Performance Measurement Framework with Formal Indicator Definitions

Aivars Niedritis, Laila Niedrite, Natalija Kozmina

Faculty of Computing, University of Latvia, Raina boulv.19, Riga
{Aivars.Niedritis, Laila.Niedrite, Natalija.Kozmina}@lu.lv

Abstract. Definition of appropriate measures of organization's performance should be conducted in a systematic way. In this paper the performance measurement and indicators are discussed not only from the side of management models, but also from the point of view of measurement theories to find out appropriate definitions. In our work we propose a formal specification of indicators. The principles of indicator reformulation from free form indicators to formal requirements are formulated and applied in several examples from performance measures database. The formally defined indicators could be used in the proposed performance measurement framework that covers five-step indicator lifecycle.

Keywords: performance measurement, key performance indicators, data warehouse.

1 Introduction

In a long-term perspective it is necessary not only to understand the current situation in an organization and to rebuild the business processes, if necessary, in the most effective way, but also to continue the improvement of business processes based on comprehensive measurement of organization's performance.

Effective organization of business processes ensures the achievement of institution's goals. During a performance measurement, the measurement results should be compared with the target values to make decisions, whether goals are achieved or not. Organizations use performance measures to align daily activities to strategic objectives [1]. The role of appropriate measures could not be underestimated, so Harrington [2] stated: "Measurements are the key. If you cannot measure it, you cannot control it. If you cannot control it, you cannot manage it. If you cannot manage it, you cannot improve it."

An important aspect is how to choose appropriate measures and how to define an appropriate measurement framework. The performance measurement should be performed from different perspectives.

Companies should use performance measurement systems, if they want to succeed in the competition with other companies. Measurement systems have to be developed according to the strategies of the company and according to some management model, e.g. Balanced Scorecard. A data warehouse could also be used for implementation of a Performance Measurement System. The related work in this field is discussed in following sections.

The definition of performance indicators should be based on the strategy of the company and for that purpose an appropriate method should be used. The indicators can be defined on various levels of formality. Some proposals on how to specify formally performance indicators exist and are described in [3], [4].

The authors of [3] propose a formal language for a modelling of goals based on performance indicators. The goal satisfaction could be controlled and the evaluation of the organizational performance could be performed. The authors of [4] propose a formal language for the indicator definition by introducing the sorts of indicators, predicates and functions included in it. Relationships between indicators could also be defined.

In our work we propose a formal specification of indicators that could be used in the performance measurement framework, which covers five-step indicator lifecycle and is also proposed in this paper.

The rest of the paper is organized as follows. The 2nd section introduces concepts of the performance measurement. The 3rd section describes performance measurement systems. The 4th section describes the performance measurement framework with an indicator lifecycle. The 5th section defines a requirement pattern and explains how it could be used for a formal definition of indicators. The 6th section ends the paper with conclusions and a description of the future work.

2 Performance Measurement Concepts

The definition of appropriate performance measures should be performed in a systemic way, based on well known approaches. The real world experience shows that companies use wrong measures [1], many of which are incorrectly treated as key performance indicators (KPIs). There is a lack of understanding what is and what is not a KPI, how success factors are connected with KPIs and organization's strategy.

2.1 Key Concepts: from Strategy to Measures

Before starting a discussion about formalization and choosing of appropriate performance measures, notions regarding performance measures have to be introduced.

Critical success factors (CSFs) [1] are issues or aspects of organizational performance that determine ongoing health, vitality and well-being. Usually from 5 to 8 CSFs are included in such list.

Success factors (SF) [1] are approximately 30 issues or aspects of organizational performance that are important in order to perform well in any given sector/industry. The most important of them are CSFs.

Performance measures [1] refer to indicators used by management to measure, report and improve the performance in an organization. Performance measures are classified as key result indicators, result indicators, performance indicators, or key performance indicators.

- Key result indicators (KRIs) represent summaries about many activities in an organization's CSF, but they do not help to understand what should be improved within organizations. KRIs can be financial and non financial.
- Result indicators (RIs) summarize some activities within CSF/SF, they are usually a result of more than one activity, but like KRIs they do not give information on what and how to improve. All financial performance measures are RIs,
- Performance indicators (PIs), on the contrary, „tell you what to improve" [1], because PIs measure a discrete activity. PIs are non financial.
- Key Performance Indicators (KPIs) „tell you what to do to increase performance dramatically" [1], KPIs represent the set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization.

The set of used performance measures is influenced by *management models* of organizations, e.g. Balanced Scorecard (BSC) [5], and *measurement perspectives* defined within these models. In [5] four measurement perspectives are defined: Financial, Customer, Internal Process, and Learning and Growth. In [1] two more perspectives are added to the above mentioned in BSC – Environment/Community and Employee Satisfaction.

2.2 Features of Indicators

Indicators can be characterized according to different measurement aspects [1], [6].

- *Perspectives*. This aspect was already mentioned above. It should be added that not all indicator types cover all perspectives, e.g. financial perspective is related to KRIs and RIs, but not to PIs.
- *Time*. Subtypes of time aspect could also be considered – measurement periods when values of indicators are assigned and reporting periods that define the amount of historical data that should be included into reports. Although, KRIs are typically measured monthly, the time period for reporting may include even longer periods, e.g. year. KPIs are measured more frequently, e.g. daily or weekly.
- *Responsibility aspect*. Persons responsible for different types of performance measures could be at different levels starting from the top management to an individual level, where, for instance, in case of PIs all required actions are known and could be performed.
- *Activities*. RIs cannot be tied to a discrete activity; PIs, on the contrary, are tied to a discrete activity.
- *Success Factors*. Performance measures influence one or many CFS or SF depending on types of performance measures, e.g. KPIs impact more than one CSF /SF.
- *Reporting aspect*. The results for different types of performance measures could be reported at different levels starting from top management to individual level. For example, in case of KPIs results are reported to top management, they help to understand the required actions, and then the responsibility can be assigned to the individual level. So, Responsibility aspect and Reporting aspect not always have the same meaning.

2.3 Measurement Concepts

We can look at the performance measurement and indicators not only from the side of management models, but also from the point of view of measurement theories to find out appropriate definitions.

Measurement is [7] "the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to characterize the attributes by clearly defined rules."

There exist ontologies, methods [8], [9], [10] for determination of indicators in some industries, e.g. Software process measurement. In business process measurement, the measurement is considered in the context of its applications, e.g. the business process reengineering or the performance measurement. The measurement concepts are not usually discussed in the case of business process measurement.

Further some definitions concerning the measurement are given to explain how the indicators are gained according to the measurement theories mentioned above.

Before the measurement, the information needs should exist, which are necessary for decision making. In the measurement, the following hierarchical structure is considered: attributes → base measures → derived measures → indicators → information.

A measured attribute is a property of an entity that can be identified.

A base measure is a measure of one attribute. The measurement method, if performed, assigns a value for this base measure.

A derived measure is defined as a function of two or more base or other derived measures.

An indicator is a measure that ensures the evaluation for particular attributes and is gained by means of an analysis which is performed according to an analysis model. Analysis model is an algorithm that combines two or more base and/or derived measures with decision criteria. Indicators provide the basis for a decision making and supply analysts with the necessary *information*.

Decision criteria are numerical goals that are used to estimate if certain activities should be performed or if further investigation of the situation is necessary.

3 Performance Measurement Systems

Companies use measurement systems to evaluate their performance. Management models, e.g. the Balanced Scorecard, should be used to develop such system to choose appropriate measures and to define the measurement framework. According to a management model, companies can be measured corresponding to different perspectives, e.g., financial, customer, internal business processes and others. Different aspects (e.g. connection to success factors, reporting and responsibilities) of performance indicators could be modelled and documented.

The advantage of using a data warehouse for the implementation of a performance measurement system is the possibility to use existing infrastructure of the company's data warehouse. Data warehouses reflect traditionally customer and financial indicators of companies. Management theories, e.g. the Balanced Scorecard, are hardly ever used in the field of data warehousing. We can conclude that internal business processes, learning and growth and other (e.g. environment/community) perspectives are typically not covered. The authors of [11] state that the next step is to integrate the relevant perspective of internal business processes into a data warehouse. Some experience in this direction is described also in [12], [13] and will be later briefly explained in this paper.

3.1 Data Warehouse as a Solution

Data warehouses as a solution for storing and analysing business data are described in a number of papers [13], [6]. This section summarizes the specific features of data warehouses, if they are used for an implementation of performance measurement systems.

- *Different new data sources*. Workflow logs are integrated with other data sources. As a special case only workflow log files are studied [14]. Workflow Management Coalition has defined three types of data that are related to workflow systems [15]: data of application programs, workflow-specific data and internal data of workflow systems. The workflow-specific data determines the choice of particular execution path; data is calculated at the moment of process execution and often not stored in workflow logs. Internal data of workflow systems is the data about the execution of the workflow; it is stored in log files.
- *Specific data analysis approaches*. Different ways for the analysis of workflow logs are described in [6], but they are applicable also in the case of other types of process data warehouses.

The data analysis could have different data analysis goals, e.g. technical or business goals. The examples of technical goals are testing of workflow systems, evaluation of response time or workload of a system. The data analysis could be performed at different levels by different users, so, personalized needs of an individual user or the interests of the whole company could be the focus of the data analysis. Individual users could have also different roles during the analysis, e.g. process owners, process performers, managers, support staff, data analyst and others.

The data analysis could have different time periods – e.g. short term analysis of processes, which presumes monitoring of process execution at the time of the analysis, and long term analysis (or process control), which means the analysis after the process execution.

These approaches could be combined, e.g. process monitoring in the case of technical goals could be used to evaluate the number of active users and the current workload. The data analysis also could have different analysis perspectives depending on the performance measurement framework used, e.g. customer, financial etc.

- *Data warehouse model*. The existence of different above mentioned aspects may determine the data items to be included into the data warehouse model for performance measurement. Important aspects that should be evaluated [13] are: the key measurements to be implemented in a data warehouse, their priorities and the situation with the necessary data to calculate these measurements. Possibly, some new data elements should be collected in the future.

3.2 Overview of Data Warehouses for Performance Measurement

Descriptions of implementations of performance measurement systems by means of a data warehouse are given in several works. Many solutions for specific aspects that arise, if data warehouses are used for implementation of performance measurement systems, are introduced (e.g. dimensional models).

The Process Data Warehouse is defined [12] „as a data warehouse which stores histories of engineering processes and products for experience reuse, and provides situated process support".

The concept of a Performance Management System (PMS) is defined in [13] as a system which "stores and manages all performance relevant data centrally, including both financial and non-financial data", and also ensures system's approach to measurement and timely access to data. The method used to build a data warehouse for the PMS is given. Performance indicators are defined based on analysis of company's goals, processes and stakeholders. Information needs are elicited. The PMS contains values of measurements and supplementary information about company structure, business processes, goals and performance measures. Besides traditional data warehousing perspectives of performance measurement, the process perspective is also analysed to some extent (e.g. execution time).

In [11] the authors propose a Corporate Performance Measurement System (CPMS), where process performance data is integrated with institution's data warehouse. Log files of a workflow system are used as data sources. Also, a method used to build a data warehouse for the CPMS is given. Goals for business processes are derived from goals of the company. Questions about measurement of goals are used, relevant indicators and data sources are described. The model of CPMS is developed as a part of an existing data warehouse model of the company.

3.3 Concept of a Data Warehouse of Processes

A category of data warehouses for performance measurement can be distinguished, where the focus is the storage and analysis of business process execution patterns. So, the concept of a Data Warehouse of Processes is introduced, however, the interpretations of the concept could differ.

In the systems mentioned already in previous section workflow data is used as a data source. The performance management system [13] stores process execution data besides other data to ensure the systematic measurement of processes. In the corporate measurement system [11], the data about the execution of processes from log files of workflow systems are integrated into the relevant company's data warehouse.

Workflow data warehouse [14] represents the concept of Data Warehouse of Processes. The authors of Workflow Data Warehouse [14] argue why and when data warehouse can be an appropriate solution for storing and analysing log files of process execution. Existing analysis tools provide limited possibilities – typical measures are number of executed process instances per time period, average execution time. The authors [14] propose a general-purpose model that is meant for storing different types of related facts, e.g. an activity is executed in the context of particular definition of workflow as a part of particular branch. The proposed model also includes Behaviour dimension that defines typical patterns of workflow execution in the past and that allows to analyse the current workflow execution, according to these predefined patterns. An important aspect of the proposed solution is that the dimensional model includes additional fact table Process Data Fact that represents the business data, where each particular process has changed.

3.4 Summary about Process Measurement Systems

On one hand, it is not enough to analyse separately some particular aspect, e.g. the workflow log files, because it does not help to evaluate the true status of the business, if we do not know anything about the data changed by business processes. On the other hand, when only business specific data is analysed, we can get information about a business situation, but there is a lack of reasonable information on why the situation is such as is shown in the business data analysis. The investigation of the workflow data can help to find out the bottlenecks of the workflow execution and to improve them.

Often in the real world not all possible data sources are used for data analysis, e.g. different log files, because of additional complexity of the integration; this data may also be underestimated as a valuable information source. The limited choice of data sources determine that the business indicators could not be freely defined.

A data warehouse as a solution for a process measurement system is appropriate, when mostly the long term analysis is performed, when integration of all possible data sources is needed and one part of necessary data already exists within a data warehouse of a company. A data warehouse will also be an appropriate solution, if analysis is performed at the level of an organization.

4 Performance Measurement Framework with Indicator Lifecycle

The previous concepts from the section about performance measurement concerning the management part of the measurement could be treated together with concepts from measurement theories to explain the nature of indicators in a most comprehensive way.

We could observe a lifecycle of indicators, which consists of 5 steps – indicator definition, measurement, analysis, reaction and improvement. Indicator definition step describes mostly different features of indicators that help to understand why that measure is introduced. The measurement step represents the process, when indicators get the values. The analysis step represents the process, when indicators are used to make decisions. The reaction step represents the process, when the decisions that are made in the previous step are implemented. The improvement step supports the evaluation of indicator definitions and values of aspects.

According to the five-step lifecycle of indicators, each step could be represented by different aspects (Figure 1). These aspects have the same meaning as described in the previous sections, but are grouped according to the particular step. We introduced some additional aspects, e.g. *Level*. This aspect describes the Level of indicators, whether it is needed for *Organization* level (team level) or for an *Individual* person. Concerning the *Process* aspect, it should be mentioned that for the reaction purposes also the goal of measurement should be clarified.

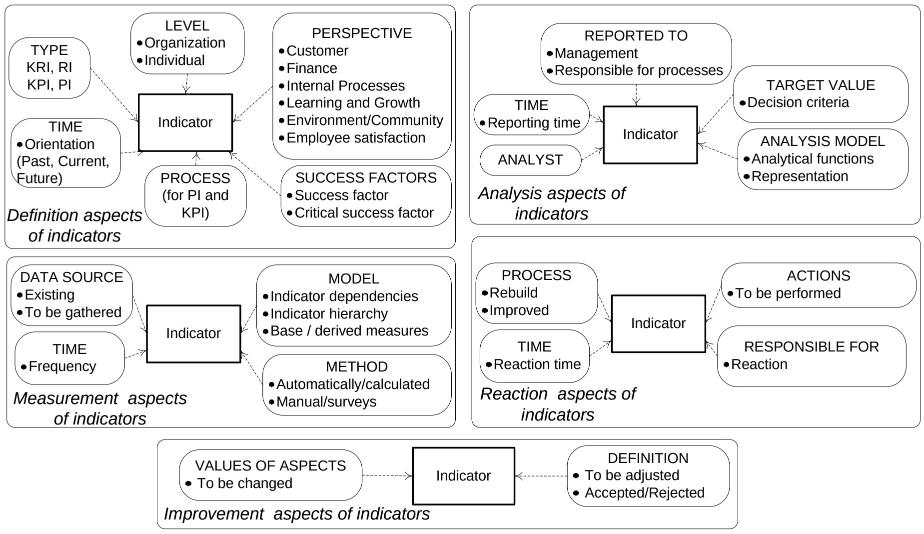

Fig. 1. Five groups of indicator aspects.

The proposed measurement framework with indicators as the focus raises two further questions – (i) which is the most appropriate implementation to support the described lifecycle, and (ii) how the indicators should be formalized to bring the maximum of clarity into the process, what, why and how is measured.

The analysis of existing Performance Measurement Systems in Section 3 shows that the data warehouse can be chosen for a possible solution for the proposed measurement framework. Further in this paper, the formalization method of sentences that express the indicators is proposed.

5 Formal Model for Indicator Definition

Software requirements are normally described using a requirement specification language with various levels of formality [16]. The approach that we have proposed in our previous work [17] is based on extracting necessary information out of business requirements defined in natural language, thus, making business requirements more precise and formal. The proposed approach in [17] is meant for expressing requirements of data warehouse systems with one particular goal – to find out similar requirements for creating similar multidimensional model for a data warehouse that is built in the same industry.

The type of an information system to be developed has some impact on a way of formulating sentences that express requirements. Before starting our study, we assumed that requirements for data warehouses and information requirements particularly have a similar structure or pattern. This assumption was based on our observations on how the information needs were described in practical data warehousing projects. Also, the typical way the data warehouse data are analysed afterwards by means of typical OLAP operations with underlying typical SELECT statements shows that this assumption could be true.

Requirements are represented by sentences (subsets of words in natural language) that are used to specify what action(-s) the system should perform and which object(-s) it will affect. The question is whether the description of action(-s) and object(-s) could be a subject of formalization, e.g. how the terms of sentences are structured and if common pattern could be observed. In [17] a metamodel is given that describes the common structure of the information requirements for data warehouses.

In the case of performance measurement, we propose a revised metamodel defined in [17] to express the indicators.

We could use similar approach because of several reasons. On one hand, indicators are the focus of data analysis in the measurement process, according to the measurement concepts. On the other hand, the data warehousing models are built to represent the information needs for data analysis. So, we could talk about indicators as an information requirement for a data warehouse system. Therefore, the formalization of indicators could be based on the nature of elements of multidimensional models, e.g. the distinction between Quantifying Data and Qualifying Data.

We based the proposed model on the structure evaluation of the sentences that formulate performance indicators taken from the performance measures database [1].

5.1 Principles of Indicator Reformulation

After considering approximately 330 different indicators in [1] that refer to customer focus (CF), environment & community (EC), employee satisfaction (ES), finance (F), internal process (IP), and learning & growth (LG), we highlighted the following principles:

- An indicator component, which is supposed to be measured, is treated as an aggregated number of all occurrences of this component. For example, *calls* is reformulated to “count (call occurrence)”, where *count* is the suitable aggregate function.
- If an indicator component is supposed to be shown in detail, then in the corresponding requirement the refinement function *show* is applied. For example, *employee* is reformulated to “show employee”.
- If an indicator contains such components as “listing of”, “list of”, or “instances of”, then in the corresponding requirement the refinement function *show* is applied. For example, *listing of customers* is reformulated to “show customers”.
- If an indicator contains such component as “number of”, then in the corresponding requirement the aggregate function *count* is applied. For example, *number of visits* is reformulated to “count (visit occurrence)”.

- If an indicator contains such components as “cost of”, “value of”, “expense”, “total expense”, “income”, “total income”, “revenue”, “investment”, etc., or the name of currency in the beginning if the indicator, then in the corresponding requirement the aggregate function *sum* is applied. For example, *dollars saved* is reformulated to “sum (dollars)”, however, *total income* is reformulated to “sum (income)”.
- If an indicator contains such component as “average”, then in the corresponding requirement the aggregate function *avg* is applied. For example, *average response time* is reformulated to “avg (response time)”.
- If an indicator contains such components as “%”, “percent”, “percentage”, or “ratio”, then the percentage is substituted by *division* of partial quantity by total quantity. For example, an indicator *IT expense as a % of total administrative expense* is reformulated to “sum (IT expense) / sum (expense)”.

Of course, the mentioned principles are supposed to be used taking into consideration the context of each indicator. One should analyse indicators to decide whether the data has to be aggregated or not and choose the appropriate aggregate function, if needed. Some of the instances of such indicators are: sales closed, initiatives completed, dates, candidates, days of production, energy consumed, etc.

5.2 Requirement Pattern Description

All indicators have common structure, for that reason it is possible to determine a pattern for re-writing business requirements formally. The approach that we use is describing indicators by means of formal grammar (EBNF notation) depicted in Figure 2. The same idea of requirement formalization may be represented as a metamodel. The metamodel is designed using UML 2.0 class diagram notation (Figure 3).

In Figure 2 business requirement is denoted by *Requirement* abstract class, which divides into a *Simple* and *Complex Requirement*. A complex requirement is composed of two or more simple requirements with an *Arithmetical Operator* between the simple requirements. A simple requirement consists of a verb (*Operation*) that denotes a command, which refers to an *Object*, and zero or one *Typified Condition*.

There are two kinds of data in data warehousing: *Quantifying* (measurements) and *Qualifying Data* (properties that characterize measurements). An object is either an instance of quantifying or qualifying data depending on the requirement.

The term „operation” describes the kind of *Action(-s)* to be performed. If some kind (or different kinds) of action should be performed more than once, then it is called a *Complex Operation*. We propose two possible types of action: an *Aggregation* (a command, used for calculation and grouping, “roll-up”) and a *Refinement* (a command, used for information selection, “drill-down”, as an opposite to an aggregation). Information refinement is either showing details, i.e., selecting information about one or more objects, or slicing, i.e., showing details, according to a certain constraint (*Typified Condition*).

<Requirement>: <Simple Requirement> | <Complex Requirement>
<Complex Requirement>: <Requirement> <Arithmetical Operator> <Requirement>
<Arithmetical Operator>: + | - | * | /
<Simple Requirement>: <Operation> <Object> <Typified Condition>?
<Object>: <Quantifying Data> | <Qualifying Data>
<Operation>: <Complex Operation> | <Action>
<Complex Operation>: <Operation>+
<Action>: <Aggregation> | <Refinement>
<Aggregation>: count | sum | average | …
<Refinement>: show
<Typified Condition>: <Condition Type> <Condition>
<Condition Type>: where
<Condition>: <Simple Condition> | <Complex Condition>
<Complex Condition>: <Condition> <Logical Operator> <Condition>
<Logical Operator>: or | and | not
<Simple Condition>: <Expression> <Comparison> <Expression>
<Comparison>: > | >= | < | <= | =
<Expression>: <Simple Expression> | <Complex Expression>
<Complex Expression>: <Expression> < Arithmetical Operator> <Expression>
<Simple Expression>: <Qualifying Data> | <Constant>

Fig. 2. Business requirements in EBNF notation.

If there is a restriction in the requirement, then it is represented by a typified condition. There are two types of conditions: *Simple Condition* and *Complex Condition*.

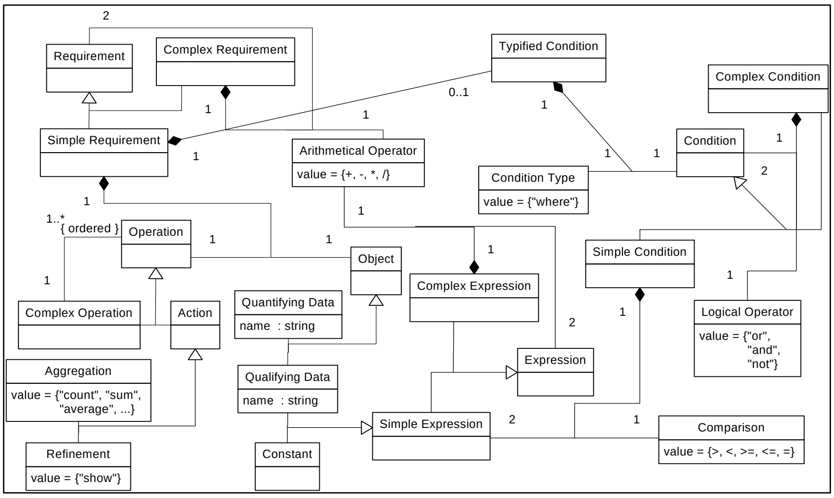

Fig. 3. Formalized requirements metamodel (UML).

Complex condition joins two or more simple conditions by *Logical Operators* (such as “and”, “or”, “not”). Simple condition consists of a *Comparison* of two *Expressions*, for example, “time is greater than last_access_time – 1 second”. An expression as well may be either a *Simple Expression* or a *Complex Expression*. A complex expression contains two or more simple expressions with an arithmetical operator between the simple expressions. A simple expression belongs either to qualifying data (for example, “last_access_time”) or to *Constants* (for example, “1 second”).

5.3 Indicator Examples

Examples given in this section illustrate the application of proposed formal model to define the indicators. We use performance indicator definitions formulated in natural language from a performance measures database [1]. This database contains a comprehensive list of performance measures from six measurement perspectives, e.g. customer, finance, etc. We have chosen indicators from different perspectives with different structure of the sentences in a natural language.

show	Refinement	Action	Operation		Simple Requirement
month	Qualifiyng Data	Object			
AVG	Aggregation	Action	Operation		
contacts ocurrance	Quantifying Data	Object			
where	Condition Type			Typified Condition	
customer type	Qualifiyng Data	Simple Expression	Simple Condition		
=	Comparison				
'key customer'	Constant	Simple Expression			

Fig. 4. Requirement formalization example of the requirement type CF.

Suppose that we would like to obtain information about average number of contacts made with key customers per month. This indicator represents the indicators regarding customer focus. This statement could be reformulated using our proposed requirement pattern metamodel (Figure 3). A statement that is “valid”, i.e., can be derived from the original indicator, is “show month, average (contact occurrence) where customer type is ‘key customer’ ”. Figure 4 demonstrates the application of requirement patterns taking as an example the business requirements mentioned above. The left column is filled with parts of the statement and all the rest columns (left to right) contain names of the metamodel levels from the bottom to the top.

Let’s take other examples from the list of indicators. Suppose that we would like to obtain information about late deliveries to key customers. This indicator represents the indicators regarding customer focus and internal processes. A statement that is derivable from the metamodel (Figure 3) is “show deliveries where delivery type is ‘late’ ”. Because of the space limitations, Figure 5.a demonstrates only the bottom level elements of our proposed requirement pattern metamodel (2nd column) and corresponding parts of the statement (1st column).

Consider that we would also like to get information about staff turnover by type (resignations, end of contract, temporary staff, and termination). This indicator represents the indicators regarding employee satisfaction. A statement that can be derived, using our proposed requirement pattern metamodel (Figure 3), is “show employment types, count (person occurrence) where employment type is ‘resignation’ or ‘end of contract’ or ‘temporary staff’ or ‘termination’ ”. Figure 5.b demonstrates the bottom level elements of our proposed requirement pattern metamodel and corresponding parts of the statement.

(a)

	Indicator Component	Metamodel Element
1	show	Refinement
2	delivery	Qualifying Data
3	where	Condition Type
4	delivery type	Qualifying Data
5	=	Comparison
6	'late'	Constant

(b)

	Indicator Component	Metamodel Element
1	show	Refinement
2	employment type	Qualifying Data
3	count	Aggregation
4	person occurrence	Quantifying Data
5	where	Condition Type
6	employment type	Qualifying Data
7	=	Comparison
8	'resignation'	Constant
9	or	Logical Operator
10	employment type	Qualifying Data
11	=	Comparison
12	'end of contract'	Constant
rows 9-12 are repeated for 2 more comparisons with different constants		

Fig. 5. Requirement formalization examples: (a) Type – CF, IP; (b) Type – ES.

Suppose that we would like to obtain information about the number of sponsorship projects in past 12 months by company. This indicator represents the indicators regarding the environment and community focus. Let's assume that the time period closes with current date. A statement that can be derived, using our proposed requirement pattern metamodel (Figure 3), is "show company, count (project occurrence) where project type is 'sponsorship' and date is greater than current_date - 356 and date is less than current_date". This requirement involves also an arithmetical operator, which is used in the expression together with qualifying data and a constant. Therefore, the formalization example is more complicated than the previous examples. Figure 6.a demonstrates the bottom level elements of our proposed requirement pattern metamodel and corresponding parts of the statement.

(a)

	Indicator Component	Metamodel Element
1	show	Refinement
2	company	Qualifying Data
3	count	Aggregation
4	project occurrences	Quantifying Data
5	where	Condition Type
6	project type	Qualifying Data
7	=	Comparison
8	'sponsorship'	Constant
9	and	Logical Operator
10	date	Qualifying Data
11	>	Comparison
12	current_date	Qualifying Data
13	-	Arithmetical Operator
14	356	Constant

(b)

	Indicator Component	Metamodel Element
1	sum	Aggregation
2	expense	Quantifying Data
3	where	Condition Type
4	expense type	Qualifying Data
5	=	Comparison
6	'IT'	Constant
7	/	Arithmetical Operator
8	sum	Aggregation
9	expense	Quantifying Data

Fig. 6. Requirement formalization examples: (a) Type – EC; (b) Type – F.

Assume that we are interested in summary information on the percentage of IT expense of total administrative expense by quarters in a year. This indicator represents the indicators regarding finance. A statement that can be derived, using our proposed requirement pattern metamodel (Figure 3), is "(sum (expense) where expense type is equal to 'IT') divide by (sum (expense))". This requirement is complex, and it is composed of two simple requirements and an arithmetical operator. Here we apply arithmetical operator '/' to calculate percentage by dividing expenses in IT by total expenses. Figure 6.b demonstrates the bottom level elements of our proposed requirement pattern metamodel and corresponding parts of the statement.

6 Conclusions and Future Work

We have already executed performance measurement tasks at our University, and we have described the approaches and techniques used for that purpose in our previous works [18], [19]. A process measurement and monitoring system (PMMS) was created; a data warehouse was used as one of the important elements of this solution. The PMMS consists of: 1) a process operational monitoring component, 2) a process measurement system, and 3) a process execution log file. The process operational monitoring component supports the analysis of indicators of the process workflow directly from the log file during the process execution. Then the indicator's data is loaded into a data warehouse and used for the process measurement. The results are provided in the case studies of our approach in [18].

Despite the fact that the PMMS is successfully used, we have searched for a more systemic approach to facilitate a more targeted measurement. As a result, the measurement framework proposed in this paper is described, and it is obvious that we will complement the existing PMMS, according to the proposed framework. Also, a data warehouse will serve as an implementation platform for the new version of the PMMS. Not only our experience, but also our literature studies showed that a data warehouse that already exists in an organization could be effectively used for performance measurement purposes.

Another important part of our research is a model for indicator formalization. The future work will be done in two directions. The first one is concerning practical case studies and evaluations of ease of use of formal patterns. The second direction is a development of a new method for semi-automated construction of data warehouse schemas based on the formal definitions of indicators, according to the model given in this paper.

Acknowledgments. This work has been supported by ESF project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

1. Parmenter, D.: Key Performance Indicators: Developing, Implementing, and Using Winning KPIs. Jon Wiley & Sons, Inc., Second Edition (2010)

2. Harrington, J. H.: Business Process Improvement – The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness. McGraw-Hill (1991)
3. Frank, U., Heise, D., Kattenstroth, H., Schauer, H.: Designing and Utilizing Business Indicator Systems within Enterprise Models - Outline of a Method. In: Loos, P., Nüttgens, M., Turowski, K., Werth, D. (eds.) Modellierung Betrieblicher Informationssysteme (MobIS 2008) - Modellierung zwischen SOA und Compliance Management, LNI, vol. 141, pp. 89-105. Saarbrücken, Germany (2008)
4. Popova, V., Treur, J.: A Specification Language for Organizational Performance Indicators. In: Applied Intelligence Journal, 27 (3), pp. 291--301 (2007)
5. Kaplan, R.S., Norton, D.P.: The Balanced Scorecard. Harvard Business School Press (1996)
6. Muehlen, M.: Process-driven Management Informations Systems – Combining Data Warehouses and Workflow Technology. In: ICECR-4 (2001)
7. Fenton, N.E., Whitty, R.: Software Quality Assurance and Measurement: A Worldwide Perspective. International Thomson Computer Press, pp. 1--19 (1995)
8. Munson, J.: Software Engineering Measurement. Auerbach Publications (2003)
9. McGarry, J., Card, D., Jones, C., et. al.: Practical Software Measurement: Objective Information for Decision Makers, Addison-Wesley (2001)
10. Garcıa, F., Bertoa, M., Calero, C., et. al.: Towards a Consistent Terminology for Software Measurement. In: Information and Software Technology**,** Vol. 48, Issue 8, pp. 631--644 (2006)
11. List, B., Machaczek, K.: Towards a Corporate Performance Measurement System. In: Proceedings of the ACM Symposium SAC'04, ACM Press, pp. 1344--1350 (2004)
12. Jarke, M., List,T., Koller, J.: The Challenge of Process Data Warehousing. In: Proceedings of the 26th International Conference VLDB 2000, pp. 473--483 (2000)
13. Kueng, P., Wettstein, T., List, B.: A Holistic Process Performance Analysis through a Process Data Warehouse. In: Proceedings of the 7th American Conference on Information Systems (AMCIS'01), pp. 349--356 (2001)
14. Bonifati, A., Casati, F., Dayal, U., Shan, M.C.: Warehousing Workflow Data: Challenges and Opportunities. In: Proceedings of the 27th International Conference VLDB2001, pp. 649--652 (2001)
15. Workflow Management Coalition, Terminology and Glossary, 3rd Edition, Document Nr. WFMC-TC-1011, Winchester (1999)
16. Videira, C., Silva, A.R.: Patterns and Metamodel for a Natural-language-based Requirements Specification Language. In: Belo, O., et. al. (eds) The 17th Conference on Advanced Information Systems Engineering (CAiSE '05), Porto, Portugal, pp. 189--194 (2005)
17. Kozmina, N., Niedrite, L., Solodovnikova, D.: A Knowledge-based Method for Transforming Requirements to Conceptual Model of a Data Warehouse. In: Proceedings of the 8th International Baltic Conference on Databases and Informational Systems (DBIS '08), Estonia, pp. 63--74 (2008)
18. Medvedis, I., Niedrite, L., Niedritis, A., et. al.: e-University Initiative: Approach, Solutions and Case Studies. In: Proceedings of the 8th International Baltic Conference on Databases and Information Systems, Estonia, pp. 141--152 (2008)
19. Niedrite, L., Solodovnikova, D., Treimanis, M., Niedritis, A.: The Development Method for Process-Oriented Data Warehouse. In: Mastorakis, N.E. (ed.) WSEAS Transactions on Computer Research, 2(2), pp. 183--190 (2007)