Performance Measurement Framework with Indicator Life-Cycle Support

Aivars NIEDRITIS, Janis ZUTERS and Laila NIEDRITE[1]
University of Latvia

Abstract. The performance measurement method introduced in this paper is based on the five-step indicator lifecycle that covers formal definition of indicators, measurement, analysis, reaction, and reformulation of indicator definitions. Performance measurement framework is introduced that support this performance measurement method and that enables the indicator lifecycle. The goal of this research is to provide a method for performance measurement that ensures timely and to given context appropriate decision making process. For the purposes of storing the information necessary for decision support a data warehouse is used as a component of the process measurement framework.

Keywords. Performance measurement, key performance indicators, indicator lifecycle, data warehouse

Introduction

Effective organization of business processes ensures the achievement of institution's goals. Performance measurement compares the measurement results with target values to discover the progress. An important aspect is how to choose appropriate measures and how to define an appropriate measurement framework.

Performance measures [1] are indicators used by management to measure, report and improve the performance in an organization. What kind of particular performance measures are used is influenced by management models of organizations and measurement perspectives of these models. For example, BSC [2] defines four measurement perspectives: Financial, Customer, Internal Process, and Learning and Growth, other approaches add more perspectives, for example, Environment/Community and Employee Satisfaction [1].

To perform effective measurement and adequate reaction on discovered situations, not only different perspectives, but also different aspects (e.g. connection to success factors, reporting, reaction, responsibilities) of performance indicators could be modeled and documented. In our previous research [3] we investigated the features of indicators and grouped them according to the indicator life-cycle. This concept helps to support appropriate usage of indicators according the values of the features.

To implement performance measurement according to the strategies of the company and some management model, the companies develop and use measurement systems. A data warehouse is an option to build a Performance Measurement System.

[1] Corresponding Author: Laila Niedrite, University of Latvia, Raina bulv. 19, LV-1586, Riga, Latvia E-mail: Laila.Niedrite@lu.lv

An 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.

Traditionally data warehouses store customer and financial indicators of the companies, but other perspectives are typically not covered. Some attempts to integrate the perspective of internal business processes into a data warehouse have been made in [4, 5, 6].

We do not try to incorporate another perspective of measurements into the data warehouse. We propose to use the data warehouse as an integral part of a performance measurement system, which can store indicators of different perspectives and can be used according to proposed measurement framework.

The measurement framework describes different measurement aspects to bring order within this important undertaking of the organization. Thereby, the quality of measurement is improved, for example, by means of performing the analysis of right indicators in right time and undertaking the right actions as a result.

The usage of an existing data warehouse gives additional advantage to the performance measurement. The analysis of indicator values can be performed using existing OLAP tools, reports and dashboards.

We start with related work described in section 1. Section 2 explains the concept of indicators life-cycle that forms the basis for proposed measurement framework. In section 3 the reporting tool and its metadata is described that is one of the ready-made data warehousing components used within the measurement framework. The architecture of performance measurement system is given in section 4. In section 5 the conclusions are given.

1. Related Work

Performance measurement systems implemented by means of a data warehouse are given in several works. The existing approaches concentrate mostly on how to build an appropriate dimensional model of the data warehouse according the process perspective of measures to be stored.

The Process Data Warehouse [5] stores histories of engineering processes and products for experience reuse. The Performance Management System (PMS) [6] stores financial and non-financial data centrally. The PMS contains values of measurements as well as supplementary information about company structure, business processes, goals and performance measures. Besides traditional data warehousing perspectives the process perspective is also analyzed. In [4], 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. The model of CPMS is developed as a part of an existing data warehouse model of the company.

A category of data warehouses for performance measurement can be distinguished, where the business process execution data is stored. The systems already mentioned use workflow data as one of data sources, but workflow data warehouse [7] represents the concept of Data Warehouse of Processes. The authors of Workflow Data Warehouse [7] argue why and when data warehouse can become an appropriate solution for storing and analyzing log files of process execution.

Methodologies how the performance should be evaluated also are a subject of research. For example, methodology [8], based on dynamic process performance evaluation, proposes measurement models for analysis of different process flows in order to control the quality of process execution. Activity flow, information flow, resource flow and others are measured using time, quality, service, cost, speed, efficiency, and importance as evaluation criteria.

Our approach uses the advantages of an existing data warehouse – ETL processes, analysis tools, data storage schemas – that allow to prepare and store indicators according to the different perspectives, as well as integrates the data warehouse with a performance measurement framework that is based on the life-cycle of indicators, which ensures the quality of the performance measurement by supplying necessary information for each measurement task.

2. Indicators and Their Life-Cycle

In our previous research [3] we defined a lifecycle of indicators, which consists of five steps – indicator definition, measurement, analysis, reaction and improvement. In each step an indicator is characterized by a different set of properties.

Indicator definition step describes the information needs of the user. In the measurement step the indicators get the values. The analysis step represents the process, when indicators are used to make decisions. The reaction step represents the implementation of decisions. The life-cycle ends with the evaluation of indicator definitions and predefined values of indicator properties during the improvement step.

2.1. Groups of Indicator Aspects

The properties of indicators are grouped in aspects according to the particular step (Figure 1). The explanation of the meaning of properties can be found in [3].

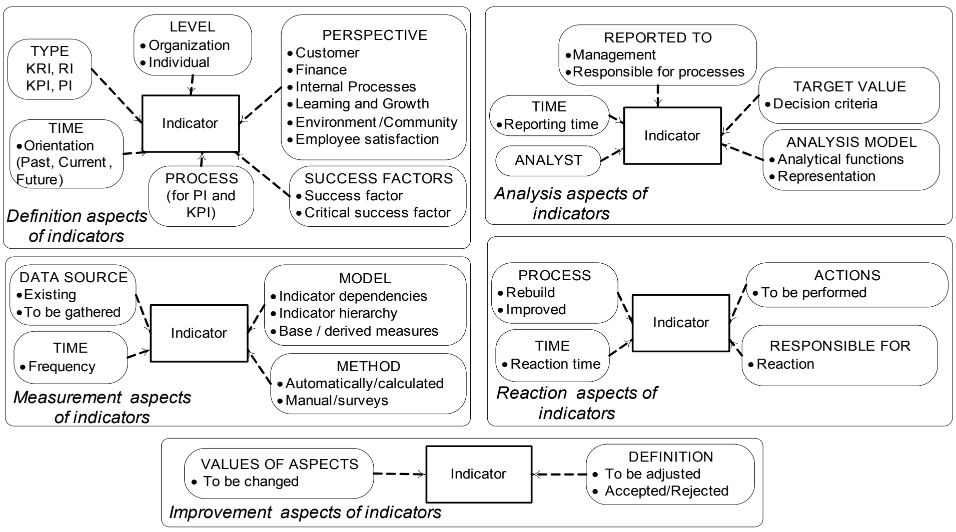

Figure 1. Five groups of indicator aspects [3]

One of the questions raised by the proposed measurement framework with indicators as the central element of interest was how the indicators should be formalized to bring the maximum of clarity into the measurement process – what, why and how is measured. Our previous research [3] is focused on the formal definition of indicators. The formalization method of sentences that expresses the indicators was proposed.

2.2. Formal Model for Indicator Definition

On one hand, indicators are the focus of data analysis in the measurement process. On the other hand, data warehousing models are built to represent the information needs for data analysis. Therefore we could talk about indicators as an information requirement for a data warehouse system.

The type of an information system to be developed has some impact on a way of formulating sentences that express requirements. We assumed that requirements for data warehouses and information requirements particularly have a similar structure or pattern. We based the proposed model on the structure evaluation of the sentences that formulate performance indicators taken from the performance measures database [1].

All indicators have common structure, for that reason it is possible to determine a pattern for re-writing business requirements formally. The requirement formalization may be represented as a metamodel. The detailed description of the metamodel and the algorithm how the sentences expressing the indicators are reformulated according to the given metamodel can be found in [3].

3. Reporting Tool and its Metamodel

One important and integral part of our process measurement framework is a reporting tool developed at the University of Latvia. This reporting tool is developed as the part of the data warehouse framework [9]. The reporting tool is based on metadata and in the latest version it has five metadata layers (Figure 2) that describe different aspects of defining and storage of a data warehouse schemata as well as defining and operating reports defined on these schemata.

Semantic, Logical and Physical metadata describe the data warehouse schemata in different levels of abstraction, starting from the business understanding of the schema elements, describing it by means of OLAP concepts at logical level and ending with the physical storage of the data warehouse tables. OLAP Preferences metadata is introduced to describe the user preferences on reports' structure and data and is used for OLAP personalization purposes. Reporting metadata contains definitions of reports on data warehouse schemata.

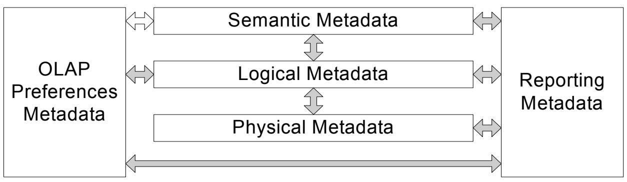

Figure 2. Metadata connections [10]

Metadata levels are interconnected by associations between particular classes of metadata. In the context of this research, the Logical and Reporting metadata are of particular interest; so both the levels as well as the connections between them will be described in more detail here. Detailed description of the rest of the metadata levels can be found in our previous research [11, 12].

3.1. Logical Level Metadata

The logical level metadata describes the data warehouse schema from the multidimensional viewpoint (Figure 3) and mostly is based on the OLAP package of Common Warehouse Metamodel (CWM) [13]. Therefore it contains the core concepts of OLAP – dimensions and fact tables (cubes in CWM).

Fact tables contain measures, but dimensions consist of attributes and hierarchies built from hierarchy levels. Fact tables and dimensions have *FactTableDimension association*. Only dimensions and fact tables having *FactTableDimension* associations can be used simultaneously in one report. More about connections with reporting metadata is given in section 3.3.

The standard OLAP package of CWM is extended by the class *AcceptableAggregation* that allows only meaningful definitions of aggregate functions (e.g. SUM, AVG) for each measure and dimension. This metadata is used to ensure correct queries by the reporting tool.

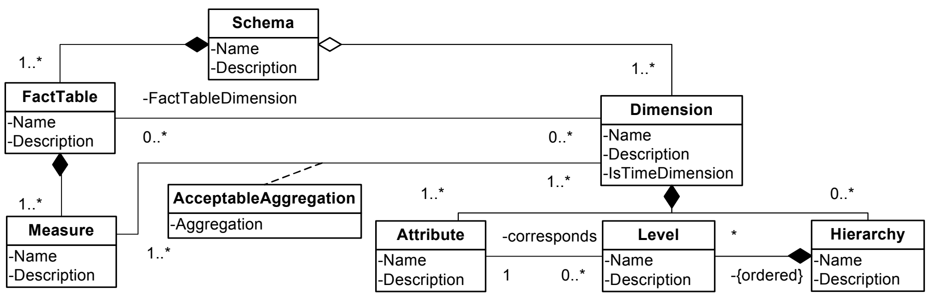

Figure 3. Logical level metadata [11]

3.2. Reporting Level Metadata

Reporting metadata describes the structure of reports (Figure 4). In the meaning of this model, reports are worksheets. Worksheets contain data items defined by calculations. Calculations in their turn specify formulas containing parameters and table columns that correspond to schema elements of the underlying data warehouse. Reports also are based on joins between tables and may have user-defined conditions.

Reports in the tool are created by choosing desired elements of a data warehouse schema and defining conditions, parameters etc. Only measures and attributes belonging to one schema could be included in the definition of one report.

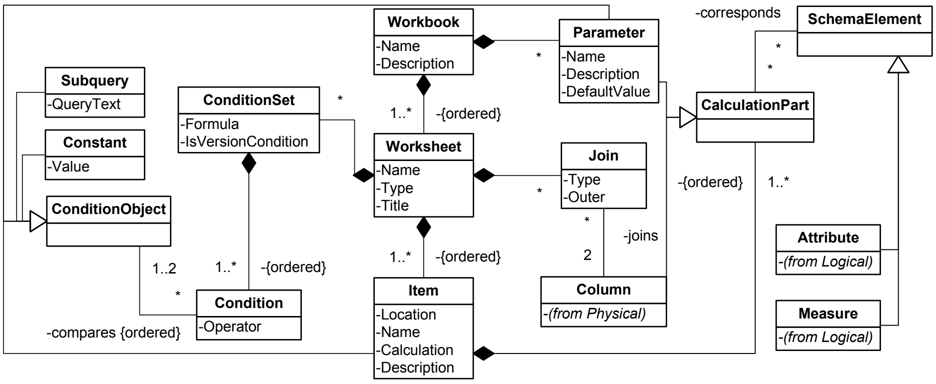

Figure 4. Reporting metadata [14, 12]

3.3. Connections between Logical and Reporting Metadata

The models of logical and reporting metadata are interrelated. Report items are defined by formulas from calculation parts. If a calculation part corresponds to a particular dimension attribute or measure, then this schema element from Logical metadata is connected to the class *CalculationPart* by the association *'corresponds'* in the reporting metadata.

4. Construction of Performance Measurement Framework

We propose an approach of building Performance measurement systems by substantially exploiting existing data warehouse technologies.

The proposed Performance measurement framework is grounded on the following principles of design and operation:

- processing of indicator information is performed in conformance with the life-cycle of indicators and formal indicator metamodel, defined in [3];
- measurement data are obtained through an ETL process and stored in a data warehouse;
- indicator analysis aspect is provided by using a ready-made data warehouse reporting tool extensively both for obtaining actual value from measurement data and providing users with detailed reports.

4.1. Architecture of Performance Measurement System

The kernel of the performance measurement framework (Figure 5) consists of performance management component and the indicator life-cycle support database, as well as the dashboard module.

Indicator life-cycle support database (detailed information is given in the next section) stores links to the formal definitions of indicators from the *Indicator formal definition database*, which is built according to the formal model for indicator definition described earlier in section 2.2. These indicator definitions are collected and formalized during the requirements gathering process for obtaining the precise and appropriate indicators for process measurement, as well as for documenting the information requirements of a data warehouse.

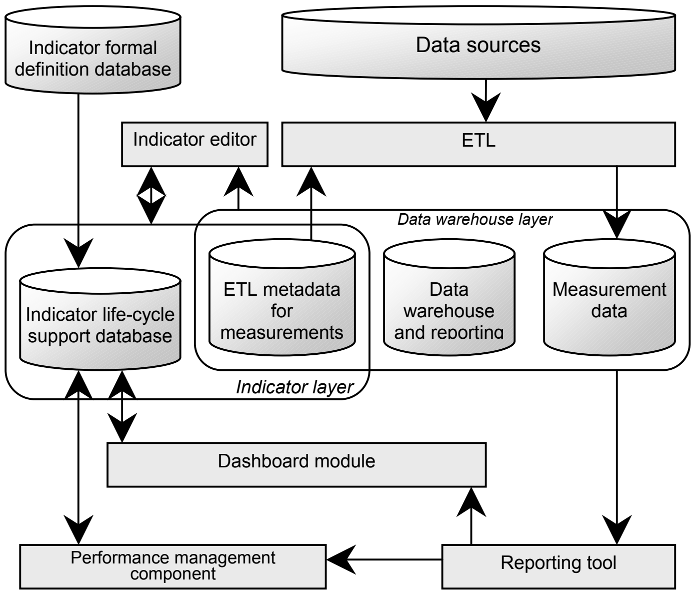

Figure 5. Data flow diagram of the performance measurement framework

Indicator editor is an administrative tool and is meant for two purposes: (1) to establish the links between the Indicator layer of the system and the Indicator formal definition database and (2) to configure the Indicator measurement (or ETL) metadata.

The measurement process, during which indicators get their values, is performed through the *ETL processes,* which use corresponding *ETL metadata for measurements*. The ETL component is an external part of the performance measurement framework. In the context of this research we assume that a set of procedures is defined for performing the data warehouse data renewal according to the values of ETL metadata for measurements (e.g. according to the planned timing schedule). During the ETL processes data from external *Data sources* are processed and loaded into the data warehouse that represents in our framework the *Measurement data*.

The remaining part of the data warehouse layer of the proposed framework is the *Data warehouse and reporting metadata* component that is developed according the previously described metadata layers in sections 2.1 and 2.2 that describe respectively the logical level of data warehouse schema and the reporting metadata.

Performance management component is the main part of the framework that is provided to coordinate the monitoring of business processes by analyzing the measurement results of indicators. Component is based on descriptions of different properties of indicators that are stored in the Indicator life-cycle support database and that allow the user to analyze the indicator values in the most appropriate way by means of two other components of the framework – *Dashboard module* and the *Reporting tool*.

The Dashboard module visualizes the most important values of indicators comparing them to the stored target values of indicators. The reporting tool provides more detailed information to the user by calling predefined reports linked to particular indicator definition. An existing reporting tool is used, which is built according to the previous mentioned reporting metadata (more information about this tool can be found in [14, 12].

4.2. Indicator Life-Cycle Support Database

Indicator life-cycle support database spins around the '*Indicator life-cycle support metadata'* (Figure 6), which define the behavior of the framework. These metadata are used by *Performance management component* designed to coordinate the workflow of the indicator life-cycle.

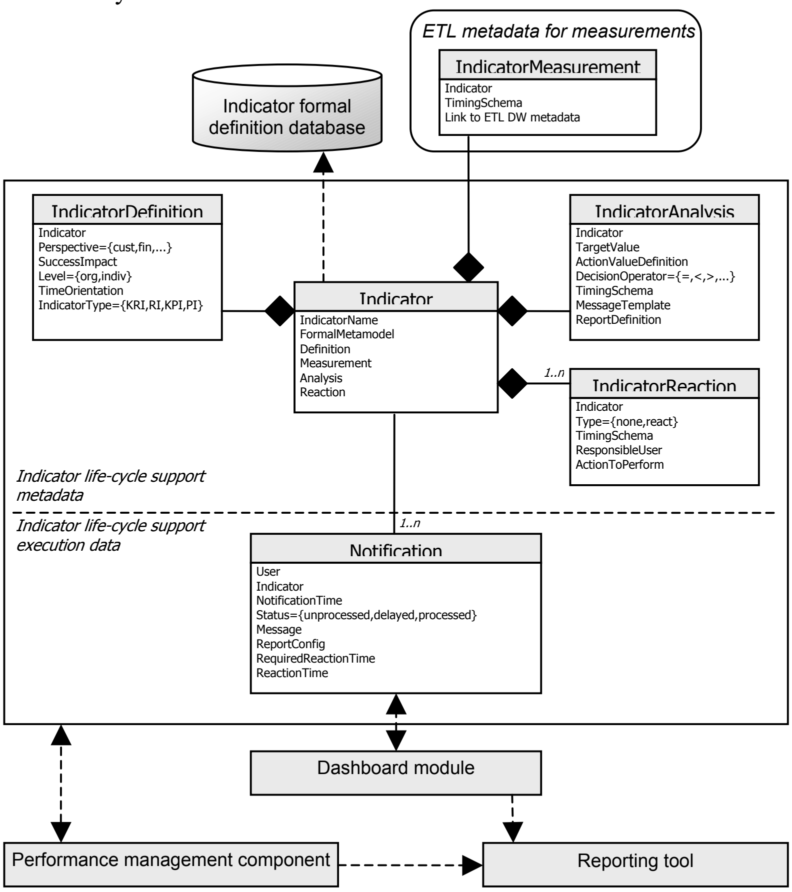

Figure 6. Indicator life-cycle support database and the context

As the duty of performing measurements is fully assigned to the data warehouse, ETL metadata are prepared and stored separately from Indicator life-cycle support database. Actually, this is one of the key points of the framework to fully connect data warehouse for such functionality.

Workflow status is stored in the 'Indicator life-cycle support execution data' and is accessible directly by users via Dashboard module. Workflow status is controlled both by Performance management component and by Dashboard module. It incorporates information about notifications by the system sent to users and the reaction of users to them.

4.3. Indicator Life-Cycle Steps in Performance Measurement Framework

According to the indicator life-cycle definition in section 2, Performance measurement framework should support all five steps of the life-cycle. This section is to describe the proposed framework according to the life-cycle steps.

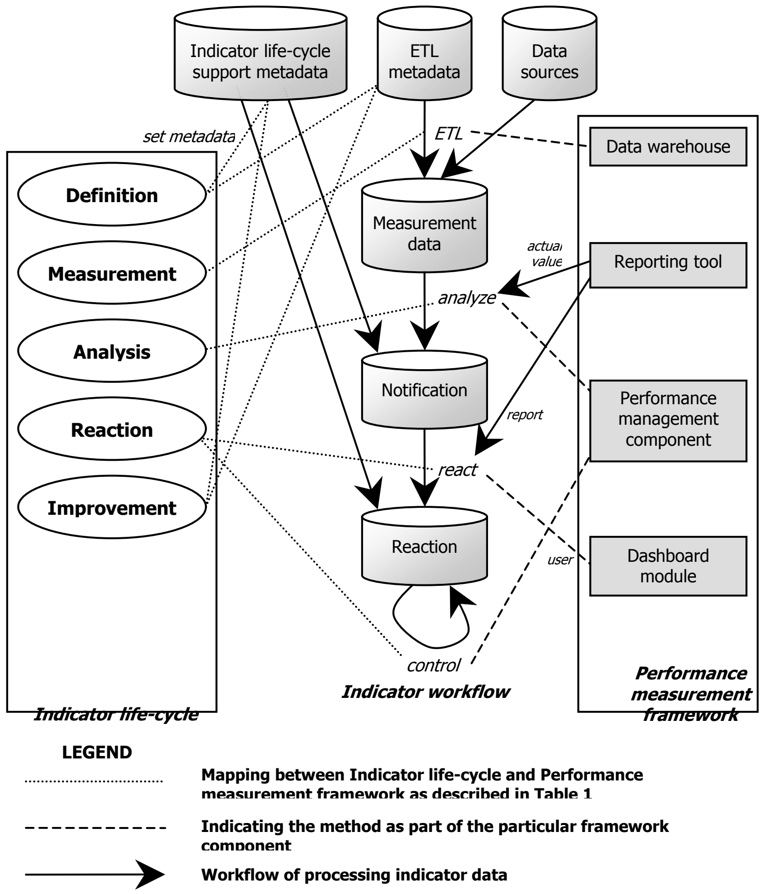

Figure 7. Indicator life-cycle support implemented by the Performance measurement framework

Figure 7 shows the connections between each step and the workflow performed by the framework components and different data. The workflow of processing indicator data is organized in the following steps:

1. Measurement step is performed by an ETL process of the data warehouse (see section 4.4).
2. In analysis step measurement data are processed according to indicator life-cycle support metadata by the Performance management component (see the algorithm in Figure 8). Reporting tool is used here to obtain the actual value of the indicator. During this step, a record is added to the indicator life-cycle execution data; thus, the information about the performed measurements of indicators in form of a notification becomes visible to appropriate users in a special dashboard.

3. User's reaction is obtained from the Dashboard module (Figure 9) and can be of two types:
   - A request for the detailed notification. Reporting tool is used here to obtain a report that describes the actual measurement in detail;
   - Reaction. If the description of an indicator provides for a response to the notification, user is required to assert this in time and in a special way.
4. In control step Performance management component checks whether users have responded to the notifications, if such reactions were appointed in the analysis step (Figure 10).

The above described processing of indicator data by Performance measurement framework is performed in conformance with the life-cycle of indicators.

Table 1 shows mapping between the indicator life-cycle and its implementation by the Performance measurement framework.

Table 1. Mapping "Indicator life-cycle ↔ Performance measurement framework"

Indicator life-cycle aspect group	***Description of implementation by the performance measurement framework***
1. Definition	Indicator definition is described in indicator formal definition database, as well as Indicator life-cycle support database. Indicator definition includes preparation of the metadata required to ensure the whole process.
2. Measurement	Measurement process is fully delegated to the data warehouse and its appropriate ETL process.
3. Analysis	Analysis is coordinated by Performance management component. Measurement data are processed and displayed to the users.
4. Reaction	A user reads and, if required, reacts to the notification. Performance management component controls the reaction.
5. Improvement	Indicator improvement technically matches indicator definition.

```
Procedure analyze
Begin
    Repeat Forever
        Foreach indicator From Indicator Do
            analysis := Indicator.Analysis
            Wait for the next report according to analysis.TimingSchema
            actual_value := run report according to analysis.ActualValueDefinition
            If analysis.DecisionOperator (actual_value, analysis.TargetValue) = True Then
                Forall reaction In Indicator.Reaction Do
                    add record to Notification With
                        User := reaction.ResponsibleUser
                        Indicator := reaction.ResponsibleUser
                        User := indicator
                        NotificationTime := current time
                        Status := 'unprocessed'
                        Message := compute according to analysis.MessageTemplate
                          and indicator.Definition and reaction.ActionToPerform
                        ReportConfig := set according to analysis.ReportDefinition
                        RequiredReactionTime := compute according to reaction.TimingSchema
                        ReadTime := Null
                        ReactionTime := Null
```

Figure 8. Algorithm of the analysis step in the Performance management component

```
Procedure react
Begin
    Foreach user
        Display all from Notification in the dashboard Where User = user
        Foreach notification From Notification Where User = user Do
            Wait for user action Do
                Case user asks to show detalized information Do
                    run report according to notification.ReportConfig and display it
                Case performs an action according to report.Indicator.Reaction.ActionToPerform Do
                    notification.ReactionTime = current time
```

Figure 9. Algorithm of the reaction step in the Dashboard module

```
Procedure control
Begin
    Repeat Forever
        Foreach notification From Notification Where ReportStatus <> 'processed' Do
            If notification.Indicator.Reaction.Type = 'none' Then
                notification.Status = 'processed'
            Else
                If notification.ReactionTime Is Not Null Then
                    notification.Status = 'processed'
                Else If current time > notification.RequiredReactionTime Then
                    notification.Status = 'delayed'
```

Figure 10. Algorithm of the control step in the Performance management component

4.4. Integration with Data Warehouse Components

ETL metadata for measurements (*IndicatorMeasurement* class) is a part of Indicator life-cycle support database (Figure 6). The *Indicator* attribute identifies a particular attribute that is measured, whereas *TimingSchema* describes the time parameters of measurement (e.g. frequency, exact starting time). The last attribute – *ETLprocess* – points to the data warehouse meatadata repository, particulary to the ETL metadata part of the repository that describes mappings between the source and data warehouse schemas. This metadata also contains calls to corresponding procedures that implement these mappings and necessary data transformations. For the proposed measurement framework we can assume that the *IndicatorMeasurement* class contains the procedure call that renews the data warehouse data schema that contains data necessary for calculation of the given indicator.

The *IndicatorAnalysis* class of Indicator life-cycle support database (Figure 6) and its *ReportDefinition* attribute is planned to be a pointer to the report definition stored in accordance to the metamodel of the reporting tool.

Reporting metadata (Figure 4) contain the *Worksheet* class that identifies a particular report that can be invoked when analysis of measurement results is performed. The report can be simple, when one particular value is retrieved to compare it with a target value, or complex, when the report is used for the detailed analysis. The complexity of the report depends on the definition of the particular report.

5. Conclusions

Using data warehouses in performance measurement systems has been already extensively explored. The proposed Performance measurement framework has been designed to obtain the maximum benefits from matured data warehouses technologies in implementing indicator life-cycle support.

The applied model of indicator life-cycle serves as a theoretical means of quality assurance for the performance measurement. The use of data warehouses as integral part of the framework covers two important aspects of ensuring the indicator life-cycle: (a) indicator measurement, and (b) part of indicator analysis (performed by Reporting module).

The provided method for performance measurement ensures timely and to given context appropriate decision making process. The indicator life-cycle support database stores metadata that define and schedule the measurement and control processes of indicators, including timing schemas, responsibilities and actions to be performed. The proposed framework provides the option to build performance control on the activities initiated from the side of the measurement system, as soon as the system recognizes the problem and so the need for more detailed analysis.

Preliminary works of implementing the framework are already in progress, so we expect the first experimental results in the near future.

Acknowledgments

This work has been supported by European Social Fund (ESF) project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

[1] D. Parmenter, *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*. Second ed. Jon Wiley & Sons, 2010.

[2] R. S. Kaplan and D. P. Norton, *The Balanced Scorecard*. Harvard Business School Press, 1996.

[3] A. Niedritis, L. Niedrite, and N. Kozmina, Performance measurement framework with formal indicator definitions. In: J. Grabis, M. Kirikova, editors, *Perspectives in Business Informatics Research*, Lecture Notes in Business Information Processing **90** (2011), Springer, Berlin, 44-58.

[4] B. List and K. Machaczek, Towards a corporate performance measurement system. In: *Proceedings of the ACM Symposium SAC'04*, ACM Press, 2004, 1344-1350.

[5] M. Jarke, T. List, and J. Koller, The challenge of process data warehousing. In: *Proceedings of the 26th International Conference VLDB'2000*, 2000, 473-483

[6] P. Kueng, T. Wettstein, and B. List, A holistic process performance analysis through a process data warehouse. In: *Proceedings of the 7th American Conference on Information Systems (AMCIS'01)*, 2001, 349-356.

[7] A. Bonifati, F. Casati, U. Dayal, and M. C. Shan, Warehousing workflow data: challenges and opportunities. In: *Proceedings of the 27th International Conference VLDB'2001*, 2001, 649-652

[8] W. Tan, W. Shen, L. Xu, B. Zhou, and L. Li, A business process intelligence system for enterprise process performance management, *IEEE Transactions on Systems, Man, and Cybernetics* **38**(6) (2008), 745-756.

[9] D. Solodovnikova, Data warehouse evolution framework. In: *Proceedings of the Spring Young Researcher's Colloquium On Database and Information Systems (SYRCoDIS'07)*, Moscow, Russia, 2007. Available from: http://ceur-ws.org/Vol-256/submission_4.pdf.

[10] N. Kozmina and D. Solodovņikova, Determining preferences from semantic metadata in OLAP reporting tool. In: L. Niedrite, R. Strazdina, B. Wangler, editors, *Perspectives in Business Informatics Research*, Local Proceedings of the 10th International Conference BIR 2011, Associated Workshops and Doctoral Consortium, Riga Technical University, 2011, 363-370.

[11] D. Solodovnikova, Metadata to support data warehouse evolution. In: *Proceedings of the 17th International Conference on Information Systems Development (ISD'08)*, Paphos, Cyprus, 2008, 627-635.

[12] D. Solodovnikova and L. Niedrite, Evolution-oriented user-centric data warehouse. In: J. Pokorny, V. Repa, and K. Richta, W. Wojtkowski, H. Linger, Ch. Barry, M. Lang, editors, *Proceedings of the 19th International Conference on Information Systems Development*, Springer, 2011, 721-734.

[13] *Common Warehouse Metamodel Specification, v1.1*. Object Management Group. Available from: http://www.omg.org/cgi-bin/doc?formal/03-03-02.

[14] D. Solodovnikova, Building Queries on Multiple Versions of Data Warehouse. In: H.-M. Haav, A. Kalja, editors, *Databases and Information Systems V*, Selected Papers from the 8th International Baltic Conference DB&IS 2008, IOS Press, 2008, 75-86.