A labelling procedure is defined for determining the degree satisfycing of each node in the goal structure. The label propagation procedure used in our approach is inspired by but different from the one used in the NFR framework. We consider only positive refinement links. The negative links are modelled using conflict links which can also relate goals at the same level in the goals hierarchy. The conflict links are propagated to the lowest levels of the hierarchy and only there they can be used in the propagation of labels. This will only be necessary if the label of some of the lowest level goals is not known. Sets of goals in conflict with one or more other goals can only be combined using an AND-relation. Furthermore, we enrich the refinement structure with one more relation in addition to AND and OR representing balanced contribution and providing tools for finer definition of how a set of goals together contributes to the satisfycing of the higher-level goal.

9.2 Performance measurement

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 performance indicator is nearly invisible in the current methodologies for enterprise modelling. Since performance measurement is a central issue for every organization, the organization's model should take it into account. This however is currently done implicitly at best. We are only aware of one methodology, GRAI [7] which explicitly models performance indicators however only in the context of decision making and without taking into account the relationships among the performance indicators and between the performance indicators and other notions such as goals.

Letier et al. on the other hand define in [22] quality variables which can be related to the performance indicators defined 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 is that in [Letier et al.] probabilistic reasoning is used to determine the partial satisfaction of goals which is reflected in the definitions of objective functions and goals.

It should be noted that sometimes measures such as customer satisfaction, profit, production costs, delivery time (typical performance indicators) are visible in other models as well – often in the definition of goals but they always remain implicit and the relationships between them are usually not discussed.

10. Conclusions

This paper presents a performance oriented organizational meta-model which constitutes a part of a general organizational meta-model aiming at providing tools for modeling all important aspects of an organization. Within the performance-oriented view in particular a diverse vocabulary and methodological guidelines are provided for expressing the available knowledge within the organization with respect to performance measurement and evaluation such as: organizational or individual goals either hard or soft and how they contribute or conflict with each other, organizational or individual performance indicators and how they influence each other, etc. This allows building structures that can be used for complex analysis both 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 will also be presented separately together with how they relate to each other.

References

1. Azzone, G., Masella, C., and Bertelè, U.: Design of performance measures for time-based companies. International Journal of Operations & Production Management, Vol. 11, No. 3, (1991) 77-85
2. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J.: Tropos: An Agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems, 8 (2004) 203-236
3. CIMOSA – Open System Architecture for CIM; ESPRIT Consortium AMICE, Springer-Verlag, Berlin (1993)
4. Dardenne, A., van Lamsweerde, A., and Fiskas, S.: Goal directed requirements acquisition. Science of Computer Programming, 20 (1993) 3-50
5. Dastani, M., Hulstijn, J., Dignum, F., and Meyer, J-J.: Issues in Multiagent System Development. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems AAMAS'04 (2004) 922-929
6. David, R., Alla, H.: Discrete, Continuous, and Hybrid Petri Nets. Springer-Verlag (2005)
7. Doumeingts, G., Vallespir, B., and Chen, D.: Decisional Modelling using the GRAI Grid, In : Bernus, P., Mertins, K. and Schmidt, G. (Eds): Handbook on Architectures of Information Systems, Springer-Verlag (1998) 313-338
8. Flapper, S.D.P., Fortuin, L., and Stoop, P.P.M.: Towards consistent performance management systems. International Journal of Operations and Production Management, 18 (1996) 604–625
9. Forrester, J. W.: Industrial dynamics, Waltham, MA: Pegasus Communications (1961)
10. Fox, M.S.: The TOVE project: towards a common-sense model of the enterprise. In Petrie, C.J., Jr.(Ed): Proceedings of ICIEMT'92, MIT Press (1992) 310-319
11. Fox, M.S., Barbuceanu, M., Gruninger M., and Lin, J.: An Organization ontology for enterprise modelling. In: M. Prietula, K. Carley and L. Gasser (Eds): Simulating Organizational: Computational Models of Institutions and Groups, AAAI/MIT Press (1997) 131-152

12. Fuxman, A., Pistore, M., Mylopoulos, J., and Traverso, P.: Model checking early requirements specification in Tropos. In: Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (RE'01) (2001) 174-181
13. Galbraith, J.R.: Organization design, Addison-Wesley Publishing Company, London Amsterdam Sydney (1978)
14. GERAM: The Generalized Enterprise Reference Architecture and Methodology. IFIP-IFAC Task Force on Architectures for Enterprise Integration, In : Bernus, P., Nemes, L. and Schmidt, G. (Eds): Handbook on Enterprise Architectures, Springer-Verlag (2003) 21-63
15. Horling, B., and Lesser, V.: A Survey of Multi-Agent Organizational Paradigms. The Knowledge Engineering Review, Volume 19, Number 4, Cambridge University Press (2005) 281-316
16. Jureta I., and Faulkner, S.: An Agent-oriented meta-model for enterprise modelling. In: J. Akoka et al (Eds): ER Workshops 2005, LNCS 3770 (2005) 151-161
17. Kaplan, R.S., and Norton, D.P.: The balanced scorecard – measures that drive performance, Harvard Business Review, January-February (1992) 71-9
18. Kavakli, E., and Loucopoulos, P.: Goal Driven Requirements Engineering: Evaluation of Current Methods. Proc. 8th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD '03) (2003)
19. Konar, A.: Artificial intelligence and soft computing.CRC Press (2000)
20. Lamsweerde, A. van: Goal-oriented requirements engineering: a guided tour. In: Proceedings of 5th IEEE International Symposium on Requirements Engineering, (2001) 249-263
21. Lamsweerde, A. van, Darimont, R., and Letier, E. Managing conflicts in goal-driven Requirements Engineering. IEEE Transaction on Software Engineering (1998)
22. Letier E., and Lamsweerde, A. van: Reasoning about partial goal satisfaction for requirements and design engineering. Proceedings of the 12th ACM SIGSOFT International symposium on Foundations of software engineering (2004) 53 – 62
23. Manzano, M.: Extensions of First Order Logic, Cambridge University Press (1996)
24. Marlow, W. H.: Mathematics for Operations Research. New York: Dover (1993)
25. Mintzberg, H.: The Structuring of Organizations. Prentice Hall, Englewood Cliffs, N.J. (1979)
26. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: a process-oriented approach. IEEE Transactions on Software Engineering, vol. 18, no. 6 (1992) 483-497
27. Scott, W.R.: Institutions and organizations. 2nd edn, SAGE Publications, Thousand Oaks London New Delhi (2001)
28. Sycara, K.: Resolving goal conflicts via negotiation. In Proceedings of the Seventh National Conference on Artificial Intelligence (1988) 245-250
29. Vázquez-Salceda, J., Aldewereld, H.M., and Dignum, F.P.M.: Norms in multiagent systems: From theory to practice. International Journal of Computer Systems Science & Engineering, 20(4) (2005) 225-236
30. Yu, E.: Towards Modelling and Reasoning Support for Early phase Requirements Engineering. 3rd IEEE International Symposium on Requirements Engineering, RE'97, Washington D.C., USA (1997)
31. Yu, E., Bois, Du Bois, P., Dubois, E., Mylopoulos, J.: From Organization Models to System Requirements - A "Cooperating Agents" Approach. 3rd International Conference on Cooperative Information Systems - CoopIS-95, Vienna, Austria (1995)